ROBUST COUNTERFACTUAL LEARNING FOR CLINICAL DECISION-MAKING USING ELECTRONIC HEALTH RECORDS

A Dissertation
Presented to
The Academic Faculty

By

Anirudh Choudhary

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Computational Science and Engineering

Georgia Institute of Technology

December 2020

ROBUST COUNTERFACTUAL LEARNING FOR CLINICAL DECISION-MAKING USING ELECTRONIC HEALTH RECORDS

Approved by:

Dr. May D. Wang
Wallace H. Coulter Department of Biomedical Engineering
Georgia Institute of Technology

Dr. Faramarz Fekri
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Umit Catalyurek
School of Computational Science and Engineering
Georgia Institute of Technology

Date Approved: December 3, 2020

To my family, friends and mentors for their love and support.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. May Wang for providing me the opportunity to be a part of the wonderful BioMIB lab. It has been a privilege to work with her and I am truly grateful for her mentorship and support throughout my Masters studies. Dr. Wang provided the perfect environment to undertake this research and gave me a lot of freedom in choosing projects and learning from my own experiences. Peers and senior students in Dr Wang's group such as Hang Wu and Li Tong have advised, helped, and inspired me. I shall always be grateful for their support.

I am thankful to Dr. Faramarz Fekri and Dr. Umit Catalyurek who spared valuable time to serve on my thesis committee and provided valuable feedback on my research.

A heartfelt gratitude to my personal support system, my wife, Deepika, who has been my constant companion, health coach and an inspiration throughout this journey. I am forever thankful to my brother, Utkarsh, for his constant love and support throughout my career and motivating me to pursue research. I am also grateful to my mother-in-law who selflessly cared for me and kept me motivated through her tasty food in difficult times of the COVID pandemic. I would like to thank my parents, Mrs. Jaya and Mr. Rameshwar Lal, for all their sacrifices to ensure my well-being and their encouragement throughout my life. This work is also a tribute to my father who retired this year from his service of 37 years at Indian Oil Corporation. Papa, you continue to inspire me with your dedication and hard work.

I would like to thank my friends - Prashant, Naresh, Arpit, Saboo, Anurag, and Nakul who have motivated me to perform to the best of my academic capabilities and given me a lifetime of fun memories to cherish. Finally, I would like to thank the CSE department at Georgia Tech for providing me the opportunity to study a subject I am truly passionate about. Go Jackets!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Building clinical decision support systems, which includes diagnosing patient's disease states and formulating a treatment plan, is an important step toward personalized medicine. The counterfactual nature of clinical decision-making is a major challenge for machine learning-based treatment recommendation, i.e., we can only observe the outcome of the clinician's actions while the outcome of alternative treatment options is unknown. The thesis is an attempt to formulate robust counterfactual learning frameworks for efficient offline policy evaluation and policy learning using observational data. We focus on the offline data scenario and leverage historically collected Electronic Health Records, since online policy testing can potentially adversely impact the patient's well-being. The problem is compounded by the inherent uncertainty in clinical decision-making due to heterogeneous patient contexts, the presence of significant variability in patient-specific predictions, smaller datasets, and limited knowledge of the clinician's intrinsic reward function and environment dynamics. This motivates the need to tackle uncertainty and enable improved clinical policy generalization via context-based policy learning. We propose counterfactual frameworks to tackle the highlighted challenges under two learning scenarios: contextual bandits and dynamic treatment regime. In the bandit setting, we focus on effectively tackling the model uncertainty inherent in inverse propensity weighting methods and highlight our approach's efficacy on oral anticoagulant dosing task. In dynamic treatment regime, we focus on sequential treatment interventions and consider the problem of imitating the clinician's policy for sepsis management. We formulate it as a multi-task problem and propose meta-Inverse Reinforcement Learning framework to jointly adapt policy and reward functions to diverse patient groups, thus enabling improved policy generalization.

CHAPTER 1
INTRODUCTION

1.1 The Importance of Clinical Decision-Making

Clinical decision support systems leverage clinical knowledge and patient-related information to improve patient care [1]. Building clinical decision support systems (CDSS), which includes diagnosing patient's disease states and formulating a treatment plan, is an important step toward personalized medicine. We have now witnessed an increasing popularity of machine learning-based diagnosis systems [2], such as in skin cancer detection [3], prediction of cardiovascular risk factors from retinal fundus photographs [4]. However, high-accuracy diagnosis by itself is not sufficient to solve the challenges of building clinical decision support systems. In addition to diagnosing patients, a major part of clinical decision-making is to recommend appropriate treatments for patients with certain diagnoses, i.e., learning policies for treatment recommendation. Clinical treatment recommendations can be grouped into two major categories. The first group focuses on making more accurate predictions for future patient outcomes, sometimes referred to as prognosis prediction, such as in cancer patients [5] and dermatology [6]. Typically, these methods apply supervised learning on historical patient data to predict disease progression, survival outcomes, and certain clinical events in response to a prescribed treatment. Consequently, treatment options with the best outcomes are suggested for decision-support. The second group focuses on building models to map the observed clinical features to treatment actions directly, such that the overall reward, which is directly related to the patient's health, is maximized. Recent biomedical studies have leveraged bandit and reinforcement learning algorithms to recommend adaptive treatment policy regimes in chronic diseases and critical care settings. Some examples include optimizing antiretroviral therapy in HIV patients [7], tailoring anti-epilepsy drugs for seizure control [8], timing ventilation support for ICU patients [9] and determining optimal antibiotic dosing for sepsis

patients [10]. In contrast to the first group which focuses on predictive modeling and suggesting actions which are close to clinician's judgement, reward-based policy learning focuses on exploring alternate optimal actions to derive policy that optimizes the probability of favourable clinical outcomes. This makes the scenario more complicated since the policy output affects both the patient's future health and future treatment plan. In this thesis, our focus is on developing efficient counterfactual learning frameworks to tackle the challenges arising from reward-based policy learning in clinical settings.

1.2 Challenges in Biomedical Domain

The counterfactual nature of clinical decision-making is one of the biggest modeling challenges for machine learning-based clinical decision-making, i.e., for a patient at any given time, we have multiple treatment options, however, we only observe the outcome of the clinician's action and have no knowledge about the efficacy of alternative treatments. Thus, understanding the effectiveness of our treatment suggestion requires us to compare the counterfactual outcome with the observed factual outcome: "Had we administered another treatment to this patient, would the patient be cured?". A common practice to address the counterfactual problem is via randomized controlled trials (RCTs), where the treatment is assigned to patients randomly and the difference between the average outcome of treatment and control groups is a consistent estimator of the effect of the studied treatment. However, to account for patient heterogeneity and be representative of the population, RCTs need to be conducted at a large scale and even then, RCTs provide limited information about treatment applicability to the individual patient [11]. Most of the clinical treatment data is available in the form of observational data, such as electronic health records (EHRs) retained by hospitals and insurance companies. In observational data, the treatment has been assigned by the physicians based on their domain expertise and the patient's condition. Majority of the policy learning algorithms based on reinforcement learning, require interacting with the environment (in our case, the patients) in real time to collect feedback and update the algorithms. This practice is unsafe as learning algorithms can make the policy output arbitrary decisions, which can

adversely affect the patient's health. Thus, it will be unethical to directly deploy the algorithm in a clinic and hence, in this work, we focus on learning decision policies in an offline manner using publicly available health records data.

Limited Data Policy learning algorithms based on reinforcement learning (RL) are often data hungry. For example, it was reported that to match human performance in playing games, an RL algorithm watched "200 million frames from each of the games", equivalent to 38 full days and 500 times as much as human players need [12]. However, in the clinical setting, collecting a sufficient amount of data itself is daunting task due to the personal health data protection regulations as well as the concerns of individuals toward sharing personal data. A typical approach to counter data paucity is to aggregate smaller datasets from various hospitals, however, due to varying clinical protocols and medical devices with different specifications, the multicenter datasets are highly hetereogeneous.

Uncertainty : The heterogeneity in the patient data gives rise to uncertainty, leading to significant variability in patient-specific model predictions and decisions [13]. Even the dataset from a particular institution comprises of patients with varying demographics. The uncertainty is compounded due to limited knowledge of the clinician's intrinsic propensity model for selecting treatments and their underlying reward function. Predictive models based on neural networks have been shown to be prone to model uncertainty under limited data scenarios. Hence, policy learning methods which rely on imputing clinician's action propensity scores to derive optimal policies are rendered prone to uncertainty. Models which achieve nearly similar performance, can disagree significantly in the final predictions, particularly in regions with little or no data. It has been previously shown that effectively capturing the model uncertainty directly translates into lower variance and better exploration during policy learning [14, 13]. Since medical decision-making requires lower risk and higher confidence in policy efficacy, this motivates us to tackle model uncertainty in propensity score estimation and incorporate that into existing off-policy learning frameworks.

Policy Generalization : The heterogeneity and uncertainty also pose a generalization challenge for the learned policy. While the patient's clinical state is defined based on dynamic vitals such as heart rate, albumin level, etc., there are static parameters such as demographics which define the context. Patients with different contexts but similar vitals can respond very differently to a particular treatment [15]. With multicenter datasets having many different underlying patient distributions, and each having limited samples, a central question in ensuring personalized medicine is : "How can our learned policies effectively generalize and adapt to different patient scenarios?". Conventionally, dataset concatenation is used to tackle this problem. This approach, while straightforward, can ignore the underlying differences between datasets, thus trying to learn treatment for the 'average patient' and ignoring context. In addition, most approaches only consider the performance of learning algorithms on historical data, however, in real-world scenarios, the datasets during deployment or test can be different from the historical patient dataset, thus, making the clinical policy adaptation to unseen scenarios a key challenge.

1.3 Contributions

In this thesis, we propose counterfactual frameworks to effectively tackle uncertainty and data heterogeneity while learning clinical policies from electronic health records in an offline setting. We consider two learning scenarios: contextual bandits and dynamic treatment regime.

In the first work, we formulate the clinical decision-making process in the framework of contextual bandits and tackle the model uncertainty inherent in propensity score-based off-policy bandit frameworks. Our frameworks enable policy learning and evaluation with lower reward variance and higher confidence. Specifically, our contributions are as follows:

- We propose bootstrapping-based inverse propensity scoring estimators (IPS_{inv}, IPS_{avg}) for policy evaluation, that can give both the reward estimate and a confidence interval, thus enabling physicians to choose actions with lower variance, when desired.

- Besides estimating confidence intervals for individual patients, bootstrapping reduces the

model uncertainty inherent in IPS-based estimators, thus leading to lesser variance in policy evaluation and improved policy optimization.

- We also tackle model uncertainty from a distributionally robust counterfactual risk minimization perspective and propose an adversarial IPS learner (IPS_{adv}) which focuses on maximizing the reward over the worst-case propensity model bounded by an uncertainty set.

- We demonstrate the efficacy of our proposed frameworks (IPS_{inv}, IPS_{avg}, IPS_{adv}) in a clinical setting involving oral dosing of two popular anticoagulants, heparin and warfarin. Our proposed frameworks help in learning better dosage initialization policies and achieve higher rewards. Moreover, we create semi-synthetic and real-world clinical bandit datasets to promote further research in this field.

In the second work, we consider a dynamic treatment regime involving sequential clinical interventions and address the problem of suboptimal generalization of RL-based treatment policy on heterogeneous patient data. Motivated by the challenges in manually defining the reward function for clinical policy learning [16], we focus on learning the reward function and imitating the clinician's policy for sepsis management using Inverse Reinforcement Learning (IRL). We propose a multitask framework, wherein patients are separated into different groups based on their context (demographics, comorbities) and the reward formulation and policy networks are jointly adapted using meta-learning. We incorporate meta-learning to enable the policy to adapt and generalize across heterogeneous patient groups. Specifically, our contributions are as follows:

- We propose a multitask formulation of offline max-margin IRL that leverages meta-learning to jointly learn and adapt a global policy and reward function to heterogeneous patient groups, thus enabling the policy to generalize better on previously unseen patient contexts.

- By running experiments on real-world clinical problem of sepsis treatment, we showcase the effectiveness of our approach in more effectively replicating the clinician's vasopressor dosage actions compared to the single task IRL, which does not account for patient context explicitly.

CHAPTER 2

BACKGROUND

In this chapter, we provide background on offline counterfactual learning frameworks in contextual bandit and reinforcement learning scenarios. Our focus is on policy learning and evaluation using purely observational data, since evaluating policies by deploying them on patients can be dangerous. In comparison to online learning, offline learning, also known as batch learning, is statistically more challenging since the collected data is generated by a historical policy different from the current policy we intend to evaluate and optimize. We first layout the problem of decision making and introduce relevant notations:

Definition 2.0.1. *We call a mapping, $h : x \to a$, a* policy*, which recommends treatments similar to how a physician makes clinical decisions. The decision-making process involves a tuple of three components*

- *x : the context of the patient, which can include information such as demographics and lab test results, drawn according to distribution λ*
- *a : the treatment action taken by the policy or the physician, for example, whether to administer a drug, or a particular dosage of the drug. The action can be either continuous or discrete, however in our work, we focus on discrete action settings.*
- *r : the reward (or feedback) the policy obtains by taking action a on patient with context x. It is implicitly determined by a function $f(x, a)$, which we typically do not have access to.*

2.1 Contextual Bandits

In the contextual bandit setting, the learner (machine-learning model) repeatedly observes a context, takes an action, and observes a reward for the chosen action (e.g., +1 if patient recovers, -1

if the patient dies). The learner receives feedback (reward) immediately on performing an action, hence bandit learning is essentially single step reinforcement learning without state transitions. It has been recently explored to study clinical decision-making in an offline setting. Kallus and Zhou [17, 18, 19] designed policy learning algorithms in a continuous action space, for recommending the dosage of the drug warfarin to patients with blood clots. Bandit models have also been used in designing and analyzing clinical trials [20, 21], as well as in mobile health applications [22]. Herein, the goal of policy learning for decision-making is to find the optimal policy h which obtains the maximized reward when applied, i.e.,

$$h^* = \arg\max_h \mathbb{E}_{x,a\sim h}[r] \tag{2.1}$$

The learner typically leverages a historical observational dataset comprising of n i.i.d. samples $D = (x_i, a_i, r_i)$, $i \in \{1, 2, 3, .., n\}$; which was collected under a behavioral policy h_0 (also known as logging policy or clinician policy). However, before deploying the treatment policy learned by a black-box clinical support system, special care must be taken to evaluate these policies due to the high-stake scenario. Hence, we first focus on the related problem of counterfactual evaluation in the bandit setting. Accurate policy evaluation is necessary to address the related policy learning problem.

2.1.1 Off-Policy Evaluation

In off-policy evaluation, we seek to estimate the quality of an alternative target policy h by estimating its expected reward, had we applied it to the dataset D:

$$\hat{R}_h = \mathbb{E}_h[r] = \sum_{i=1}^{n} \mathbb{E}_{a\sim h(.|x_i)} \mathbb{E}_{r\sim\mathcal{F}(\cdot|a_i,x_i)}[r] \tag{2.2}$$

Various statistical approaches have been developed to access the quality of target policies based on historical data. There are primarily two classes of evaluation approaches: 1) the direct method (DM) based estimator, also known as regression adjustment and; 2) importance sampling-based

estimator. Direct method uses regression approach to fit a parametric or nonparametric approximation to the true reward function as $\hat{r}(x, a; \theta)$, and the reward of a new policy h is estimated as:

$$\hat{R}_h^{DM} = \frac{1}{n} \sum_{i=1}^{N} \sum_{a \sim h} p_h(a|x_i) \hat{r}(x_i, a) \tag{2.3}$$

where p_h also known as the propensity score, is the probability of selecting action a under policy h, given the observed features x. This approach is simple in its design, but suffers from several biases. The first bias results from the possible mis-specification of the reward function $\hat{r}$ (linear vs nonlinear models), and the second bias arises from the sampling distribution: the target policy might be choosing different actions compared to the logging policy h_0. If h_0 is biased towards a particular region in the action space, the logged data will contain a lot of samples from that region, and the resulting imbalanced dataset will create bias in the reward function estimation.

A common approach to correct for the mismatch in the action distributions under h and h_0 is importance weights, defined as $w(x, a) = \frac{p_h(a|x)}{p_{h_0}(a|x)}$, where p_h and p_{h_0} are the probability of selecting the action a given the observed features, under policies h and h_0 respectively. Importance sampling-based estimators are built on importance weighting with a widely popular estimator being the inverse propensity scoring (IPS) estimator[23]:

$$\hat{R}_h^{IPS} = \frac{1}{n} \sum_{i=1}^{N} \frac{p_h(a_i|x_i)}{p_{h_0}(a_i|x_i)} r_i \tag{2.4}$$

From the formulation, it can be noted that IPS estimator is an unbiased estimator of R i.e. $\mathbb{E}[\hat{R}_h^{IPS}] = R_h$, which makes it well-suited to policy optimization. However, the estimator suffers from high variance in reward estimation, especially when $p_h(a|x) >> p_{h_0}(a|x)$. For consistent estimation, it is standard to assume that whenever $p_h > 0$, then $p_{h_0} > 0$ also and we assume this throughout our analysis. To reduce the variance of IPS, several techniques have been proposed in the bandit literature. A line of work focuses on regularizing the variance of IPS [24, 25, 26] with the POEM estimator being widely used. Another straightforward approach is capping propensity weights [27,

28], which leads to the estimator

$$IPS^M : \hat{R}_h = \sum_{i=1}^{n} \frac{p(a_i|x_i)}{max(M, p_{h_0}(a_i|x_i))} r_i; 0 < M < 1 \tag{2.5}$$

Smaller values of M reduce the variance of $\hat{R}_h$ but introduce bias. Given that the IPS estimator is not equivariant [29], thresholding propensity weights exacerbates this effect. Moreover, IPS estimator is prone to overfitting of propensity weights, i.e., for positive reward, policies which avoid actions in the dataset D are selected; for negative reward, policies that overrepresent actions in D are selected. Hence, Swaminathan and Joachims [29] proposed the self-normalized estimator (SNIPS), which uses weight normalization to counter the propensity overfitting problem of IPS.

$$SNIPS : \hat{R}_h = \frac{\sum_{i=1}^{n} r_i w_i}{\sum_{i=1}^{N} w_i} \text{ with } w_i = \frac{p(a_i|x_i)}{p_{h_0}(a_i|x_i)} \tag{2.6}$$

SNIPS has lower variance than the vanilla IPS estimator because of its ability to normalize and bound the propensity weights between 0 and 1. Additionally, another line of work focuses on reducing both the bias and variance of off-policy estimators by combining the direct method and IPS-based methods in a linear fashion, leading to the doubly-robust estimator [30].

In clinical settings, the behaviour policy is typically unknown. Since IPS-based approaches require the behaviour policy's propensity score p_{h_0}, we need to impute these scores using a behaviour propensity model. The model must accurately represent the clinician's treatment action probability distribution. If the behaviour policy is estimated incorrectly, IPS-based estimators suffer from significant bias and variance. Given that we do not know the parametric class of behaviour policy, we can leverage universal function approximators such as neural networks to estimate the propensity scores. Neural networks often lead to a reduced approximation error with an increasing number of layers and neurons and have been shown to work well in off-policy bandit scenarios [29, 31]. However, learning a highly accurate model for imputing behaviour policy is not enough, our model should provide well-calibrated probability estimates which represent true probabilities.

Using overparameterized approximators such as neural networks, which are capable of expressing a wide range of functions, along with the limited size and heterogeneity of clinical datasets leads to model uncertainty i.e. uncertainty regarding the true underlying parameters. Multiple neural networks can achieve similar accuracy, however, the probability estimates can widely differ and every model might not be able to capture the true conditional probability for the clinician's actions. Therefore, the question which we ask here is: *How can we confidently estimate the propensity score in the presence of model uncertainty due to the limited scale and heterogeneity of clinical data?*

2.2 Uncertainty of Predictive Models

There are two types of uncertainty in machine learning and deep learning models: data uncertainty and model uncertainty [32]. Consider a binary classification setting in which we have $y \sim \text{Bernoulli}(\lambda)$, where y is the binary classification target, and $\lambda(\cdot|x;\theta)$ is the logit representing the conditional distribution $p(y|x;\theta)$ with feature x and parameters θ.

In data uncertainty, the logit λ is a deterministic function of x and θ i.e., $\lambda = g(x,\theta)$, and the uncertainty in data is reflected in the feature x. This uncertainty might be due to inherent noise in the process which generated the data or unaccounted factors which created variability in the targets. This is often referred to as irreducible or aleatoric uncertainty.

On the other hand, model or epistemic uncertainty refers to the uncertainty in the values of the parameters θ for modeling the prediction i.e. we are unable to properly constrain our model's parameters. More specifically, we can model λ as a distribution over a plausible values instead of a point estimate, as $\lambda \sim \mathcal{P}(\lambda|x,w)$ and are unsure of which distributions better explain the data. This could be due to the use of a complex model relative to the amount of training data. Additionally, our choice of model structure might be wrong and is unable to reflect the process which generated the data (here, the clinician). Model uncertainty can be reduced by observing more data, however, typical clinical datasets for bandit learning have limited size ($\leq$ 5,000 patients). Our focus here is on tackling model uncertainty caused by uncertainty in the parameters. To quantify model

uncertainty in clinical setting, we explore the use of two popular approaches: Model Ensembling and Bayesian Neural Networks.

2.2.1 Model Ensembling

Deep ensembles proposed by Lakshminarayan *et al.*[33] is a simple yet powerful method in characterizing the model uncertainty. It has been shown to yield high quality predictive uncertainty estimates, requires little hyperparameter tuning, and is readily parallelizable. Ensembles tackle uncertainty by collecting predictions from M independently trained deterministic models (ensemble components). We train an ensemble of neural networks (NNs) ($NN_1, ..., NN_M$) by varying the random seed in our training process. The seed affects the initialization of the neural network's weights and the order of mini-batch samples seen by the neural network during training. At the test time, for a given patient, we output the ensembled action prediction as $p(a|x) = \frac{1}{M} \sum_{m=1}^{M} p_{NN_m}(a|x)$. In addition, the collection of prediction values $p_{NN_i}(x); i = \{1, 2, ..., M\}$ can be seen as samples from the distribution $p(\lambda|x, \theta)$ describing the model uncertainty.

2.2.2 Bayesian Neural Networks

Bayesian inference is a principled approach to model the distribution over possible outcomes and estimate the uncertainty in the prediction of a machine learning model. Bayesian Neural Networks (BNNs) are neural networks whose parameters θ are represented by probability distributions, so the uncertainty of weights characterizes the uncertainty of models. Given a dataset $D = (x_i, y_i)_{i=1}^{N}$, BNN is defined in terms of a prior $p(w)$ on the weights and the data likelihood $p(D|w)$. By sampling from the posterior weight distributions, BNN could train an infinite number of different realizations of the NNs, and these realizations capture the model uncertainty in the predictive distribution $p(\lambda|x, \theta)$. However, training BNNs is much more challenging since we need to compute the posterior distribution. Various approximate inference methods are proposed to efficiently train BNNs, such as MC-Dropout [34], Variational Inference, [14] and Noisy Natural Gradient method [35]. Bayesian approaches to uncertainty estimation have been proposed to assess the reliability

of clinical predictions [13] but have been applied to very few real-world policy learning settings using clinical data.

Variational Inference

Variational approximation methods aim to estimate the weight posterior by maximizing the evidence lower bound (ELBO) to fit an approximate posterior $q(w|\theta)$, given data D. Variational inference is formulated as an optimization problem of minimizing the Kullback-Leiber (KL) divergence between the approximate $p(w)$ and exact $q(w|\theta)$ posterior. The loss function embodies a trade-off between data-dependent likelihood cost and prior-dependent complexity cost as follows:

$$L(D, \theta) = \mathbb{E}_{q(w|\theta))}[log(p(D|w)] - KL[q(w|\theta)||p(w)] \tag{2.7}$$

where $p(w)$ is the prior distribution on weights, which enforces simplicity. The most common approach to learn an approximate posterior over the weights $q_\theta(w)$ given the prior is mean-field variational inference wherein we assume a fully factorized Gaussian prior and posterior, $q(w) = \prod_{i=1}^{m} q_i(x)$. This reduces the computational complexity of estimating ELBO. To reduce the time complexity of computing KL-divergence during a forward pass through the network, we leverage Monte Carlo estimates. Blundell et al. [14] proposed Bayes-by-Backprop by applying the re-parametrization trick from Kingma *et al.* [36] to variational inference and reduced the computational complexity involved in calculating the data likelihood expectation $E_q[log(p(D|w))]$ over $q(w|D)$. They estimate the variational inference loss function by sampling weights from the posterior $q(w|D)$:

$$L(D, \theta) \approx \sum_{i=1}^{n} log[q(w^i|\theta)] - logP(w^i) - logP(D|w^i) \tag{2.8}$$

where w^i are the sampled weights. To enable training by backpropagation, they choose a Gaussian variational posterior on weights given as : $q(w|\theta) = \prod_{i=1}^{n} \mathcal{N}(w^i|\mu, \sigma^2)$. To perform inference using BNNs, Monte Carlo sampling is performed from the weight distribution. Multiple networks are sampled from the variational posterior q and their predictions are averaged to compute the

network output. In BNNs learned using variational inference, typically both the mean and variance of weights are learnable.

MC-Dropout

Gal *et al.* [34] showed that optimising a standard neural network with dropout and L_2 regularization techniques is equivalently a form of variational inference in a probabilistic interpretation. MC-Dropout is quite popular due to the simplicity of the idea: by enabling dropout during testing and applying different dropout masks, multiple networks can be sampled to predict the output and related uncertainty. This contrasts with performing inference using deterministic neural network wherein the dropout approximation is fixed at the test time. However, in practical applications, MC-Dropout faces some challenges such as the choice of dropout probability and L_2 regularization, the position to insert the dropout layers at, etc.

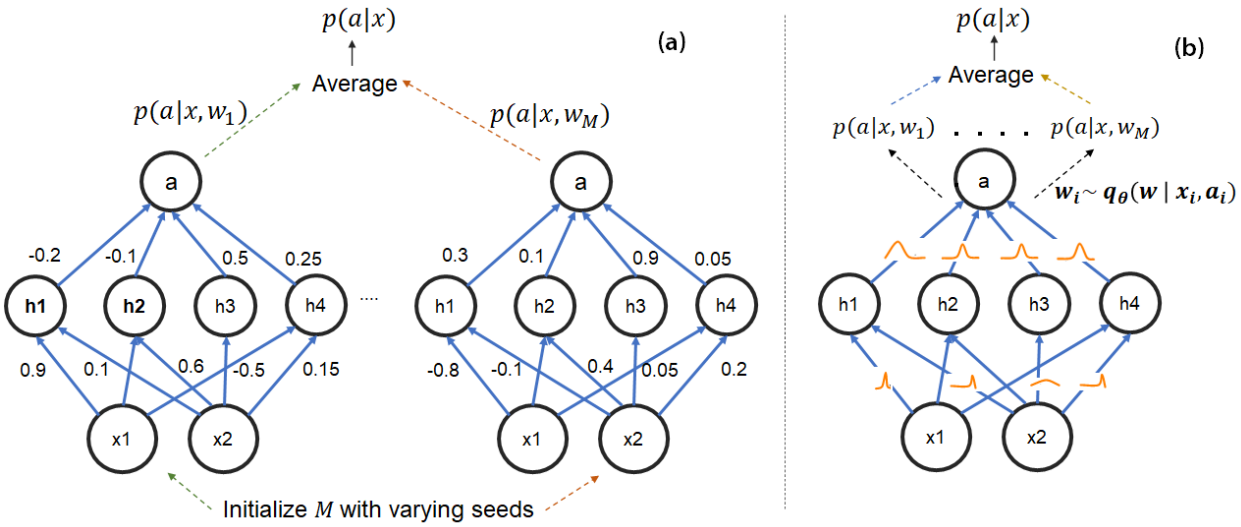


Figure 2.1: (a) Neural Network Ensemble: Each network is initialized with a different seed and network weights are point-estimate (b) Bayesian Neural Networks represent weights by probability distributions and sample networks from learned weight posterior during inference

2.3 Dynamic Treatment Regime

Many clinical settings involve sequential decision-making, also known as dynamic treatment regime, wherein the clinician prescribes certain treatments to improve the patient's condition over a period of time [37]. In such settings, offline policy learning is typically formulated as a Markov Decision Process (MDP) and approached using Reinforcement Leaning with a manually specified proxy-reward function. Manually specifying the reward function is challenging and a poorly specified reward function can adversely affect policy training [38]. Hence, in our second work, we focus on Inverse Reinforcement Learning (IRL) [39] to efficiently recover the clinician's underlying reward function, such that our learned policy imitates the clinician's actions in the best possible manner. Existing offline-IRL approaches [40, 41] have focused on recovering the overall reward function, without explicitly accounting for the patient context. This limits the ability of the policy agent to generalize across unseen patient contexts. Instead, we focus on learning contextual rewards for different patient groups by formulating the IRL problem as a multitask setting. Instead of learning separate policies for different groups, we meta-learn a global generalizable policy by adapting it along with the reward function to individual tasks(patient groups). Meta learning refers to building models that can learn from a distribution of tasks and can quickly adapt to unseen tasks. In this section, we introduce the reinforcement learning problem and provide background on Inverse Reinforcement Learning and multitask learning.

2.3.1 Reinforcement Learning and Markov Decision Process

In a typical online sequential decision-making problem, the policy learner interacts with the environment and optimizes its actions for a sequence of states. The learner takes actions which influence future states and receives rewards for its actions. The reward can be received either at the end of sequential process or after each action. Sequential decision-making is typically modeled using a Markov Decision Process (MDP), which is a mathematical framework for modeling discrete-time sequential processes. A MDP is a tuple (X, A, T, γ, R) comprising of states $x_t \in X$; actions

$a_t \in A$; the probability of transitioning to x_{t+1} from x_t after taking action a_t, $T(x_{t+1}|x_t, a_t)$; the initial state distribution $d(x_0)$; discount factor $\gamma \in [0, 1)$ and reward function $R(x, a)$. The state transitions proceed in a stochastic manner and the process continues until the agent reaches a terminal state x_T. In a MDP, the transition probability distribution of the next state depends only on the current state-action pair. The sequence of state-action pairs observed for a particular starting state is also called an episode or trajectory. The goal of the learner is to learn an optimal policy h by maximizing the value function

$$h^* = \arg\max_h V(\mathbf{x}) = \mathbb{E}_{a(\cdot|x)}[\sum_{t=0}^{T} \gamma^t r(x_t, a_t)] \tag{2.9}$$

Given that we rely on observational data, we focus on offline RL i.e., the learner has access to precollected expert(clinician) demonstrations D^e = (x_t, a_t, x_{t+1}). The performance of RL depends on accurately defining the reward function corresponding to the optimal treatment. In clinical settings, manually defining rewards is challenging and rewards based on the ultimate outcome such as mortality might not capture the objective of improving the patient's condition [42]. Hence, we focus on recovering the reward function from demonstrations which explains the clinician's behaviour and leverage it to derive our policy. This would also help in identifying elements which the expert optimizes unknowingly or might have missed.

2.3.2 Apprenticeship Learning

Apprenticeship learning refers to learning to act from expert (clinician) demonstrations. Learning from demonstrations enables an agent to query an expert as it starts to learn and potentially train itself in an offline setting without access to simulators. There are two major approaches:

1. Behaviour Cloning: In this case, the goal of the agent is to learn the policy directly from the demonstrations using a multi-class supervised learning approach. Such policies can be biased towards actions taken by the clinician and might suffer from error accumulation, wherein the learned policy starts taking actions not encountered during training for a partic-

ular state space.

2. Inverse Reinforcement Learning: In this case, the goal of the agent is to imitate the expert by recovering their underlying reward function. The key assumption is that the expert behaves optimally with respect to some unknown reward function, which encodes knowledge of optimally performing a task. The reward function allows us to succinctly represent a task and IRL focuses on recovering it such that policies trained on the recovered reward function take actions which match with the expert's actions.

Inverse Reinforcement Learning

Inverse Reinforcement Learning (IRL) observes and tries to mimic an expert, instead of manually defining the rewards to learn optimal policies and produce the desired behaviour. Abbeel and Ng [39] proposed one of the initial formulations of IRL, known as max-margin IRL. Max-margin IRL algorithms [39, 43] leverage the feature expectation of a policy μ_θ as a proxy for evaluating the similarity between an expert policy and a policy learned using IRL. The reward function $R = w^T \cdot \mathcal{F}(x, a)$; $||w||_1 = 1$, is typically assumed to be a linear function of a set of known features, which could correspond to raw state-action feature set (x, a) or be derived from a feature mapping $\mathcal{F}$. We compute the feature expectation of a policy as follows:

$$\mu^h(s, a) = \mathcal{F}(s, a) + \mathbb{E}_{a_t \sim h(\cdot|x_t)}\big[\sum_{t=1}^{T} \gamma \mathcal{F}(x_t, a_t)\big] \tag{2.10}$$

The feature expectation for the expert μ^e is estimated from the set of demonstrated trajectories. Max-margin IRL uses $||\mu^h - \mu^e||_2$ as the objective to determine the reward weights. Abbeel and Ng showed that the convergence of feature expectations implies similar expected reward between the two policies and proposed a projection method to determine the weights when they are bounded by euclidean norm. IRL algorithms often require engineering feature functions $\mathcal{F}$ since linear estimators do not have enough representational power to model real-world tasks.

2.3.3 Multitask Learning

As we discussed in Chapter 1, in biomedical settings, due to practical constraints in data collection, often, we only have a limited number of samples per dataset, which greatly limits the power of our learning algorithm. Thus, it is often desirable to integrate several small-scale datasets from different clinical institutions to improve the performance of our policy learning. A straight-forward approach in such data integration is to concatenate all datasets together and apply a base off-policy learning algorithm. However, this approach ignores the difference between different tasks and could lead to suboptimal learning performance. Multitask learning, on the other hand, allows the model to leverage common knowledge from related tasks and thus prevents the model from overfitting to a single task.

As a thought experiment, imagine we are learning decision-making algorithms for cancer patients, whether to suggest chemotherapy or not, i.e., $a \in \{0, 1\}$. We have only two datasets $\{D^1, D^2\}$ from two regions, one colored in black and another in blue. The best actions for patients represented as in circles are $a = 1$, while the best actions for patients in squares are $a = 0$. We cannot directly observe the shape, but only a two dimensional measurement $X1, X2$. If we concatenate the two datasets together, we cannot find a linear function $h : x \to a$ that assigns the best treatment. However, if we can allow our policy to act differently with respect to datasets, we can identify the optimal policy as follows:

$$a^* = h(x) = \begin{cases} \mathbb{I}[x_1 > 3] & \text{x from dataset 1} \\ \mathbb{I}[x_1 < 3] & \text{x from dataset 2} \end{cases} \tag{2.11}$$

In a clinical setting, a simple multitask approach would be to apply a single-task IRL algorithm to recover the reward functions for each task and subsequently learn a global policy. However, in practice, the reward functions for patients with similar underlying disease have similar structures which multitask learning can leverage. However, this approach might favor tasks with significantly larger amounts of data and is sample inefficient [44] (Figure 2.3). Moreover, this is not exactly how

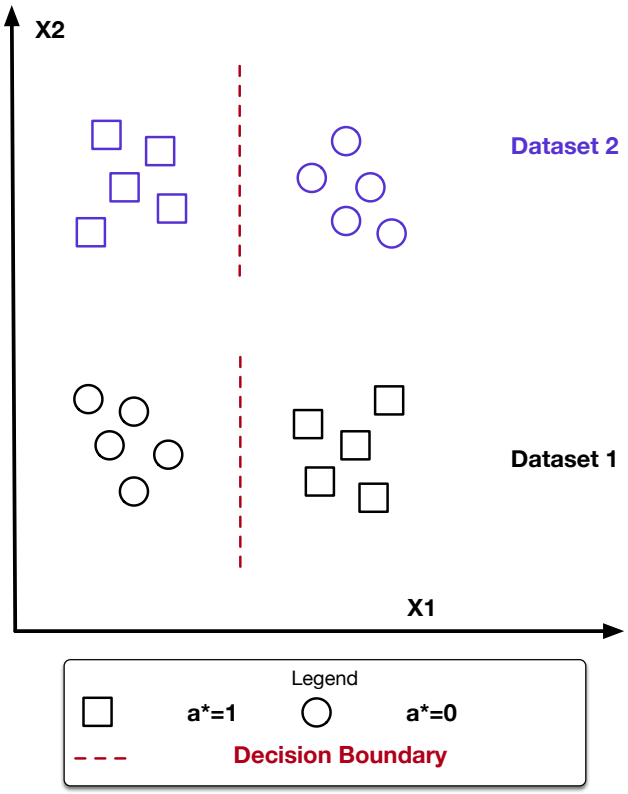


Figure 2.2: An illustration of how multitask learning can help us to identify optimal policies

clinicians learn to make decisions, interacting with the patients daily, updating their knowledge and continuously personalizing their treatments. This motivates us to think: how can we build a learning algorithm that can simultaneously learn common knowledge from all tasks and exploit to learn to adapt to individual tasks differently? The adaptation is important for the policy to learn to generalize. Meta-learning is an appealing alternative to multitask learning that enables rapid generalization by learning a good initialization and fine-tuning on multiple tasks with limited training data.

2.3.4 Meta Learning and Biomedical Informatics

Meta learning, also called learning to learn, describes the machine learning paradigm that extracts knowledge from a set of tasks to learn and allows for rapid adaptation to new tasks. The meta-learner learns to leverage task-specific information to generalize better on training tasks and learn new tasks with fewer samples. Meta-learning approaches can be gradient-based or recurrence-

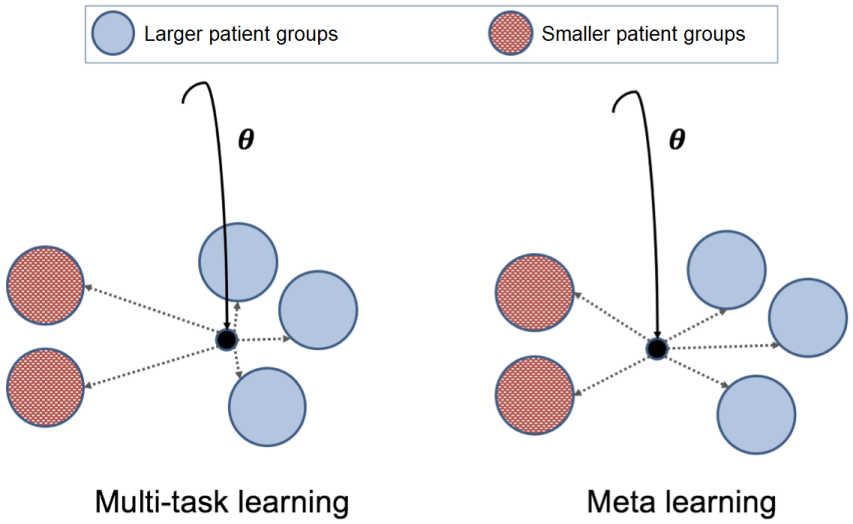


Figure 2.3: Multitask learning vs meta learning: Meta learning aims at learning an optimal model initialization θ which can be easily adapted to unseen patient groups with lesser samples. Multitask learning may be biased towards larger patient groups. Image adapted from Gu *et al.*[45].

based [46]. Here, we leverage gradient-based approach which tries to learn a good initialization of the model's meta-parameter θ, which is iteratively updated on each training task, such that the model learns to perform new tasks quickly with few gradient steps. Examples include second-order algorithms like Model-Agnostic Meta-Learning (MAML) [47] or first-order methods such as FOMAML [48] and Reptile [49]. While MAML tries to optimize the efficiency of the learning algorithm such that it can customize the model's parameter with few gradient steps on test-task, Reptile tries to optimize the model such that it can generalize well on all training tasks. Reptile is also computationally more efficient than MAML and is easily extendable to the offline clinical setting.

Recently, meta-learning has shown great progress in few-shot image classification, reinforcement learning, and hyper-parameter tuning. In biomedical informatics, meta-learning has attracted interest recently (since 2018). Zhang *et al.* [50] used meta-learning for clinical risk prediction when there are several datasets, each with limited patient health records, and showed improvement over conventional data integration methods. Liu *et al.* [51] and Li *et al.* [52] studied a similar problem in rare disease prediction via learning shared initialization properties when we have several tasks. Jiang *et al.* [53] proposed a baseline procedure that aims to evaluate different

meta-learning algorithms in medical imaging settings. Sharma *et al.* [54] studied how to apply meta-learning for treatment effect estimation. Recently, meta-learning has been applied to online IRL problems, for instance, Gleave & Habryka [55] explore adversarial-IRL and Reptile on continuous control tasks while Xu *et al.* [56] leverage MAML to learn reward prior for grid-based environment. However, limited studies have explored meta-learning for recovering reward using IRL in offline settings.

Reptile : Reptile is a first-order gradient-based meta-learning algorithm and is similar to joint training. Reptile adapts the global model's parameters towards task-specific parameters by multiple gradient descent steps. The meta-parameter update step is given by:

$$\theta = \theta + \beta \frac{1}{N_{\mathcal{T}_i}} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} (\theta_i^{(k)} - \theta) \tag{2.12}$$

where $\theta_m^{(k)}$ is obtained after apply k steps of SGD on training task $\mathcal{T}_i$ and β is the meta-learning rate.

CHAPTER 3

TACKLING THE UNCERTAINTY IN OFFLINE POLICY LEARNING

In this chapter, we focus on offline clinical policy evaluation using inverse propensity scoring(IPS) based estimators in a contextual bandit scenario. As described in Chapter 1, directly testing policies in the real-world (online) is not possible because it can adversely impact a patient's well-being, thus, offline evaluation works as a good surrogate for evaluating policies which can be advanced to the next stage of a clinical trial. However, IPS-based approaches require clinician's propensity scores, which we typically don't have access to from the observational data. Thus, a common approach is to impute the propensity score by fitting a behaviour policy model $\hat{h}_0$ to predict the clinician's conditional probability of choosing a medical intervention, given the patient's physiological features. Such a model can take a parametric or nonparametric form, however, because of our limited knowledge of the behaviour policy h_0, we typically rely on universal function approximators such as neural networks. Given the limited scale of clinical datasets and the potential over-parameterization of neural networks, this introduces model uncertainty over the parameters, i.e., equally likely models with nearly equivalent accuracy can have diverse parameter distributions and thus, lead to varying final propensity score estimates. This chapter begins by defining a boostrapping-based approach to tackle the model uncertainty inherent in IPS-based approaches. We initially focus on off-policy evaluation problem and subsequently extend our framework to policy optimization. We conclude by accessing the quality of our proposed estimators over the clinical task of optimal dosage initialization of orally-administered anticoagulant drugs, warfarin and Heparin.

3.1 Bootstrapped Counterfactual Estimator

In this section, we first introduce the off-policy evaluation problem and then present our framework for bootstrapped-based evaluation. For each patient, the physiological feature x, policy treatment

recommendation $a \sim h(x; w)$, and a reward r was observed. The collection of these triplets $\{(x_i, a_i, r_i)\}_{i=1,\ldots,N}$ forms an offline dataset $\mathcal{D}$. The goal of offline evaluation is to estimate its expected reward $R(h)$ using the dataset $\mathcal{D}$ only. This problem is important as it represents a majority of scenarios arising during the evaluation of a clinical decision support system. Suppose we use our machine learning algorithm to build a new treatment policy h using EHRs obtained from a hospital, how do we ensure this policy is advantageous before deploying it in the clinic? Traditionally, to obtain an unbiased estimate of $R(h)$, we can use the inverse propensity scoring estimator as follows:

$$IPS : R(h) = \sum_{i=1}^{N} \frac{p_h(a_i|x_i)}{p_{h_0}(a_i|x_i)} r_i \tag{3.1}$$

where p_h is the propensity score of the policy h.

In clinical setting, however, $p_{h_0}(a|x)$, is not available, as physicians will not record the exact probability of them choosing a treatment. Modeling h_0 via supervised learning using a maximum likelihood-based approach, is possible, but introduces additional model-uncertainty: *There can be multiple versions of h_0 that are equally-likely and evaluate the same on a finite training set of N data points, however having totally different behaviors on other data points(test set).* To see this, imagine our policy is only a polynomial of degree '$N + 1$', and with N data points x, we can fit infinite number of functions $f(x, w)$ attaining zero error and satisfying the learning objective, thus, giving out a diverse range of model parameters w. The distribution over the model parameters $w \sim p(w)$ induces uncertainty in the learned function, characterized by $\hat{h_0} \sim U(f_w)$, subsequently leading to variance in the marginalized predictive probability distribution $p_{h_0}(a|x)$. If we consider more complex functions such as neural networks, the potential solutions for h_0 are even more.

Thus, we propose to reduce such model uncertainty in IPS-based estimators using a bootstrapping-based approach. We highlighted the various approaches for addressing model uncertainty in Chapter 2. By bootstrapping over multiple resamples of the dataset D and using model ensembling, we can reduce the uncertainty from learning h_0 and obtain a better estimate of the policy reward. In addition, we also obtain a confidence interval for the overall performance of the new policy h, and when we have multiple policies, we can choose not only based on the mean reward, but also the

tightness of the reward confidence interval as a criterion for the stability of the policy. We present our bootstrapped policy evaluation framework in Algorithm 1.

Algorithm 1 Bootstrapped Policy Evaluation

Require: The number of bootstrap evaluations B
Require: An off-policy dataset D
Require: A new policy h to evaluate
  Init $Result \leftarrow [\,]$
  for $b \leftarrow 1\text{to}B$ **do**
    Resample a dataset D_b from D
    Fit a propensity score model h_b
    Compute R_b, for example, using Eq. (3.1)
    Append R_b to the result array $Result$
  end for
  return Mean and standard deviation of $Result$

To tackle model uncertainty, we explore both deterministic NN ensembles and probabilistic BNN-based approaches. For simplicity, we discretize the clinician actions a and formulate the propensity score imputation problem as a multiclass classification problem. Specifically, we train a classifier on $(x_i, a_i) \in D$ and derive the propensity scores from softmax-layer probability scores. For NN ensembles, we train a deterministic MLE classifier by minimizing the cross-entropy loss and obtain the ideal hyperparameter values(network size, dropout probability) via 5-fold cross-validation.

$$\theta_{NN} \sim \arg\min_{\theta} L(x, a) = -\sum_{k} a_k log(f(x_k; \theta) + (1 - a_k) log(1 - f(x_k; \theta)) \tag{3.2}$$

Subsequently, we initialize B replica networks with varying random seeds and train using mini-batch stochastic gradient descent. During testing, we obtain B probability score predictions $p_{h_0}(a|x)$ corresponding to clinician's actions in the dataset.

For BNNs, we learn a posterior distribution over weights $p(w|x)$ and sample network during inference by sampling w. We train the BNN using Mean-field variational inference and MC-Dropout approaches, described in Chapter 1. In training BNN using variational inference, we leverage a scale mixture of two Gaussian distributions as prior over weights $p(w)$ with zero mean and tunable

standard deviation. We assume factorized weight posteriors, i.e., $q(w|\mu, \Sigma) = \prod_i q(w_i|\mu_i, \Sigma_i)$, where each weight w_i follows a normal distribution with learnable mean μ_i and diagonal covariance Σ_i. The BNN is trained using 'Bayes-By-Backprop' approach described in Chapter 2, which combines variational approximation with the re-parameterization trick. In MC-Dropout, we train a deterministic neural network using cross-entropy loss with dropout and L_2 regularization, and subsequently sample neural networks during inference using Monte-Carlo sampling i.e., randomly varying dropout masks. After bootstrapping B networks, we propose our counterfactual estimators based on the propensity score estimates obtained from those models:

$$\textit{IPS}_{avg} : \hat{R}(h) = \frac{1}{N} \sum_{i=1}^{N} \frac{p_h(a|x)}{\frac{1}{M} \sum_{m=1}^{M} p_{h_0}^m(a|x)} \tag{3.3}$$

$$\textit{IPS}_{inv} : \hat{R}(h) = \frac{1}{N} \frac{1}{M} \sum_{i=1}^{N} \sum_{m=1}^{M} \frac{p_h(a|x)}{p_{h_0}^m(a|x)} \tag{3.4}$$

where $p_{h_0}^b$ is the propensity score derived from b^{th} bootstrapped model.

The simplest approach is to average the propensity scores from bootstrapped models to reduce the variance of p_{h_0} leading to IPS$_{avg}$. The inverse estimator, IPS$_{inv}$, computes a harmonic mean of propensity scores and is equivalent to averaging the estimated rewards $\hat{R}(h)^m$ from each bootstrapped model. The average estimator can also be seen as a special case of multiple importance sampling and is equivalent to the Balance Heuristic estimator (Veach *et al.* [57]) when $N = KN_k$:

$$\hat{R} = \sum_{k=1}^{K} \sum_{i=1}^{N} \frac{p(x_{ik})}{\sum_{j=1}^{K} N_j p_0^j(x_{ik})} r_i \textit{ ; where } \sum_{k=1}^{K} N_k = N \tag{3.5}$$

3.2 Bootstrapped Counterfactual Learning

In the policy evaluation scenario, we have a collection of data and a clinical policy which we want to evaluate. However, the more common scenario in clinical settings is: *Given we have an offline dataset collected from a clinician's policy h_0, we would like to find a policy h that attains*

a high reward. The policy might be defined by weights belonging to a certain class of parametric functions such as a linear model or nonparametric method such as neural networks. Thus, the goal of counterfactual learning is to estimate a set of weights such that the expected reward $R(h(w))$ of the corresponding policy is maximized.

$$h_w^* = \arg\max_w R[a \sim h(x; w), x] \tag{3.6}$$

In the previous section, we defined how we can evaluate the expected reward $\hat{R}(h)$ for any h. Intuitively, if we have a finite set of h to choose from, by an exhaustive evaluation of all h, we can find the optimum h^*. However, here our focus is on policies defined using a neural network. This renders the optimal parameter search to be NP-complete with an infinite space of weights and corresponding network h, hence, we leverage stochastic gradient-based optimization. Using the IPS formulation, we can evaluate the gradient of policy h as

$$\nabla R(h) = \sum_{i=1}^{N} \frac{\nabla h(a_i|x_i)}{h_0(a_i|x_i)} r_i \tag{3.7}$$

When we model h with neural networks, we can automatically compute its gradient via back-propagation, so gradient-based optimization can be applied. Based on our bootstrapped evaluation algorithm, we define our bootstrapped counterfactual learning formulation as follows:

$$\textit{IPS}_{avg} : h_w^* = \arg\max_w \frac{1}{N} \sum_{i=1}^{N} \frac{p_h(a|x; w)}{\frac{1}{B} \sum_{b=1}^{B} p_{h_0}^b(a|x)} \tag{3.8}$$

$$\textit{IPS}_{inv} : h_w^* = \arg\max_w \frac{1}{N} \frac{1}{B} \sum_{i=1}^{N} \sum_{b=1}^{B} \frac{p_h(a|x; w)}{p_{h_0}^b(a|x)} \tag{3.9}$$

where B is the number of bootstrapped models. As in policy evaluation, we bootstrap models using deterministic NN ensembles and Bayesian approaches. We outline the bootstrapped learning algorithm in Algorithm 2. The benefit of this algorithm is that by adding bootstrapping, we reduce the variance in propensity scores while learning h_0 and optimize h against multiple possible pro-

posal distribution, thus improving its performance and stability. In addition, we can also populate the confidence intervals.

Algorithm 2 Bootstrapped Policy Optimization

Require: The number of bootstrap evaluations B
Require: An off-policy dataset D
  Init $h_0 \leftarrow [\,]$
  for $b \leftarrow 1$ to B **do**
    Resample a dataset D_b from D
    Fit a propensity score model h_b
    Append h_b to the collection array h_0
  end for
  Init h randomly
  while Not converged **do**
    Compute gradient as
    Update $h \leftarrow h - \alpha \nabla h$
  end while
  return h as the optimum h^*

3.3 Adversarial Bandit Learner

While, bootstrapping optimizes the learned policy against an ensemble of propensity models with varying propensity scores, the empirical reward $\hat{R}[h_w]$ cannot be used a performance certificate for the optimal true reward. This is because we are not explicitly optimizing for tackling the uncertainty due to the worst-case propensity model h_0^{worst}. To circumvent this limitation, we treat the propensity model parameter $P(\theta)$ distribution with skepticism and replace it with an uncertainty set $U_\epsilon(P)$ with ϵ controlling the size of uncertainty set. Given, that the propensity model h_0 is already constrained by the cross-entropy loss by virtue of behaviour policy imputation (the goal of h_0 is to model clinician's actions accurately), we can derive a distributionally robust counterfactual learning objective as follows:

$$\textit{IPS}_{adv} : h_w^* \sim \arg\max_{w} \min_{w_0} \frac{1}{N} \sum_{i=1}^{n} \frac{p_h(a|x;w)}{p_{\hat{h}_0}(a|x;w_0)} r_i + \lambda * CE(a_i, \hat{h}_0(x_i;w_0)) \tag{3.10}$$

where CE is the standard multiclass cross-entropy loss, λ is a hyperparameter which define trade-off between accurate behaviour policy imputation (2^{nd} term) vs reward maximization (1^{st} term). Consequently, we propose an adversarial policy learning framework (IPS$_{adv}$) with an iterative optimization scheme, which simultaneously optimizes networks corresponding to h and h_0. h is optimized to maximize reward against the worst-case possible h_0, which acts in an adversarial manner to h : the goal of h_0 is to learn a classification model to impute clinician's action probabilities accurately and at the same time, reduce the reward achieved by learned policy h. We present the pseudocode of our adversarial policy optimization algorithm in Algorithm 3.

Algorithm 3 Adversarial Policy Optimization

Require: An off-policy dataset D
  Initialize policy network $h(w)$ randomly
  Initialize propensity model $h_0(w_0)$ by training for $\mathcal{K}$ steps over (x_i, a_i) with CE loss
  while Not converged **do**
    Train h_0:
    Sample minibatch of *m* examples $(x_i, a_i, r_i)_{i=\{1,2,...,m\}}$ from dataset D
    Update h_0 by SGD: $\nabla_{w_0} \sum_{i=1}^{m} \frac{h(a_i|x_i,w)}{h_0(a_i|x_i,w_0)} r_i + \lambda * CE(\hat{a_i}, a)$ where $\hat{a_i} = h_0(x_i; w_2)$
    Train h_1:
    Sample minibatch of *m* examples $(x_i, a_i, r_i)_{i=\{1,2,...,m\}}$ from dataset D
    Update h by SGD: $\nabla_{w} \sum_{i=1}^{m} \frac{h(a_i|x_i,w)}{h_0(a_i|x_i,w_0)} r_i$
  end while
  return h as the optimum h^*

3.4 Experiments

We evaluate the efficacy of our proposed frameworks on a synthetic bandit dataset as well as on the clinical task of dosing initialization for orally administered anticoagulant drugs. Anticoagulants are blood thinners administered to remove blood clots and their dosage during treatment initiation varies significantly across patients. Moreover, incorrect dosing can have significant side effects, thus making it a challenging clinical setting for treatment recommendation systems. We consider two commonly used anticoagulants in hospitals, namely, warfarin and heparin. We use two freely available electronic health records databases to derive the clinical bandit datasets: 1) PharmGKB

(Consortium 2009) [58] for warfarin dosing and 2) Multiparameter Intelligent Monitoring in Intensive Care (MIMIC-III v1.4) [59] for heparin dosing. For Warfarin dosing, we have access to counterfactuals and artificially simulate the logging policy to derive semi-synthetic bandit dataset. For heparin dosing, we derive a true real-world bandit dataset without access to counterfactuals.

3.4.1 Datasets

Non-clinical Datasets

Synthetic Dataset : We simulate a synthetic dataset with 3 discrete actions by sampling patient context and clinician action propensities from multivariate standard normal distributions. Similar to context x, the actions a are represented using a 10-dimensional representation and reward is computed via an outer-product of x and a.

$$\begin{aligned} x_i &\sim N(\mu = \mathbf{0}, \Sigma = \mathbf{I}_{10x10}) \; where \, x_i \in \mathcal{R}^{10} \\ p_0(a_i|x_i) &\sim N(\mu = \mathbf{0}, \Sigma = \mathbf{I}_{3x3}) \; where \, p_0 \in \mathcal{R}^{3} \end{aligned} \tag{3.11}$$

UCI Dataset : We select 3 multiclass classification datasets from UCI repository previously used for off-policy bandit evaluation[60] and convert them to contextual bandits by choosing actions derived from a multiclass logistic regression policy trained on 5% of the dataset, similar to Dudik *et al.* [60].

Clinical Datasets

Warfarin Dosing (Semi-synthetic) Using the PharmGKB [61] dataset, we develop a case study to evaluate our framework on Warfarin dosing. Warfarin dosing is concerned with determining the correct dosage of the blood anticoagulant drug for a heart patient. The dataset includes patient information (demographics, physiological, and genotype features) with final ideal therapeutic dosage. Warfarin's administration needs to be monitored closely, since incorrect dosage can lead to adverse side effects such as heart attacks. The therapeutic dosage varies widely across patients due to different contextual features. Physicians typically prescribe an initial dose which is adjusted

according to the patientś response. Previous work [62] on predicting dosage policies using bandits, discretize the dosage into three categories, 'low'($<$ 21 mg/wk), 'medium' ($\geq$ 21 mg/wk, $\leq$ 49 mg/wk) or 'high' ($>$ 49 mg/wk). Although, recently warfarin dosing has been approached in the continuous domain which allows for finer adjustments, we focus on tackling uncertainty in propensity score estimation under discrete dosage setting to keep the overall formulation simple. With dosage discretization, Warfarin dataset was converted to a supervised classification dataset $D = (x_i, y_i)_{n_i=1}$ with access to treatment counterfactuals. This provides us with the ground-truth action for each patient. Since the dataset is supervised, we simulate a contextual bandit environment by simulating clinician's policy and using a custom reward function. We follow the Supervised $\rightarrow$ Bandit conversion approach highlighted in [30, 63, 24] and simulate expert (clinicianś) behavior using stochastic logging policy to sample $y_i^* = h(*|x_i)$ with reward defined based on the match between ground-truth and sampled actions, $r_i = I(y_i = y_i^*)$. We simulate the following stochastic logging policies with 3 and 5 discrete dosage levels(policy actions). These policies are also referred to as expert policies.

1. LR: We follow the experimental design specified in [24] and use multi-class logistic regression model trained on 5% data, as logging policy. For different simulations, we randomly sample 5% data from our training set and fit a multi-class logistic regression model to obtain weight vector w_{lr}. To introduce further stochasticity, we randomly perturb w_{lr} using random noise drawn from a standard normal distribution $u \sim N[0, 1]$.

2. PHARMA: We adopt the clinical policies (WPGA, WCGA)[61] as our base deterministic policies (h_1, h_2). Both WPGA and WCGA are clinically motivated linear models with WPGA incorporating genotype features to improve over WCGA. Our aim was to emulate clinicians using WPGA or WCGA for dosage recommendation and combine them in a stochastic manner. Motivated from the friendly softnening approach proposed by Farajtabar *et al.* [63], we transform the deterministic policy into stochastic policy by drawing

actions $a_i = h_0(x_i)$ from a mixture of these models with equal probability.

$$a_i = \begin{cases} h_1(x_i), & r_i <= 0.5 \\ h_2(x_i), & otherwise \end{cases} \tag{3.12}$$

where $r_i \in [0, 1]$ is a random number for patient x_i.

Heparin Dosing (True Bandit) : One of the most commonly used anticoagulant medications in hospitals and ICUs is Heparin. The dosage of intravenous unfractionated heparin is commonly based on the patient's weight, as per most clinical practice guidelines [64]. Such weight-based approach alone may result in improper dosage for obese patients. Although some works have recommended using an adjusted body weight [65], in practice, activated partial thromboplastin time (aPTT) is a good indicator of blood coagulation level. There is significant variation in the guidelines for the initial loading dose of heparin, the rate of dosage, and the time measurement intervals of aPTT. A higher aPTT level reflects slow blood clotting, whereas a low level indicates fast clotting. Samples of blood are usually taken every 4-6 hours to measure the levels of aPTT, and the result of anticoagulation therapy is analysed by observing whether aPTT reaches the therapeutic window timely. Typically, aPTT between 60s and 100s is considered therapeutic with aPTT $>$ 100s being supra-therapeutic and aPTT $<$ 60s being subtherapeutic. While machine learning techniques have tried to develop the ability to provide clinical decision support for heparin dosing, the high patient variability has led to the underperformance of multinomial logistic regression-based models [66]. Here, we formulate heparin dosing as an offline bandit problem by considering the aPTT after 6 hours of dosage initialization as the reward outcome. We discretize the Heparin dosages into 3 categories(actions) 'low'($<$ 10 mg/wk), 'medium' ($\geq$ 10 mg/wk, $\leq$ 15 mg/wk) or 'high' ($>$ 15 mg/wk). The outcome of interest was the aPTT value 6 hours after initial heparin infusion and the

rewards were defined as:

$$r_i = \begin{cases} 1, & 60s <= aPTT_{t=6hrs} <= 100s \\ 0, & otherwise \end{cases} \tag{3.13}$$

Patient demographics and physiological features of interest used to define the context included : age, height, weight, ethnicity, gender, obesity, creatinine concentration, SOFA score, type of ICU admission, end-stage-renal-disease(ESRD) and pulmonary embolism. These features contribute collectively to patient's response to Heparin dose, for instance, creatinine concentration reflects the filtration function of glomeruli and together with ESRD serves as an indicator of renal function. We selected these features in line with the previous studies [66, 67], with most of the features being statistically significant for predicting aPTT outcomes. To create the patient cohort, we follow the scheme proposed by Ghassemi *et al.* [66]. A total of 4,761 adult patients, who had undergone Heparin dosing during ICU stay, are extracted from MIMIC-III database. We included only those patients with aPTT measurements 6 hours after the initial Heparin infusion, reducing the cohort size to 2,981 patients. Further, some patients had missing covariates and removing these patients, we obtained 2,136 patients. Lastly, we removed patients who were transferred from another hospital,since their Heparin infusion might have started prior to the ICU admission and we have limited knowledge of medical interventions taken before transfer. Our final cohort comprises of 1,378 patients.

3.4.2 Baselines

We consider two popular off-policy estimators: IPS [28] and SNIPS [29] and use the propensity score imputed from a single neural network with vanilla IPS/SNIPS formulation as the baseline. The single neural network can be a deterministic neural network in case of NN ensemble or a network obtained by sampling once from the posterior weight distribution of Bayesian NN.

Our logging policy imputation model is a single hidden-layer perceptron network with ReLU activation units. We establish baseline estimators by selecting one of bootstrapped models as

propensity score estimator. We denote these baseline estimators as Vanilla SNIPS/Vanilla IPS. To bootstrap deterministic NN model, we initialize the model weights randomly and use dropout (0.25) for fitting the models. To train BNN with variational inference framework, we follow 'Bayes-by-Backpropagation' approach [14] assuming a scale mixture of two Gaussian densities as the prior distribution for weights $w_{h_0} \sim 0.5N(0, 0.5) + 0.5N(0, 0.002)$. The network configurations are different for Warfarin dosing (hidden units = 20) and Heparin dosing (hidden units = 40). We use the Adam optimizer [68] (β_1 = 0.999, β_2 = 0.9) with a learning rate of $1e^{-3}$ and mini-batch size of 50 for both datasets, and use progressive validation to detect convergence. We determined the optimal training hyperparameters using 5-fold cross-validation on both datasets. For adversarial IPS learner, we determined λ = 1 to be optimal after experimenting with multiple values (0.5, 1, 1.5, 2).

3.4.3 Policy Evaluation

In this experiment, we evaluate whether bootstrapping-based framework leads to more confident reward estimation of a custom clinical policy. Here, we leverage the synthetic non-clinical dataset and semi-synthetic Warfarin dosing dataset, since they allow comparing estimated policy reward with the ground-truth reward (estimated from counterfactuals). We perform 20 simulations and report the root-mean squared error (RMSE = $E[\hat{R}(h) - R(h)]^2$) of our proposed estimators and baselines over these 20 sampled datasets, where $R(h)$ is the ground-truth reward. We follow the following methodology of Dudik *et al.* [30] during each simulation to derive the semi-synthetic bandit dataset for Warfarin dosing

1. For each logging policy, we create a partially-labeled bandit dataset by applying the transformations described in section 3.4.1.

2. We randomly subsample 70% of the synthetic-bandit dataset as our evaluation dataset and divide it into train/validation sets in 80/20 ratio for fitting the propensity model.

3. We obtain the evaluation policy h by training a multiclass logistic regression model on full

classification dataset and define its classification accuracy as the ground truth reward $R(h)$.

4. We bootstrap 10 models ($\hat{h}_0^b, b \in \{1, 2, ...10\}$) for imputing logging policy propensity scores as described in 3.1. For ensemble approach, we initialize 10 models with seeds in multiples of 2. In variational inference, we train 10 BNNs and sample 10 models, one from each of the 10 weight distributions. For MC-Dropout model, we apply dropout randomly to sample networks during inference. We also resample data while bootstrapping a NN ensemble model or training a new BNN.

Results

We present the policy evaluation results for SNIPS and IPS estimators on Warfarin bandit dataset (LR and PHARMA policies) in Table 3.1 and Table 3.2 respectively. We highlight both bias and variance of the estimated policy rewards. Using bootstrapping leads to significantly lower bias and variance, even in the case of SNIPS which typically has lower variance due to weight normalization. Comparing the two bootstrap-based estimators, we find that $average$ propensity score estimator is able to achieve lower policy evaluation bias compared to the $inverse$ estimator. We also observe that NN ensemble and MC-Dropout based networks lead to slightly better variance reduction compared to BNNs, which is in line with the uncertainty reduction results observed in [33]. In the case of NN ensemble, we also evaluate the impact of bootstrap count on the reduction in bias and variance of SNIPS-based reward estimators (Figure 3.1). We observe that an ensemble of 5 neural networks performs sufficiently well in reducing both the variance and bias. As the number of bootstrapped models increases, the bias and variance of SNIPS$_{inv}$ and SNIPS$_{avg}$ estimators reduce significantly with SNIPS$_{avg}$ achieving lower bias and variance. Thus, bootstrapping multiple models allows to sample from multiple proposal distributions and avoids the situation wherein a single propensity score model suffers from very low probability coverage over certain regions of the action space.

Table 3.1: Policy Evaluation: Mean Average Error ($\mu \pm \sigma$) of SNIPS-based estimators

Dataset	**Logging Policy**	**SNIPS(h_0^{true})**	$\hat{h}_0$ **- NN**			$\hat{h}_0$ **- BNN**			
			Baseline SNIPS	**NN Ensemble**		**Baseline SNIPS**	**Variational Inf.**		**MC-Dropout**
				SNIPS_{inv}	SNIPS_{avg}		SNIPS_{inv}	SNIPS_{avg}	SNIPS_{avg}
Synthetic	Gaussian (3 actions)	215.1 ± 209.3	3.7 ± 11.1	4.5 ± 9.7	$\mathbf{2.3 \pm 6.2}$	11.5 ± 26.4	$\mathbf{5.0 \pm 5.0}$	6.6 ± 14.2	$\mathbf{2.7 \pm 6.8}$
Warfarin	LR (3 actions)	6.7 ± 0.7	16.7 ± 18.5	9.3 ± 3.2	$\mathbf{7.7 \pm 0.7}$	27.6 ± 18.2	30.0 ± 7.3	$\mathbf{8.0 \pm 1.0}$	$\mathbf{7.0 \pm 0.7}$
	LR (5 actions)	10.0 ± 0.6	11.6 ± 6.7	9.3 ± 2.3	$\mathbf{10.1 \pm 0.9}$	19.4 ± 11.8	17.7 ± 8.0	$\mathbf{9.4 \pm 1.5}$	$\mathbf{10.3 \pm 0.6}$
	PHARMA (3 actions)	21.2 ± 0.4	19.7 ± 16.7	17.8 ± 5.2	$\mathbf{15.6 \pm 1.4}$	17.2 ± 12.0	18.3 ± 4.8	$\mathbf{15.4 \pm 1.1}$	$\mathbf{12.0 \pm 0.7}$
	PHARMA (5 actions)	12.2 ± 1.7	15.2 ± 9.1	14.4 ± 3.1	$\mathbf{12.9 \pm 0.9}$	9.7 ± 3.1	13.6 ± 3.4	$\mathbf{12.3 \pm 1.0}$	$\mathbf{11.6 \pm 0.6}$

Table 3.2: Policy Evaluation: Mean Average Error ($\mu \pm \sigma$) of IPS-based estimators

Dataset	**Logging Policy**	**IPS(h_0^{true})**	$\hat{h}_0$ **- NN**			$\hat{h}_0$ **- BNN**			
			Baseline IPS	**NN Ensemble**		**Baseline IPS**	**Variational Inf.**		**MC-Dropout**
				IPS_{inv}	IPS_{avg}		IPS_{inv}	IPS_{avg}	IPS_{avg}
Synthetic	Gaussian (3 actions)	482.9 ± 505.4	3.7 ± 11.1	4.6 ± 9.6	$\mathbf{2.4 \pm 6.1}$	1172.2 ± 3542.1	616.5 ± 1138.5	$\mathbf{7.2 \pm 9.2}$	71.9 ± 11.9
Warfarin	LR (3 actions)	28.3 ± 0.9	47.7 ± 0.8	47.4 ± 1.5	$\mathbf{47.8 \pm 0.6}$	1504.1 ± 6105.1	215.0 ± 85.4	45.8 ± 0.8	$\mathbf{12.5 \pm 1.1}$
	LR (5 actions)	41.9 ± 1.0	66.4 ± 38.0	55.8 ± 12.6	$\mathbf{63.0 \pm 1.0}$	1734.5 ± 5800.5	378.0 ± 120.1	61.5 ± 1.8	$\mathbf{17.8 \pm 1.3}$
	PHARMA (3 actions)	17.3 ± 2.0	13.7 ± 9.1	14.4 ± 5.7	$\mathbf{20.7 \pm 1.6}$	18.4 ± 24.8	43.9 ± 21.2	19.1 ± 1.7	$\mathbf{4.0 \pm 1.5}$
	PHARMA (5 actions)	13.0 ± 3.6	14.3 ± 6.2	11.5 ± 5.7	$\mathbf{19.9 \pm 1.6}$	5.9 ± 6.8	13.0 ± 8.3	9.9 ± 2.4	$\mathbf{12.1 \pm 1.1}$

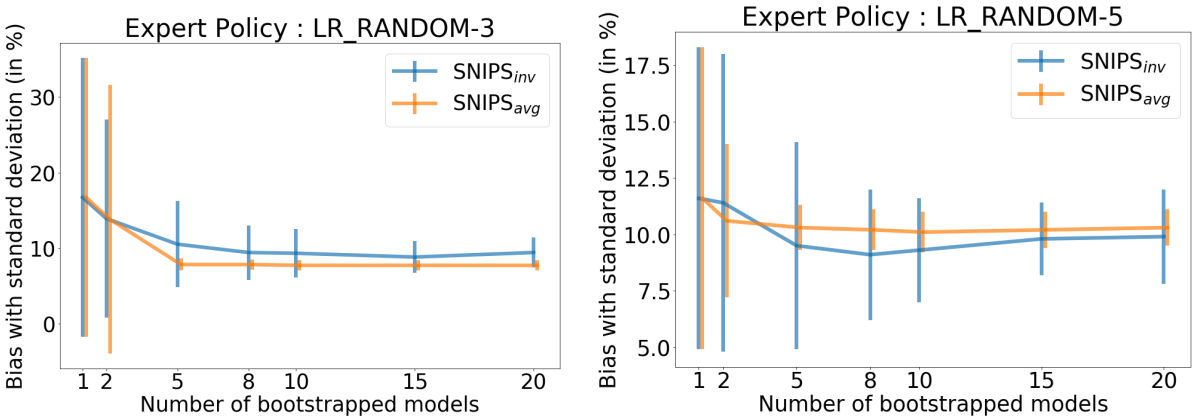


Figure 3.1: Policy Evaluation: Impact of bootstrap count on mean average error and standard deviation of reward estimates ($\hat{h}_{NN}$)

3.4.4 Policy Learning

In this experiment, we use the bootstrapping and adversarial learning frameworks to learn optimal policies with maximum reward. Based on the performance of our bootstrapped estimators for policy evaluation, we expect that addressing uncertainty with bootstrapping and adversarial formulations will translate to learning better policies. We perform 10 simulations and learn the dosing policy h using IPS-based loss formulation and minibatch stochastic gradient descent. To evaluate our frameworks, we report the mean reward achieved by our learned policies along with their variance ($\mu_{R(h)} \pm \sigma_{R(h)}$). We follow the following steps during each simulation:

1. We randomly split the data into training (70%) and test (30%) sets.

2. For each logging policy type, we obtain partially labelled semi-synthetic bandit dataset for Warfarin dosing by applying the transformations described in section 3.4.1. Moreover, we also consider the Heparin dosing dataset which is a true bandit dataset and allows us to evaluate policy learning on non-simulated real-world clinical setting.

3. Bootstrapping: We bootstrap 10 models for imputing the logging policy. By incorporating the average and inverse learning formulations described in section 4 into IPS and SNIPS estimators, we learn optimum policies h_{avg} and h_{inv} respectively.

4. Adversarial Bandit Learner: We train the models h_0 and h alternately using the IPS$_{adv}$ loss formulation. Before initiating the adversarial training, we initialize propensity model h_0 by training it for 4 epochs on the bandit dataset. This assures that h_0 initializes with parameters not widely different from the optimal propensity model, which stabilizes the subsequent adversarial learning process. We train both networks alternately for 100 epochs with a learning rate of 0.001.

Evaluation Setup

For Warfarin dosing, we execute the learned policy on the test dataset and compare the predicted actions with ground truth dosage actions from the full classification dataset. For Heparin dosing, since we have access to a real-world bandit dataset, we do not have access to counterfactuals, i.e., the optimal ground-truth dosage for each patient. Hence, we leverage the SNIPS estimator for evaluating the performance of our learned policies, given that offline SNIPS estimates have been shown to be highly correlated to the true (online) performance for a wide range of policies by Zenati *et al.* [69]. In our evaluation experiments (Table 3.1), we found SNIPS to have both lower variance and bias compared to IPS.

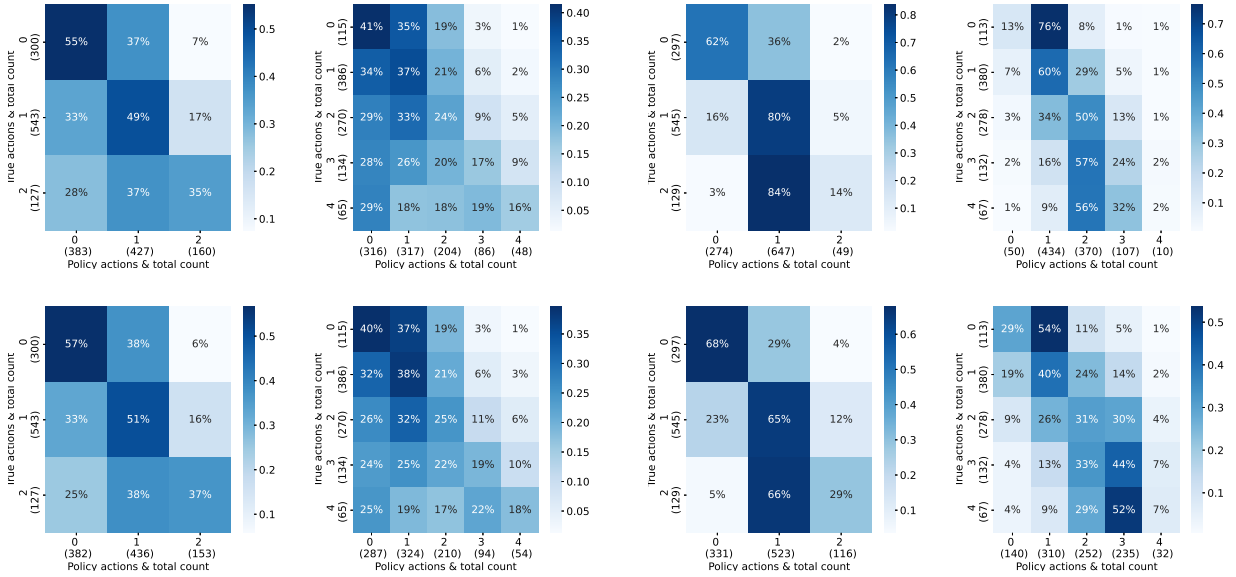
Results

In Table 3.3, we highlight the results of policy learning on clinical datasets. Using bootstrapping leads to improved policy learning both on semi-synthetic data (LR logging policy in Warfarin dosing) as well as true-bandit data (Heparin dosing). Moreover, we observe that IPS_{inv} outperforms IPS_{avg} across multiple datasets. Consistent with the policy evaluation results, we find that NN ensemble is more effective at reducing uncertainty than BNNs. An interesting observation is that bootstrapping leads to lower rewards for warfarin datasets simulated using PHARMA logging policy. However, on further analysis, we find that this is because the PHARMA policy actions are heavily biased towards certain actions (dosage 1 in 3-action case and dosages 1 & 2 in 5-action case). This bias in the simulated actions of the logging policy leads to the learned policy being substantially biased towards action '1'. However, the bootstrapped framework leads to a policy which is less-biased and more balanced in its actions, although it achieves a lower overall reward. As observed in figure 3.2, policy learning using IPS_{inv} achieves higher accuracy for infrequent actions (dosages 0 & 2 in 3-action and dosages 3, 4 & 5 in 5-action scenarios).

In the case of heparin dosing, all learned policies outperform the actual clinician policy which achieves a reward of 0.27 (based on actual aPTT outcomes). This highlights that formulating heparin dosing as a bandit problem is a promising approach to develop dosage recommendation

Table 3.3: Policy Learning: Rewards ($\mu \pm \sigma$) of clinical policies learned using bootstrapped IPS frameworks (IPS_{inv} , IPS_{avg})

Dataset	**Clinician Policy/ Logging Policy**	$\hat{h}_0$ **- NN**			$\hat{h}_0$ **- BNN**
		Vanilla IPS (Single NN)	**NN Ensemble**		**Varational Inf.**
			IPS_{inv}	IPS_{avg}	IPS_{avg}
Synthetic	None	2.169 ± 1.048	2.166 ± 1.038	2.182 ± 1.032	$\mathbf{2.206 \pm 1.005}$
Warfarin (semi-synthetic)	LR (3 actions)	0.493 ± 0.040	$\mathbf{0.506 \pm 0.037}$	0.492 ± 0.040	0.500 ± 0.041
	LR (5 actions)	0.457 ± 0.034	$\mathbf{0.469 \pm 0.033}$	0.458 ± 0.032	0.462 ± 0.030
	PHARMA (3 actions)	0.656 ± 0.017	0.610 ± 0.028	0.640 ± 0.018	0.661 ± 0.015
	PHARMA (5 actions)	0.596 ± 0.020	0.525 ± 0.022	0.556 ± 0.020	0.578 ± 0.018
Heparin (true bandit)	Unknown	0.295 ± 0.043	$\mathbf{0.317 \pm 0.033}$	0.311 ± 0.043	0.295 ± 0.051



LR(3 actions) LR(5 actions) PHARMA(3 actions) PHARMA (5 actions)

Figure 3.2: Policy Learning: Comparison of true clinical actions and policy actions for Vanilla IPS (1^{st} row) and IPS_{inv} (2^{nd} row). Each cell represents predicted count as a percentage of true action total count.

Table 3.4: Policy Learning: Rewards ($\mu \pm \sigma$) of policies learned on UCI datasets using IPS_{adv}

Methods	SatImage	Letter	OptDigits
# train samples	3858	12000	3372
# actions	6	26	10
Vanilla IPS	0.859 ± 0.010	0.520 ± 0.057	0.935 ± 0.034
Adversarial IPS_{adv}	0.859 ± 0.009	$\mathbf{0.666 \pm 0.027}$	$\mathbf{0.944 \pm 0.009}$

Table 3.5: Policy Learning: Comparison of rewards ($\mu \pm \sigma$) of clinical policies learned using our proposed frameworks (IPS$_{inv}$, IPS$_{avg}$ and IPS$_{adv}$)

Dataset	**Clinician Policy/ Logging Policy**	$\hat{h}_0$ **- NN**			
		Vanilla IPS	**NN Ensemble** IPS$_{inv}$	**NN Ensemble** IPS$_{avg}$	**Adversarial** IPS$_{adv}$
Warfarin (semi-synthetic)	LR (3 actions)	0.493 ± 0.040	0.506 ± 0.037	0.492 ± 0.040	$\mathbf{0.515 \pm 0.038}$
	LR (5 actions)	0.457 ± 0.034	0.469 ± 0.033	0.458 ± 0.032	$\mathbf{0.471 \pm 0.030}$
	PHARMA (3 actions)	0.656 ± 0.017	0.610 ± 0.028	0.640 ± 0.018	$\mathbf{0.657 \pm 0.019}$
	PHARMA (5 actions)	0.596 ± 0.020	0.525 ± 0.022	0.556 ± 0.020	$\mathbf{0.626 \pm 0.012}$
Heparin (true bandit)	Unknown	0.295 ± 0.043	0.317 ± 0.033	0.311 ± 0.043	0.306 ± 0.035

systems in addition to the traditional prediction-driven supervised learning approaches, which typically are more effective in replicating the clinician's policy. In Table 3.5, we highlight the results of our adversarial learning framework. We observe that explicitly optimizing policy h for the worst-case propensity-scoring model h_0 leads to more optimal policy learning with learned h achieving higher rewards compared to bootstrapping. In Table 3.4, we also observe that adversarial IPS achieves higher rewards and explicitly optimizing for the worst-case propensity scoring model acts as a regularization , leading to significantly lower variance in nonclinical settings involving UCI datasets.

CHAPTER 4

IMPROVING POLICY GENERALIZATION WITH MULTITASK META-LEARNING

This chapter explores the use of multitask formulation and meta-learning based adaptation in reinforcement learning (RL) to learn generalizable clinical policies in dynamic treatment regime. Here our focus is on learning from demonstrations, i.e., we have access to a fixed set of expert trajectories generated by the clinician following a near-optimal policy with unknown underlying reward function. Our goal is to imitate the clinician in this offline setting by recovering the reward function from the demonstrations using Inverse Reinforcement Learning (IRL). As highlighted in Chapter 2, clinical policy learning can be formulated as a multitask problem with clinician demonstrations coming from multiple intrinsic reward functions, each contextualized to a specific patient group. To achieve rapid generalizability across multiple reward functions, we leverage meta-learning technique with IRL. We present the key components of our meta-IRL framework and evaluate its efficacy on the clinical problem of sepsis management in ICU patients, comparing it with single-task IRL and behaviour-cloning baselines.

4.1 Meta-IRL Framework

Multitask Formulation

We assume that we have a collection of tasks $\{\mathcal{D}^1, ..., \mathcal{D}^M\}$ over which we want our clinical policy to generalize, and we have access to the task distribution $\mathbb{P}(\mathcal{D})$ from which we sample the tasks. Within each learning episode, a set of expert trajectories $\mathcal{D}^j = (\tau_1^j, \tau_2^j, ..., \tau_K^j)$ from an unknown new MDP environment are drawn, where each trajectory is a sequence of state-action pairs $\{(s_1^j, a_1^j), (s_2^j, a_2^j), ...(s_T^j, a_T^j)\}$. The goal of our learning algorithm is to learn a linear reward function $R = w^T \cdot \mathcal{F}(s, a)$, where $\mathcal{F}$ is a feature map, that enables the corresponding policy to imitate the clinician's actions on all tasks: h^* = $\arg\max \mathbb{E}_{\mathcal{D}\sim\mathbb{P}} R(h; \mathcal{D}^j)$

We leverage an off-policy variant of max-margin IRL formulation proposed by Abbeel and Ng [39] to recover reward function and learn clinical policy. Specifically, we leverage Batch-IRL approach proposed by Lee *et al.* [40] which leverages deep network-based feature expectation instead of directly sampling from the expert's state space. In a multi-task IRL setting, the reward $R(\mathcal{D}^j)$ varies with each task(patient group) and a simple approach would be to apply single-task IRL to each task separately. This works well when the tasks are unrelated to each other, however, in clinical data, patients with different contexts, but suffering from a particular disease, have common disease dynamics with a similar structure of underlying reward function for recommending treatment. By leveraging the shared transferable knowledge between tasks, we can influence the parameter trajectories of our reward and policy networks to enable smooth convergence with fewer clinician's trajectories. Moreover, by exposing our reward model to multiple tasks with sufficient training on each task, we can create a meta-learner which efficiently infers the contextual reward function and learns generalizable clinical policies. To leverage shared knowledge for quick task adaptation of reward and policy networks, we setup a set of parameters θ^0 that are shared among all tasks. We meta-learn the shared parameter θ^0 with a procedure $\mathcal{B}^2$ using all patient groups in training data $\{\mathcal{D}^1, ..., \mathcal{D}^M\}$. We selected Reptile [49] as the basis for our meta-learning procedure due to its computational efficiency and ability to extend to offline settings, where online gradient adaptation-based methods cannot be applied. To capture the different characteristics of each task $\mathcal{D}^j$, we define a procedure $\mathcal{B}^2$ that uses θ^0 and D^j to effectively leverage information across task trajectories and derive task-specific parameters θ^j.

$$\begin{aligned} \text{Task-specific Learning: } \mathcal{B}^2 : \theta_i^j &= \text{Batch-IRL}(\theta^0, \mathcal{D}^j) \\ \text{Task Adaptation } \mathcal{B}^1 : \theta^0 &= \textit{MetaLearn}(\theta^0, \theta_i^j) \end{aligned} \tag{4.1}$$

Note that here we cannot directly assign a separate θ^j to each task because, in practice, during testing, the tasks can come from distributions not in the M training tasks. Before formally introducing our IRL algorithm, we introduce the key components of our Meta-IRL framework:

Task Creation

In our model, a task represents a group of patients with similar contextual features. We define the context based on 7 static features at the time of ICU admission: gender, age, weight, GCS, Elixhauser comorbidity score, whether the patient was mechanically ventilated at $t = 0$ and patient readmission. The tasks are derived by clustering the patients into 20 groups using K-Means algorithm [70] on the contextual features.

IRL Components

Since our IRL framework is based on Batch-IRL framework, we leverage their feature expectation network (DSFN), reward formulation, and warm-start network(TRIL) to initialize our algorithm with near-optimal policy. However, instead of relying on Deep Q-network for policy optimization, we leverage Batch-Constrained Q-Learning [71] due to its superior performance in offline settings.

1. Deep-Successor Feature Expectation Network (DSFN): As discussed in Chapter 2, matching feature expectations between expert and learned policies is a key optimization metric for max-margin IRL. Batch-IRL approaches feature expectation as a policy evaluation problem and parameterizes the feature expectation estimator using a deep neural network. DSFN has been inspired from the linear least-squares approach of LSTD and uses a training procedure analogous to that of Deep Q-learning [72]. Given the expert trajectories $\mathcal{D}^e = (s_i, a_i, s_{i+1})$ where $i \in \{1, 2, ...N\}$ sampled from unknown clinician policy h_e, DSFN aims to learn a feature expectation estimator network parameterized by θ, $\mu_h(s, a; \theta)$, for a learned policy h such that $\mu_h(s, a; \theta) \approx \mu_e(s, a)$. The network is trained using MSE-Loss based on TD-errors derived from Bellman equation [73].

$$\begin{aligned} \text{Estimate } \mathbf{u}(s_j, a_j) &= \mathcal{F}(s_j, a_j) + \gamma\, \mu_h(s_{j+1}, a \sim h(s_{j+1}); \theta) \\ \text{DSFN Loss } \mathcal{L}(\theta) &= \sum_{j=1}^{N} \Big[||\mathbf{u}(s_j, a_j) - \mu_h(s_j, a_j; \theta)||^2 \Big] \end{aligned} \tag{4.2}$$

where $\mathcal{F} \in \mathcal{R}^d$ is a feature map defined for state-action pairs over $S \times A$.

2. Reward Function: To reduce the dimension of the problem and keep the formulation simple, the reward function is assumed to be linear in feature $\mathcal{F}$ over the state-action pairs: $R(s, a; w) = w^T \cdot \mathcal{F}(s, a)$. The weights for the reward function are learned by solving the max-margin QP:

$$\begin{aligned} w_i &= min_{w \in \mathcal{R}^d} ||w||^2 \\ s.t.\, w_i^T \mu_j &\leq w^T \mu_e + 1 \quad j \in \{1, 2, 3, ..., i-1\} \end{aligned} \tag{4.3}$$

3. Warm-Start Network: Online IRL typically starts with a random policy. However, since the feature expectation is computed basis actions sampled from learned policy h, in offline scenario, if h significantly differs from expert policy h_e, the action support could be nearly disjoint. Since, it is impossible to collect additional transitions in offline setting, the gradient updates for μ_h could be heavily-biased. Thus, Batch-IRL initializes IRL with a near-optimal policy which has decent overlap with expert policy in the action space $\mathcal{A}$. The warm-start policy is learned using regularized imitation learning named TRIL, which leverages a two-channel network to jointly predict clinician's action as well as the next state transition. Moreover, the intermediate shared layers of the TRIL network are used as feature encoders to derive corresponding feature representations $\mathcal{F}(s, a)$ in IRL.

$$\text{TRIL Loss} : \mathcal{L}(\theta_0, \theta_{\mathcal{T}}) = \mathcal{L}_{CE}[= \Big[a_j, h_0(s; \theta_0)\Big] + \lambda\, \mathcal{L}_{MSE}\Big[T(s_j, a_j; \theta_{\mathcal{T}}), s_{j+1}\Big] \tag{4.4}$$

where $\mathcal{L}_{CE}$ is the cross-entropy loss for predicting clinician action, $\mathcal{L}_{MSE}$ is the mean-squared error for next-state prediction, given current state and clinician action; and λ is the regularization parameter.

4. Policy Network: In our model, we use batch-constrained Q-learning (BCQ) as a MDP solver for policy optimization. Fujimoto *et al.* [71] showed that off-policy deep Q-learning fails due

to extrapolation error i.e. state-action pairs outside the expert trajectories can have arbitrarily inaccurate Q-values. This error is propagated via temporal-difference(TD) update of off-policy Q-learning, thus causing extreme overestimation and adversely affecting the training. While exploring actions would correct for such values in an online setting, it is impossible to do so in an offline setting. Hence, during TD-update, BCQ constrains the action space to eliminate actions which are unlikely to be present in the provided expert data. We leverage a discrete-action version of BCQ [74] which uses a state-conditioned generative model to sample policy actions during TD-update. We specify the learned policy network by $h(\phi)$, where ϕ denotes the Q-learning network parameters.

In our Meta-IRL framework, we implement the idea of jointly adapting both the policy network $h(\phi)$ and the feature expectation network $\mu(\theta)$ along with the reward weights w, by applying gradient-based meta-update derived from REPTILE. This is because we want to obtain an optimal global policy h^* which imitates clinician accurately for all tasks $\mathcal{D}^j$ and generalizes well during test-time by relying on contextual reward $R(w)$ which is also meta-learned. Thus, our shared meta-learnable parameter space is given by $\theta^0 = \{\theta, \phi, w\}$. The pseudocode for Meta-IRL is presented in Algorithm 4.

4.2 Sepsis Management

In this section, we evaluate our algorithm on real-world medical task of sepsis management in ICU patients. Sepsis is a leading cause of cost and mortality in ICU [75]. Sepsis management is extremely complex and includes several strategies such as controlling infection via antibiotics, correction of hypovolemia by administering intravenous fluids (IV fluids) and administration of vasopressors to counter sepsis-induced vasodilation. Multiple dosing strategies have been shown to lead to patient mortality, highlighting the importance of carefully timing these interventions [76]. From a learning perspective, RL models have been developed to determine optimal strategies in both continuous and discrete settings [77, 78, 42]. These studies make assumptions about reasonable patient behavior over subsequent steps and incorporate them into the reward function. How-

ever, incorrect reward specifications can lead to adverse behaviors, for instance, sudden changes in drug dosage. Thus, in this case study, we focus on recovering the clinician's reward function using Inverse Reinforcement Learning (IRL) and using it to train a policy which mimics the clinician. Successful reward recovery would help in understanding clinician's implicit goals and help develop robust generalizable agents. We demonstrate the efficacy of Meta-IRL in imitating the clinician's policy for sepsis treatment using MIMIC-III dataset.

Algorithm 4 Meta-IRL

Input: Expert Demonstrations: $D_e = (s_i, a_i)$; Task (Patient Group): $\tau \in \{\tau_1, \tau_2, \tau_3 ....., \tau_m\}$; $\mathcal{N}$ (max iterations)
Parameters: w^0, θ^0, ϕ^0
Randomly initialize feature expectation network μ_θ and reward function R_w
Initialize policy network $\pi(\phi)$ and feature map $\mathcal{F}$ using TRIL
for *i = 1 to* $\mathcal{N}$ **do**
  Set weights of $\mu_\theta, R_w, \pi_\phi$ to be θ^0, w^0, ϕ^0 respectively
  Randomly sample task τ_j and sample expert trajectories: $D_e^j = (s_k^j, a_k^j, s_{k+1}^j)$
  for *n = 1 to* $\mathcal{M}$ **do**
    $\mathcal{B}^2$**: Run one iteration of Batch-IRL on task-specific learning**
    Estimate true feature expectation from D_e^j : $\mu_e^j = \mathbb{E}_t[\gamma^{t-1}\, \mathcal{F}(s_t^j, a_t^j)]$
    Estimate task-specific $\mu(\theta_n^j)$ with DSFN and μ_e^j
    Solve QP for reward formulation and obtain task-specific reward weights w_n^j
    Run BCQ to obtain task-specific policy $\pi(\phi_n^j)$ with $R_w = [w_n^j]^T \cdot \mathcal{F}(s, a)$
  end for
  $\mathcal{B}^1$**: Perform Meta Update**
  $\theta^0 = \theta^0 + \beta(\theta_M^j - \theta^0)$
  $w^0 = w^0 + \beta(w_M^j - w^0)$
  $\phi^0 = \phi^0 + \beta(\phi_M^j - \phi^0)$
end for

4.3 Experiments

4.3.1 Datasets

Our input data comprises of a cohort of 17,000 adult patients from the Multiparameter Intelligent Monitoring in Intensive Care (MIMIC-III v1.4) database [59]. The patients fulfill the Sepsis-3 criteria, namely the presence of a suspected infection along with evidence of organ dysfunction [75]. We follow the formulation of Raghu *et al.* [77] and define each trajectory using a continuous

state-space which combines 46 static and dynamic physiological features including demographics, vitals, lab measurements and ventilation/fluid output related events. The dynamic features are collected at an interval of 4 hours and mortality or successful discharge from the ICU is the terminal state. Typically, IV fluid administration and vasopressor dosing are the two actions under policy control, while other treatments such as antibiotic dosing are out of scope of our study. Here, we further limit the action space to the amount of vasopressor dosage given to a patient at each 4-hour interval. The dosages are discretized into five bins, the first representing no treatment (zero dosage) while non-zero dosages being represented by quartiles. Limiting the action space makes the challenging IRL problem more tractable.

4.3.2 Training Details

To derive multiple tasks, we cluster the patients into 20 groups using K-means algorithm using their static contextual features at the time of admission. We divide the tasks in 70/30 ratio to create the training and test datasets. As shown in Figure 4.1 the different clusters are of varying siz and are mostly well-separated. To ensure stable training and effective adaptation, we limit the maximum number of patients sampled from a group during task-specific training to 100. A particular patient group is randomly selected during each iteration. We use single-task formulation as our baseline setting with all patient groups concatenated into a single dataset. We consider two baseline policies, derived from vanilla Batch-IRL and TRIL-based behaviour cloning. We consider two learning settings under Meta-IRL framework: 1) we apply the meta-update on the policy network only, while training feature expectation network μ_{theta} and reward function R_w from scratch on each task; 2) we meta-learn both the policy and feature expectation networks along with the reward formulation. We employ Adam optimizer [68] for training all networks with a learning rate of $3e^{-4}$ and execute $N = 600$ total iterations for both Batch-IRL and Meta-IRL. The training configurations for Batch-IRL and Meta-IRL are described in Table 4.1. We also use an isotropic multivariate Gaussian output layer for sampling feature expectation values from DSFN, as it has been shown to improve training in stochastic clinical setting. To do a fair comparison,

both Batch-IRL and Meta-IRL were initialized with the same initial policy π_ϕ and feature space $\mathcal{F}$ using TRIL. To determine the optimal meta-learning rate, we experimented with 5 different rates (0.005, 0.01, 0.025, 0.05, 0.1) and found the rate of 0.01 to perform best.

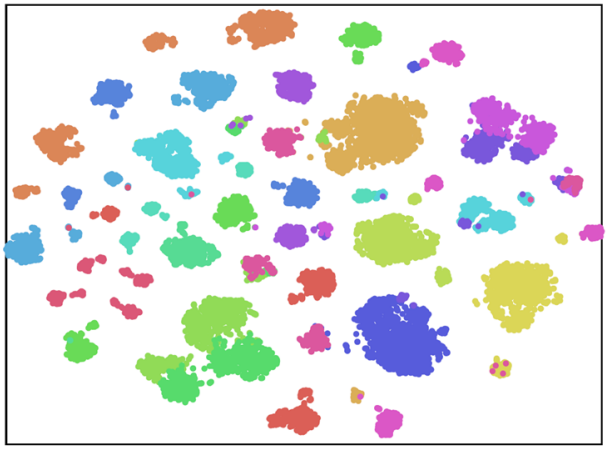


Figure 4.1: Visualization of tasks: tSNE plot of patient groups

Hyperparameters	**Warm Start**	**Batch-IRL**		**Meta-IRL**	
	TRIL	**DSFN**	**BCQ**	**DSFN**	**BCQ**
number of hidden layers	2	2	2	2	2
hidden node size	128	64	64	64	64
max training iterations	50,000	10,000	20,000	5,000	5,000
activation function	tanh	ReLU	ReLU	ReLU	ReLU
mini-batch size	64	32	128	32	128
λ (regularization)	1.4	-	-	-	-
prioritized experience replay	N	N	Y	N	Y
moving average for target network	-	0.01	0.01	0.01	0.01
discount rate	0.99	0.99	0.99	0.99	0.99
β (meta-update rate)	-	-	-	0.01	0.01

Table 4.1: Hyperparameter settings for multilayer neural networks employed in Batch-IRL and Meta-IRL

Evaluation : In a real-world offline setting, we cannot evaluate the reward of the IRL-based policy by simulating it in an environment, we can only measure its accuracy. We evaluate our approach by comparing the actions recommended by our policy network with the clinician's actions. The action-matching accuracy is defined as the proportion of transitions in the test data in which the policy action matches with the clinician's action. While accuracy does not necessarily

Method	All dosages	Zero dosage	Non-zero dosages
Behaviour Cloning (single-task)	$70.3 \pm 0.4\%$	$86.3 \pm 0.7\%$	$15.4 \pm 0.8\%$
Batch-IRL (single-task)	$73.9 \pm 1.4\%$	$93.7 \pm 2.2\%$	$6.6 \pm 0.6\%$
Meta-IRL (policy adaptation)	$67.8 \pm 5.1\%$	$85.8 \pm 8.3\%$	$13.1 \pm 2.3\%$
Meta-IRL (policy and reward adaptation)	$69.8 \pm 4.1\%$	$86.6 \pm 5.8\%$	$12.2 \pm 3.2\%$

Table 4.2: Action matching accuracy ($\mu \pm \sigma$)

imply optimal reward values, these measures are often correlated and a higher action match with clinician would ensure a reward close to that achieved by the clinician. We perform 5 simulations by varying the initialization seed during training. Since the expert action space is highly biased towards zero (no vasopressor dosage), in addition to the overall accuracy on for all actions, we also evaluate the accuracy on non-zero dosages.

4.3.3 Results

Meta-IRL imitates clinicians better than the Batch-IRL baseline. Table 4.2 shows the mean action-matching accuracy (with standard deviation) over 5 simulations on the 6 test groups. We observe that our multitask framework outperforms single-task IRL significantly on non-zero dosages. For zero dosage, our network slightly underperforms single-task IRL formulation. However, we believe that imitating the clinician on non-zero dosages is more challenging since in expert trajectories, non-zero dosage actions occur very infrequently, accounting for $\approx 20\%$ of all transitions. We also observe that adapting both the policy and reward networks helps in preventing the reward network from overfitting on the training tasks and thus, leads to 1% improvement in overall accuracy. Meta-IRL performs comparably to behaviour cloning in terms of overall accuracy but underperforms on non-zero dosages. This is because Meta-IRL undertakes the harder task of learning the underlying reward function, while behavioural cloning just mimics the clinician and doesn't need to reason about the underlying process. We also found that leveraging BCQ significantly stabilizes policy training, while the DDQN network used in the original Batch-IRL approach suffers from poor Q-value estimation and high variance during policy optimization.

CHAPTER 5
CONCLUSIONS AND FUTURE WORK

In this thesis, we propose counterfactual frameworks for robust offline clinical policy learning using observational data. Our proposed frameworks effectively tackle the inherent uncertainty in Electronic Health Records and enable improved policy generalization over heterogeneous patient groups. We observe that our frameworks outperform baseline methods on multiple clinical tasks, both in contextual bandit and dynamic treatment regime settings. Our frameworks can be valuable in designing robust clinical decision support systems by enabling more confident clinical policy evaluation as well as personalized treatment recommendations for patient cohorts with limited data.

In the first work, we described boostrapping and adversarial learning-based variants of IPS to tackle model uncertainty due to the clinician's propensity score imputation in an offline setting. Our work is one of the initial studies to highlight the importance of robust propensity score modeling for policy evaluation with higher confidence. Moreover, while existing research in off-policy learning has primarily focused on synthetic or semi-synthetic setups with access to true propensity scores, our estimators do not make such assumptions. We are the first to formulate the Heparin dosing problem in a bandit framework and derive a real-world clinical dataset with unknown clinician's propensity. In the second work, we presented a meta-Inverse Reinforcement Learning framework to learn sequential treatment policies with improved generalization over patients with varying contexts. While meta learning has shown great success in supervised learning tasks and online policy learning problems, it has not been investigated in the study of offline policy learning problems, particularly in clinical settings. Our work highlights that the multitask formulation with meta-learning based adaptation is a promising framework for recovering a physician's intrinsic contextual reward function in a large-scale chaotic clinical setting.

There are multiple promising future directions worth pursuing. Our frameworks can be com-

bined with more optimal off-policy bandit (e.g., doubly-robust method [30]) or IRL algorithms (e.g., f-IRL [79]). Our bootstrapping and adversarial learning frameworks could be extended to a more practical continuous drug dosage scenario which is difficult than the discrete action setting [80]. Another interesting direction for meta-learning-based IRL would be to leverage context-based meta-learning and encode the context via latent probabilistic embeddings. This would enable meta-learning of rewards and policy with unstructured multitask demonstrations and would not require us to explicitly create the task distribution, thus making it more amenable to real-world clinical data. Lastly, a better clinical policy algorithm by itself is not sufficient to practically deploy it in hospitals, it is important that all aspects of our framework, including policy, reward formulation and state feature representation, be critically evaluated by experts.

REFERENCES

[1] J. A. Osheroff, E. A. Pifer, D. F. Sittig, R. A. Jenders, and J. M. Teich, "Clinical decision support implementers' workbook", *Chicago: HIMSS*, p. 68, 2004.

[2] A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, and J. Dean, "A guide to deep learning in healthcare", *Nature Medicine*, vol. 25, no. 1, pp. 24–29, 1 Jan. 2019.

[3] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks", *Nature*, vol. 542, no. 7639, pp. 115–118, 2017.

[4] R. Poplin, A. V. Varadarajan, K. Blumer, Y. Liu, M. V. McConnell, G. S. Corrado, L. Peng, and D. R. Webster, "Prediction of cardiovascular risk factors from retinal fundus photographs via deep learning", *Nature Biomedical Engineering*, vol. 2, no. 3, p. 158, 2018.

[5] W. Zhu, L. Xie, J. Han, and X. Guo, "The Application of Deep Learning in Cancer Prognosis Prediction", *Cancers*, vol. 12, no. 3, p. 603, 2020.

[6] K. Thomsen, L. Iversen, T. L. Titlestad, and O. Winther, "Systematic review of machine learning for diagnosis and prognosis in dermatology", *Journal of Dermatological Treatment*, vol. 31, no. 5, pp. 496–510, 2020.

[7] S. Parbhoo, J. Bogojeska, M. Zazzi, V. Roth, and F. Doshi-Velez, "Combining kernel and model based learning for hiv therapy selection", *AMIA Summits on Translational Science Proceedings*, vol. 2017, p. 239, 2017.

[8] A. Guez, R. D. Vincent, M. Avoli, and J. Pineau, "Adaptive Treatment of Epilepsy via Batch-mode Reinforcement Learning.", presented at the AAAI, 2008, pp. 1671–1678.

[9] N. Prasad, L.-F. Cheng, C. Chivers, M. Draugelis, and B. E. Engelhardt, "A reinforcement learning approach to weaning of mechanical ventilation in intensive care units", *arXiv preprint arXiv:1704.06300*, 2017.

[10] M. Komorowski, L. A. Celi, O. Badawi, A. C. Gordon, and A. A. Faisal, "The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care", *Nature medicine*, vol. 24, no. 11, pp. 1716–1720, 2018.

[11] T. A. Carey and W. B. Stiles, "Some problems with randomized controlled trials and some viable alternatives", *Clinical Psychology & Psychotherapy*, vol. 23, no. 1, pp. 87–95, 2016.

[12] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, "Building machines that learn and think like people", *Behavioral and brain sciences*, vol. 40, 2017.

[13] M. W. Dusenberry, D. Tran, E. Choi, J. Kemp, J. Nixon, G. Jerfel, K. Heller, and A. M. Dai, "Analyzing the role of model uncertainty for electronic health records", in *Proceedings of the ACM Conference on Health, Inference, and Learning*, 2020, pp. 204–213.

[14] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight Uncertainty in Neural Network", in *International Conference on Machine Learning*, Jun. 1, 2015, pp. 1613–1622.

[15] B. Conroy, M. Xu-Wilson, and A. Rahman, "Patient similarity using population statistics and multiple kernel learning", in *Machine Learning for Healthcare Conference*, 2017, pp. 191–203.

[16] M. Komorowski, L. A. Celi, O. Badawi, A. C. Gordon, and A. A. Faisal, "The Artificial Intelligence Clinician learns optimal treatment strategies for sepsis in intensive care", *Nature Medicine*, vol. 24, no. 11, pp. 1716–1720, Nov. 2018.

[17] N. Kallus and A. Zhou, "Policy Evaluation and Optimization with Continuous Treatments", presented at the International Conference on Artificial Intelligence and Statistics, 2018, pp. 1243–1251.

[18] D. Bertsimas and C. McCord, "Optimization over continuous and multi-dimensional decisions with observational data", presented at the Advances in Neural Information Processing Systems, 2018, pp. 2962–2970.

[19] N. Kallus, "Balanced policy evaluation and learning", presented at the Advances in Neural Information Processing Systems, 2018, pp. 8895–8906.

[20] S. S. Villar, J. Bowden, and J. Wason, "Multi-armed bandit models for the optimal design of clinical trials: Benefits and challenges", *Statistical science: a review journal of the Institute of Mathematical Statistics*, vol. 30, no. 2, p. 199, 2015.

[21] Y. Varatharajah, B. Berry, S. Koyejo, and R. Iyer, "A Contextual-bandit-based Approach for Informed Decision-making in Clinical Trials", *arXiv preprint arXiv:1809.00258*, 2018.

[22] A. Tewari and S. A. Murphy, "From ads to interventions: Contextual bandits in mobile health", in *Mobile Health*, Springer, 2017, pp. 495–517.

[23] D. G. Horvitz and D. J. Thompson, "A generalization of sampling without replacement from a finite universe", *Journal of the American statistical Association*, vol. 47, no. 260, pp. 663–685, 1952.

[24] A. Swaminathan and T. Joachims, "Counterfactual Risk Minimization: Learning from Logged Bandit Feedback", in *International Conference on Machine Learning*, Jun. 1, 2015, pp. 814–823.

[25] H. Wu and M. Wang, "Variance Regularized Counterfactual Risk Minimization via Variational Divergence Minimization", in *International Conference on Machine Learning*, Jul. 3, 2018, pp. 5353–5362.

[26] L. Faury, U. Tanielian, F. Vasile, E. Smirnova, and E. Dohmatob, "Distributionally Robust Counterfactual Risk Minimization", Jun. 14, 2019. arXiv: 1906.06211 [cs, stat].

[27] E. L. Ionides, "Truncated importance sampling", *Journal of Computational and Graphical Statistics*, vol. 17, no. 2, pp. 295–311, 2008.

[28] L. Bottou, J. Peters, J. Quiñonero Candela, D. X. Charles, D. M. Chickering, E. Portugaly, D. Ray, P. Simard, and E. Snelson, "Counterfactual Reasoning and Learning Systems: The Example of Computational Advertising", *Journal of Machine Learning Research*, vol. 14, pp. 3207–3260, 2013.

[29] A. Swaminathan and T. Joachims, "The self-normalized estimator for counterfactual learning", in *advances in neural information processing systems*, 2015, pp. 3231–3239.

[30] M. Dudík, D. Erhan, J. Langford, L. Li, *et al.*, "Doubly robust policy evaluation and optimization", *Statistical Science*, vol. 29, no. 4, pp. 485–511, 2014.

[31] Y. Xie, B. Liu, Q. Liu, Z. Wang, Y. Zhou, and J. Peng, "Off-policy evaluation and learning from logged bandit feedback: Error reduction via surrogate policy", *arXiv preprint arXiv:1808.00232*, 2018.

[32] M. W. Dusenberry, D. Tran, E. Choi, J. Kemp, J. Nixon, G. Jerfel, K. Heller, and A. M. Dai, "Analyzing the Role of Model Uncertainty for Electronic Health Records", Jun. 10, 2019. arXiv: 1906.03842 [cs, stat].

[33] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles", Dec. 5, 2016. arXiv: 1612.01474 [cs, stat].

[34] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning", Jun. 6, 2015. arXiv: 1506.02142 [cs, stat].

[35] G. Zhang, S. Sun, D. Duvenaud, and R. Grosse, "Noisy Natural Gradient as Variational Inference", in *International Conference on Machine Learning*, Jul. 3, 2018, pp. 5852–5861.

[36] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick", in *Advances in neural information processing systems*, 2015, pp. 2575–2583.

[37] B. Chakraborty and S. A. Murphy, "Dynamic treatment regimes", *Annual review of statistics and its application*, vol. 1, pp. 447–464, 2014.

[38] J. Leike, M. Martic, V. Krakovna, P. A. Ortega, T. Everitt, A. Lefrancq, L. Orseau, and S. Legg, *Ai safety gridworlds*, 2017. arXiv: 1711.09883 [cs.LG].

[39] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning", in *Proceedings of the twenty-first international conference on Machine learning*, 2004, p. 1.

[40] D. Lee, S. Srinivasan, and F. Doshi-Velez, "Truly batch apprenticeship learning with deep successor features", *arXiv preprint arXiv:1903.10077*, 2019.

[41] D. Jarrett, I. Bica, and M. van der Schaar, "Strictly batch imitation learning by energy-based distribution matching", *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[42] M. Komorowski, L. A. Celi, O. Badawi, A. C. Gordon, and A. A. Faisal, "The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care", *Nature medicine*, vol. 24, no. 11, pp. 1716–1720, 2018.

[43] N. Radcliffe, "Using control groups to target on predicted lift: Building and assessing uplift models", *Direct Market J Direct Market Assoc Anal Council*, vol. 1, pp. 14–21, 2007.

[44] X. Liu, P. He, W. Chen, and J. Gao, "Multi-task deep neural networks for natural language understanding", *arXiv preprint arXiv:1901.11504*, 2019.

[45] J. Gu, Y. Wang, Y. Chen, V. O. Li, and K. Cho, "Meta-learning for low-resource neural machine translation", in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 3622–3631.

[46] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, "Meta-learning in neural networks: A survey", *arXiv preprint arXiv:2004.05439*, 2020.

[47] C. Finn, P. Abbeel, and S. Levine, "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks", in *International Conference on Machine Learning*, Jul. 17, 2017, pp. 1126–1135.

[48] A. Nichol, J. Achiam, and J. Schulman, "On First-Order Meta-Learning Algorithms", Mar. 8, 2018. arXiv: 1803.02999 [cs].

[49] A. Nichol and J. Schulman, "Reptile: A scalable metalearning algorithm", *arXiv preprint arXiv:1803.02999*, vol. 2, no. 3, p. 4, 2018.

[50] X. S. Zhang, F. Tang, H. H. Dodge, J. Zhou, and F. Wang, "Metapred: Meta-learning for clinical risk prediction with limited patient electronic health records", presented at the Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2487–2495.

[51] L. Liu, Z. Liu, H. Wu, Z. Wang, J. Shen, Y. Song, and M. Zhang, "Multi-task Learning via Adaptation to Similar Tasks for Mortality Prediction of Diverse Rare Diseases", May 11, 2020. arXiv: 2004.05318 [cs, stat].

[52] X. Li, L. Yu, C.-W. Fu, and P.-A. Heng, "Difficulty-aware Meta-Learning for Rare Disease Diagnosis", *arXiv preprint arXiv:1907.00354*, 2019.

[53] X. Jiang, L. Ding, M. Havaei, A. Jesson, and S. Matwin, "Task Adaptive Metric Space for Medium-Shot Medical Image Classification", in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, D. Shen, T. Liu, T. M. Peters, L. H. Staib, C. Essert, S. Zhou, P.-T. Yap, and A. Khan, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2019, pp. 147–155, ISBN: 978-3-030-32239-7.

[54] A. Sharma, G. Gupta, R. Prasad, A. Chatterjee, L. Vig, and G. Shroff, "MetaCI: Meta-Learning for Causal Inference in a Heterogeneous Population", *arXiv preprint arXiv:1912.03960*, 2019.

[55] A. Gleave and O. Habryka, "Multi-task maximum entropy inverse reinforcement learning", *arXiv preprint arXiv:1805.08882*, 2018.

[56] K. Xu, E. Ratner, A. Dragan, S. Levine, and C. Finn, "Learning a prior over intent via meta-inverse reinforcement learning", in *International Conference on Machine Learning*, PMLR, 2019, pp. 6952–6962.

[57] E. Veach and L. J. Guibas, "Optimally combining sampling techniques for monte carlo rendering", in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 1995, pp. 419–428.

[58] R. P. Owen, R. B. Altman, and T. E. Klein, "PharmGKB and the International Warfarin Pharmacogenetics Consortium: The changing role for pharmacogenomic databases and single-drug pharmacogenetics", *Human mutation*, vol. 29, no. 4, pp. 456–460, 2008.

[59] A. E. W. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "MIMIC-III, a freely accessible critical care database", *Sci Data*, vol. 3, no. 1, pp. 1–9, May 24, 2016.

[60] M. Dudík, J. Langford, and L. Li, "Doubly robust policy evaluation and learning", *arXiv preprint arXiv:1103.4601*, 2011.

[61] I. W. P. Consortium, "Estimation of the warfarin dose with clinical and pharmacogenetic data", *New England Journal of Medicine*, vol. 360, no. 8, pp. 753–764, 2009.

[62] H. Bastani and M. Bayati, "Online Decision Making with High-Dimensional Covariates", *Operations Research*, Nov. 7, 2019.

[63] M. Farajtabar, Y. Chow, and M. Ghavamzadeh, "More robust doubly robust off-policy evaluation", in *International Conference on Machine Learning*, 2018, pp. 1447–1456.

[64] J. W. Schurr, A.-M. Muske, C. A. Stevens, S. E. Culbreth, K. W. Sylvester, and J. M. Connors, "Derivation and validation of age-and body mass index-adjusted weight-based unfractionated heparin dosing", *Clinical and Applied Thrombosis/Hemostasis*, vol. 25, 2019.

[65] J. Fan, B. John, and E. Tesdal, "Evaluation of heparin dosing based on adjusted body weight in obese patients", *American Journal of Health-System Pharmacy*, vol. 73, no. 19, pp. 1512–1522, 2016.

[66] M. M. Ghassemi, S. E. Richter, I. M. Eche, T. W. Chen, J. Danziger, and L. A. Celi, "A data-driven approach to optimized medication dosing: A focus on heparin", *Intensive care medicine*, vol. 40, no. 9, pp. 1332–1339, 2014.

[67] S. Nemati, M. M. Ghassemi, and G. D. Clifford, "Optimal medication dosing from suboptimal clinical examples: A deep reinforcement learning approach", in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 2016, pp. 2978–2981.

[68] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization", *arXiv preprint arXiv:1412.6980*, 2014.

[69] H. Zenati, A. Bietti, M. Martin, E. Diemert, and J. Mairal, "Counterfactual learning of continuous stochastic policies", 2020.

[70] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations", in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Oakland, CA, USA, vol. 1, 1967, pp. 281–297.

[71] S. Fujimoto, D. Meger, and D. Precup, "Off-policy deep reinforcement learning without exploration", in *International Conference on Machine Learning*, PMLR, 2019, pp. 2052–2062.

[72] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning", *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[73] R. S. Sutton, A. G. Barto, *et al.*, *Introduction to Reinforcement Learning*, 4. MIT press Cambridge, 1998, vol. 2.

[74] S. Fujimoto, E. Conti, M. Ghavamzadeh, and J. Pineau, "Benchmarking batch deep reinforcement learning algorithms", *arXiv preprint arXiv:1910.01708*, 2019.

[75] M. Singer, C. S. Deutschman, C. W. Seymour, M. Shankar-Hari, D. Annane, M. Bauer, R. Bellomo, G. R. Bernard, J.-D. Chiche, C. M. Coopersmith, *et al.*, "The third international consensus definitions for sepsis and septic shock (sepsis-3)", *Jama*, vol. 315, no. 8, pp. 801–810, 2016.

[76] J. Waechter, A. Kumar, S. E. Lapinsky, J. Marshall, P. Dodek, Y. Arabi, J. E. Parrillo, R. P. Dellinger, A. Garland, C. A. T. of Septic Shock Database Research Group, *et al.*, "Interaction between fluids and vasoactive agents on mortality in septic shock: A multicenter, observational study", *Critical care medicine*, vol. 42, no. 10, pp. 2158–2168, 2014.

[77] A. Raghu, M. Komorowski, I. Ahmed, L. Celi, P. Szolovits, and M. Ghassemi, "Deep reinforcement learning for sepsis treatment", *arXiv preprint arXiv:1711.09602*, 2017.

[78] A. Raghu, M. Komorowski, and S. Singh, "Model-Based Reinforcement Learning for Sepsis Treatment", Nov. 23, 2018. arXiv: `1811.09602 [cs, stat]`.

[79] T. Ni, H. Sikchi, Y. Wang, T. Gupta, L. Lee, and B. Eysenbach, "F-irl: Inverse reinforcement learning via state marginal matching", *arXiv preprint arXiv:2011.04709*, 2020.

[80] N. Kallus and A. Zhou, "Policy evaluation and optimization with continuous treatments", *arXiv preprint arXiv:1802.06037*, 2018.