Thesis for The Degree of Licentiate of Engineering

Understanding and Evaluating Policies for Sequential Decision-Making

Anton Matsson

Department of Computer Science and Engineering
Chalmers University of Technology | University of Gothenburg
Gothenburg, Sweden, 2023

Understanding and Evaluating Policies for Sequential Decision-Making

Anton Matsson

ISSN 1652-876X

Department of Computer Science and Engineering
Division of Data Science and AI
Healthy AI Lab
Chalmers University of Technology | University of Gothenburg
SE-412 96 Göteborg,
Sweden
Phone: +46(0)31 772 1000

Printed by Chalmers Digitaltryck,
Gothenburg, Sweden 2023.

To my family.

Understanding and Evaluating Policies for Sequential Decision-Making

Anton Matsson

Department of Computer Science and Engineering
Chalmers University of Technology | University of Gothenburg

Abstract

Sequential-decision making is a critical component of many complex systems, such as finance, healthcare, and robotics. The long-term goal of a sequential decision-making process is to optimize the policy under which decisions are made. In safety-critical domains, the search for an optimal policy must be based on observational data, as new decision-making strategies need to be carefully evaluated before they can be tested in practice. In this thesis, we highlight the importance of understanding sequential decision-making at different stages of this procedure. For example, to assess which policies can be evaluated with the available data, we need to understand the policy that actually generated the data. And once we are given a policy to evaluate, we need to understand how it differs from current practice.

First, we focus on the evaluation process, where a target policy is evaluated using off-policy data collected under a different so-called behavior policy. This problem, commonly referred to as off-policy evaluation, is often solved with importance sampling (IS) techniques. Despite their popularity, IS-based methods suffer from high variance and are hard to diagnose. To address these issues, we propose estimating the behavior policy using prototype learning. Using the learned prototypes, we describe differences between target and behavior policies, allowing for better assessment of the IS estimates.

Next, we take a clinical direction and study the sequential treatment of patients with rheumatoid arthritis (RA). The armamentarium of disease-modifying anti-rheumatic drugs (DMARDs) for RA patients has greatly expanded over the past decades. However, it is still unclear which treatment work best for individual patients. To examine how observational data can be used to evaluate new policies, we describe the most common patterns of DMARDs in a large patient registry from the US. We find that the number of unique patterns is large, indicating a significant variation in clinical practice which can be exploited for evaluation purposes. However, additional assumptions may be required to arrive at statistically sound results.

Keywords

Sequential Decision-Making, Off-Policy Evaluation, Observational Data, Rheumatoid Arthritis

List of Publications

Appended Publications

This thesis is based on the following publications:

[**Paper I**] **A. Matsson**, F. D. Johansson, *Case-Based Off-Policy Evaluation Using Prototype Learning.*
Proceedings of the 38th Conference on Uncertainty in Artificial Intelligence, PMLR 180 (2022), 1339–1349.

[**Paper II**] **A. Matsson**, D. H. Solomon, M. M. Crabtree, R. W. Harrison, H. J. Litman, F. D. Johansson, *Patterns in the Sequential Treatment of Rheumatoid Arthritis Patients Starting a b/tsDMARD: 10-Year Experience from a US-Based Registry.*
Submitted, under review.

Other Publications

The following publications were published during my PhD studies, or are currently in submission/under revision. However, they are not appended to this thesis, due to contents overlapping that of appended publications or contents not related to the thesis.

[**a**] **A. Matsson**, F. D. Johansson, *Evaluating Policies for Sepsis Management: Decomposing Value Estimates Using Prototypes.*
AAAI 2022 Workshop on Trustworthy AI for Healthcare.

[**b**] A. Breitholtz, **A. Matsson**, F. D. Johansson, *Unsupervised Domain Adaptation by Learning Using Privileged Information.*
Submitted, under review.

Acknowledgment

First of all, I would like to thank my supervisor, Fredrik Johansson, for his continuous support and guidance. I am truly fortunate to have such a dedicated supervisor. I would also like to thank my co-supervisor, Morteza Haghir Chehreghani, and my examiner, Dag Wedelin, for their feedback and encouragement.

I am grateful to work in the stimulating environment that the DSAI division at Chalmers offers. Thanks to all my PhD colleagues who brighten up the time in the office: Markus, Hampus, David, Firooz, Mehrdad, Christopher, Tobias, Riccardo, Juan, Emilio, Tobias, Mena, Hannes, Hanna, Simon, Fazeleh, Niklas, Arman, Linus, Emil, Alexander, Filip, and Daniel. A special thanks to my great office buddies Lena, Newton, Adam, and Lovisa. A final thanks to the rest of the division: faculty, postdocs, and administrative staff.

During the last year, I have been involved in a research collaboration with a team from the US. I would like to thank Dan Solomon, Heather Litman, Ryan Harrison, and Margaux Crabtree for interesting discussions about the treatment of rheumatoid arthritis.

Last but not least, I would like to thank my family. Thank you Gunnel and Erik, my parents, and Julia, my sister, for your invaluable support and love during this journey. And thank you Sara, my wonderful partner, for always being my biggest supporter.

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The computations were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC) at Chalmers Centre for Computational Science and Engineering (C3SE) partially funded by the Swedish Research Council through grant agreement no. 2018-05973.

Contents

Part I

Introductory Chapters

Chapter 1

Introduction

Making decisions is central to many domains, from finance to healthcare. In finance, an investor must decide how to allocate investments across a variety of assets. In healthcare, a physician is faced with the decision of choosing the medication that works best for a particular patient. In many cases, decisions constitute a sequential process, where each decision may affect the outcome of future decisions, and the outcome of the entire process is dependent on the sequence of decisions made.

A decision-making policy is a mapping from a basis for decision to a probability distribution over available decisions, here called actions. Depending on the problem, such a basis may not only consist of information about the current state of the system but also a history of previous states, actions, and rewards. For example, previous medications and their associated rewards are valuable information for selecting the next treatment for a patient, along with their current health status.

The goal of sequential decision-making is to optimize the sequence of actions to maximize the cumulative reward. Achieving this goal translates into finding a policy with the highest possible value, where the value of a policy is defined as the expected cumulative reward when acting as suggested by the policy. For autoimmune diseases, for example rheumatoid arthritis, the reward is often measured by negative disease activity, and treating a patient according to an optimal policy means selecting medications that keep the disease activity as low as possible through the course of the disease.

In safety-critical domains, the search for an optimal policy is limited by the fact that it is generally not possible to deploy a policy without having an estimate of its value. Since no data of actions and rewards are available for an unverified policy, its value must be estimated based on data generated by another policy. This problem is known as off-policy evaluation. The data-generating policy and the policy to be evaluated are called the behavior policy and the target policy, respectively. The true behavior policy is often unknown, but it can be estimated from data. In healthcare, observational data of decisions and rewards are plentiful in, for example, electronic health records.

The most popular approach to off-policy evaluation is importance sampling

(IS) (Precup, 2000), where observed rewards are re-weighted to account for the distributional mismatch between the target and behavior policies. The standard IS value estimator is unbiased but known to suffer from high variance (Mandel et al., 2014). The weighted IS estimator reduces variance at the cost of introducing bias (Precup, 2000). A different set of methods perform off-policy evaluation using a learned model of the decision-making environment (Mannor et al., 2007). This approach has generally low variance but the bias induced by the model approximation is hard to quantify. Finally, there are also doubly robust methods which combine these ideas to lower variance and avoid bias (Jiang & Li, 2016).

Regardless of how the off-policy evaluation problem is attacked, there is a fundamental difficulty in the fact that the ground truth value is unknown. After all, the estimated value is just a number, sometimes supplemented by a confidence interval. At most we can compare it to the value of the behavior policy, which is relatively easy to estimate. When using IS-based methods, we can try to assess the estimate by, e.g., inspecting individual weights, but this approach does not answer some of the most critical questions: In which type of situations do the policies differ from each other? How do the differences manifest themselves in the estimated value?

In the first paper of this thesis, Paper I, we answer these questions by estimating the behavior policy using prototype learning. Prototype learning is an interpretable machine learning technique mostly used for classification (Chen et al., 2019; Li et al., 2018). Here, we use the prototypical cases, which are learned during model training and interpretable by a domain expert, as a diagnostic tool for off-policy evaluation. We utilize that the prototypes are representative cases of the space of decision bases and actions. By comparing how the target and behavior policies suggest acting in each of the cases, we obtain a compact summary of any differences between the policies. We can also stratify estimated values into prototype-based contributions, allowing us to understand in which situations the target policy yields higher reward than the behavior policy, and vice versa.

The second paper, Paper II, focuses on a clinical application of sequential decision-making, namely the sequential treatment of patients with rheumatoid arthritis (RA). Although there are many effective drugs for RA, the search for a working medication for individual patients is often based on trial-and-error. A first step in developing new clinical guidelines is to understand current practice. In particular, what opportunities do observational data offer to evaluate new policies for clinical decision-making? In this work, we describe the most common patterns in the sequential treatment of RA in a large US-based patient registry. Our results show that there is a large practice variation, which allows for evaluating new treatment strategies but also presents statistical challenges in the evaluation process.

Chapter 2

Background

In this chapter, we provide necessary background for the papers included in this thesis. We first formulate sequential decision-making in general terms before discussing the specific case of sequential decision-making in healthcare. Then, we introduce reinforcement learning as a tool for solving such problems and learning optimal policies. Finally, we describe policy evaluation, focusing on off-policy evaluation, before concluding the chapter with a brief description of interpretable policy representations.

2.1 Sequential Decision-Making

Sequential decision-making can be seen as a sequence of interactions between an agent and an environment, see Figure 2.1(a). At each stage $t \in \{0, \dots, T\}$ of the process, the agent executes an action $A_t \in \mathcal{A}$ based on an observed state $S_t \in \mathcal{S}$ of the environment.[1] The length of the process, T, is a finite random variable, and $\mathcal{S}$ and $\mathcal{A} = \{1, ..., K\}$, where K is the number of actions, denote the state space and the action space, respectively. Next, the environment transitions into a new state S_{t+1}, and the agent receives a reward $R_t \in \mathbb{R}$ as a quality measure of the action taken. At all stages, states, actions and rewards are random variables, of which observed values are denoted by lower-case letters.

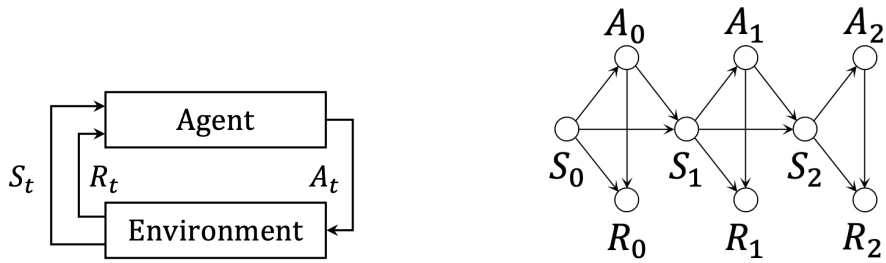
State transitions and rewards are defined by the dynamics $p(S_{t+1}, R_t \mid S_0, A_0, R_0, \dots, S_t, A_t)$ of the environment. A common assumption is that the decision process is Markov. In a Markov decision process (MDP), transitions and rewards depend only on the most recent state-action pair, that is,

$$p(S_{t+1}, R_t \mid S_0, A_0, R_0, \dots, S_t, A_t) = p(S_{t+1}, R_t \mid S_t, A_t).$$

Figure 2.1(b) shows the probabilistic graphical model (Koller & Friedman, 2009) for an MDP with $T = 2$.

To navigate through the potentially high-dimensional space of states and actions, the agent follows a policy $\pi \in \Pi$. A policy can be either deterministic

[1] We assume that the same set of actions are available at all time steps.



(a) Sequential decision-making.

(b) Markov decision process.

Figure 2.1: Left: Sequential decision-making is often described as an interaction between the decision-maker, the agent, and an environment. At each stage t of the process, the agent executes an action A_t based on the current state S_t of the environment. The environment responds with a reward R_t and transitions into a new state S_{t+1}. The long-term goal of the agent is to learn a policy that maximizes the expected cumulative reward. Right: A common assumption is that the decision process is Markov, meaning that state transitions, actions, and rewards depend only on the most recent state-action pair.

or stochastic. A deterministic policy $\pi : \mathcal{S} \to \mathcal{A}$ is a mapping from a state S to an action A, while a stochastic policy $\pi : \mathcal{S} \to \Delta_{\mathcal{A}}$ maps the state S to a probability distribution over the action space. We use $\pi(s)$ to denote the action selected by a deterministic policy. For a stochastic policy, the probability of taking action a in state s is denoted $p_\pi(A = a \mid S = s)$.

A policy π induces a probability distribution $p_\pi(S_0, A_0, R_0, \ldots, S_T, A_T, R_T)$. In an MDP, represented by the graph in Figure 2.1(b), this distribution can be factorized as

$$p(S_0)p_\pi(A_0 \mid S_0) \prod_{t=0}^{T-1} p(S_{t+1}, R_t \mid S_t, A_t)p_\pi(A_{t+1} \mid S_{t+1})p(R_T \mid S_T, A_T),$$

where components not influenced by the policy lack the subscript π. Let $\mathbb{E}_\pi$ denote the expectation with respect to this distribution. Now, the value of π, V^π, is defined as the expected sum of rewards,

$$V^\pi \coloneqq \mathbb{E}_\pi \left[\sum_{t=0}^{T} R_t \right]. \tag{2.1}$$

The ultimate goal of a sequential-decision making process is to find a policy with the highest possible value.[2]

2.2 Sequential Decision-Making in Healthcare

As a concrete example of sequential decision-making in healthcare, we consider the sequential treatment of patients with rheumatoid arthritis (RA). RA is an

[2]The value of a policy is also called the average value function under that policy.

autoimmune disease which affects the joints of the body, often with painful inflammations and stiffness as a result. Patients with RA are typically treated with different types of disease-modifying anti-rheumatic drugs (DMARDs), for example biologic or targeted-synthetic DMARDs (b/tsDMARDS).

In this example, the decision-making agent corresponds to a rheumatologist and each stage of the decision process to a clinical visit of a patient. The state S_t contains information about the patient and their health status, for example demographics, number of swollen joints, and comorbidities, while the action A_t is a selected treatment. The reward R_t could be defined as the reduction of disease activity between two visits (note that the reward may be negative). In other applications, it is also common that there is only a final reward $R = R_T$ awarded at the end of the sequence. As in most medical settings, the causes of the patient's response to the treatment, i.e., the dynamics of the environment, are unknown.

Above, we formulated the sequential-decision making problem as a Markov decision process, which is the standard setting in many domains. In healthcare, however, there is generally no reason to believe that the Markov assumption holds. For example, in RA treatment, the reward at stage t may not only depend on the immediately preceding state and the most recent treatment alone, but also on previous treatment strategies.

To obtain a more realistic model, we introduce the notion of *history*. The history at stage t, H_t, comprises the states, actions and rewards up until this point in time, i.e., $H_t \coloneqq (S_0, A_0, R_0, S_1, A_1, R_1, \ldots, S_t)$. Let $H_0 \coloneqq S_0$. The fact that the history grows with time presents a potential difficulty, and it may be necessary to consider only the most recent history or learn a function that summarizes the history. In this setting, we define a stochastic policy as a mapping from a history space $\mathcal{H}$ to a probability distribution over the action space.[3] Furthermore, the expectation in Equation (2.1) is computed with respect to the distribution

$$p(H_0)p_\pi(A_0 \mid H_0) \prod_{t=0}^{T-1} p(S_{t+1}, R_t \mid H_t, A_t)p_\pi(A_{t+1} \mid H_{t+1})p(R_T \mid H_T, A_T).$$

Note that this expression is the complete factorization of the joint distribution, without independence assumptions.

2.3 Reinforcement Learning

Reinforcement learning (RL) is a branch of machine learning which provides methods for solving sequential-decision making problems and learning optimal policies. RL has drawn attention due to its recent success in, for example, mastering board and computer games (Mnih et al., 2013; Silver et al., 2018), adapting robots (Cully et al., 2015), and discovering faster algorithms for matrix computation (Fawzi et al., 2022). RL has also been applied to problems

[3] In a healthcare context, a policy may also be referred to as a dynamic treatment regime (Chakraborty, 2013).

from the medical domain, for example the treatment of sepsis (Komorowski et al., 2018), the management of mechanical ventilation (Prasad et al., 2017), and mobile health (Tewari & Murphy, 2017). Here, we introduce two standard RL algorithms, policy iteration and Q-learning, and explain how they can be used in an offline setting when no data collection is allowed.

2.3.1 Policy Iteration

We have already defined the marginal value of a policy π, V^π, in Equation (2.1). Often, we are interested in the value of a single state s, defined as the expected sum of rewards when following a policy π starting in s at time t: $V_t^\pi(s) \coloneqq \mathbb{E}_\pi \left[\sum_{t'=t}^{T} R_{t'} \mid S_t = s \right]$. Assuming a Markov decision process with dynamics $p(S_{t+1}, R_t \mid S_t, A_t)$, the value is the same for all t, and we usually write $V^\pi(s)$ to simplify notation. A well-known property of $V^\pi(s)$ is that it satisfies the Bellman equation

$$V^\pi(s) = \sum_a p_\pi(a \mid s) \sum_{s',r} p(s', r \mid s, a) \left[r + V^\pi(s') \right],$$

where $s' \in \mathcal{S}$.

Policy iteration is one of the most fundamental RL algorithms and relies on the Bellman equation in the search for an optimal policy (Sutton & Barto, 2018). As input, the algorithm takes an initial policy π and an initial value $V_0^\pi(s)$, both chosen arbitrarily. For simplicity, we exemplify with a deterministic policy $\pi(s)$ here. The algorithm iteratively approximates $V_1^\pi(s), V_2^\pi(s), \ldots$ according to

$$V_{k+1}^\pi(s) \coloneqq \sum_{s',r} p(s', r \mid s, \pi(s)) \left[r + V_k^\pi(s') \right].$$

As $k \to \infty$, it can be shown that the sequence $\{V_k^\pi(s)\}$ converges to $V^\pi(s)$. When the value of all states $s \in \mathcal{S}$ have been approximated, the policy is updated according to

$$\pi(s) = \arg\max_a \sum_{s',r} p(s', r \mid s, \pi(s))[r + V(s')].$$

This update results in a greedy policy, i.e., a policy that always takes the best-looking action. The policy improvement theorem ensures that such a policy is at least as good as the initial policy. Policy iteration alternatively approximates the values and improves the policy until convergence of the policy, resulting in an optimal policy $\pi^\star$.

2.3.2 Q-Learning

Policy iteration and its close relative "value iteration" require full knowledge of the environment, including its dynamics, which is often not available in practice. Instead, most RL algorithms that have practical utility learn a policy based on collected experience of the environment. One of the most popular

algorithms that belongs to this category is Q-learning (Sutton & Barto, 2018). "Q" stands for quality, and the Q-function, or action-value function, for a policy π is defined as $Q^\pi(s,a) \coloneqq \mathbb{E}_\pi\left[\sum_{t'=t}^{T} R_{t'} \mid S_t = s, A_t = a\right]$. It is the expected sum of rewards when following π starting in state s *and* taking action a.

In Q-learning, starting with an arbitrarily initialized Q-function, the agent interacts with the environment by following an exploration policy derived from the Q-function. Typically, this policy is the ϵ-greedy policy which in state s selects the action $\arg\max_a Q(s,a)$ with probability $1-\epsilon$ and with probability ϵ selects an action uniformly at random. Every time the agent executes an action A in a state S and observes the next state S' and the reward R, the Q-function is updated according to

$$Q(S,A) = Q(S,A) + \alpha\left[R + \max_a Q(S',a) - Q(S,A)\right],$$

where α is a step size parameter.[4] Note that the greedy action chosen in the update may not be the action that would have been chosen under the exploration policy. Since the Q-values are updated independently of the agent's policy, Q-learning is a so-called off-policy algorithm. With sufficient exploration, the learned Q-function converges to the optimal Q-function, from which an optimal policy can easily be derived.

2.3.3 Offline Reinforcement Learning

The continuous exploration of the environment in the Q-learning procedure is not always practical. In safety-critical domains, for example, rolling out the agent's policy while learning the Q-function may be both dangerous and unethical. Instead, the search for an optimal policy must be based on already collected data. This setting is usually called offline RL or batch RL, in contrast to traditional online RL. In offline RL, the task is to learn a policy from a static dataset of transition tuples $\left(s_t^{(i)}, a_t^{(i)}, s_{t+1}^{(i)}, r_t^{(i)}\right)$.

There are many approaches to offline RL and we will not cover all of them here. Instead, we briefly outline two simple methods that are connected to the previous sections on online RL. First, Q-learning can be used in an offline setting if we instead of letting the agent interact with the environment repeatedly sample transition tuples from the static dataset. While this method works in principle, it may suffer from distributional shift, see e.g., (Kumar et al., 2019). Second, we can try to estimate the unknown dynamics of the environment using the available data. Assuming an MDP, we can then use policy iteration as described in Section 2.3.1, replacing the true dynamics $p(s', r \mid s, a)$ with an estimate $\hat{p}(s', r \mid s, a)$. For a more complete overview of offline RL, we refer to, e.g., Levine et al. (2020).

[4] This version of the Q-learning algorithm assumes discrete states and actions.

2.4 Policy Evaluation

Applying reinforcement learning to a sequential decision-making problem yields a policy, which hopefully is at least "good" for the problem at hand. A different problem is to understand how "good" a *given* policy is. This task translates to computing the value of the given policy——the target policy——according to Equation (2.1). Without complete knowledge of the system dynamics, the expectation in Equation (2.1) is intractable. However, if it is possible to collect trajectories, often called roll-outs, of the target policy, a simple Monte Carlo estimator provides an unbiased estimate of its value:

$$\hat{V}^{\pi} = \frac{1}{n} \sum_{i=1}^{n} \sum_{t=0}^{T} r_t^{(i)}.$$

Here, n denotes the number of roll-outs and $r_t^{(i)}$ is the reward awarded at stage t in roll-out i.

As we already have pointed out, rolling out an unverified target policy π is unacceptable in many situations. Instead, we must rely on data that have already been collected under a different policy μ, commonly referred to as the behavior policy. Estimating the value of π based on data from μ is referred to as off-policy evaluation. To ensure identifiability of V^{π}, we need to make assumptions about the policies and the data-generating process. Before stating these assumptions and formalizing off-policy evaluation, we introduce a concept from causality: potential outcomes.

2.4.1 Potential Outcomes

The notion of potential outcomes represents what *would* happen under a certain action (Rubin, 2005). For example, consider a one-stage decision process in healthcare where two treatments a and a' are available in state s, corresponding to an individual patient. The potential outcomes under a and a' are defined as $R(a)$ and $R(a')$, respectively. By selecting treatment a, we observe the reward r corresponding to the potential outcome $R(a)$. The potential outcome or counterfactual $R(a')$, like the treatment effect $\Delta \coloneqq R(a) - R(a')$, remains unobserved. Consequently, treatment effects must be studied at population level, for example through the conditional average treatment effect $\mathbb{E}\left[\Delta \mid S\right]$.

The potential outcomes framework can be extended to multi-stage decision processes (Robins, 1997). Let $\bar{A}_t$ denote a sequence of actions up to stage t. The potential outcomes under $\bar{A}_t$ include states $S_0, \ldots, S_t(\bar{A}_{t-1})$ and rewards $R_0(A_0), \ldots, R_t(\bar{A}_t)$. Clearly, as t grows, the total number of potential outcomes quickly becomes very large. In this framework, the value of a policy π is defined as $\mathbb{E}_{\pi}\left[\sum_{t=0}^{T} R_t(\bar{A}_t)\right]$.

2.4.2 Off-Policy Evaluation

Off-policy evaluation raises the following question: What would happen if we executed the unobserved actions as suggested by the target policy π?

Answering this question requires counterfactual reasoning since we only observe potential outcomes associated with actions taken under the behavior policy μ. Specifically, we assume that we have access to a dataset $\mathcal{D}$ of n samples $\left(h_T^{(i)}, a_T^{(i)}, r_T^{(i)}\right)$ from the distribution $p_\mu(H_T, A_T, R_T)$. To ensure that we can estimate V^π using observed data, we need to assume sequential ignorability and overlap.

Assumption 1 (Sequential ignorability)**.** *For all stages $t = 0, \ldots, T$ and for any sequence of actions $\bar{a}_T$, conditional on the history H_t, the action A_t generated by the behavior policy μ is independent of future potential outcomes $R_t(\bar{a}_t), S_{t+1}(\bar{a}_t), \ldots, S_T(\bar{a}_{t-1}), R_T(\bar{a}_T)$. We say that the behavior policy μ satisfies sequential ignorability.*

Assumption 2 (Overlap)**.** *For all pairs of actions $A_t \in \mathcal{A}$ and histories $H_t \in \mathcal{H}$, $p_\mu(A_t \mid H_t) > 0$ whenever $p_\pi(A_t \mid H_t) > 0$. We say that overlap holds between the target policy and the behavior policy.*

In addition, we assume that any uncertainty in the target policy $p_\pi\left(A_t \mid H_t\right)$ originates from an exogenous variable. Under these assumptions, the value of the target policy π can be written as

$$V^\pi \coloneqq \mathbb{E}_\pi\left[\sum_{t=0}^{T} R_t\right] = \mathbb{E}_\mu\left[W \sum_{t=0}^{T} R_t\right], \tag{2.2}$$

with importance weight

$$W \coloneqq \prod_{t=0}^{T} \frac{p_\pi\left(A_t \mid H_t\right)}{p_\mu\left(A_t \mid H_t\right)}. \tag{2.3}$$

In practice, the behavior policy $p_\mu\left(A_t \mid H_t\right)$ is often unknown. To compute the importance weight in Equation (2.3), we must first fit an estimator $\hat{p}_\mu\left(A_t \mid H_t\right)$ to the observed data. Now, we can compute a sample-based estimate of Equation (2.2), the importance sampling (IS) estimate:

$$\hat{V}^\pi = \frac{1}{n} \sum_{i=1}^{n} w_i \sum_{t=0}^{T} r_t^{(i)} \quad \text{with} \quad w_i = \prod_{t=0}^{T} \frac{p_\pi\left(a_t^{(i)} \mid h_t^{(i)}\right)}{\hat{p}_\mu\left(a_t^{(i)} \mid h_t^{(i)}\right)}.$$

If the target policy is deterministic, we replace $p_\pi\left(a_t^{(i)} \mid h_t^{(i)}\right)$ with the indicator $\mathbb{I}\left[a_t^{(i)} = \pi\left(h_t^{(i)}\right)\right]$.

By inspecting the importance weight in Equation (2.3), we note potential problems with the IS estimate. For example, if $\hat{p}_\mu\left(A_t \mid H_t\right)$ is small and $p_\pi\left(A_t \mid H_t\right)$ is large, the estimate suffers from high variance. This problem is often aggravated for long sequences. As a diagnostic, we can compute the effective sample size

$$n_e = \frac{\left(\sum_i w_i\right)^2}{\sum_i w_i^2},$$

which tells us how many samples contribute to the estimate (Owen, 2013). Ideally, n_e should be close to n, but it can be significantly less.

2.5 Interpretable Policy Representations

Once we have evaluated a new candidate policy with promising results, should we then feel safe to deploy the policy in practice? Not without caution. Ideally, we would also like to reason about how the new policy differs from the behavior policy. To enable such reasoning when the behavior policy is unknown, we need to model it with interpretability in mind. An interpretable model allows us to understand the behavior policy, and this understanding is crucial if we then want to examine how the behavior policy differs from the new policy.

Learning an interpretable representation of the behavior policy can also be a starting point for the process of developing a new policy for future decision-making based on observational data. To obtain reliable off-policy value estimates, we need to ensure that candidate policies have enough support in the observed data. An interpretable description of the current policy helps us to identify candidates that meet the criteria.

The most common types of interpretable machine learning models include logical models, for example, decision trees, and scoring systems, which essentially are linear models with integer coefficients; see (Rudin et al., 2022) for an overview. A potential drawback with these models is that they require a Markov state. They are less suitable for modeling $p(A \mid H)$, i.e., the probability of taking action A given the entire history H. Here, we focus on so-called prototype-based techniques, which are more flexible in this regard.

In prototype learning, the probability $p(A \mid H)$ is estimated by comparing the input history to a few prototypical cases from the training data; the closer the input history is to a certain prototype, the more that prototype influences the estimated probability. Being real training examples, the prototypes are interpretable by a domain expert. The prototype-based approach resembles how a clinician selects treatment for a patient based on the patient's similarities with previously treated patients. The prototypes, which are learned during training, can be represented as a prototype layer in a deep neural network (Chen et al., 2019; Li et al., 2018), allowing for flexible and powerful models. The encoding part of the network, which maps the input to the space of prototypes, can be a sequence learning model, e.g., a recurrent neural network, capable of handling sequential data (Ming et al., 2019).

Chapter 3

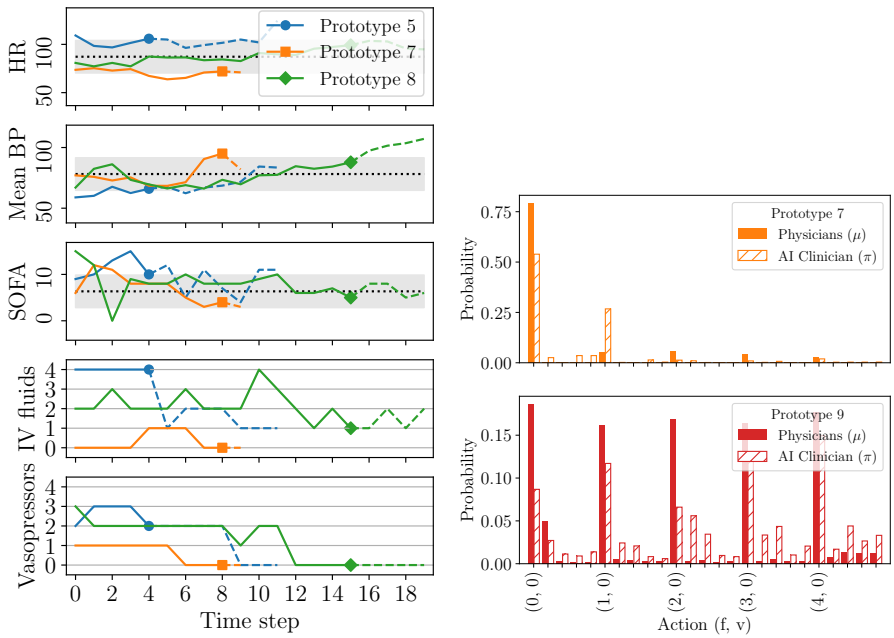
Summary of Included Papers

In this chapter, we summarize the two papers that are included in this thesis. Both papers are related to the process of developing new policies from observational data. The overall theme of the papers is to understand sequential decision-making at different stages of this process. We focus on applications in healthcare, where, for example, electronic health records provide a rich source of observational data.

3.1 Paper I

In Paper I, we study off-policy evaluation in more detail. We focus on importance sampling (IS) methods and the common case where the behavior policy is unknown and must be estimated from data. While IS-based methods are preferred due to their simplicity, they suffer from high variance when there are significant differences between target and behavior policies. The analyst can inspect individual IS-weights to detect outliers and compute the effective sample size to get an idea of the reliability of the estimate. However, these methods do not describe patterns in the difference between the policies. To reuse the healthcare example, it may not be clear for which patients, and at what stages, the treatment suggested by the target policy is different from that suggested by the behavior policy.

To address this problem, we propose estimating the unknown behavior policy $p_\mu(A \mid H)$ using prototype learning. The prototypes induce a soft clustering of the space of histories, and intuitively, each prototype is representative of a certain type of histories. By inspecting the behavior and target policies for each of the prototype cases, we get an overview of which action the policies would suggest in different key situations. A domain expert can reason about any differences between the policies and detect cases where the target policy suggests questionable actions. We can also divide estimated values into prototype-based components to describe situations where it would be beneficial to follow the



(a) Example prototypes. (b) Policy inspection.

Figure 3.1: In Paper I, we estimate the behavior policy using prototype learning. We then use the learned prototypes as a diagnostic tool for off-policy evaluation. Left: Visualization of three of the prototypes in the sepsis experiment. Each prototype corresponds to a real patient and is interpretable by a domain expert. Right: Comparison of the target policy, the "AI Clinician", with the behavior policy estimated from the data. We see that the target policy suggests actions that are rare under the behavior policy.

target policy instead of the behavior policy, and vice versa.

We demonstrate our framework in a real-world example of the management of sepsis in the intensive care unit. As target policy, we consider the so-called AI Clinician (Komorowski et al., 2018), which is learned from offline data using MDP estimation and value iteration, as described in Section 2.3. Using learned prototypes as a diagnostic tool, we identity groups of patients who at the first stage of treatment would have received an AI-recommended treatment that is rare in the observed data, see Figure 3.1(b), highlighting the need for a transparent comparison of the policies. A few of the prototypes are visualized in Figure 3.1(a).

While the prototypes give interpretability, they also pose a limitation to the model. We therefore compare the prototype model to standard baseline models and plain neural networks. Evaluating the models on the sepsis data, we find that the prototype model has the capacity to perform similarly to the other models. The key hyperparameter is the number of prototypes controlling the trade-off between interpretability and performance. In practice, we suggest

using a small number of prototypes when comparing the policies and reasoning about the data. Then, if needed, a more flexible model can be learned for the importance sampling estimator.

3.2 Paper II

In Paper II, we take a clinical direction and study the sequential treatment of patients with rheumatoid arthritis (RA) with disease-modifying anti-rheumatic drugs (DMARDs). Most RA patients receive a conventional synthetic DMARD (csDMARD) as their first treatment, and non-responders are further treated with biologic and targeted synthetic DMARDs (b/tsDMARDs), usually tumor necrosis factor inhibitors (TNFi) or Janus kinase inhibitors (JAKi) (Smolen et al., 2020). However, how to continue treatment after initial csDMARD failure is often unclear. As the number of available drugs grows, there is an increasing need to improve clinical guidelines. Paper II takes a first step in this direction by describing common treatment patterns in RA. As we have previously discussed, having a good understanding of current practice is often a prerequisite for developing new policies.

Using data from the CorEvitas RA registry (Kremer, 2016), we study 6015 b/tsDMARD-naïve patients, of whom 77 % are female, who started their first b/tsDMARD therapy, defined as the first line of therapy, between 2012 and the end of 2021. We define a treatment pattern as a unique sequence of therapy changes following and including the first-line therapy. Statistical estimates and visualizations are used to provide a quantitative picture of standard practice in the US, including first-line therapy selection and selection of sequential therapies. A strength of our study is that we consider the whole sequences of therapies and not only transitions between consecutive therapies, which is more common in the literature (Fletcher et al., 2022; Zhao et al., 2022).

As first-line therapy, most patients start with a TNFi therapy, but we observe a recent shift towards JAKi therapies. There is high variation in subsequent treatment choices, leading to a large number of treatment patterns, especially for longer sequences of therapies. Interestingly, we observe that therapy cycling, i.e., returning to a therapy from a previously used therapeutic class, is relatively common. We also find that the duration of the first lines of therapy decreased over the past decade.

Our results indicate that there is a wide variation in clinical practice. Such variation is necessary to evaluate new strategies that deviate from current guidelines. However, it also presents challenges to evaluate new policies with statistical reliability. For example, to evaluate later lines of treatment for RA, we would need to make assumptions about patients with different treatment histories to increase the amount of evaluation data. An alternative is to group patients with respect to the set of drugs they have been treated with, without regard to the actual order of drugs.

Chapter 4

Concluding Remarks and Future Directions

In this thesis, we studied sequential decision-making from the perspective of policy understanding and policy evaluation. The need for policy understanding is what ties this thesis together. In Paper I, we proposed learning an interpretable model of the behavior policy to *understand* how it differs from a target policy in off-policy evaluation. In Paper II, we took the first steps in developing new clinical guidelines for the treatment of rheumatoid arthritis by contributing with an *understanding* of current practice

As a continuation of the work presented in Paper II, we would like to understand current practice even better by describing the contextual behavior policy $p_\mu(A \mid H)$. The idea is to learn an interpretable model, which allows for verification by a domain expert. We also hope to use an interpretable representation of the behavior policy to identify new policy candidates in consultation with domain experts. Finally, we want to evaluate these candidate using off-policy evaluation methods, hopefully providing new insights into how patients with rheumatoid arthritis can be better treated.

One possibility is to model the behavior policy $p_\mu(A \mid H)$ for rheumatoid arthritis using prototype-based techniques as in Paper I. To obtain a well-calibrated model, however, it may be necessary to build domain knowledge into the model. For example, some treatments are not given to patients with certain comorbidities. It is not certain that such relationships are captured by, e.g., a logistic regression model, which is commonly used to estimate the behavior policy. An interesting question, which extends beyond the specific case of rheumatoid arthritis, is how such constraints can be learned from data, without requiring available domain expertise.

Another interesting direction is to study off-policy evaluation from a different perspective. Instead of asking if the available data allow us to evaluate a given policy, we can ask: Given these data, what are the policies that we can evaluate with statistical certainty? One possible approach is to identify policies that are sufficiently similar to the behavior policy.

Bibliography

Chakraborty, B. (2013). *Statistical methods for dynamic treatment regimes*. Springer. (Cit. on p. 7).

Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C., & Su, J. K. (2019). This looks like that: Deep learning for interpretable image recognition. *Advances in neural information processing systems*, *32* (cit. on pp. 4, 12).

Cully, A., Clune, J., Tarapore, D., & Mouret, J.-B. (2015). Robots that can adapt like animals. *Nature*, *521*(7553), 503–507 (cit. on p. 7).

Fawzi, A., Balog, M., Huang, A., Hubert, T., Romera-Paredes, B., Barekatain, M., Novikov, A., R Ruiz, F. J., Schrittwieser, J., Swirszcz, G., et al. (2022). Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, *610*(7930), 47–53 (cit. on p. 7).

Fletcher, A., Lassere, M., March, L., Hill, C., Barrett, C., Carroll, G., & Buchbinder, R. (2022). Patterns of biologic and targeted-synthetic disease-modifying antirheumatic drug use in rheumatoid arthritis in Australia. *Rheumatology*, *61*(10), 3939–3951 (cit. on p. 15).

Jiang, N., & Li, L. (2016). Doubly robust off-policy value evaluation for reinforcement learning. *International Conference on Machine Learning*, 652–661 (cit. on p. 4).

Koller, D., & Friedman, N. (2009). *Probabilistic graphical models: Principles and techniques*. MIT press. (Cit. on p. 5).

Komorowski, M., Celi, L. A., Badawi, O., Gordon, A. C., & Faisal, A. A. (2018). The Artificial Intelligence Clinician learns optimal treatment strategies for sepsis in intensive care. *Nature medicine*, *24*(11), 1716–1720 (cit. on pp. 8, 14).

Kremer, J. M. (2016). The Corrona US registry of rheumatic and autoimmune diseases. *Clinical and Experimental Rheumatology*, *34*(5 (Suppl. 101)), S96–S99 (cit. on p. 15).

Kumar, A., Fu, J., Soh, M., Tucker, G., & Levine, S. (2019). Stabilizing off-policy Q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, *32* (cit. on p. 9).

Levine, S., Kumar, A., Tucker, G., & Fu, J. (2020). Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643* (cit. on p. 9).

Li, O., Liu, H., Chen, C., & Rudin, C. (2018). Deep learning for case-based reasoning through prototypes: A neural network that explains its pre-

dictions. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*(1) (cit. on pp. 4, 12).

Mandel, T., Liu, Y.-E., Levine, S., Brunskill, E., & Popovic, Z. (2014). Offline policy evaluation across representations with applications to educational games. *AAMAS*, *1077* (cit. on p. 4).

Mannor, S., Simester, D., Sun, P., & Tsitsiklis, J. N. (2007). Bias and variance approximation in value function estimates. *Management Science*, *53*(2), 308–322 (cit. on p. 4).

Ming, Y., Xu, P., Qu, H., & Ren, L. (2019). Interpretable and steerable sequence learning via prototypes. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 903–913 (cit. on p. 12).

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing Atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (cit. on p. 7).

Owen, A. B. (2013). *Monte carlo theory, methods and examples.* Stanford. (Cit. on p. 11).

Prasad, N., Cheng, L.-F., Chivers, C., Draugelis, M., & Engelhardt, B. E. (2017). A reinforcement learning approach to weaning of mechanical ventilation in intensive care units. *33rd Conference on Uncertainty in Artificial Intelligence* (cit. on p. 8).

Precup, D. (2000). Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, 80 (cit. on p. 4).

Robins, J. M. (1997). Causal inference from complex longitudinal data. *Latent variable modeling and applications to causality*, 69–117 (cit. on p. 10).

Rubin, D. B. (2005). Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association*, *100*(469), 322–331 (cit. on p. 10).

Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L., & Zhong, C. (2022). Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistic Surveys*, *16*, 1–85 (cit. on p. 12).

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. (2018). A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, *362*(6419), 1140–1144 (cit. on p. 7).

Smolen, J. S., Landewé, R. B., Bijlsma, J. W., Burmester, G. R., Dougados, M., Kerschbaumer, A., McInnes, I. B., Sepriano, A., Van Vollenhoven, R. F., De Wit, M., et al. (2020). EULAR recommendations for the management of rheumatoid arthritis with synthetic and biological disease-modifying antirheumatic drugs: 2019 update. *Annals of the rheumatic diseases*, *79*(6), 685–699 (cit. on p. 15).

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction.* MIT press. (Cit. on pp. 8, 9).

Tewari, A., & Murphy, S. A. (2017). From ads to interventions: Contextual bandits in mobile health. *Mobile Health: Sensors, Analytic Methods, and Applications*, 495–517 (cit. on p. 8).

Zhao, S. S., Kearsley-Fleet, L., Bosworth, A., Watson, K., Group, B.-R. C., & Hyrich, K. L. (2022). Effectiveness of sequential biologic and targeted disease modifying anti-rheumatic drugs for rheumatoid arthritis. *Rheumatology*, *61*(12), 4678–4686 (cit. on p. 15).