REINFORCEMENT LEARNING FOR NON-STATIONARY PROBLEMS

A Dissertation Presented

by

YASH CHANDAK

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2022

Robert and Donna Manning College of
Information and Computer Sciences

REINFORCEMENT LEARNING FOR NON-STATIONARY PROBLEMS

A Dissertation Presented

by

YASH CHANDAK

Approved as to style and content by:

Philip S. Thomas, Chair

Bruno Castro da Silva, Member

Shlomo Zilberstein, Member

Emma Brunskill, Member

James Allan, Chair of the Faculty
Robert and Donna Manning College of
Information and Computer Sciences

ABSTRACT

REINFORCEMENT LEARNING FOR NON-STATIONARY PROBLEMS

MAY 2022

YASH CHANDAK

B.Tech., VELLORE INSTITUTE OF TECHNOLOGY, CHENNAI

M.Sc., UNIVERSITY MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Philip S. Thomas

Reinforcement learning (RL) has emerged as a general-purpose technique for addressing problems involving sequential decision-making. However, most RL methods are based upon the fundamental assumption that the transition dynamics and reward functions are fixed, that is, the underlying Markov decision process is stationary. This limits the applicability of such RL methods because real-world problems are often subject to changes due to external factors (*passive* non-stationarity), or changes induced by interactions with the system itself (*active* non-stationarity), or both (*hybrid* non-stationarity). For example, personalized automated healthcare systems and other automated human-computer interaction systems need to constantly account for changes in human behavior and interests that occur over time. Further, when the stakes associated with financial risks or human life are high, the cost associated with

a false stationarity assumption may be unacceptable. In this work, we address several challenges underlying (off-policy) policy evaluation, improvement, and safety amidst such non-stationarities. Our approach merges ideas from reinforcement learning, counterfactual reasoning, and time-series analysis.

When the stationarity assumption is violated, using existing algorithms may result in a performance lag and false safety guarantees. This raises the question: how can we use historical data to optimize for future scenarios? To address this challenges in the presence of *passive* non-stationarity, we show how future performance of a policy can be *evaluated* using a forecast obtained by fitting a curve to counter-factual estimates of policy performances over time, without ever directly modeling the underlying non-stationarity. We show that this approach further enables policy *improvement* to proactively search for a good future policy by leveraging a policy gradient algorithm that maximizes a forecast of future performance. Building upon these advances, we present a Seldonian algorithm that provides the first steps towards ensuring safety, with high confidence, for smoothly-varying non-stationary decision problems.

The presence of *active* and *hybrid* non-stationarity pose additional challenges by exposing a completely new feedback loop that allows an agent to potentially control the non-stationary aspects of the environment. This makes the outcomes of future decisions dependent on all of the past interactions, thereby resulting in effectively a *single* lifelong sequence of decisions. We propose a method that provides the first steps towards a general procedure for on-policy and off-policy evaluation amidst structured changes due to active, passive, or hybrid non-stationarity.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

> Intelligence is the ability to adapt to change.
>
> Stephen Hawkins

Throughout the history of evolution it can be observed that a hallmark of intelligence has been the ability to adapt to changes. As we strive to build systems that exhibit characteristics of intelligence, an important research challenge is to develop methods that can autonomously adapt to and proactively reason about changes that will occur in the future. This dissertation takes a step towards addressing this challenge.

One need not have a vivid imagination to see the advantages of AI systems that showcase such characteristics. Even current basic AI systems have enormous potential, and the way we address this challenge will pave the way for future AI systems. For example, in recent years, there has been a surge of interest in developing automated *sequential decision making* algorithms for a wide variety of real-world applications. To highlight one, consider prior work that proposed using sequential decision making methods to provide automated healthcare for patients (Yu et al., 2019). Because it is such a high-stakes application, researchers have also proposed developing methods that provide high-confidence safety guarantees on the performance of such algorithms (Thomas et al., 2019a). While these methods open exciting new avenues towards high-impact applications, it is vital to understand the assumptions made in these works. Particularly, the results established by prior works rely on the assumption that the problem is stationary, i.e., (a) when considering patients individually, the physiology

and behavior of the patient remains fixed across days, or (b) when considering the patient *population*, healthcare facilities and public health remain fixed across time.

Is this assumption valid for such real-world problems? If not, what are the consequences for a patient when algorithms blindly rely on such assumptions?

Clearly such assumptions of stationarity are often violated. (a) When considering patients *individually*, notice that age is an important aspect that changes constantly, never returning to a previous value. This results in changes in physiology and behavior of a patient with age. (b) At the *population* level, when considering data collected over extended periods, not only do healthcare facilities change over time, but public health also continuously evolves based on the treatments made available in the past.

Violations of the stationarity assumption can be observed across a plethora of applications. For example, many medical support systems for the treatment of health problems like type-1 diabetes (Bastani, 2014), sepsis (Saria, 2018), HIV (Ernst et al., 2006), etc. involve sequential decision making under conditions similar to those discussed above. Given the high stakes of such applications, the cost associated with a false stationarity assumption may be unacceptable, necessitating the development of algorithms that can adequately account for non-stationarity. Further, almost all human-computer interaction systems have a common non-stationary component: humans. In tutoring systems, a student's behavior changes over time. For personalized music and video recommendation systems, a user's interests change. In driving assistance systems, a driver's response to warnings changes depending on how frequently the system correctly and incorrectly alerts the driver. In robotic control applications, motors and joints suffer wear and tear over time that can change the dynamics of a robot. Lastly, power management systems need to account for non-stationarity at multiple scales for power supply and demand, ranging from day-night fluctuations to weekday-weekend fluctuations, to yearly seasonal fluctuations.

These examples capture the broad idea that for a system that is deployed in the real world, parts of the problem specification change over time, and *will* violate the stationarity assumption. The goal of this dissertation is to address challenges that stem from non-stationarity and develop methods that inform an array of such applications.

1.1 Contributions

This dissertation makes three main contributions, each organized as a chapter.

1. The primary contribution of Chapter 3 is to present a *Prognosticator* procedure that, in the presence of structured non-stationarity due to external factors, can (a) provide a model-free estimate of the performance of a policy if that policy were to be deployed in the future, and (b) proactively search for a good future policy through a gradient based procedure that maximizes estimates of the *future* performance. Perhaps surprisingly, we observe that *minimizing* performance at some times in the past can be beneficial when searching for a policy that *maximizes* future performance. We also show how *Prognosticator* is an unbiased and a strongly consistent estimator in the stationary setting, thereby generalizing several existing methods for the stationary setting.

2. The primary contribution of Chapter 4 is to formalize the conditions under which safety[1] can be ensured in the presence of structured non-stationarity due to external factors. Under these conditions we propose *SPIN*, the first procedure for safe policy improvement under such non-stationarities. SPIN first constructs asymptotically valid confidence intervals of a policy's future performance and

[1]Safety has many definitions in the literature. Later we formally define what *safety* means in this dissertation. In short, we say an algorithm is safe if it provides a high-confidence guarantee that it will only change the current policy when doing so would increase the expected discounted return in the future.

then searches for a policy that maximizes the lower bound obtained from this confidence interval. Empirically, we observe that SPIN provides safe policy improvement even in the finite sample setting and even when the structure resulting from non-stationarity is misspecified. In comparison, existing methods for ensuring safety that do not account for non-stationarity result in up to five times more unsafe behavior than desired.

3. The primary contribution of Chapter 5 is to account for a more general class of non-stationarity, where the changes may occur due to both external factors and due to the past decisions made by the agent. This setting is particularly challenging as it exposes a completely new feedback loop that allows an agent to influence and control the non-stationary aspects of the environment. In this setting, we formalize the fundamental problem of (off-policy) policy evaluation, establish additional assumptions for tractability, and propose a method, *OPEN*, to address this challenge. OPEN presents the first steps towards a unified procedure that can tackle general forms of structured non-stationarities (while remaining effective in the stationary setting).

1.2 Layout

The remainder of this dissertation is structured as follows,

- **Chapter 2** (Background and Related Work) This chapter provides background on different types of non-stationarities and the roots of the challenges that arise when dealing with non-stationary problems. It also sets up notation relevant for all of the subsequent chapters and discusses work related to the overarching topic of non-stationarity.

- **Chapter 3** (Optimizing for the Future) This chapter first reviews existing work on off-policy evaluation in the stationary setting and then introduces the core

idea for how the stationary methods can be generalized to the non-stationary setting. The ideas developed in this chapter are foundational and form the pre-requisite for the following chapters.

- **Chapter 4** (Towards Safe Policy Improvement) This chapter reviews existing literature for ensuring safety in the stationary setting and techniques for obtaining confidence intervals for time-series. The proposed method merges concepts from these past methods with the foundations laid in Chapter 3. Subsequent chapters remain coherent if this chapter is skipped.

- **Chapter 5** (Action-dependent Non-stationarity) Building upon the foundations laid in Chapter 3, this chapter proposes a generalized method to tackle different forms of structured non-stationarity. The contributions in this chapter are the culmination of this thesis.

- **Chapter 6** (Conclusion and Future Work) This brief chapter summarizes the dissertation and proposes directions for future research.

CHAPTER 2

BACKGROUND AND RELATED WORK

This chapter first discusses the roots of the problems that stem from non-stationarity and introduces the notation for non-stationary decision processes that will be used throughout this dissertation. We then summarize prior work related to the overarching problem of non-stationarity.

2.1 Roots of the Problem

Reinforcement learning (RL) approaches have emerged as a ubiquitous class of methods for tackling decision-making problems. However, the foundations of RL are built upon the assumption that the problem specification is stationary. That is, RL work typically assumes that a decision *always* results in the same (distribution of) consequence(s) when taken at any given state. While this assumption was a cornerstone when laying the theoretical foundations of the field and developing initial RL algorithms, it is rarely true for real-world problems. Hence, it is imperative that we remove this assumption if we hope to transition towards practical algorithms. However, being such a fundamental assumption, it is so deeply rooted within the current state-of-the-art methods that it is often not even explicitly mentioned.

To understand the roots of this problem, let us revisit the use of RL algorithms for type 1 diabetes management (Bastani, 2014). For this application, the RL agent must decide how much insulin should be injected to keep a patient's blood glucose levels near-ideal levels. Unfortunately, successful use of popular RL algorithms here is contingent on one of the following two instantiations of the stationarity assumption:

(a) existence of a fixed resetting procedure for sampling independent *episodes*, or (b) existence of a *stationary distribution* for the Markov chain induced by executing any policy. For the first condition, when considering treatment for a single patient, one way for resetting might be to consider each day as a single episode (Thomas et al., 2019a). However, because the blood glucose level at any day is dependent on the insulin injected during the previous day, the independence assumption is violated. Additionally, as human physiology keeps changing with age, the effective transition dynamics for the Markov chain induced by any decision making rule keeps changing. This further inhibits any policy from reaching a stationary distribution during a patient's lifetime, thereby violating the second condition as well.

2.2 Partially Observable Markov Decision Processes

In this dissertation, random variables are denoted using capital letters and sets are denoted using calligraphic letters. For example, $X \in \mathcal{X}$, is a random variable taking on values in the set $\mathcal{X}$ with $|\mathcal{X}|$ elements. Below we define common functions used in this dissertation.

Throughout the dissertation, for simplicity of notation, we will consider partially observable Markov decision process (POMDPs), defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathscr{O}, \mathscr{P}, \mathscr{R}, \mu)$ with which an agent interacts for a finite number of time steps T (Kaelbling et al., 1998). Here $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, and $\mathcal{O}$ is the set of observations. While all of our results extend to the setting where the state set $\mathcal{S}$, observation set $\mathcal{O}$, and the action set $\mathcal{A}$ are continuous (or of infinite cardinality), we assume that these sets are finite for notational simplicity. When an agent interacts with an environment modeled as a POMDP, $S_t \in \mathcal{S}$, $O_t \in \mathcal{O}$, $A_t \in \mathcal{A}$, and $R_t \in \mathbb{R}$ correspond to the random variables for the state, observation, action, and reward at time step $t \in \{0, 1, ..., T\}$. The function $\mathscr{O} : \mathcal{S} \times \mathcal{O} \to [0, 1]$ is the observation function which (stochastically) maps states to observations, i.e., $\mathscr{O}(s, o) \coloneqq \Pr(O_t = o | S_t = s)$. The *transition function*

$\mathscr{P}\colon \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ specifies the probability of the agent transitioning to a state s' when taking action a in state s, i.e., $\mathscr{P}(s, a, s') \coloneqq \Pr(S_{t+1} = s' | S_t = s, A_t = a)$. The reward distribution $\mathscr{R}\colon \mathcal{S} \times \mathcal{A} \times \mathcal{S} \times \mathcal{R} \to [0, 1]$, specifies the distribution over rewards the agent receives for taking action a in state s, and transitioning to s', i.e., $\mathscr{R}(s, a, s', r) = \Pr(R_t = r | S_t = s, A_t = a, S_{t+1} = s')$, where $\mathcal{R} \subseteq [-R_{\max}, R_{\max}]$ for some finite constant $R_{\max}$.[1] The *initial state distribution* is defined by $\mu\colon \mathcal{S} \to [0, 1]$, i.e., $\mu(s) = \Pr(S_0 = s)$.

The interaction proceeds as the following: S_0 is sampled from μ and mapped to O_0 using the observation function $\mathscr{O}$. The agent uses O_t, for all $t \in \{0, 1, ..., T\}$. On executing action A_t, S_t transitions to state S_{t+1} under the transition dynamics $\mathscr{P}$, and the agent is provided with a reward R_t generated using $\mathscr{R}$ and the observation O_{t+1}. After T steps of interaction, the state is reset to a starting state using μ and the entire process repeats. The method that an agent uses to select actions is called a *policy*. We define a policy to be a function $\pi\colon \mathcal{O} \times \mathcal{A} \to [0, 1]$, which specifies the probability of the agent taking actions a upon observing o, i.e., $\pi(o, a) \coloneqq \Pr(A_t = a | O_t = o)$. For policy optimization problems, the agent will use a *parameterized policy* $\pi_\theta : \mathcal{S} \times \mathcal{A} \times \mathbb{R}^n \to [0, 1]$, which is also a function of d parameters $\theta \in \mathbb{R}^d$. For brevity of notation and when clear from context, the parameters θ may be left implicit, e.g., $\Pr(A_t = a | O_t = o) = \Pr(A_t = a | O_t = o; \theta)$.

We call the discounted sum of rewards the *return*, i.e., $G = \sum_{t=0}^{T} \gamma^t R_t$, where $\gamma \in [0, 1]$ is the *discount factor*. The start sate objective for a policy π at a given state s is $J(\pi) \coloneqq \mathbb{E}_\pi[G]$, where the subscript of π denotes that π was used to select actions during interaction with the POMDP.

---

[1] For notational simplicity, we assume that $\mathcal{R}(s, a, s', \cdot)$ has finite support for all s, a, and s'. This allows us to sum over possible trajectories and discuss probabilities of trajectories rather than using the more general but complex measure theoretic notation for probability.

2.3 Non-stationary Decision Processes

We define a non-stationary decision process (NSDP) as a sequence of POMDPs. Formally, let $\mathcal{M}$ be a finite set of POMDPs. The observation function $\mathscr{O}_i$, the transition function $\mathscr{P}_i$, the reward function $\mathscr{R}_i$, and the initial state distribution μ_i may differ for each POMDP M_i. For clarity, we will use O_i^t, A_i^t, and R_i^t to denote the random variables corresponding to the observation, action, and reward at timestep t when the agent interacts with POMDP M_i. Let $H_i \coloneqq (S_i^t, O_i^t, A_i^t, R_i^t)_{i=0}^T$ be the entire sequence of interactions in M_i. Similarly, let $G_i \coloneqq \sum_{t=0}^T R_i^t$ be an observed return and $J_i(\pi) \coloneqq \mathbb{E}_\pi[G_i|M_i]$ be the performance of π on M_i. Let $\mathcal{H}$ be the set of possible interaction sequences, and finally let $\mathcal{T} : \mathcal{M} \times \mathcal{H} \times \mathcal{M} \to [0, 1]$ be the 'meta-transition' function that governs the non-stationarity in the POMDPs. That is, $\mathcal{T}(m, h, m') = \Pr(M_{i+1}{=}m'|M_i{=}m, H_i{=}h)$.

The interaction with an NSDP proceeds as the following: Agent is first presented with a POMDP M_0, with which it interacts for T steps (one episode). Upon termination of interaction with M_i, instead of resetting the state using the initial state distribution of POMDP M_i, the state is reset using the initial state distribution μ_{i+1} of the next POMDP M_{i+1}, and the process continues. In other words, i indexes the episode and there is a different POMDP for each episode, i.e., M_i is the POMDP the agent interacts with during episode i. Importantly, this is a lifelong process which need not ever reset back to M_0.

We provide an illustration of the control process in Figure 2.1. In the stationary POMDP setting, resampling from the starting state distribution permits interacting with the same POMDP multiple times, making finding a policy that is effective in the future possible. However, for an NSDP learning a good policy for the future, or even evaluating a policy's future performance, would be intractable without additional assumptions, as the future POMDP could be arbitrary and very different from the POMDPs the agent has interacted with so far. To make the problem tractable, we

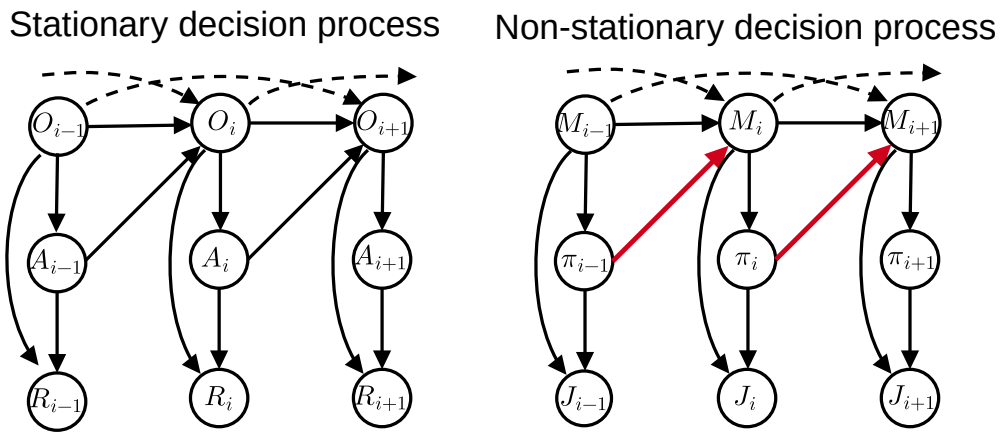


Figure 2.1. **(Left)** Control graph for interaction in a stationary POMDP, where each column corresponds to one time step. Here, *independent* episodes from the *same* POMDP can be resampled using μ. **(Right)** Control graph that we consider for a non-stationary decision process, where each column corresponds to one episode. Here, the agent interacts with a sequence of related POMDPs. In the absence of red arrows, the change from M_i to M_{i+1} is independent of the past decisions and is governed only by external factors (passive non-stationarity). Presence of red arrows indicate that M_{i+1} can *also* be dependent on the past decisions made in M_i (active non-stationarity).

will need some structural assumptions on the POMDPs that will enable an agent to use data from past interactions to infer potential outcomes in the future. We will discuss such structural assumptions in the later chapters.

Using terminology introduced by Khetarpal et al. (2020), non-stationarity can be further categorized as passive, active or hybrid. Below we provide intuitive and mathematical formulations of these three settings (as well as the stationary setting).

2.3.1 Stationarity

It can be immediately observed that the stationary POMDP setting is a special case of the NSDP setting. Specifically, when the size of the set of possible POMDPs $|\mathcal{M}| = 1$, then

$$\forall i, j > 0,\ M_i = M_j,$$

and all H_i's are sampled from the same POMDP. Further, if $O_t = S_t$ for all t then the setting reduces to the stationary Markov decision process (MDP) setting (Puterman, 1990; Sutton and Barto, 2018b).

2.3.2 Passive Non-stationarity

When the non-stationarity is caused only by external (exogenous) factors, then we refer to it as *passive* non-stationarity. That is, an agent's interactions with past POMDPs do not influence the POMDPs that the agent will face in the future. Formally, regardless of which policy is used to select actions,

$$\forall(m, m') \in \mathcal{M}^2, \forall(h, h') \in \mathcal{H}^2, \qquad \mathcal{T}(m, h, m') = \mathcal{T}(m, h', m'). \tag{2.1}$$

For example, consider a product recommendation system interacting with a user (Theocharous et al., 2020). If the interactions during each day i are interactions with POMDP M_i, i.e., each day corresponds to an episode, then seasonal changes across days that cause change in the user's interests for different products result in passive non-stationarity. As another example, consider social media platforms that provide personalised content recommendations to its users. Because the relevance of content constantly changes based on external events, e.g., elections and sporting events, the recommender systems need to constantly account for the non-stationarity of the user's interests.

2.3.3 Active (Action-dependent) and Hybrid Non-stationarity

When the non-stationarity is dependent on the past interactions of the agent with the environment, we refer to it as *active* (or action-dependent) non-stationarity. In the most general form, non-stationarity can be dependent on both external changes and past interactions of the agent. We refer to this general form of non-stationarity

as *hybrid* non-stationarity, which is modeled by the meta-transition function $\mathcal{T} : \mathcal{M} \times \mathcal{H} \times \mathcal{M} \to [0, 1]$ without additional restrictions like (2.1).

As an example of hybrid non-stationarity, social media platforms need to constantly account for the partisan biases of their users that changes not only due to both external political developments but also from increased self-validation resulting from previous posts/ads suggested by the recommender system itself (Cinelli et al., 2021; Gillani et al., 2018).

2.4 Related Work

In the following we summarize various related directions that fall under the overarching topic of non-stationarity in reinforcement learning. A more exhaustive survey can be found in the works by Padakandla (2020) and Khetarpal et al. (2020).

2.4.1 (Stationary) POMDPs

It might often seem natural to consider the factors that induce non-stationarity (e.g., body's glucose absorption rate during automated diabetes treatment as that rate changes with age, or a user's partisan bias during social media recommendations) as an *unobserved* variable and model the problem as a *partially-observable Markov decision process* (POMDPs). While a general POMDP might be adequate to *model* the problem setup, wherein the agent's interactions with the environment correspond to one long episode, often the end goal is not to model the problem but to obtain an optimal policy for it (or provide safety guarantees, etc.). Consequently, searching for an optimal policy requires a policy search algorithm. Unfortunately, the success of these search algorithms is typically contingent on the additional stationarity assumption that either independent episodes can be sampled, or transitions from a stationary distribution can be sampled (Ghavamzadeh et al., 2016). Neither of these is true when a non-stationary episodic problem is modeled as a POMDP with one long episode.

Therefore, just modeling the problem as a general POMDP is not sufficient to tackle non-stationarity, unless we also have algorithms that do not rely upon stationarity of the initial state distribution, transition function, and/or reward function across episodes.

For instance, in the episodic setting, a typical POMDP would require that the starting state is drawn i.i.d. from a fixed distribution. In the diabetes treatment example above, where the body's glucose absorption rate is the unobserved variable, a fixed starting state distribution would imply that the glucose absorption rate is drawn i.i.d. at the start of the episode. However, as the patient's physiology changes with age and induces non-stationarity, this i.i.d. condition may be *invalid* as the glucose absorption rate will constantly drift across time.

In contrast, instead of the episodic setting, it might be natural to consider the continuing/average-reward setting as well. Typically, this setting requires assumptions of ergodicity (Puterman, 1990) of the domain such that the Markov chain induced by any policy reaches a stationary distribution. In the diabetes treatment example above the patient's physiology changes constantly with age and it is not possible to revisit a past age, which results in violation of the ergodicity assumption.

Additionally, POMDP methods employ some procedure to infer the unobserved variables to estimate the underlying true state. However, *even* if an oracle provides access to the complete state (which includes the unobserved variables), it can be observed that the above problems remain unadressed (as the starting states will not be drawn i.i.d. in the episodic setting, nor will the ergodicty assumption hold in the continuing setting). This problem is further exacerbated when only off-policy data is available. Inferring unobserved variables of a POMDP in the off-policy setting is equivalent to inferring latent confounders from observational data during causal inference. Unless strong assumptions are made, it is known that it may not be possible to consistently estimate the confounding variables even with infinite data (Pearl

et al., 2000). In this thesis, we will look at methods that never require inferring the unobserved/confounding variables.

Further, in the non-stationary setting, not only the type of distribution over the unobserved variable, but also the support of those distributions may change over time, i.e., the system might encounter environments that have new values for the unobserved variables. For example, In the diabetes treatment example above, with age the patient's blood glucose absorption rate might drift to values that the system might not have encountered in the past. Tackling such settings would inevitably require the ability to generalize. To models such settings, we build upon the (stationary) POMDP setup and provide the setup for non-stationary decision processes in Chapter 2.3.

2.4.2 Algorithmic Non-stationarity in Stationary Domains

In the face of uncertainty, prior works often opt for exploratory or safe behavior by acting optimistically or pessimistically, respectively. This is often achieved by using the collected data to dynamically modify the observed rewards for any state-action pair by either providing bonuses (Agarwal et al., 2020; Taiga et al., 2021) or penalties (Buckman et al., 2020; Cetin and Celiktutan, 2021). One could view this as an instance of active non-stationarity. Similarly, in temporal-difference (TD) methods the target for the value function keeps changing and such changes are also dependent on the data collected in the past (Sutton and Barto, 2018a). However, we note that such non-stationarities are only artifacts of the learning algorithm as the underlying domain remains stationary throughout. In contrast, the focus of our work is on settings where the underlying domain is non-stationary.

2.4.3 Meta and Continual Learning

The problem of adapting to non-stationarity is also related to continual learning (Ring, 1994), lifelong-learning (Thrun, 1998), and meta-learning (Schmidhuber, 1999). Several meta-learning based approaches for fine-tuning a (mixture of) trained model(s)

using samples observed during a similar task at test time have been proposed (Nagabandi et al., 2018a;b). Other works have also shown how models of the environment can be useful for continual learning (Lu et al., 2019) or for model predictive control (Wagener et al., 2019).

A work that is more closely related (to our contribution in Chapter 3) is that of Al-Shedivat et al. (2017). They consider a setting where an agent is required to solve test tasks that have different transition dynamics than the training tasks. Using meta-learning, they aim to use training tasks to find an initialization vector for the policy parameters that can be quickly fine-tuned when facing tasks in the test set. In many real-world problems, however, access to such independent training tasks may not be available *a priori*. In this work, we are interested in the continually changing setting where there is no boundary between training and testing tasks. As such, we show how their proposed online adaptation technique that fine-tunes parameters, by discarding past data and only using samples observed online, can create performance lag and can therefore be data-inefficient. In settings where training and testing tasks do exist, our method can be leveraged to better adapt during test time, starting from any desired parameter vector.

Recent work by Finn et al. (2019) aims at bridging both the continuously changing setting and the train-test setting for supervised-learning problems. They propose continuously improving an underlying parameter initialization vector and running a Follow-The-Leader (FTL) algorithm (Shalev-Shwartz et al., 2012) every time new data is observed. A naive adaption of this for RL would require access to all the underlying MDPs in the past for continuously updating the initialization vector, which would be impractical. Doing this efficiently remains an open question and our method is complementary to choosing the initialization vector. Additionally, FTL based adaptation always lags in tracking optimal performance as it uniformly maximizes performance over all the past samples that might not be directly related to

the future. Further, in Chapter 3 we show that by explicitly capturing the trend in the non-stationarity, we can mitigate this performance lag resulting from the use of an FTL algorithm during the adaptation process.

More importantly, in many real-world applications, it can be infeasible to update the system frequently if it involves high computational or monetary expense. In such a case, even optimizing for the immediate future might be greedy and sub-optimal. The system should optimize for a longer term in the future, to compensate for the time until the next update is performed. None of the prior approaches mentioned above can efficiently tackle this problem.

2.4.4 Multi-Agent Systems and Games

Non-stationarity also occurs in multiplayer games, like rock-paper-scissors, where each episode is a single one-step interaction (Singh et al., 2000; Bowling, 2005; Conitzer and Sandholm, 2007) and the opponent can change their strategy as a response to the agent's previous decisions. These types of changes are related to active non-stationarity (which we consider in Chapter 5). In such games, opponent modeling has been shown to be useful and regret bounds for multi-player games have also been established (Zhang and Lesser, 2010; Mealing and Shapiro, 2013; Foster et al., 2016; Foerster et al., 2018). Efficiently learning *sequential* strategies in a non-stationary setting is still an active research problem. Further, often these games still assume that the underlying system/environment (excluding other players) is stationary and focus on searching for (Nash) equilibria. However, under general non-stationarity, the underlying system may also change and thus there may not even exist any fixed equilibria.

Perhaps a more relevant setting would be that of evolutionary games, where the pay-off matrix and specification of the game can change over time. In such settings, methods involving replicator dynamics (Hennes et al., 2019) have been used to adapt to the changed game. Such methods, however, do not leverage any underlying structure

in how the game is changing nor do they account for settings where the changes might be a consequence of past interactions of the agent.

2.4.5 Hidden-Parameter MDP

A Hidden-Mode MDP is an alternate setting that assumes that the environment changes are confined to a small number of hidden modes, where each mode represents a unique MDP. This provides a more tractable way to model a limited number of MDPs (Choi et al., 2000; Basso and Engel, 2009; Mahmud and Ramamoorthy, 2013), or perform updates using mode-change detection (Da Silva et al., 2006; Alegre et al., 2021).

Hidden-parameter (HiP) MDPs (Doshi-Velez and Konidaris, 2016) build upon this direction by assuming that there exist hidden real-valued features that parameterize the MDP. Changes in these features cause the changes in the environment. To tackle non-stationarity, Xie et al. (2020a) proposed modeling the problem as a HiP-MDP and estimating the hidden parameter from the observed trajectories. Our work provides a complementary perspective by using purely model-free approach that does not require inferring or modeling any environment parameters.

2.4.6 Tracking

Tracking has also been shown to play an important role in non-stationary domains. Thomas et al. (2017) and Jagerman et al. (2019b) have proposed policy evaluation techniques for the passive non-stationary setting by tracking a policy's past performances. However, they do not provide any procedure for searching for a good future policy. To adapt quickly in non-stationary tasks, TIDBD (Kearney et al., 2018) and AdaGain (Jacobsen et al., 2019) perform TD-learning while also automatically (de-)emphasizing updates to (ir)relevant features by modulating the learning rate of the parameters associated with the respective features. Similarly, Abdallah and Kaisers (2016) propose

repeating a Q-value update inversely proportional to the probability with which an action was chosen to obtain a transition tuple.

For episodic non-stationary MDPs, researchers have also looked at providing regret bounds for algorithms that exploit oracle access to the current reward and transition functions (Even-Dar et al., 2005; Yu and Mannor, 2009; Abbasi et al., 2013; Lecarpentier and Rachelson, 2019; Li et al., 2019). Alleviating oracle access by performing a count-based estimate of the reward and transition functions based on the recent history of interactions has also been proposed (Gajane et al., 2018; Cheung et al., 2019). For tabular MDPs, past data from a non-stationary MDP can be used to construct a maximum-likelihood estimate model (Ornik and Topcu, 2019) or a full Bayesian model (Jong and Stone, 2005) of the transition dynamics. Our focus is on the setting which is not restricted to tabular representations. Further, we go beyond tracking and proactively optimize for the future.

2.4.7 One-step Decision Making

Non-stationary multi-armed bandits (NMAB) capture the setting where the horizon length is one, but the reward distribution changes over time (Moulines, 2008; Besbes et al., 2014). Many variants of NMAB, like *cascading non-stationary bandits* (Wang et al., 2019b; Li and de Rijke, 2019) and *rotting bandits* (Levine et al., 2017; Seznec et al., 2018) have also been considered. In optimistic online convex optimization, researchers have shown that better performance can be achieved by updating the parameters using predictions (which are based on the past gradients) of the gradient of the future loss (Rakhlin and Sridharan, 2013; Yang and Mohri, 2016; Mohri and Yang, 2016; Wang et al., 2019a). In contrast, the focus of this dissertation is on methods for sequential decision making.

2.4.8 Operations Research

In the operations research community, many dynamic sequential decision-making problems are modeled using infinite horizon *non-homogeneous* MDPs (Hopp et al., 1987). While estimating an optimal policy is infeasible under an infinite horizon setting when the dynamics are changing and a stationary distribution cannot be reached, several researchers have studied the problem of identifying sufficient forecast horizons for performing near-optimal planning (Garcia and Smith, 2000; Cheevaprawatdomrong et al., 2007; Ghate and Smith, 2013) or robust policy iteration (Sinha and Ghate, 2016). These methods often require either a known model or a procedure for estimating the entire model, which could be prohibitively difficult for many applications of interest.

CHAPTER 3

OPTIMIZING FOR THE FUTURE

Policy optimization algorithms in RL are promising for obtaining general purpose control algorithms. However, as discussed earlier, most existing algorithms assume that the transition dynamics and reward functions are fixed, that is, the underlying decision process is stationary. This assumption is often violated in practical problems of interest. For example, consider an assistive driving system. Over time, tires suffer from wear and tear, leading to changes in friction. Similarly, in almost all human-computer interaction applications, e.g., automated medical care, dialogue systems, and marketing, human behavior changes over time. In such scenarios, if the automated system is not adapted to take such changes into account, or if it is adapted only after observing such changes, then the system might quickly become sub-optimal, incurring severe loss (Moore et al., 2014). This raises the main question of this chapter:

How do we build systems that proactively search for a policy that will be good when deployed in the future?

To address this question, in this chapter, we restrict our focus to structured changes resulting from *passive* non-stationarity (see Chapter 2). Under this setting, we search for a policy that is expected to have the highest performance in the future, where the future performance of any policy is proactively anticipated by leveraging (estimates of) the trend of that policy's historical performance. The crux of the proposed idea is based on merging concepts from reinforcement learning and counter-factual reasoning with time-series forecasting.

Formally, we present a policy gradient based approach to search for a policy that maximizes the *forecasted* future performance in the presence of passive non-stationarity. To capture the impact of changes in the environment on a policy's performance, first, the performance of the policy during the past episodes is estimated using counterfactual reasoning. Subsequently, a regression curve is fit to these estimates to model the performance trend of the policy over time, thereby enabling the forecast of future performance. By differentiating this performance forecast with respect to the parameters of the policy being evaluated, we obtain a gradient-based optimization procedure that proactively searches for a policy that will perform well in the future.

Advantages: The proposed method has the following advantages:

- It does not require modeling the transition function, reward function, or how either of them change in an environment with passive non-stationarity; and thus holds the potential to scale well with respect to the number of states and actions in the environment.

- Irrespective of the complexity of the environment or the policy parameterization, it concisely models the *effect* of changes in the environment on a policy's performance using a *univariate* time-series.

- It is data-efficient in that it leverages all available data.

- It mitigates performance lag by proactively optimizing performance for interactions in both the immediate and near future.

- It degenerates to an estimator of the ordinary policy gradient if the system is stationary, meaning that there is little reason not to use our approach if there is a possibility that the system *might* be non-stationary.

This chapter is organized as follows. Section 3.1 provides a quick overview of the notation, followed by the problem statement in Section 3.2. Section 3.2 contains the

core idea that forms the foundation for this and the following chapters. Section 3.5 provides theoretical support regarding how the proposed method can generalize to the stationary setting and Section 3.6 provides empirical results on several domains inspired by real-world applications.

3.1 Notation

Recall from Chapter 2 that a non-stationary decision process (NSDP) is a sequence of POMDPs $(M_i)_{i=1}^{\infty}$. Let $\mathcal{M}$ be a set of possible POMDPs, where each POMDP is defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathscr{O}, \mathscr{P}, \mathscr{R}, \mu)$. Only the observation function $\mathscr{O}_i$, the transition function $\mathscr{P}_i$, the reward function $\mathscr{R}_i$, and the initial state distribution μ_i may differ for each POMDP M_i. Recall that O_i^t, A_i^t, and R_i^t denote the random variables corresponding to the observation, action, and reward at time t in POMDP M_i. The sequence of interactions in M_i is denoted by $H_i := (S_i^t, O_i^t, A_i^t, R_i^t)_{i=1}^{T}$, the observed return is denoted by $G_i := \sum_{t=i}^{T} R_i^t$, and $J_i(\pi) := \mathbb{E}_\pi[G_i|M_i]$ is the performance of π on M_i. The set of possible interaction sequences (the possible values of H_i) is denoted by $\mathcal{H}$, and $\mathcal{T} : \mathcal{M} \times \mathcal{H} \times \mathcal{M} \to [0, 1]$ is the 'meta-transition' function that governs the non-stationarity in the POMDPs. That is, $\mathcal{T}(m, h, m') = \Pr(M_{i+1}{=}m' | M_i{=}m, H_i{=}h)$. In this chapter we consider the restricted case where the non-stationarity is passive, i.e., only caused by external factors,

$$\forall (m, m') \in \mathcal{M}^2, \forall (h, h') \in \mathcal{H}^2, \qquad \mathcal{T}(m, h, m') = \mathcal{T}(m, h', m').$$

3.2 Problem Statement

In many problems, like adapting to friction in robotics, human-machine interaction, etc., the transition dynamics and reward functions change, but every other aspect of the POMDP remains the same throughout. Therefore, we assume that for any two POMDPs, M_k and M_{k+1}, the state set $\mathcal{S}$ and the action set $\mathcal{A}$ are the same.

If the exogenous process causing non-stationarity is arbitrary and changes from M_i to M_{i+1} in unreasonable ways, then there is little hope of finding a good policy for the future as M_{k+1} can be wildly different from everything that the agent has observed by interacting with the past POMDPs $M_1, ..., M_k$. However, in many practical problems of interest, such changes are gradual and have an underlying (unknown) structure. To make the problem tractable, we therefore assume that both the transition dynamics $(\mathscr{P}_1, \mathscr{P}_2, ...)$, and the reward functions $(\mathscr{R}_1, \mathscr{R}_2, ...)$ vary gradually over time in a way that ensures there are no abrupt jumps in the performance of any policy.

Problem Statement. We seek to find a sequence of policies that minimizes lifelong regret:

$$\underset{\{\pi_1, \pi_2, ... \pi_k, ...\}}{\text{argmin}} \sum_{k=1}^{\infty} J_k^* - \sum_{k=1}^{\infty} J_k(\pi_k),$$

where $J_k^* = \max_\pi \ J_k(\pi)$.

3.3 Background and Preliminaries

In this section we review some of the most related work, and summarize the background needed for the concepts used in the chapter. A detailed summary of other approaches can be found in Section 2.

3.3.1 Related Work

Recent works by Al-Shedivat et al. (2017) and Finn et al. (2019) present meta-learning methods that search for initial policy parameters that can be quickly fine-tuned when the objective is changing over time. However, they require *a priori* known boundaries between train and test tasks, which are not available in the continually changing setting. Further, these approaches are complementary to our own, as they could be additionally applied to set the initial parameters of our algorithms. In our empirical study, we show how the adaptation procedure of their methods can result in

a performance lag that is mitigated by our method by explicitly capturing the trend in the objective resulting due to non-stationarity.

Concurrent to our contributions in this chapter (Chandak et al., 2020c), work by Xie et al. (2020a) demonstrated how modeling the changes in a *dynamic-parameter* MDP can be useful to tackle non-stationarity. We focus on the model-free paradigm and our approach is complementary to these (partially) model-based methods.

3.3.2 Per-decision Importance Sampling

Consider the stationary setting, where $M_i = M$ and $J_i(\pi) = J(\pi)$ for all i. For a given POMDP M, per-decision importance sampling (PDIS) allows us to estimate the performance $J(\pi)$ of a policy π for M, when the trajectory data might have been collected/sampled using a *different* policy β. Formally, the PDIS estimator for $J(\pi)$ using N observed trajectories is defined as follows,

$$\hat{J}(\pi) := \frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{T} \left(\prod_{l=0}^{t} \frac{\pi(O_i^l, A_i^l)}{\beta_i(O_i^l, A_i^l)} \right) \gamma^t R_i^t. \tag{3.1}$$

If, $\forall o \in \mathcal{O}$ and $\forall a \in \mathcal{A}$, $\pi(o,a) > 0$ implies that $\beta_i(o,a) > 0$, then it is known that $\hat{J}(\pi)$ is an unbiased estimator of $J(\pi)$ (Thomas, 2015). Further, if the ratio $\pi(o,a)/\beta(o,a)$ is bounded above by a fixed constant, then $\hat{J}(\pi)$ is known to be a strongly consistent estimator of $J(\pi)$ as well (Thomas, 2015). Intuitively, to correct for the mismatch between the policy β_i that was used to collect the trajectory data and the policy π whose performance $J(\pi)$ we wish to estimate, $\hat{J}(\pi)$ re-weights the observed rewards such that if that reward is more likely under the policy π compared to β_i then it up-weights the reward, and vice-versa. A more detailed discussion can be found in the work by Thomas (2015, Chapter 3.6).

3.3.3 Weighted Importance Sampling

While (3.1) provides an unbiased estimate of $J(\pi)$, it typically suffers from high-variance. To observe this, notice that if the denominator in the importance ratio $\prod_{l=0}^{t} \frac{\pi(O_i^l, A_i^l)}{\beta_i(O_i^l, A_i^l)}$ is small then the ratio will have very large value, which can result in high variance. To mitigate this issue, weighted importance sampling (WIS) normalizes the importance ratios such that they are always bounded between zero and one. Formally, WIS estimator is defined as:

$$\bar{J}(\pi) := \frac{\sum_{i=1}^{N} \sum_{t=0}^{T} \gamma^t R_i^t \left(\prod_{l=0}^{T} \frac{\pi(O_i^l, A_i^l)}{\beta_i(O_i^l, A_i^l)} \right)}{\sum_{i=1}^{N} \left(\prod_{l=0}^{T} \frac{\pi(O_i^l, A_i^l)}{\beta_i(O_i^l, A_i^l)} \right)}. \tag{3.2}$$

Normalizing the importance ratios can help in mitigating the variance issue of PDIS, but incurs bias. That is, for finite values of N, (3.2) may no longer be an unbiased estimator of $J(\pi)$. However, it is known that under similar conditions as discussed for PDIS above, WIS is also a strongly consistent estimator of $J(\pi)$ (Thomas, 2015).

3.4 Optimizing for the Future

The problem of minimizing lifelong regret is straightforward if the agent has access to sufficient samples, in advance, from the future environment, M_{k+1}, that it is going to face (where k denotes the current episode number). That is, if we could estimate the start-state objective, $J_{k+1}(\pi)$, for the future POMDP M_{k+1}, then we could search for a policy π whose performance is close to J_{K+1}^*. However, obtaining even a single sample from the future is impossible, let alone getting a sufficient number of samples. This necessitates rethinking the optimization paradigm for searching for a policy that performs well when faced with the future unknown POMDP. There are two immediate challenges here:

1. *How can we estimate $J_{k+1}(\pi)$ without any samples from M_{k+1}?*

2. *How can gradients, $\partial J_{k+1}(\pi)/\partial\theta$, of this future performance be estimated?*

In this section we address both of these issues using the following idea. When the transition dynamics $(\mathscr{P}_1, \mathscr{P}_2, ...)$, and the reward functions $(\mathscr{R}_1, \mathscr{R}_2, ...)$ are changing gradually, the performances $(J_1(\pi), J_2(\pi), ...)$ of any policy π can also be expected to vary smoothly over time. The impact of smooth changes in the environment can thus often manifest as smooth changes in the performance of any policy, π. In cases where there is an underlying, unknown, structure in the changes of the environment, one can now ask:

If the performances $J_{1:k}(\pi) \coloneqq (J_1(\pi), ..., J_k(\pi))$ of π over the course of past episodes were known, can we analyze the trend in its past performances to find a policy that maximizes future performance $J_{k+1}(\pi)$?

3.4.1 Forecasting Future Performance

In this section we address the first challenge of estimating future performance $J_{k+1}(\pi)$ and pose it as a time series forecasting problem. Broadly, this requires two components: (a) A procedure to compute past performances, $J_{1:k}(\pi)$, of π. (b) A procedure to create an estimate, $\hat{J}_{k+1}(\pi)$, of π's future performance, $J_{k+1}(\pi)$, using the estimated values from component (a). An illustration of this idea is provided in Figure 3.1.

Component (a). As we do not have access to the past POMDPs for computing the true values of past performances, $J_{1:k}(\pi)$, we propose computing estimates, $\hat{J}_{1:k}(\pi)$, of them from the observed data. That is, in a non-stationary decision process, starting with the fixed transition matrix $\mathscr{P}_1$ and the reward function $\mathscr{R}_1$, we want to estimate the performance $J_i(\pi)$ of a given policy in episode $i \leq k$. Leveraging the fact that the changes to the underlying POMDP are due to an exogenous processes, we can estimate $J_i(\pi)$ by estimating,

$$J_i(\pi) = \sum_{t=0}^{T} \gamma^t \mathbb{E}_\pi \left[R_i^t \middle| M_i \right], \tag{3.3}$$

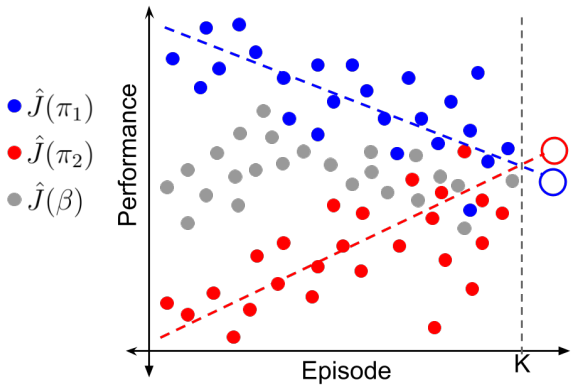


Figure 3.1. An illustration, where the blue and red filled circles represent estimates of the performances of policies π_1 and π_2 at different episodes in the past, using data collected from a given policy β. The open circles represent the forecasted performance of π_1 and π_2 estimated by fitting a curve on the past performance estimates.

where M_i is also a random variable. Next we describe how an estimate of $J_i(\pi)$ can be obtained from (3.3) using information only from the i^{th} episode.

To get an unbiased estimate, $\hat{J}_i(\pi)$, of π's performance during episode i, consider the past trajectory H_i of the i^{th} episode that was observed when executing a policy β_i. By using counter-factual reasoning (Rosenbaum and Rubin, 1983) and leveraging the per-decision importance sampling (PDIS) estimator (Precup, 2000), an unbiased estimate of $J_i(\pi)$ is thus given by:[1]

$$\hat{J}_i(\pi) := \sum_{t=0}^{T} \left(\prod_{l=0}^{t} \frac{\pi(O_i^l, A_i^l)}{\beta_i(O_i^l, A_i^l)} \right) \gamma^t R_i^t. \tag{3.4}$$

It is worth noting that computing (3.4) does not require storing all the past policies β_i, one need only store the actions and the probabilities with which these actions were chosen.

Component (b). To obtain the second component, which captures the structure in $\hat{J}_{1:k}(\pi) := (\hat{J}_1(\pi), ..., \hat{J}_k(\pi))$ and predicts future performances, we make use of a

[1]We assume that $\forall i \in \mathbb{N}$ the distribution of H_i has full support over the set of all possible trajectories of the POMDP M_i.

forecasting function Ψ that estimates future performance $\hat{J}_{k+1}(\pi)$ conditioned on the past performances:

$$\hat{J}_{k+1}(\theta) \coloneqq \Psi(\hat{J}_1(\pi), \hat{J}_2(\pi), ..., \hat{J}_k(\pi)). \tag{3.5}$$

While Ψ can be any forecasting function, we consider Ψ to be an ordinary least squares (OLS) regression model with parameters $w \in \mathbb{R}^{d\times 1}$, and the following input (X) and output (Y) variables,

$$\begin{aligned} X &\coloneqq [1, 2, ..., k]^\top && \in \mathbb{R}^{k\times 1}, \\ Y &\coloneqq [\hat{J}_1(\pi), \hat{J}_2(\pi), \hat{J}_2(\pi), ..., \hat{J}_k(\pi)]^\top && \in \mathbb{R}^{k\times 1}. \end{aligned}$$

For any $x \in X$, let $\phi(x) \in \mathbb{R}^{1\times d}$ denote a d-dimensional function for encoding the time index. For example, using an identity basis $\phi(x) \coloneqq \{x, 1\}$, or Fourier basis functions

$$\phi(x) \coloneqq \{\sin(2\pi n x | n \in \mathbb{N}\} \cup \{\cos(2\pi n x) | n \in \mathbb{N}\} \cup \{1\},$$

where $\mathbb{N}$ is the set of natural numbers. Let $\Phi \in \mathbb{R}^{k\times d}$ be the corresponding basis matrix. If we consider the following least-squares problem,

$$w^* \in \underset{w\in\mathbb{R}^{d\times 1}}{\operatorname{argmin}} \|\Phi w - Y\|_2^2,$$

then using its solution $w^* = (\Phi^\top\Phi)^{-1}\Phi^\top Y$ (Strang et al., 1993) a forecast of the future performance can be obtained using,

$$\begin{aligned} \hat{J}_{k+1}(\pi) =& \phi(k+1)w^* \\ =& \phi(k+1)(\Phi^\top\Phi)^{-1}\Phi^\top Y. \end{aligned} \tag{3.6}$$

This procedure enjoys an important advantage – by only using a univariate time-series to estimate future performance, it bypasses the need for modeling the

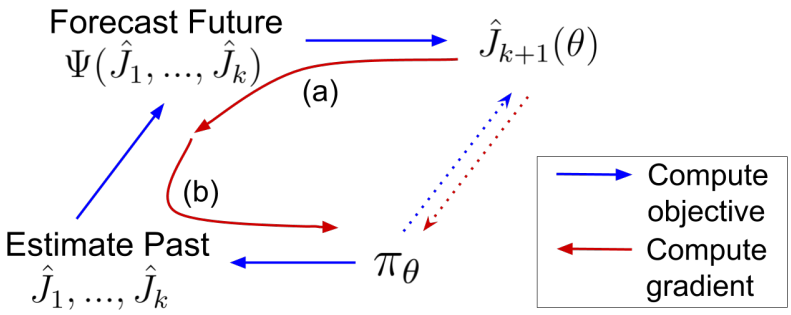


Figure 3.2. The proposed method from the lens of differentiable programming. At any time k, we aim to optimize the policy's parameters, θ, to maximize its performance in the future, i.e., to maximize $J_{k+1}(\theta)$. However, conventional methods (dotted arrows) can not be used to directly optimize for this. In this work, we achieve this as a composition of two programs: one which connects the policy's parameters to its past performances, and the other which forecasts future performance as a function of these past performances. The optimization procedure then corresponds to taking derivatives through this composition of programs to update policy parameters in a direction that maximizes future performance. Arrows (a) and (b) correspond to the respective terms marked in 3.7.

environment, which can be prohibitively hard or even impossible. Further, note that $\Phi^\top\Phi \in \mathbb{R}^{d\times d}$, where $d << k$ typically, and thus the cost of computing the matrix inverse is negligible. These advantages allow this procedure to scale to more challenging problems, while being robust to the sizes of the state and action sets, $|\mathcal{S}|$ and $|\mathcal{A}|$.

3.4.2 Differentiating Forecasted Future Performance

In the previous section, we addressed the first challenge and showed how to proactively estimate future performance, $\hat{J}_{k+1}(\theta)$, of a policy π_θ by explicitly modeling the trend in its past performances $\hat{J}_{1:k}(\theta)$. In this section, we address the second challenge to facilitate a complete policy improvement procedure. An illustration of the idea is provided in Figure 3.2.

Gradients for $\hat{J}_{k+1}(\theta)$ with respect to θ can be obtained as follows,

$$\frac{d\hat{J}_{k+1}(\theta)}{d\theta} = \frac{d\Psi(\hat{J}_1(\theta), ..., \hat{J}_k(\theta))}{d\theta}$$
$$= \sum_{i=1}^{k} \underbrace{\frac{\partial\Psi(\hat{J}_1(\theta), ..., \hat{J}_k(\theta))}{\partial\hat{J}_i(\theta)}}_{\text{(a)}} \underbrace{\frac{d\hat{J}_i(\theta)}{d\theta}}_{(b)}. \tag{3.7}$$

The decomposition in (3.7) has an elegant intuitive interpretation. The terms assigned to (a) in (3.7) correspond to how the future prediction would change as a function of past outcomes, and the terms in (b) indicate how the past outcomes would change due to changes in the parameters of the policy π_θ. In the next paragraphs, we discuss how to obtain the terms (a) and (b).

To obtain term (a), note that in (3.6), $\hat{J}_i(\theta)$ corresponds to the i^{th} element of Y, and so using (3.5) the gradients of the terms (a) in (3.7) are,

$$\frac{\partial\hat{J}_{k+1}(\theta)}{\partial\hat{J}_i(\theta)} = \frac{\partial\phi(k+1)(\Phi^\top\Phi)^{-1}\Phi^\top Y}{\partial Y_i}$$
$$= [\phi(k+1)(\Phi^\top\Phi)^{-1}\Phi^\top]_i, \tag{3.8}$$

where $[Z]_i$ represents the i^{th} element of a vector Z. Therefore, (3.8) is the *gradient of predicted future performance with respect to an estimated past performance*.

The term (b) in (3.7) corresponds to the gradient of the PDIS estimate $\hat{J}_i(\theta)$ of the past performance with respect to policy parameters θ. The following property provides a form for (b) that makes its computation straightforward.

Property 1 (PDIS gradient)**.** *Let* $\rho_i(0,l) \coloneqq \prod_{j=0}^{l} \frac{\pi_\theta(O_i^j, A_i^j)}{\beta_i(O_i^j, A_i^j)}$.

$$\frac{d\hat{J}_i(\theta)}{d\theta} = \sum_{t=0}^{T} \frac{\partial \log \pi_\theta(O_i^j, A_i^t)}{\partial\theta} \left(\sum_{l=t}^{T} \rho_i(0,l)\gamma^l R_i^l \right).$$

Proof. Here we provide complete derivations for obtaining a straightforward equation for computing the gradients of the PDIS estimator with respect to the policy parameters.

$t \backslash l$	0	1	2	...	T
0	$\gamma^0 \rho_i(0,0)\Psi_i^0 R_i^0$				
1	$\gamma^1 \rho_i(0,1)\Psi_i^0 R_i^1$	$\gamma^1 \rho_i(0,1)\Psi_i^1 R_i^1$			
2	$\gamma^2 \rho_i(0,2)\Psi_i^0 R_i^2$	$\gamma^2 \rho_i(0,2)\Psi_i^1 R_i^2$	$\gamma^2 \rho_i(0,2)\Psi_i^2 R_i^2$		
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
T	$\gamma^T \rho_i(0,T)\Psi_i^0 R_i^T$	$\gamma^T \rho_i(0,T)\Psi_i^1 R_i^T$	$\gamma^T \rho_i(0,T)\Psi_i^2 R_i^T$	...	$\gamma^T \rho_i(0,T)\Psi_i^T R_i^T$

Table 3.1. let $\Psi_i^t = \partial \log \pi_\theta(O_i^t, A_i^t)/\partial\theta$. This table represents all the terms in 3.9 required for computing $\nabla \hat{J}_i(\theta)$. Gray color denotes empty cells.

These might also be of independent interest when dealing with off-policy policy optimization for stationary (PO)MDPs.

Recall from (3.4) that,

$$\hat{J}_i(\theta) = \sum_{t=0}^{T} \left(\prod_{l=0}^{t} \frac{\pi(O_i^l, A_i^l)}{\beta_i(O_i^l, A_i^l)} \right) \gamma^t R_i^t.$$

Computing the gradient of $\hat{J}_i(\theta)$,

$$\begin{aligned}
\nabla \hat{J}_i(\theta) &= \sum_{t=0}^{T} \frac{\partial}{\partial\theta} \left(\prod_{l=0}^{t} \frac{\pi_\theta(O_i^l, A_i^l)}{\beta(O_i^l, A_i^l)} \right) \gamma^t R_i^t \\
&= \sum_{t=0}^{T} \left(\prod_{l=0}^{t} \frac{\pi_\theta(O_i^l, A_i^l)}{\beta(O_i^l, A_i^l)} \right) \frac{\partial \log \left(\prod_{l=0}^{t} \pi_\theta(O_i^l, A_i^l) \right)}{\partial\theta} \gamma^t R_i^t \\
&= \sum_{t=0}^{T} \left(\prod_{l=0}^{t} \frac{\pi_\theta(O_i^l, A_i^l)}{\beta(O_i^l, A_i^l)} \right) \left(\sum_{l=0}^{t} \frac{\partial \log \pi_\theta(O_i^l, A_i^l)}{\partial\theta} \right) \gamma^t R_i^t \\
&= \sum_{t=0}^{T} \rho_i(0,t) \left(\sum_{l=0}^{t} \frac{\partial \log \pi_\theta(O_i^l, A_i^l)}{\partial\theta} \right) \gamma^t R_i^t \\
&= \sum_{t=0}^{T} \frac{\partial \log \pi_\theta(O_i^t, A_i^t)}{\partial\theta} \left(\sum_{l=t}^{T} \rho_i(0,l) \gamma^l R_i^l \right),
\end{aligned} \tag{3.9}$$

where, in the last step, instead of the summation over the partial derivatives of $\log \pi_\theta$ for each weight $\rho(\cdot,\cdot)$, we consider the alternate form where the summation is over the importance weights $\rho(\cdot,\cdot)$ for each partial derivative of $\log \pi_\theta$. To see this step

clearly, let $\Psi_i^t = \partial \log \pi_\theta(O_i^t, A_i^t)/\partial\theta$, then Table 3.1 shows all the terms in (3.9). The last step above corresponds to taking the column-wise sum instead of the row-wise sum in Table 3.1.

□

3.4.3 Algorithm

Algorithm 1 provides a sketch of our proposed procedure, called *Prognosticator*, for optimizing the future performance of the policy. To make the method more practical, we incorporated two additional modifications to reduce computational cost and variance.

First, it is often desirable to perform an update only after a certain episode interval δ to reduce computational cost. This raises the question: if a newly found policy will be executed for the next δ episodes, should we choose this new policy to maximize performance on just the single next episode, or to maximize the average performance over the next δ episodes? An advantage of our proposed method is that we can easily tune how far in the future we want to optimize for. Thus, to minimize lifelong regret, we propose optimizing for the mean performance over the next δ episodes. That is, $\arg\max_\theta (1/\delta) \sum_{\Delta=1}^{\delta} \hat{J}_{k+\Delta}(\theta)$, where $\hat{J}_{k+\Delta}$ is the forecast (made using all of the past data) of the performance of π_θ for episode $k+\Delta$.

Second, notice that if the policy becomes too close to deterministic, there would be two undesired consequences. (a) The policy will not explore, thereby precluding the agent from observing any changes to the environment in states that it no longer revisits—changes that might make entering those states worthwhile to the agent. (b) In the future when estimating $\hat{J}_{k+1}(\theta)$ using the *past* performance of θ, importance sampling will have high variance if the policy executed during episode $k+1$ is close to deterministic. To mitigate these issues, we add an entropy regularizer $\mathcal{H}$ during policy optimization.

Algorithm 1: Prognosticator

Input Learning-rate η, time-duration δ, entropy-regularizer λ
Initialize Forecasting function Ψ, Buffer $\mathbb{B}$
while *True* **do**
    # Record a new batch of trajectories using π_θ
    for $episode = 1, 2, ..., \delta$ **do**
        $h = (O_t, A_t, \Pr(A_t|O_t), R_t)_{t=0}^{T}$
        $\mathbb{B}$.insert(h)
    # Update for future performance
    for $i = 1, 2, ...$ **do**
        # Evaluate past performances using (3.4)
        for $k = 1, 2, ..., |\mathbb{B}|$ **do**
            $\hat{J}_k(\theta) = \sum_{t=0}^{T} \rho(0,t)\gamma^t R_k^t$
        # Future forecast and its gradient using using (3.6) and (3.7)
        $\mathcal{L}(\theta) = \frac{1}{\delta}\sum_{\Delta=1}^{\delta} \hat{J}_{k+\Delta}(\theta)$
        $\theta \leftarrow \theta + \eta \frac{\partial}{\partial\theta}(\mathcal{L}(\theta) + \lambda\mathcal{H}(\theta))$

3.4.4 Understanding the Behavior of Prognosticator

Notice that as the scalar term (a) is multiplied by the gradient of the PDIS term (b) in (3.7), the gradient of future performance can be viewed as a weighted sum of off-policy policy gradients. In Figure 3.3, we provide visualization of the weights $\zeta_i := \partial\hat{J}_{100}(\theta)/\partial\hat{J}_i(\theta)$ for PDIS gradients of each episode i, when the performance for 100^{th} episode is forecasted using data from the past 99 episodes. For the specific setting when Ψ is an OLS estimator, these weights are independent of Y in (3.8) and their pattern remains constant for any given sequence of POMDPs.

Importantly, note the occurrence of negative weights in Figure 3.3 when the identity basis function or Fourier basis functions is used, suggesting that the optimization procedure should move towards a policy that had *lower* performance in some of the past episodes. While this negative weighting seems unusual at first glance, it has an intriguing interpretation.

To better understand these negative weights, consider a qualitative comparison when weights from different methods in Figure 3.3 are used along with the performance

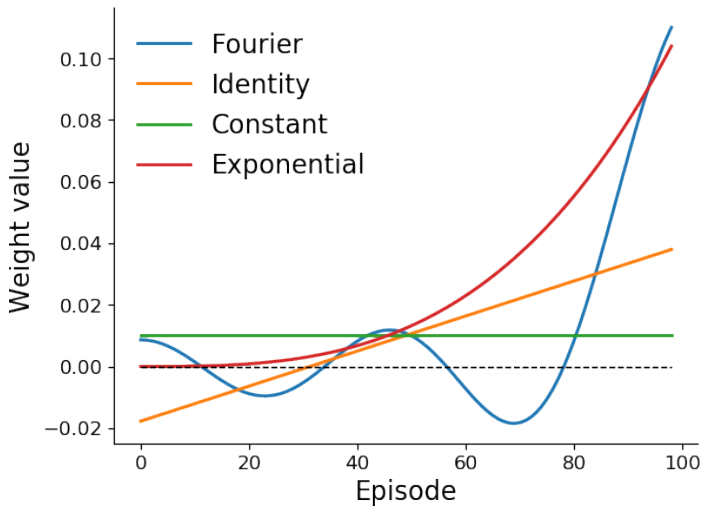


Figure 3.3. The value of weights ζ_i for all values of $i \in [1, 99]$ using different functions to encode the time index. Notice that many weights are negative when using the identity or Fourier bases.

estimates of policies π_1 and π_2 in Figure 3.1. Despite having lower estimates of return everywhere, π_2's rising trend suggests that it might have higher performance in the future, that is, $J_{k+1}(\pi_2) > J_{k+1}(\pi_1)$. Existing online learning methods like FTL, maximize performance on all the past data uniformly (green curve in Figure 3.3). Similarly, the exponential weights (red curve in Figure 3.3) are representative of approaches that only optimize using data from recent episodes and discard previous data (Peters and Schaal, 2008). Either of these methods that use only non-negative weights can *never* capture the trend to forecast $J_{k+1}(\pi_2) > J_{k+1}(\pi_1)$. However, the weights obtained when using the identity basis would facilitate *minimization* of performances in the distant past and maximization of performance in the recent past. Intuitively, this means that it moves towards a policy whose performance is on a linear *rise*, as it expects that policy to have better performance in the future.

While weights from the identity basis are useful for forecasting whether $J_{k+1}(\pi_2) > J_{k+1}(\pi_1)$, it cannot be expected that the trend will always be linear as in Figure 3.1. To be more flexible and allow for any smooth trend, we opt to use the Fourier basis functions in our experiments. Observe the alternating sign of weights in Figure 3.3

when using the Fourier basis functions. This suggests that the optimization procedure will take into account the *sequential differences* in performances over the past, thereby favoring the policy that has shown the most performance *increments* in the past. This also avoids restricting the performance trend of a policy to be linear.

3.4.5 Mitigating Variance

While model-free algorithms for finding a good policy are scalable to large problems, they tend to suffer from high-variance (Greensmith et al., 2004). In particular, the use of importance sampling estimators can increase the variance further (Guo et al., 2017). In our setup, high variance in estimates of past performances $\hat{J}_{1:k}(\pi)$ of π can hinder capturing π's performance trend, thereby making the forecasts less reliable.

Notice that a major source of variance is the availability of only a *single* trajectory sample per POMDP M_i, for all $i \in \mathbb{N}$. If this trajectory H_i, generated using β_i is likely when using β_i, but has near-zero probability when using π then the estimated $\hat{J}_i(\pi)$ is also nearly zero. While $\hat{J}_i(\pi)$ is an unbiased estimate of $J_i(\pi)$, information provided by this single H_i is of little use to evaluate $J_i(\pi)$. Subsequently, discarding this from time-series analysis, rather than setting it to be 0, can make the time series forecast more robust against outliers. In comparison, if trajectory H_i is unlikely when using β_i but likely when using π, then not only is H_i very useful for estimating $J_i(\pi)$ but it also has a lower chance of occurring in the future, so this trajectory must be emphasized when making a forecast. Such a process of (de-)emphasizing estimates of past returns using the collected data itself can introduce bias, but this bias might be beneficial in this few-sample regime.

To capture this idea formally, we build upon the insights of Hachiya et al. (2012) and Mahmood et al. (2014), who draw an equivalence between weighted least-squares (WLS) estimation and the weighted importance sampling (WIS) (Precup, 2000) estimator. Particularly, let $G_i := \sum_{t=0}^{T} \gamma^t R_i^t$ be the discounted return of the i^{th}

trajectory observed from a stationary POMDP, and $\rho_i^\ddagger := \rho_i(0,T)$ be the importance ratio of the entire trajectory. Then the WIS estimator, $\hat{J}^\ddagger(\pi)$, of the performance of π in a stationary POMDP is,

$$\hat{J}^\ddagger(\pi) := \underset{c\in\mathbb{R}}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n} \rho_i^\ddagger (G_i - c)^2 = \frac{\sum_{i=1}^n \rho_i^\ddagger G_i}{\sum_{i=1}^n \rho_i^\ddagger}.$$

To mitigate variance in our setup, we propose extending WIS. In the non-stationary setting, to perform WIS while capturing the trend in performance over time, we use a modified forecasting function $\Psi^\ddagger$, which is a weighted least-squares regression model with a $d-$dimensional basis function ϕ, and parameters $w^\ddagger \in \mathbb{R}^{d\times 1}$,

$$w^\ddagger := \underset{c\in\mathbb{R}^{d\times 1}}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n} \rho_i^\ddagger (G_i - c^\top \phi(i))^2. \tag{3.10}$$

Let $\Lambda \in \mathbb{R}^{k\times k}$ be a diagonal weight matrix such that $\Lambda_{ii} = \rho_i^\ddagger$, let $\Phi \in \mathbb{R}^{k\times d}$ be the basis matrix, and let the following be input and output variables,

$$\begin{aligned} X &:= [1, 2, ..., k]^\top && \in \mathbb{R}^{k\times 1}, \\ Y &:= [G_1, G_2, ..., G_k]^\top && \in \mathbb{R}^{k\times 1}. \end{aligned}$$

The solution to the weighted least squares problem in (3.10) is then given by $w^\ddagger = (\Phi^\top \Lambda \Phi)^{-1}\Phi^\top \Lambda Y$ and the forecast of the future performance can be obtained using,

$$\hat{J}_{k+1}^\ddagger(\pi) := \phi(k+1)w^\ddagger = \phi(k+1)(\Phi^\top \Lambda \Phi)^{-1}\Phi^\top \Lambda Y.$$

$\hat{J}_{k+1}^\ddagger(\pi)$ has several desired properties:

- It incorporates a notion of how relevant each observed trajectory is towards forecasting, while also capturing the trend in performance.

- The forecasts are less sensitive to the importance sampling variances.
- The entire process is still differentiable.

3.5 Generalizing to the Stationary Setting

As the agent is unaware of how the environment is changing, a natural question to ask is: What if the agent wrongly assumed a stationary environment was non-stationary? What is the quality of of the agent's performance forecasts? What is the impact of the negative weights on past evaluations of a policy's performance? Here we answer these questions.

Before stating the formal results, we introduce some necessary notation and some additional conditions. Let $J(\pi)$ be the performance of policy π for a stationary POMDP. Let $\hat{J}_{k+\delta}(\pi)$ and $\hat{J}^{\ddagger}_{k+\delta}(\pi)$ be the non-stationary importance sampling (NIS) and non-stationary weighted importance sampling (NWIS) estimators of performance δ episodes in future.

Before proceeding towards the main results, we impose the following constraints on the set of policies, and the basis functions $\phi_i : \mathbb{N} \to \mathbb{R}$ used for encoding the time index in both Ψ and $\Psi^{\ddagger}$, with $\phi(\cdot) = [\phi_1(\cdot), ..., \phi_{d-1}(\cdot), 1]$.

(a) $\phi(\cdot)$ always contains 1 to incorporate a bias coefficient in least-squares regression (for example, $\phi(\cdot) = [\phi_1(\cdot), ..., \phi_{d-1}(\cdot), 1]$, where $\forall i \in [1, d-1]$, $\phi_i(\cdot)$ is a basis function).[2]

(b) There exists a finite constant C_1, such that $\forall i,\ |\phi_i(\cdot)| < C_1$.

(c) Φ has full column rank such that $(\Phi^\top \Phi)^{-1}$ exists.

(d) We only consider a set of policies Π that have non-zero probability of taking any action in any state. That is, $\exists C_2 > 0$, such that $\forall \pi \in \Pi, \forall s \in \mathcal{S}, \forall a \in \mathcal{A}, \pi(a|s) > C_2$.

---

[2]If additional domain knowledge is available to select an appropriate basis function that can be used to represent the performance trend of all the policies for the given *non-stationary* environment, then all the following finite-sample and large-sample properties can be extended for that environment as well, using that basis function.

Satisfying condition (a) is straightforward as it is typically already satisfied by the set of basis functions used popularly. This constraint ensures that the regression based forecasting function can capture a fixed constant that is required to model the absence of any trend. This constraint is useful for our purpose as in the stationary setting there exists no trend in the expected performance across episodes for any given policy.

Conditions (b) and (c) are also readily satisfied by popular sets of basis functions. For example, features from the Fourier basis are bounded by $[-1, 1]$, and features from polynomial/identity bases are also bounded when inputs are adequately normalized. Further, when there are no repeated basis functions, and the number of samples is more than the number of basis functions ($k \geq d$), condition (c) is satisfied. This ensures that the least-squares problem is well-defined and has a unique-solution.

Condition (d) ensures that the denominator in any importance ratio is always bounded below, such that the importance ratios are bounded above. This implies that the importance sampling estimator for any policy has finite variance. Use of entropy regularization with common policy parameterizations (e.g., softmax) can prevent violation of this condition.

With this notation and these conditions, we first formalize the stationarity assumption:

Assumption 1 (Stationarity)**.** *For all i, $M_i = M_{i+1}$.*

This implies that $\mathbb{E}[\hat{J}_i(\pi)] = J(\pi)$ for all i. Following prior literature (Precup, 2000; Thomas, 2015; Mahmood et al., 2014) we also make a simplifying assumption that allows us to later apply a standard form of the laws of large numbers:

Assumption 2 (Independence)**.** *$\hat{J}_i(\pi)$ are independent for all $i \in \{1, \ldots, k\}$.*

This assumption is satisfied if there is only one behavior policy (i.e., $\forall i, \beta_i = \beta_{i+1}$) or if the sequence of behavior policies does not depend on the data. This assumption

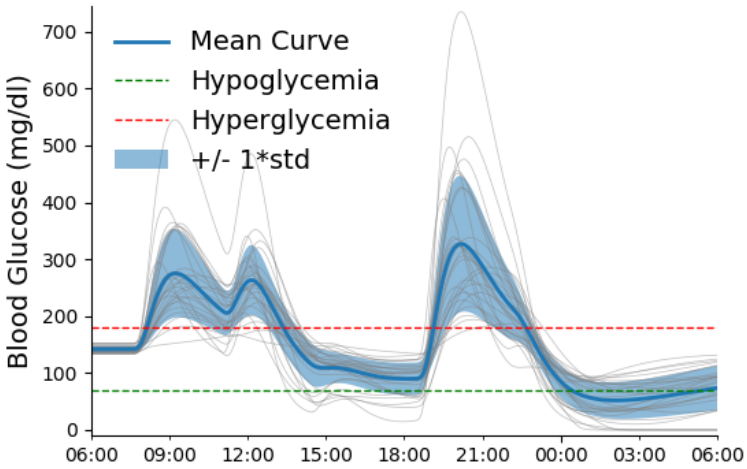


Figure 3.4. Blood-glucose level of an *in-silico* patient for 24 hours (one episode). Humps in the graph occur at times when a meal is consumed by the patient.

is not satisfied when the sequence of behavior policies depends on the data because then episodes are not independent. While we expect that the following theorems apply even without Assumption 2, we have not established this result formally.

We then have the following results indicating that NIS is unbiased and consistent like ordinary importance sampling and NWIS is biased and consistent like weighted importance sampling.

Theorem 1 (Unbiased NIS)**.** *Under Assumptions 1 and 2, for all* $\delta \geq 1$, $\hat{J}_{k+\delta}(\pi)$ *is an unbiased estimator of* $J(\pi)$*. That is,* $\mathbb{E}[\hat{J}_{k+\delta}(\pi)] = J(\pi)$.

Theorem 2 (Biased NWIS)**.** *Under Assumptions 1 and 2, for all* $\delta \geq 1$, $\hat{J}^{\ddagger}_{k+\delta}(\pi)$ *may be a biased estimator of* $J(\pi)$*. That is, it is possible that* $\mathbb{E}[\hat{J}^{\ddagger}_{k+\delta}(\pi)] \neq J(\pi)$.

Theorem 3 (Consistent NIS)**.** *Under Assumptions 1 and 2, for all* $\delta \geq 1$, $\hat{J}_{k+\delta}(\pi)$ *is a consistent estimator of* $J(\pi)$*. That is, as* $N \to \infty$, $\hat{J}_{N+\delta}(\pi) \xrightarrow{a.s.} J(\pi)$.

Theorem 4 (Consistent NWIS)**.** *Under Assumptions 1 and 2, for all* $\delta \geq 1$, $\hat{J}^{\ddagger}_{k+\delta}(\pi)$ *is a consistent estimator of* $J(\pi)$*. That is, as* $N \to \infty$, $\hat{J}^{\ddagger}_{N+\delta}(\pi) \xrightarrow{a.s.} J(\pi)$.

Proof. See Section 3.9 for all of these proofs. □

NWIS is biased and consistent like the WIS estimator, and our experiments show that it also has similar variance reduction properties that can make the optimization process more efficient for non-stationary POMDPs when the variance of $\hat{J}_i(\pi)$ is high.

Remark 1. *It is worth observing that NIS and the NWIS estimators reduce exactly to the IS and WIS estimators (Precup, 2000) when* $\phi(\cdot) \coloneqq [1]$.

3.6 Empirical Analysis

This section presents empirical evaluations using several environments inspired by real-world applications that exhibit non-stationarity. In the following paragraphs, we briefly discuss each environment.

3.6.1 Environments

We provide empirical results on three non-stationary environments: *in-silico* diabetes treatment, a recommender system, and a goal-reacher task. Details for each of these environments are provided in this section. For all of the above environments, we regulate the *speed* of non-stationarity to characterize an algorithms' ability to adapt. Higher speed corresponds to a greater amount of non-stationarity; A speed of zero indicates that the environment is stationary.

Non-stationary Diabetes Treatment: This domain models the problem of type-1 diabetes management. The body of a person with type-1 diabetes does not produce enough *insulin*, a hormone that promotes absorption of glucose from the blood. Consumption of a meal increases the blood-glucose level in the body, and if the blood-glucose level becomes too high, then the patient can suffer from *hyperglycemia*. Insulin injections can reduce the blood-glucose level, but if the level becomes too low, then the patient suffers from *hypoglycemia*. While either of the extremes is undesirable,

hypoglycemia is more dangerous and can triple the five-year mortality rate for a person with diabetes (Man et al., 2014).

Autonomous medical support systems have been proposed to decide how much insulin should be injected to keep a person's blood glucose levels near ideal levels (Bastani, 2014). Currently, the parameters of such a medical support system are set by a doctor specifically for each patient. However, due to non-stationarities induced over time as a consequence of changes in the body mass index, the insulin sensitivity of the pancreas, diet, etc., the parameters of the controller need to be readjusted regularly. Currently, this requires revisiting the doctor. A viable reinforcement learning solution to this non-stationary problem could enable the automatic tuning of these parameters for patients who lack regular access to a physician.

To model this domain, we use an open-source implementation (Xie, 2019) of the U.S. Food and Drug Administration (FDA) approved Type-1 Diabetes Mellitus simulator (T1DMS) (Man et al., 2014) for treatment of Type-1 diabetes, where we induce non-stationarity by oscillating the body parameters between two known configurations. Each episode consists of a day (1440 timesteps, where each timestep corresponds to a minute) in an *in-silico* patient's life and the transition dynamics of a patient's body for each second is governed by a continuous time ordinary differential equation (ODE) (Man et al., 2014). After each minute the insulin controller is used to inject the desired amount of insulin for controlling blood glucose.

However, the insulin sensitivity of a patient's internal body organs vary over time, inducing non-stationarity that should be accounted for. In the T1DMS simulator, we induce this non-stationarity by oscillating the body parameters (e.g., insulin sensitivity, rate of glucose absorption, etc.) between two known configurations available in the simulator.

For controlling the insulin injection, we use a parameterized policy based on the amount of insulin that a person with diabetes is instructed to inject prior to eating a

meal (Bastani, 2014):

$$\text{injection} = \frac{\text{current blood glucose} - \text{target blood glucose}}{CF} + \frac{\text{meal size}}{CR},$$

where 'current blood glucose' is the estimate of the person's current blood-glucose level, 'target blood glucose' is the desired blood glucose, 'meal size' is the estimate of the size of the meal the patient is about to eat, and CR and CF are two real-valued parameters, that must be tuned based on the body parameters to make the treatment effective.

Non-stationary Recommender System: During online recommendation of movies, tutorials, advertisements and other products, a recommender system needs to interact and personalize for each user. However, the user's interest in different items that can each be recommended fluctuates over time. For example, interests during online shopping can vary based on seasonality or other unknown factors (Thomas et al., 2017; Theocharous et al., 2020).

This environment models the desired recommender system setting where reward (interest of the user) associated with each item changes over time. Figure 3.8 (left) shows how the reward associated with each item changes over time, for each of the considered 'speeds' of non-stationarity. The goal for the reinforcement learning agent is to maximize revenue by recommending the item which the user is most interested in at any time.

Non-stationary Goal Reacher: For an autonomous robot dealing with tasks in the open-world, it is natural for the problem specification to change over time. An ideal system should quickly adapt to the changes and still complete the task.

To model the above setting, this environment considers a task of reaching a non-stationary goal position. That is, the location of the goal position keeps slowly moving around with time. The goal of the reinforcement learning agent is to control the four (left, right, up, and down) actions to move the agent towards the goal as quickly as

possible given the real valued Cartesian coordinates of the agent's current location. The maximum time given to the agent to reach the goal is 15 steps.

3.6.2 Algorithms Compared

We consider the following algorithms for comparison:

Prognosticator: Two variants of our algorithm, **Pro-OLS** and **Pro-WLS**, which use OLS and WLS estimators for Ψ.

ONPG: Similar to the adaptation technique presented by Al-Shedivat et al. (2017), this baseline performs purely online optimization by fine-tuning the existing policy using only the trajectory being observed online.

FTRL-PG: Similar to the adaptation technique presented by Finn et al. (2019), this baseline performs follow-the-(regularized)-leader optimization by maximizing performance over both the current and all of the past trajectories.

3.6.3 Hyper-parameters

For both the variants of the proposed *Prognosticator* algorithms, we use the Fourier basis functions to encode the time index while performing (ordinary/weighted) least squares estimation. Since Fourier basis functions require inputs to be normalized with $|x| \leq 1$, we normalize each time index by dividing it by $K + \delta$, where K is the current time and δ is the maximum time into the future that we will forecast for. Further, as we are regressing only on time (which are all positive values), is does not matter whether the function for the policy performance over time is odd ($\Psi(x) = -\Psi(-x)$) or not. Therefore, we drop all the terms in corresponding to $\sin(\cdot)$, which are useful for modeling odd functions. This halves the number of model parameters. Finally, instead of letting $n \in \mathbb{N}$, we restrict it to a finite set $\{1, ..., d-1\}$, where d is a fixed constant that determines the size of the feature vector for each input. In all our experiments, d was a hyper-parameter chosen from $\{3, 5, 7\}$.

Other hyper-parameter ranges were common for all the algorithms. The discounting factor γ was kept fixed to 0.99 and learning rate η was chosen from the range $[5 \times 10^{-5}, 5 \times 10^{-2}]$. The entropy regularizer λ was chosen from the range $[0, 1 \times 10^{-2}]$. The batch size δ was chosen from the set $\{1, 3, 5\}$. Inner optimization over past data for the proposed methods and FTRL-PG was run for $\{10, 20, 30\} \times \delta$ iterations. Inner optimization for ONPG corresponds to one iteration over all the trajectories collected in the current batch. Past algorithms have shown that while clipping the importance weights make the estimators biased, clipping can improve stability of reinforcement learning algorithms (Schulman et al., 2017). Similarly, we clip the maximum value of the importance ratio to a value chosen from $\{5, 10, 15\}$. Note that the use of clipping also violates the unbiased properties of our estimators. As the non-stationary diabetes treatment problem has a continuous action space, the policy was parameterized with a Gaussian distribution having a variance chosen from $[0.5, 2.5]$. For the non-stationary goal-reacher environment, the policy was parameterized using a two-layer neural network with number of hidden nodes chosen from $\{16, 32, 64\}$.

In total, for each algorithm-domain pair, 2000 settings were uniformly sampled (loguniformly for learning rates and λ) from the mentioned hyper-parameter ranges/sets. Results from the best performing settings are reported in all plots. Each hyper-parameter setting was run using 10 seeds for the non-stationary diabetes treatment (as it was time intensive to run a continuous time ODE for each step) and 30 seeds for the other two environments to get the standard error of the mean performances. The authors had shared access to a computing cluster, consisting of 50 compute nodes with 28 cores each, which was used to run all the experiments.[3]

[3]Code for our algorithm can be accessed using the following link: https://github.com/yashchandak/OptFuture_NSMDP.

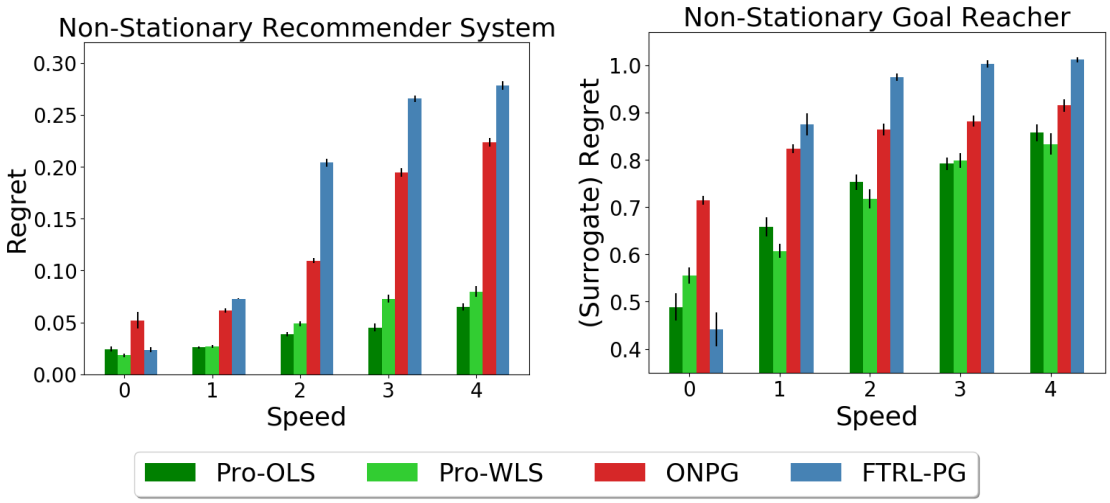


Figure 3.5. Best performances of all the algorithms obtained by conducting a hyper-parameter sweep over 2000 hyper-parameter combinations per algorithm, per environment. For each hyper-parameter setting, 30 trials were executed for the recommender system and the goal reacher environments. Error bars correspond to the standard error. The x-axis represents how fast the environment is changing and the y-axis represents regret (lower is better).

3.6.4 Results

In the non-stationary recommender system, as the exact value of J_k^* is available from the simulator, we can compute the true value of regret. However, for the non-stationary goal reacher and diabetes treatment environment, as J_k^* is not known for any k, we use a surrogate measure for regret. That is, let $\tilde{J}_k^*$ be the maximum return obtained in episode k by any algorithm, then we use $(\sum_{k=1}^{N}(\tilde{J}_k^* - J_k(\pi)))/(\sum_{k=1}^{N}\tilde{J}_k^*)$ as the surrogate regret for a policy π.

In the non-stationary recommender system, all the methods perform nearly the same when the environment is stationary. FTRL-PG has a slight edge over ONPG when the environment is stationary as all the past data is directly indicative of the future POMDP. It is interesting to note that while FTRL-PG works the best for the stationary setting in the recommender system and the goal reacher task, it is not the best in the diabetes treatment task as it can suffer from high variance. We discuss the impact of variance in later paragraphs.

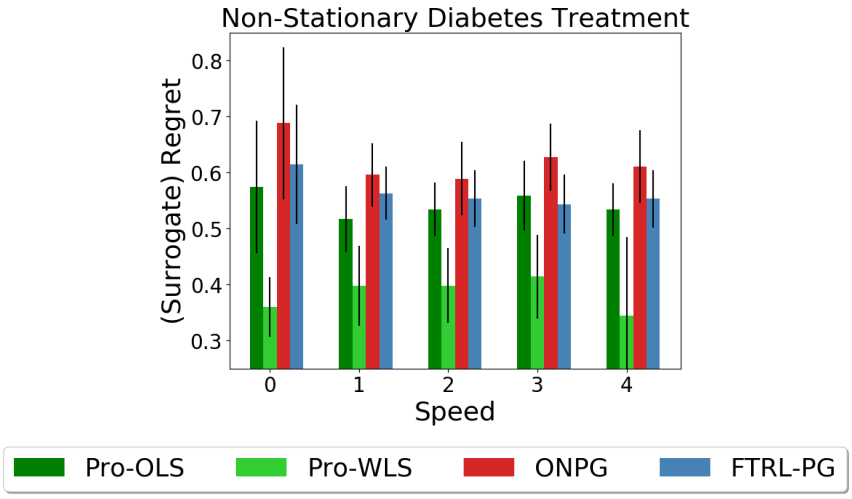


Figure 3.6. Best performances of all the algorithms obtained by conducting a hyper-parameter sweep over 2000 hyper-parameter combinations per algorithm, per environment. For each hyper-parameter setting, 10 trials for the diabetes treatment environment. Error bars correspond to the standard error. The x-axis represents how fast the environment is changing and the y-axis represents regret (lower is better).

With the increase in the speed of non-stationarity, performance of both the baselines deteriorate quickly. Of the two, ONPG is better able to mitigate performance lag as it discards all the past data. In contrast, both the proposed methods, Pro-OLS and Pro-WLS, can leverage all the past data to better capture the impact of non-stationarity and thus are consistently more robust to the changes in the environment.

In the non-stationary goal reacher environment, a similar trend as above is observed. While considering all the past data equally is useful for FTRL-PG in the stationary setting, it creates drastic performance lag as the speed of the non-stationarity increases. Between Pro-OLS and Pro-WLS, in the stationary setting, once the agent nearly solves the task all subsequent trajectories come from nearly the same distribution and thus the variance resulting from the importance sampling ratio is not severe. In such a case, where the variance is low, Pro-WLS has less advantage over Pro-OLS and additionally suffers from being biased. However, as the non-stationarity increases, the optimal policy keeps changing and there is a higher discrepancy between distributions

of past and current trajectories. This makes the lower variance property of Pro-WLS particularly useful. Having the ability to better capture the underlying trend, both Pro-OLS and Pro-WLS consistently perform better than the baselines when there is non-stationarity.

The non-stationary diabetes treatment environment is particularly challenging as it has a continuous action set. This makes importance sampling based estimators subject to much higher variance. Consequently, Pro-OLS is not able to reliably capture the impact of non-stationarity and performs similar to FTRL-PG. In comparison, ONPG is data-inefficient and performs poorly on this domain across all the speeds. The most advantageous algorithm in this environment is Pro-WLS. Since Pro-WLS is designed to better tackle variance stemming from importance sampling, it is able to efficiently use the past data to capture the underlying trend and performs well across all the speeds of non-stationarity.

3.6.5 Computational Complexity (Memory and Time)

The space requirement for our algorithms and FTRL-PG is linear in the number of episodes seen in the past, whereas it is constant for ONPG as it discards all the past data. The computational cost of our algorithm is also similar to FTRL-PG as the only additional cost is that of differentiating through least-squares estimators which involves computing $(\Phi^\top\Phi)^{-1}$ or $(\Phi^\top\Lambda\Phi)^{-1}$. This additional overhead is negligible as these matrices are of the size $d \times d$, where d is the size of the feature vector for the time index and $d << N$, where N is the number of past episodes. Figures 3.5, 3.6, 3.8, and 3.9 present an empirical estimate for the sample efficiency.

3.6.6 Ablation Study

In Figure 3.7 we show the impact of the choice of basis function, $\phi(\cdot)$, on the performance of both of our proposed algorithms: Pro-OLS and Pro-WLS. Dimension d for both the Fourier basis and the set of polynomial basis was chosen from $\{3, 5, 7\}$.

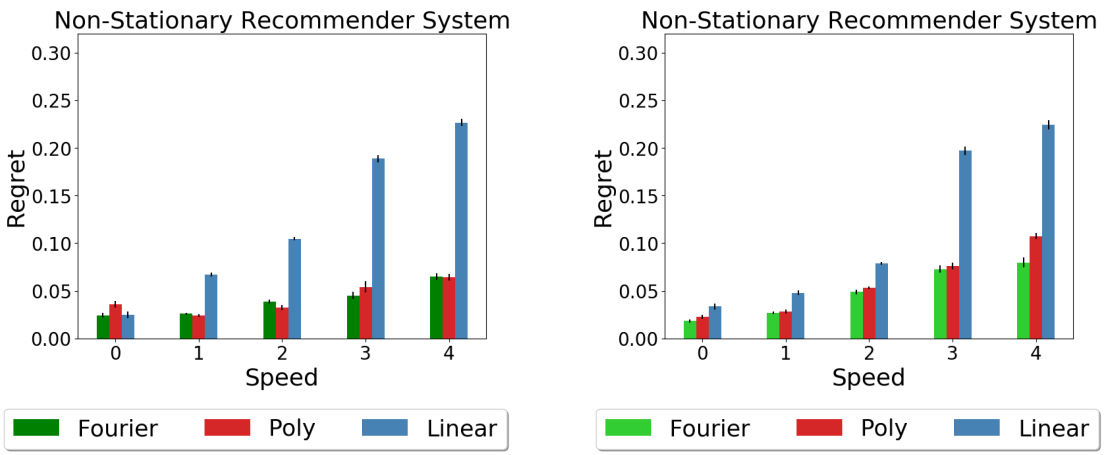


Figure 3.7. Best performances of all the algorithms for the non-stationary recommender system environment, obtained by conducting a hyper-parameter sweep over 1000 hyper-parameter combinations per algorithm. For each hyper-parameter setting, 30 trials were executed. Error bars correspond to the standard error. (Left) Performance of Pro-OLS with Fourier, polynomial, and linear basis functions. (Right) Performance of Pro-WLS with Fourier, polynomial, and linear basis functions.

All other hyper-parameters were searched as described in Section 3.6.3. It can be seen that both the Fourier and polynomial basis functions provide sufficient flexibility for modeling the trend, whereas the linear basis offers limited flexibility and results in poor performance.

3.6.7 Performance Over Time

Figures 3.5 and 3.6 present summary statistics of the results. In this section we present all the results in detail. Figure 3.8 shows the performances of all the algorithms for individual episodes as the user interests change over time in the recommender system environment. In this environment, as the true reward for each of the items is directly available, we provide a plot showing how the rewards change over time in Figure 3.8 (left). Notice that the shapes of the performance curves for the proposed methods closely resemble the trend of the maximum reward attainable across time.

Figure 3.9 shows the performances of all the algorithms for the non-stationary goal-reacher and the diabetes treatment environments. In these environments, the maximum achievable performance for each episode is not readily available.

3.7 Conclusion

We presented a policy gradient-based algorithm that combines counter-factual reasoning with curve-fitting to proactively search for a good policy for future POMDPs. Irrespective of the environment being stationary or non-stationary, the proposed method can leverage all the past data, and in non-stationary settings it can proactively optimize for future performance as well. Therefore, our method provides a single solution for mitigating performance lag and being data-efficient.

3.8 Limitations and Future Work

The method that we propose is limited to settings where (a) non-stationarity is governed by an exogenous process (i.e., past actions do not impact the underlying non-stationarity), which has no auto-correlated noise, and (b) performance of every policy changes smoothly over time and has no abrupt breaks/jumps.

While the proposed algorithm has several desired properties, many open questions remain. In our experiments, we noticed that the proposed algorithm is particularly sensitive to the value of the entropy regularizer λ. Keeping λ too high prevents the policy from adapting quickly. Keeping λ too low lets the policy overfit to the forecast and become close to deterministic, thereby increasing the variance for subsequent importance sampling estimates of policy performance. While we resorted to hyper-parameter search, leveraging methods that adapt λ automatically might be fruitful (Haarnoja et al., 2018).

Our framework highlights new research directions for studying bias-variance trade-offs in the non-stationary setting. While tackling the problem from the point of view

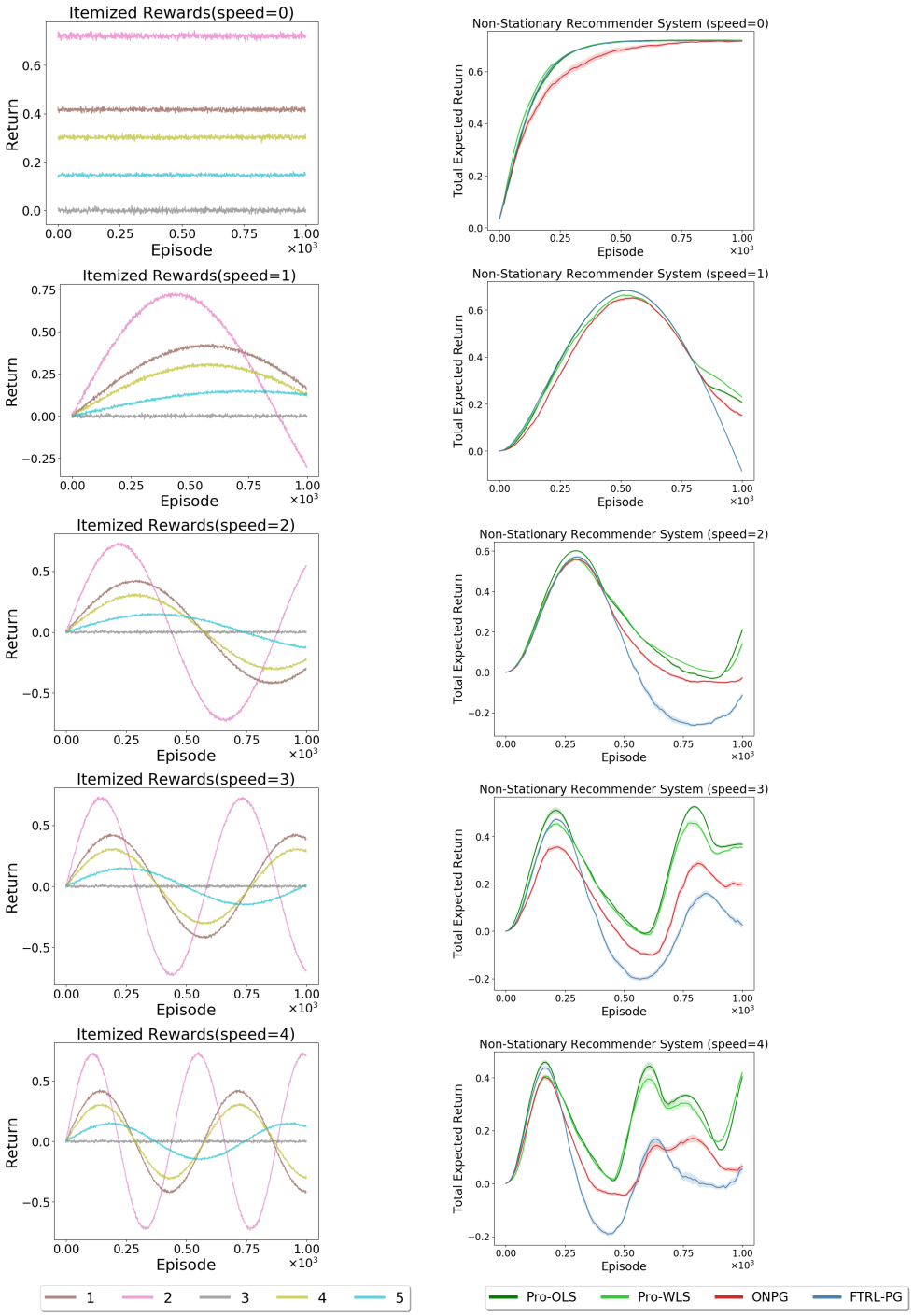


Figure 3.8. (Left) Fluctuations in the reward associated with each of the 5 items that can be recommended, for different speeds. (Right) Running mean of the best (among different hyper-parameters) performance of all the algorithms for different speeds; higher total expected return is better. Shaded regions correspond to the standard error of the mean obtained using 30 trials. Notice the shape of the performance curve for the proposed methods, which closely captures the trend of the maximum reward attainable over time.

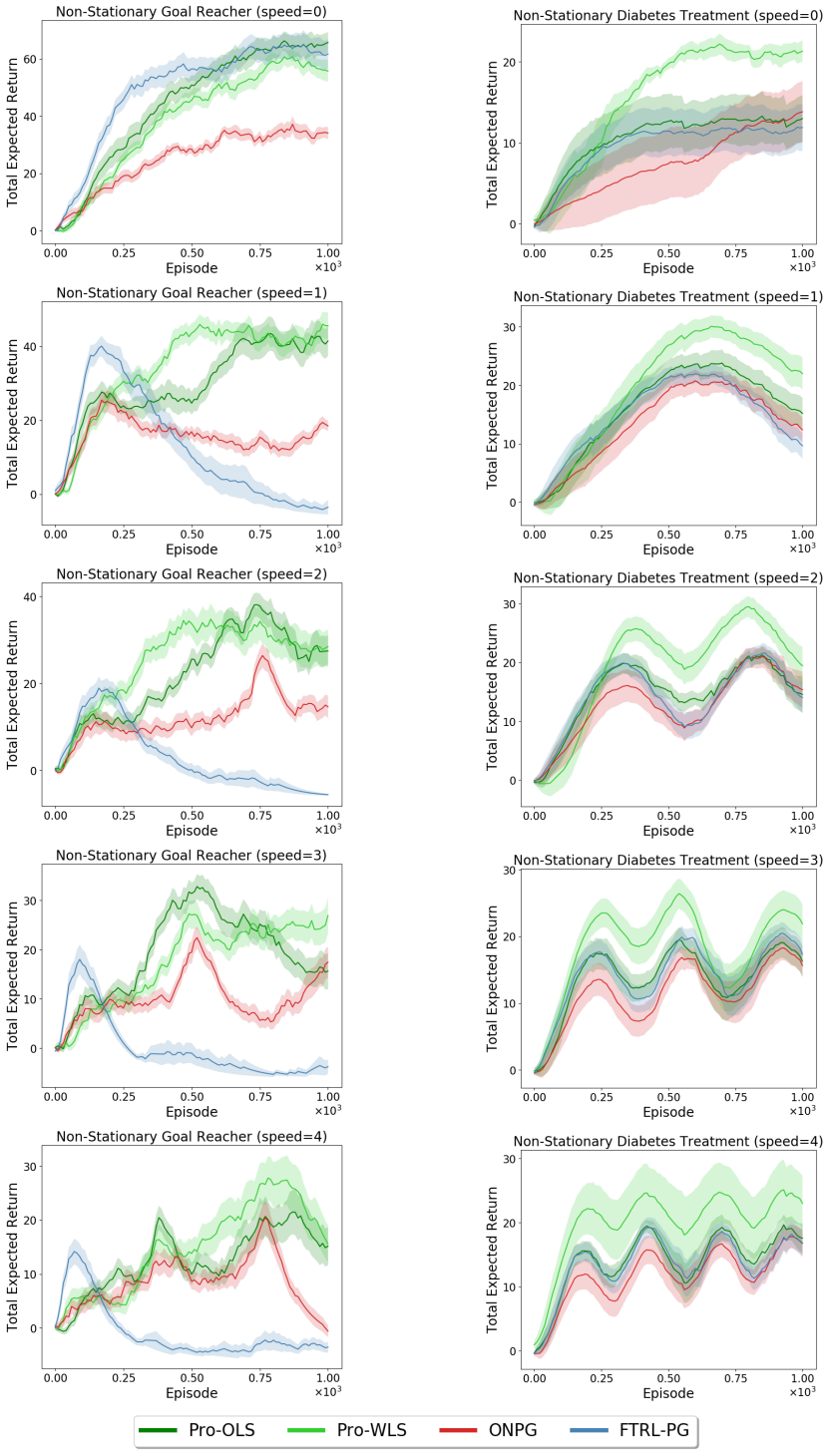


Figure 3.9. Running mean of the best performance of all the algorithms for different speeds; higher total expected return is better. Shaded regions correspond to the standard error of the mean obtained using 30 trials for NS Goal Reacher and 10 trials for NS Diabetes Treatment.

of a univariate time-series is advantageous as the model-bias of the environment can be reduced, this can result in higher variance in the forecasted performance. Developing lower variance off-policy performance estimators is also an active research direction which directly complements our algorithm. In particular, often a partial model of the environment is available and using it through doubly-robust estimators (Jiang and Li, 2015; Thomas and Brunskill, 2016) is an interesting future direction.

Further, there are other forecasting functions, like kernel regression, Gaussian Processes, ARIMA, etc., and some change-point detection algorithms that can potentially be used to incorporate more domain knowledge in the forecasting function Ψ, or make Ψ robust to jumps and auto-correlations in the time series.

3.9 Proofs

Here we provide proofs for the properties of the NIS and NWIS estimators. While NIS and NWIS are developed for the non-stationary setting, these properties ensure that these estimators generalize to the stationary setting as well. That is, when used in a stationary setting, the NIS estimator is both unbiased and consistent like the PDIS estimator, and the NWIS estimator is biased and consistent like the WIS estimator.

Our proof technique draws inspiration from the results presented by Mahmood et al. (2014). The key modification that we make to leverage their proof technique is that instead of using the features of the state as the input and the observed return from that corresponding state as the target to the regression function, we use the features of the *time index of an episode* as the input and the observed return for that corresponding episode as the target. In their setup, because states are drawn stochastically from a distribution, their analysis is not directly applicable to our setting where inputs are time indices that form a deterministic sequence. For analysis of our estimators, we leverage techniques discussed by Greene (2003) for analyzing properties of the ordinary least squares estimator.

In the following, we first establish the finite-sample properties and then we establish the large-sample properties for the NIS and NWIS estimators. Before proceeding further, recall from (3.6) and (3.4.5) that the NIS and NWIS estimators are given by:

$$\hat{J}_{k+\delta}(\pi) = \phi(k+\delta)w = \phi(k+\delta)(\Phi^\top\Phi)^{-1}\Phi^\top Y$$
$$\hat{J}^{\ddagger}_{k+\delta}(\pi) = \phi(k+\delta)w^{\ddagger} = \phi(k+\delta)(\Phi^\top\Lambda\Phi)^{-1}\Phi^\top\Lambda Y.$$

3.9.1 Finite Sample Properties

In this subsection, finite sample properties of NIS and NWIS are presented. Specifically, it is established that NIS is an unbiased estimator, whereas NWIS can be a biased estimator of $J(\pi)$, where $J(\pi)$ is the performance of a policy π in a stationary POMDP.

Theorem 5 (Unbiased NIS)**.** *Under Assumptions 1 and 2, for all $\delta \geq 1$, $\hat{J}_{k+\delta}(\pi)$ is an unbiased estimator of $J(\pi)$. That is, $\mathbb{E}[\hat{J}_{k+\delta}(\pi)] = J(\pi)$.*

Proof. Recall from (3.6) that

$$\hat{J}_{k+\delta}(\pi) = \phi(k+\delta)w = \phi(k+\delta)(\Phi^\top\Phi)^{-1}\Phi^\top Y.$$

Therefore, the expected value of $\hat{J}_{k+\delta}(\pi)$ is

$$\begin{aligned}\mathbb{E}[\hat{J}_{k+\delta}(\pi)] &= \mathbb{E}\left[\phi(k+\delta)(\Phi^\top\Phi)^{-1}\Phi^\top Y\right] \\ &= \phi(k+\delta)\left(\Phi^\top\Phi\right)^{-1}\left(\Phi^\top\mathbb{E}\left[Y\right]\right). \end{aligned} \tag{3.11}$$

As $Y = [\hat{J}_0(\pi), ..., \hat{J}_k(\pi)]^\top$ and the POMDP is stationary, the expected value of each element of Y is $J(\pi)$. Further, since $\phi(\cdot)$ always contains the bias co-efficient, and the performance of any policy is invariant to the episode number in a stationary

POMDP (Assumption 1), the optimal parameter for the regression model is $w^* = [0, 0, ..., 0, J(\pi)]^\top$, such that for any k,

$$\phi(k)w^* = [\phi_1(k), ..., \phi_{d-1}(k), 1][0, ..., 0, J(\pi)]^\top = J(\pi). \tag{3.12}$$

Therefore, $\mathbb{E}[Y] = \Phi w^*$. Using this observation in (3.11),

$$\begin{aligned}
\mathbb{E}[\hat{J}_{k+\delta}(\pi)] &= \phi(k+\delta)\left(\Phi^\top\Phi\right)^{-1}\left(\Phi^\top\Phi w^*\right) \\
&= \phi(k+\delta)\left(\Phi^\top\Phi\right)^{-1}\left(\Phi^\top\Phi\right)w^* \\
&= \phi(k+\delta)w^* \\
&= J(\pi).
\end{aligned}$$

□

Proof. (Alternate) Here we present an alternate proof for Theorem 5 which does not require invoking w^*.

$$\begin{aligned}
\mathbb{E}\left[\hat{J}_{k+\delta}(\pi)\right] &= \mathbb{E}\left[\phi(k+\delta)(\Phi^\top\Phi)^{-1}\Phi^\top Y\right] \\
&\overset{(a)}{=} \mathbb{E}\left[\sum_{i=0}^{k}\left[\phi(k+\delta)(\Phi^\top\Phi)^{-1}\Phi^\top\right]_i Y_i\right] \\
&\overset{(b)}{=} \sum_{i=0}^{k}\left[\phi(k+\delta)(\Phi^\top\Phi)^{-1}\Phi^\top\right]_i \mathbb{E}\left[Y_i\right],
\end{aligned}$$

where (a) is the dot product written as summation, and (b) holds because the multiplicative constants are fixed values, as given in (3.8). Since the environment is stationary, $\forall i\ \mathbb{E}\left[Y_i\right] = J(\pi)$, therefore,

$$\mathbb{E}\left[\hat{J}_{k+1}(\pi)\right] = J(\pi)\sum_{i=0}^{k}\left[\phi(k+\delta)(\Phi^\top\Phi)^{-1}\Phi^\top\right]_i. \tag{3.13}$$

In the following we focus on the terms inside the summation in (3.13). Without loss of generality, assume that for a given matrix of features Φ, the feature corresponding

to value 1 is in the last column of Φ. Let $\mathbf{A} := (\Phi^\top\Phi)^{-1}\Phi^\top \in \mathbb{R}^{d\times k}$, and let $B := \Phi[1:k, 1:d-1] \in \mathbb{R}^{k\times(d-1)}$ be the submatrix of Φ such that $\mathbf{B}$ has all features of Φ except the ones column, $\mathbb{1} \in \mathbb{R}^{k\times 1}$. Let $\mathbf{I}$ be the identity matrix in $\mathbb{R}^{d\times d}$, then it can seen that $(\Phi^\top\Phi)^{-1}(\Phi^\top\Phi)$ can be expressed as,

$$\left[\begin{array}{c} \mathbf{A} \end{array}\right]\left[\begin{array}{c|c} \mathbf{B} & \mathbb{1} \end{array}\right] = \mathbf{I},$$

This implies $[\mathbf{AB} \;\; \mathbf{A}\mathbb{1}] = \mathbf{I}$. Therefore, as the j^{th} row in last column of $\mathbf{I}$ corresponds to the dot product of the j^{th} row of $\mathbf{A}$, $\mathbf{A_j}$, with $\mathbb{1}$,

$$\mathbf{A}_j\mathbb{1} = \begin{cases} 0 & j \neq d, \\ 1 & j = d. \end{cases} \tag{3.14}$$

Equation (3.14) ensures that the summation of all rows of $\mathbf{A}$, except the last, sum to 0, and the last one sums to 1. Now, let $\phi(k+\delta) := [\phi_1(k+\delta), \phi_2(k+\delta), ..., \phi_{d-1}(k+\delta), 1] \in \mathbb{R}^{1\times d}$. Therefore,

$$
\begin{aligned}
\sum_{i=1}^{k}\left[\phi(k+\delta)(\Phi^\top\Phi)^{-1}\Phi^\top\right]_i &= \sum_{i=1}^{k}\left[\phi(k+\delta)\mathbf{A}\right]_i \\
&= \sum_{i=1}^{k}\sum_{j=1}^{d}\left[\phi(k+\delta)\right]_j \mathbf{A}_{j,i} \\
&= \sum_{j=1}^{d}\left[\phi(k+\delta)\right]_j \sum_{i=1}^{k}\mathbf{A}_{j,i} \\
&= \left(\sum_{j=1}^{d-1}\left[\phi(k+\delta)\right]_j \sum_{i=1}^{k}\mathbf{A}_{j,i}\right) + \left(\left[\phi(k+\delta)\right]_d \sum_{i=1}^{k}\mathbf{A}_{d,i}\right) \\
&= \left(\sum_{j=1}^{d-1}\left[\phi(k+\delta)\right]_j \left(\mathbf{A}_j \mathbb{1}\right)\right) + \left(\left[\phi(k+\delta)\right]_d \left(\mathbf{A}_d \mathbb{1}\right)\right) \\
&= \left(\sum_{j=1}^{d-1}\left[\phi(k+\delta)\right]_j \cdot 0\right) + \left(\left[\phi(k+\delta)\right]_d \cdot 1\right) \\
&= \left[\phi(k+\delta)\right]_d \\
&= 1. \qquad (3.15)
\end{aligned}
$$

Therefore, combining (3.15) with (3.13), $\mathbb{E}\left[\hat{J}_{k+\delta}(\pi)\right] = J(\pi)$. □

Theorem 6 (Biased NWIS)**.** *Under Assumptions 1 and 2, for all $\delta \geq 1$, $\hat{J}^{\ddagger}_{k+\delta}(\pi)$ may be a biased estimator of $J(\pi)$. That is, it is possible that $\mathbb{E}[\hat{J}^{\ddagger}_{k+\delta}(\pi)] \neq J(\pi)$.*

Proof. We prove this result using a simple counter-example. Consider the following basis function, $\phi(\cdot) = [1]$:

$$
\begin{aligned}
J^{\ddagger}_{k+\delta}(\pi) &= \phi(k+\delta)w^{\ddagger} \\
&= \phi(k+\delta)\ \operatorname*{argmin}_{c\in\mathbb{R}^{1\times 1}} \frac{1}{n}\sum_{i=1}^{n}\rho_i(0,T)(G_i - c\phi(i))^2 \\
&= \operatorname*{argmin}_{c\in\mathbb{R}^{1\times 1}} \frac{1}{n}\sum_{i=1}^{n}\rho_i(0,T)(G_i - c)^2 \\
&= \frac{\sum_{i=1}^{n}\rho_i(0,T)G_i}{\sum_{i=1}^{n}\rho_i(0,T)},
\end{aligned}
$$

which is the WIS estimator. Therefore, as WIS is a biased estimator (Precup, 2000), NWIS is also a biased estimator of $J(\pi)$. □

3.9.2 Large Sample Properties

In this subsection, large sample properties of NIS and NWIS are presented. Specifically, it is established that both NIS and NWIS are consistent estimators of $J(\pi)$, the performance of a policy π for a stationary POMDP.

Theorem 7 (Consistent NIS)**.** *Under Assumptions 1 and 2, for all $\delta \geq 1$, $\hat{J}_{k+\delta}(\pi)$ is a consistent estimator of $J(\pi)$. That is, as $N \to \infty$, $\hat{J}_{N+\delta}(\pi) \xrightarrow{a.s.} J(\pi)$.*

Proof. The proof follows from the standard consistency result for ordinary least-squares regression (Greene, 2003). Formally, using (3.6),

$$\begin{aligned}\lim_{N\to\infty} \quad \hat{J}_{N+\delta}(\pi) &= \lim_{N\to\infty} \quad \phi(N+\delta)w \\ &= \lim_{N\to\infty} \quad \phi(N+\delta)(\Phi^\top\Phi)^{-1}\Phi^\top Y.\end{aligned}$$

Since $Y = [\hat{J}_0(\pi), ..., \hat{J}_N(\pi)]^\top$ and the POMDP is stationary, each element of Y is an unbiased estimate of $J(\pi)$. In other words, $\forall i \in [0, N], \hat{J}_i(\pi) = J(\pi) + \epsilon_i$, where ϵ_i is a mean zero error. Let $\epsilon \in \mathbb{R}^{N\times 1}$ be the vector containing all the error terms ϵ_i. Now, using (3.12),

$$\begin{aligned}\lim_{N\to\infty} \quad \hat{J}_{N+\delta}(\pi) &= \lim_{N\to\infty} \quad \phi(N+\delta)\left(\Phi^\top\Phi\right)^{-1}\left(\Phi^\top(\Phi w^* + \epsilon)\right) \\ &= \lim_{N\to\infty} \quad \phi(N+\delta)\left(\Phi^\top\Phi\right)^{-1}\left(\left(\Phi^\top\Phi\right)w^* + \left(\Phi^\top\epsilon\right)\right) \\ &= \lim_{N\to\infty} \quad \phi(N+\delta)w^* + \phi(N+\delta)\left(\Phi^\top\Phi\right)^{-1}\left(\Phi^\top\epsilon\right) \\ &= \lim_{N\to\infty} \quad J(\pi) + \phi(N+\delta)\left(\Phi^\top\Phi\right)^{-1}\left(\Phi^\top\epsilon\right) \\ &= \lim_{N\to\infty} \quad J(\pi) + \phi(N+\delta)\left(\frac{1}{N}\Phi^\top\Phi\right)^{-1}\left(\frac{1}{N}\Phi^\top\epsilon\right).\end{aligned} \tag{3.16}$$

If both $Q^{-1} \coloneqq \left(\lim_{N\to\infty} \frac{1}{N}\Phi^\top\Phi\right)^{-1}$ and $\left(\lim_{N\to\infty} \frac{1}{N}\Phi^\top\epsilon\right)$ exist, then using Slutsky's Theorem,

$$\lim_{N\to\infty} \quad \hat{J}_{N+\delta}(\pi) = J(\pi) + \phi(N+\delta)Q^{-1}\left(\lim_{N\to\infty} \quad \frac{1}{N}\Phi^\top\epsilon\right). \tag{3.17}$$

To validate conditions for Slutsky's Theorem, notice that it holds from Grenander's conditions that Q^{-1} exists. Informally, Grenander's conditions require that no feature degenerates to a sequence of zeros, no feature of a single observation dominates the sum of squares of its series, and the $\Phi^\top\Phi$ matrix always has full rank. These conditions are easily satisfied for most popular basis functions used to create input features. For formal definitions of these conditions, we refer the reader to Chapter 5 of the work by Greene (2003).

In the following, we restrict our focus to the term inside the brackets in the second term of (3.17) and show that it exists, so that (3.17) is valid. Notice that the mean of that term is,

$$\mathbb{E}\left[\frac{1}{N}\Phi^\top\epsilon\right] = \frac{1}{N}\Phi^\top\mathbb{E}\left[\epsilon\right] = 0.$$

Since the mean is 0, the variance is

$$\mathbb{V}\left[\frac{1}{N}\Phi^\top\epsilon\right] = \frac{1}{N^2}\mathbb{V}\left[\Phi^\top\epsilon\right] = \frac{1}{N^2}\mathbb{E}\left[\left(\Phi^\top\epsilon\right)\left(\Phi^\top\epsilon\right)^\top\right] = \frac{1}{N^2}\left(\Phi^\top\mathbb{E}\left[\epsilon\epsilon^\top|\Phi\right]\Phi\right).$$

As each policy has a non-zero probability of taking any action in any state, the variance of PDIS (or the standard IS) estimator is bounded and thus each element of $\mathbb{E}[\epsilon\epsilon^\top|\Phi]$ is bounded. Further, as $\phi_i(\cdot)$ is bounded, each element of Φ is also bounded. Therefore,

$$\lim_{N\to\infty} \quad \mathbb{V}\left[\frac{1}{N}\Phi^\top\epsilon\right] \to 0.$$

Since the mean is 0 and the variance asymptotes to 0, by Kolmogorov's strong law of large numbers it follows that as $N \to \infty$, $\frac{1}{N}\Phi^\top \epsilon \xrightarrow{\text{a.s.}} 0$. Combining this with (3.17),

$$\lim_{N\to\infty} \hat{J}_{N+\delta}(\pi) \overset{a.s.}{\to} J(\pi) + \phi(N+\delta)Q^{-1}0 = J(\pi).$$

□

Theorem 8 (Consistent NWIS)**.** *Under Assumptions 1 and 2, for all $\delta \geq 1$, $\hat{J}^{\ddagger}_{k+\delta}(\pi)$ is a consistent estimator of $J(\pi)$. That is, as $N \to \infty$, $\hat{J}^{\ddagger}_{N+\delta}(\pi) \xrightarrow{a.s.} J(\pi)$.*

Proof. Recall from (3.10) that

$$\hat{J}^{\ddagger}_{N+\delta}(\pi) = \phi(N+\delta)w^{\ddagger} = \phi(N+\delta)(\Phi^\top \Lambda \Phi)^{-1}\Phi^\top \Lambda Y.$$

Consistency of $\hat{J}^{\ddagger}_{N+\delta}(\pi)$ can be proven similarly to the proof of Theorem 7. Note that here $Y = [G_0, ..., G_k]^\top$ contains the returns for each episode, and ΛY denotes the unbiased estimates for $J(\pi)$. Therefore, similar to (3.16),

$$\begin{aligned}
\lim_{N\to\infty} \hat{J}^{\ddagger}_{N+\delta}(\pi) &= \lim_{N\to\infty} \phi(N+\delta)(\Phi^\top \Lambda \Phi)^{-1}(\Phi^\top(\Phi w^* + \epsilon)) \\
&= \lim_{N\to\infty} \phi(N+\delta)(\Phi^\top \Lambda \Phi)^{-1}((\Phi^\top \Phi)w^* + \Phi^\top \epsilon) \\
&= \lim_{N\to\infty} \phi(N+\delta)\left(\frac{1}{N}\Phi^\top \Lambda \Phi\right)^{-1}\left(\left(\frac{1}{N}\Phi^\top \Phi\right)w^* + \frac{1}{N}\Phi^\top \epsilon\right) \quad (3.18)
\end{aligned}$$

In the following, we will make use of Slutsky's Theorem. To do so, we first restrict our focus to the terms in the first bracket in (3.18), and show existence of its limit. Let $\tilde{\rho}_k \coloneqq \rho^{\ddagger}_k - \mathbb{E}[\rho^{\ddagger}_k]$ be a mean 0 random variable, then

$$\begin{aligned}
\lim_{N\to\infty} \quad \frac{1}{N}\Phi^\top \Lambda \Phi &= \lim_{N\to\infty} \quad \frac{1}{N}\sum_{k=1}^{N} \rho_k^{\ddagger}\phi(k)^\top\phi(k). \\
&= \lim_{N\to\infty} \quad \left(\frac{1}{N}\sum_{k=1}^{N} \tilde{\rho}_k\phi(k)^\top\phi(k) + \frac{1}{N}\sum_{k=1}^{N} \mathbb{E}\left[\rho_k^{\ddagger}\right]\phi(k)^\top\phi(k)\right). \\
&\overset{(a)}{\to} \lim_{N\to\infty} \quad \frac{1}{N}\sum_{k=1}^{N} \mathbb{E}\left[\rho_k^{\ddagger}\right]\phi(k)^\top\phi(k) \\
&\overset{(b)}{=} \lim_{N\to\infty} \quad \frac{1}{N}\sum_{k=1}^{N} \phi(k)^\top\phi(k) \\
&= \lim_{N\to\infty} \quad \frac{1}{N}\Phi^\top\Phi, \qquad (3.19)
\end{aligned}$$

where (a) follows from the Kolmogorov's strong law of large numbers. To see this, let $Z_k = \tilde{\rho}_k\phi(k)^\top\phi(k)$. Notice that $\mathbb{E}[Z_k] = \mathbb{E}[\tilde{\rho}_k]\phi(k)^\top\phi(k) = 0$, and as both $\tilde{\rho}$ and $\phi(\cdot)$ are bounded, the variance of Z_k is also bounded. Therefore, $(1/N)\sum \tilde{\rho}_k\phi(k)^\top\phi(k) \to 0$ almost surely as $N \to \infty$. Consequently, (b) is obtained using the fact that the expected value of importance ratios is 1 (Thomas, 2015, Lemma 3). Notice that (3.19) reduced to Q (which was defined in the proof of Theorem 7) and we know that its limit exists because $\phi(\cdot)$ is bounded. Further, we also know that Q^{-1} and $\left(\lim_{N\to\infty}\frac{1}{N}\Phi^\top\epsilon\right)$ exist (see the proof of Theorem 7). Therefore, using Slutsky's Theorem and substituting (3.19) in (3.18),

$$\begin{aligned}
\lim_{N\to\infty} \quad \hat{J}_{N+\delta}^{\ddagger}(\pi) &= \phi(N+\delta)\left(\lim_{N\to\infty} \quad \frac{1}{N}\Phi^\top\Phi\right)^{-1}\left(\left(\lim_{N\to\infty} \quad \frac{1}{N}\Phi^\top\Phi\right)w^* + \lim_{N\to\infty} \quad \frac{1}{N}\Phi^\top\epsilon\right) \\
&= \phi(N+\delta)w^* + \phi(N+\delta)\left(\lim_{N\to\infty} \quad \frac{1}{N}\Phi^\top\Phi\right)^{-1}\left(\lim_{N\to\infty} \quad \frac{1}{N}\Phi^\top\epsilon\right) \\
&= J(\pi) + \phi(N+\delta)\left(\lim_{N\to\infty} \quad \frac{1}{N}\Phi^\top\Phi\right)^{-1}\left(\lim_{N\to\infty} \quad \frac{1}{N}\Phi^\top\epsilon\right) \\
&\xrightarrow{\text{a.s.}} J(\pi), \qquad (3.20)
\end{aligned}$$

where (3.20) follows from the simplification used for (3.17) in the proof of Theorem 7.

□

CHAPTER 4

TOWARDS SAFE POLICY IMPROVEMENT

This chapter is not a pre-requisite for the following chapters. Therefore, readers who are familiar with Chapter 3 and aim to read Chapter 5 can skip this chapter.

Reinforcement learning (RL) methods have shown potential for several real-world sequential decision-making problems such as diabetes management (Bastani, 2014), sepsis treatment (Saria, 2018), and budget constrained bidding (Wu et al., 2018). For such real-world applications, safety guarantees are critical to mitigate serious risks in terms of both human-life and monetary assets. More concretely, here, by *safety* we mean that any update to a system should not reduce the performance of an existing system (e.g., a doctor's initially prescribed treatment). As discussed in Chapter 2, a further complication is that many such practical applications of interest are non-stationary, thereby violating the foundational assumption (Sutton and Barto, 2018b) of *stationarity* required by most RL algorithms. This raises the main question we aim to address:

How can we build sequential decision-making systems that provide safety guarantees for problems with non-stationarities?

Conventionally, RL algorithms designed to ensure safety (Pirotta et al., 2013; Garcıa and Fernández, 2015; Thomas, 2015; Zhang and Cho, 2016; Laroche et al., 2017; Chow et al., 2018) rely upon the *stationarity assumption*. That is, they assume that a decision made by an *agent* always results in the same (distribution of) consequence(s)

when the environment is in a given state. Consequently, safety is only ensured by prior methods when the stationarity assumption holds, which is rare in real-world problems.

In this chapter, we take the first steps towards developing a method to address the issue of safety in the presence of *passive* non-stationarity, as discussed in Section 2.3. The proposed method builds upon the ideas established in Chapter 3 to construct a method for producing confidence intervals for the forecasted performance and use this method to search for a policy that can provide performance improvement with high-confidence.

Formally, using the Prognosticator procedure developed in Chapter 3, we first obtain obtain point-estimates for the forecast of the future performance for a policy π. Subsequently, we use a bootstrap based technique to obtain a high-confidence lower bound on the future performance. Using a gradient based technique, we show how this high-confidence lower bound can be leveraged to search for a policy that can provide improvement over the baseline (safe) policy with a high-confidence.

Contributions: The primary contributions of this chapter are:

- We formalize the *safe policy improvement* problem in the presence of passive non-stationarity and provide an algorithm for addressing it. Additionally, a user-controllable knob is provided to set the desired *confidence level*: the maximum admissible probability that an unsafe policy will be deployed.

- The proposed method only relies upon estimates of future performance, with associated confidence intervals. It does not require building a model of a non-stationary domain (NS-MDP), and so it is applicable to a broader class of problems, as modeling a non-stationary decision process can often be prohibitively difficult.

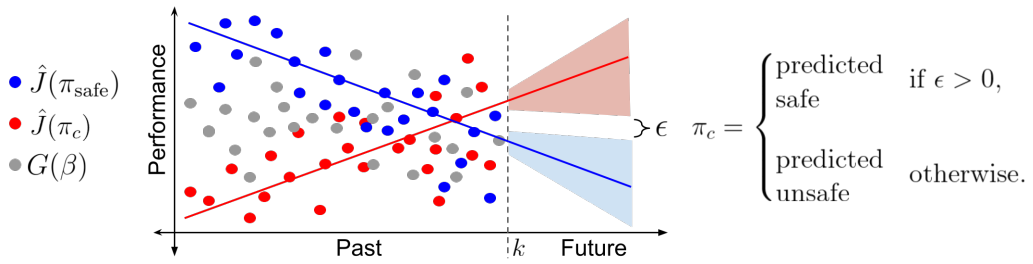


Figure 4.1. An illustration of the proposed idea where *safety* is defined to ensure that the future performance of a proposed policy π_c is never worse than that of an existing, known, safe policy π^{safe}. The gray dots correspond to the returns, $G(\beta)$, observed for a policy β. The red and the blue dots correspond to the counterfactual estimates, $\hat{J}(\pi_c)$ and $\hat{J}(\pi^{\text{safe}})$, for performance of π_c and π^{safe}, respectively. The shaded regions correspond to the uncertainty in future performance obtained by analysing the trend of the counterfactual estimates for past performances.

- The proposed method provides an efficient gradient based procedure to search for a policy that maximizes a bootstrap based high-confidence lower bound on future performance.

- Safety guarantees of the proposed method generalize to the stationary setting, meaning that there is little reason not to use our approach if there is a possibility that the system might be non-stationary. In Figure 4.1, we provide an illustration of the proposed approach for ensuring safe policy improvement amidst passive non-stationarity.

This chapter is organized as follows. Section 4.1 provides an overview of the notation and Section 4.2 provides the formal problem statement. Discussion of related work and additional background on some preliminary concepts for the proposed idea are presented in Section 4.3. In Section 4.4 we discuss the hardness of the problem and introduce an assumption to make the problem more tractable. An overview of the proposed idea is presented in Section 4.5 and Section 4.6 contains the procedure for obtaining confidence intervals on the forecasted performance, which is the key

component of the proposed method. Finally, Section 4.7 provides the complete algorithm and results are presented in Section 4.8.

4.1 Notation

Symbol	Meaning
M_i	POMDP for episode i.
$\mathscr{P}_i$	Transition dynamics for M_i.
$\mathscr{R}_i$	Reward function for M_i.
$\mathscr{O}_i$	Observation function for M_i.
μ_i	Starting state distribution for M_i.
γ	Discount factor.
π	Policy.
π^{safe}	Given baseline safe policy.
π_c	A candidate policy that can possibly be used for policy improvement.
β_i	Behavior policy used to collect data for episode i.
$G(\pi, m)$	Discounted episodic return of π for POMDP m.
$J(\pi, m)$	Expected discounted episodic return of π for POMDP m.
$J(\pi, i)$	Expected discounted episodic return for episode i.
$\hat{J}(\pi, i)$	An estimate of $J(\pi, i)$.
$\hat{J}^{\text{lb}}(\pi)$	High-confidence lower bound on the future performance of π.
$\hat{J}^{\text{ub}}(\pi)$	High-confidence upper bound on the future performance of π.
k	Current episode number.
δ	Number of episodes into the future.
H_i	Trajectory during episode i.
$\mathcal{D}$	Set of trajectories.
$\mathcal{D}_{\text{train}}$	Partition of $\mathcal{D}$ used for searching π_c.
$\mathcal{D}_{\text{test}}$	Partition of $\mathcal{D}$ used for safety test.
`alg`	An algorithm.

Table 4.1. List of symbols used in this chapter, and their associated meanings.

This chapter includes new notation beyond what was established in Chapter 2. For convenience, we provide Tables 4.1 and 4.2 containing the list of symbols used in this chapter. Some of these will be defined later in this chapter, as and when needed.

Recall from Chapter 2 that a non-stationary decision process (NSDP) is a sequence of POMDPs $(M_i)_{i=1}^{\infty}$. Let $\mathcal{M}$ be a set of POMDPs, where each POMDP is defined by

Symbol	Meaning
α	Quantity to define the desired safety level $1 - \alpha$.
X	Time indices for time-series.
Y	Time series values corresponding to X.
$\hat{Y}$	Estimates for Y.
ϕ	Basis function for time series forecasting.
Φ	Matrix containing basis for different episode numbers.
w	Parameters for time series forecasting.
ξ	Noise in the observed performances.
$\hat{\xi}$	Estimate for ξ.
$\hat{\Omega}$	Diagonal matrix containing $\hat{\xi}^2$.
$\hat{s}$	Standard deviation of the forecast.
$\mathtt{t}$	$\mathtt{t}$-statistic for the forecast.
$\mathtt{t}_\alpha$	α-quantile of the $\mathtt{t}$ distribution.
$\mathscr{C}$	Function to obtain confidence interval on future performance.
σ^*	Rademacher random variable.
Y^*	Pseudo-variable for Y.
$\hat{Y}^*$	Pseudo-variable for $\hat{Y}$.
ξ^*	Pseudo-variable for ξ.
$\hat{\xi}^*$	Pseudo-variable for $\hat{\xi}$.
$\mathtt{t}^*$	Pseudo-variable for $\mathtt{t}$.
$\mathtt{t}^*_\alpha$	Pseudo-variable for $\mathtt{t}_\alpha$.
$\hat{s}^*$	Pseudo-variable for $\hat{s}$.

Table 4.2. List of symbols used in this chapter, and their associated meanings.

the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathscr{O}, \mathscr{P}, \mathscr{R}, \mu)$. The observation function $\mathscr{O}_i$, the transition function $\mathscr{P}_i$, the reward function $\mathscr{R}_i$, and the initial state distribution μ_i can differ for each POMDP M_i. Recall that O_i^t, A_i^t, and R_i^t to denote the random variables corresponding to the observation, action, and reward at timestep t in POMDP M_i. The sequence of interactions in M_i is denoted by $H_i \coloneqq (S_i^t, O_i^t, A_i^t, R_i^t)_{i=1}^T$. For clarity in this chapter, we will define variables associated with return slightly differently from how they were defined in Chapter 2. Let a *return* of π for any $m \in \mathcal{M}$ be $G(\pi, m) \coloneqq \sum_{t=0}^{\infty} \gamma^t R^t$ and the *expected return* $J(\pi, m) \coloneqq \mathbb{E}[G(\pi, m)]$. With a slight overload of notation, let the *performance* of π for episode i be $J(\pi, i) \coloneqq \mathbb{E}[J(\pi, M_i)]$. We will use k to denote the

most recently finished episode, such that episode numbers $[1, k]$ are in the past and episode numbers $(k, \infty]$ are in the future.

The set of possible interaction sequences is denoted by $\mathcal{H}$, and $\mathcal{T} : \mathcal{M} \times \mathcal{H} \times \mathcal{M} \to [0, 1]$ is the 'meta-transition' function that governs the non-stationarity in the POMDPs. That is, $\mathcal{T}(m, h, m') = \Pr(M_{i+1}{=}m' | M_i{=}m, H_i{=}h)$. In this chapter we consider the restricted case where the non-stationarity is passive, i.e., only caused by external factors:

$$\forall (m, m') \in \mathcal{M}^2, \forall (h, h') \in \mathcal{H}^2, \qquad \mathcal{T}(m, h, m') = \mathcal{T}(m, h', m').$$

4.2 Problem Statement

Let $\mathcal{D} \coloneqq ((i, H_i) : i \in [1, k])$ be a random variable denoting a set of trajectories observed in the past. With slight abuse of notation, here we define $H_i \coloneqq (O_i^t, A_i^t, R_i^t)_{i=1}^T$ without the state variables as those variables are unobserved. Let `alg` be an algorithm that takes $\mathcal{D}$ as input and returns a policy π. Let π^{safe} be a known safe policy, and let $(1 - \alpha) \in [0, 1]$ be a constant selected by a user of `alg`, which we call the *safety level*. We aim to create an algorithm `alg` that ensures with high probability that $\texttt{alg}(\mathcal{D})$, the policy proposed by `alg`, does not perform worse than the existing safe policy π^{safe} during the *future* episode $k + \delta$. That is, we aim to ensure the following *safety guarantee*,

$$\Pr\Big(J(\texttt{alg}(\mathcal{D}), k + \delta) \geq J(\pi^{\text{safe}}, k + \delta)\Big) \geq 1 - \alpha. \tag{4.1}$$

4.3 Background and Preliminaries

In this section, we discuss related work and provide brief overviews of Seldonian algorithms (Thomas et al., 2019a) and wild bootstrap (Wu et al., 1986; Mammen, 1993) methods that the proposed method builds upon.

4.3.1 Related Work

While some works for lifelong-reinforcement learning (Brunskill and Li, 2014; Abel et al., 2018; Chandak et al., 2020a;c) or meta-reinforcement learning (Al-Shedivat et al., 2017; Xie et al., 2020a) do aim to address the problem of non-stationarity, they do not provide any safety guarantees. Perhaps the work most closely related to ours is by Ammar et al. (2015), which aims to find a policy that satisfies a safety constraint in the lifelong-learning setting. They use a follow-the-regularized-leader (FTRL) (Shalev-Shwartz et al., 2012) approach to first perform an unconstrained maximization over the *average* performance over all the trajectories collected in the past, and then project the resulting solution onto a safe set. However, as shown by Chandak et al. (2020c), FTRL based methods can suffer from a significant performance lag in non-stationary environments. Further, the parameter projection requires *a priori* knowledge of the set of safe policy parameters, which might be infeasible to obtain for many problems, especially when the constraint is to improve performance over an existing policy or when the safe set is non-convex (e.g., when using policies parameterized using neural networks). Additionally, the method proposed by Chandak et al. (2020c) for policy improvement does not provide safety guarantees, and thus it would be irresponsible to apply it to safety-critical problems.

4.3.2 Wild Bootstrap

The goal of this section is to provide a brief introduction to the (wild) bootstrap that we later use within our proposed method. Therefore, this section contains a summary of existing works and has no original technical contribution. We begin by first discussing the idea behind any general bootstrap and the wild bootstrap method. Subsequently, we discuss alternatives to wild bootstrap.

In many practical applications, it is often desirable to infer distributional properties (e.g., CIs) of a desired statistic of data (e.g., sample mean). However, in practice, it is

often not possible to get multiple estimates of the desired statistic in a data-efficient way. To address this problem, bootstrap methods have received wide popularity in the field of computational statistics (Efron and Tibshirani, 1994).

The core principle of any bootstrap procedure is to *re-sample* the observed data-set $\mathcal{D}$ and construct multiple *pseudo data-sets* $\mathcal{D}^*$ in a way that closely mimics the original *data generating process* (DGP). This allows the creation of an *empirical distribution* of the desired statistic by leveraging multiple pseudo data-sets $\mathcal{D}^*$ (Efron and Tibshirani, 1994). For example, an empirical distribution containing B estimates of the sample mean can be obtained by generating B pseudo data-sets, where each data-set contains N samples uniformly drawn (with replacement) from the original data-set of size N.

For an excellent introduction to bootstrap CIs, refer to the works by Efron and Tibshirani (1994) and DiCiccio and Efron (1996). The book by Hall (2013) provides a thorough treatment of these methods using *Edgeworth expansion*, illustrating when and how bootstrap methods can provide significant advantage over other methods. For a very readable practitioner's guide that touches upon several important aspects, refer to the work by Carpenter and Bithell (2000).

Wild bootstrap: The original idea of wild bootstrap was proposed by Wu et al. (1986) and later developed by Liu et al. (1988), Mammen (1993), and Davidson and Flachaire (1999; 2008). The following summary about the wild bootstrap process is based on an excellent tutorial by MacKinnon (2012).

Consider the system of equations in (4.10). The key idea of wild-bootstrap is that the uncertainty in regression estimates (of parameters/predictions) is due to the noise ξ in the observations. Therefore, if the pseudo-data Y^* is generated such that the noise ξ^* in the data generating process for Y^* resembles the properties of the true underlying noise ξ, then with multiple redraws of such Y^* one can obtain an empirical distribution of the desired statistic (which for our case, corresponds to the forecast of a policy π's performance). This can then be used to estimate the CIs.

As true noise ξ is unobserved, it raises a question about how to estimate its properties to generate Y^*. Fortunately, as ordinary least-squares is an unbiased estimator of parameters/predictions (Wasserman, 2013), regression errors $\hat{\xi}$ can be used as a substitute for the true noise. Therefore, to mimic the underlying data generating process, it would be ideal to have bootstrap error terms ξ^* that have similar moments as $\hat{\xi}$. Following the work by Davidson and Flachaire (1999), we set $Y^* := \hat{Y} + \xi^*$, where $\xi^* := \hat{\xi} \odot \sigma^*$, and $\sigma^* \in \mathbb{R}^{k\times 1}$ is the independent Rademacher random variable (i.e., $\forall i \in [1,k],\ \Pr(\sigma_i^* = +1) = \Pr(\sigma_i^* = -1) = 0.5$). This choice of σ_i^*, for all $i \in [1,k]$, ensures that ξ_i^* has the desired zero mean and the same higher-order *even* moments as $\hat{\xi}_i$ because,

$$\forall i,\ \mathbb{E}[\sigma_i^*] = 0,\ \mathbb{E}[\sigma_i^{*2}] = 1,\ \mathbb{E}[\sigma_i^{*3}] = 0,\ \mathbb{E}[\sigma_i^{*4}] = 1.$$

Therefore, for the purpose of this paper, pseudo performances Y^* generated using pseudo-noise ξ^* allow generating a distribution of $\hat{J}(\pi, k+\delta)^*$ that closely mimics the distribution of forecasts $\hat{J}(\pi, k+\delta)$ that would have been generated if we had the true underlying data generating process. Here, different pseudo noises ξ^* are created using different σ^*, which subsequently allows obtaining different pseudo performances Y^*.

4.4 Hardness of the Problem

In this section, we discuss the difficulty of the problem defined in (4.1), and introduce a smoothness assumption that we leverage to make the problem tractable.

While it is desirable to ensure the safety guarantee in (4.1), obtaining a new policy from $\texttt{alg}(\mathcal{D})$ that meets the requirement in (4.1) might be impossible unless some more regularity assumptions are imposed on the problem. To see why, notice that if the environment can change arbitrarily, then there is not much hope of estimating $J(\pi, k+\delta)$ accurately since $J(\pi, k+\delta)$ for any π could be any value between the

extremes of all possible outcomes, regardless of the data collected during episodes 1 through k.

To avoid arbitrary changes, previous works typically require the transition function $\mathscr{P}_k$ and the reward function $\mathscr{R}_k$ to be Lipschitz smooth over time (Lecarpentier and Rachelson, 2019; Jagerman et al., 2019a; Lecarpentier et al., 2020; Cheung et al., 2020). The bound on the change in performance given such Lipschitz conditions can be prohibitively loose. Even without the complexities of the full POMDP setting, this looseness can be observed even with MDPs (i.e., $\forall t,\ O_t = S_t$), as we show below in Theorem 9. Unfortunately, the bound on how much the performance of a policy can change can be quite large when the transition or reward function changes only a little): unless the Lipschitz constants are so small that they effectively make the problem stationary, the performance of a policy π across consecutive episodes can still fluctuate wildly. Notice that due to the inverse dependency on $(1-\gamma)^2$, if γ is close to one, then the Lipschitz constant L can be enormous even when ϵ_P and ϵ_R are small. In Section 4.11.1 we also provide an example of a non-stationary decision process for which Theorem 9 holds with exact equality, illustrating that the bound is tight.

Theorem 9 (Lipschitz smooth performance)**.** *If* $\exists \epsilon_P \in \mathbb{R}$ *and* $\exists \epsilon_R \in \mathbb{R}$ *such that for any* M_k *and* M_{k+1}, $\forall s \in \mathcal{S}, \forall a \in \mathcal{A},\ \|\mathscr{P}_k(\cdot|s,a) - \mathscr{P}_{k+1}(\cdot|s,a)\|_1 \leq \epsilon_P$ *and* $|\mathbb{E}[\mathscr{R}_k(s,a)] - \mathbb{E}[\mathcal{R}_{k+1}(s,a)]| \leq \epsilon_R$*, then the performance of any policy* π *is Lipschitz smooth over time, with Lipschitz constant* $L \coloneqq \left(\frac{\gamma R_{max}}{(1-\gamma)^2}\epsilon_P + \frac{1}{1-\gamma}\epsilon_R\right)$*. That is,* $\forall k \in \mathbb{N}_{>0}, \forall \delta \in \mathbb{N}_{>0},\ |J(\pi,k) - J(\pi,k+\delta)| \leq L\delta$.

Proof. See Section 4.11.1. □

4.4.1 An Alternate Assumption

In many real-world sequential decision-making problems, when there is non-stationarity the performance of a policy π does not fluctuate wildly between consecutive episodes. Examples where performance changes are likely more regular include the

effect of a medical treatment on a patient; the usefulness of online recommendations based on the interests of a user; or the quality of a controller as a robot's motor friction or battery capacity degrades. Therefore, instead of considering smoothness constraints on the transition function $\mathscr{P}_k$ and the reward function $\mathscr{R}_k$ like above, we consider more direct smoothness constraints on the performance $J(\pi, i)$ of a policy π. Similar assumptions have been considered for analyzing trends for digital marketing (Thomas et al., 2017) and remain popular among policymakers for designing policies based on forecasting (Wieland and Wolters, 2013).

If $J(\pi, i)$ changes smoothly with episode i, then the performance trend of a given policy π can be seen as a *univariate time-series*, i.e., a sequence of *scalar* values corresponding to performances $(J(\pi, i))_{i=1}^{k}$ of π during episodes 1 to k. Leveraging this observation, we propose modeling the performance trend using a linear regression model that takes an episode number as input and provides a performance prediction as output. To ensure that a wide variety of trends can be modeled, we use a d-dimensional non-linear *basis function* $\phi : \mathbb{N}_{>0} \to \mathbb{R}^{1\times d}$. For example, ϕ can be the Fourier basis, which has been known to be useful for modeling a wide variety of trends and is fundamental for time-series analysis (Bloomfield, 2004). We state this formally in the following assumption,

Assumption 3 (Smooth performance)**.** *For every policy π, there exists a sequence of mean-zero and independent noises $(\xi_i)_{i=1}^{k+\delta}$, and $\exists w \in \mathbb{R}^{d\times 1}$, such that, $\forall i \in [1, k+\delta]$,* $J(\pi, M_i) = \phi(i)w + \xi_i$.

Recall that the stochasticity in $J(\pi, M_i)$ is a manifestation of stochasticity in M_i, and thus this assumption requires that the performance of π during episode i is $J(\pi, i) = \mathbb{E}[J(\pi, M_i)] = \phi(i)w$.

Assumption 1 is reasonable for several reasons. The first is that the noise assumptions are not restrictive. The distribution of ξ_i does not need to be known and the ξ_i can be non-identically distributed. Additionally, both w and $(\xi_i)_{i=1}^{k+\delta}$ can be different

for different policies. The independence assumption only states that at each time step, the variability in performance due to sampling M_i is independent of the past (i.e., there is no auto-correlated noise).

The strongest requirement is that the performance trend be a linear function of the basis ϕ; but because ϕ is a generic basis, this is satisfied for a large set of problems. Standard methods that make stationarity assumptions correspond to our method with $\phi(s) = [1]$ (fitting a horizontal line). Otherwise, ϕ is generic: we might expect that there exist sufficiently rich features (e.g., Fourier basis (Bloomfield, 2004))) for which Assumption 1 is satisfied. In practice, we may not have access to such a basis, but like any time-series forecasting problem, goodness-of-fit tests (Chen et al., 2003) can be used by practitioners to check whether Assumption 3 is reasonable before applying our method.

The basis requirement, however, can be a strong condition and could be violated. This assumption is *not* applicable for settings where there are jumps or breaks in the performance trend. For example, performance change is sudden when a robot undergoes physical damage, its sensors are upgraded, or it is presented with a completely new task. The other potential violation is the fact that the basis is a function of time. Since the dimension d of the basis ϕ is finite and fixed, but k can increase indefinitely, this assumption implies that performance trends of the policies must exhibit a global structure, such as periodicity. This can be relaxed using auto-regressive methods that are better at adapting to the local structure of any time-series. We discuss this and other potential future research directions in Section 4.10.

4.5 SPIN: Safe Policy Improvement for Non-Stationary Settings

To ensure safe policy improvement, we adapt the generic template of the Seldonian framework (Thomas et al., 2019b) to the non-stationary setting. The overall approach

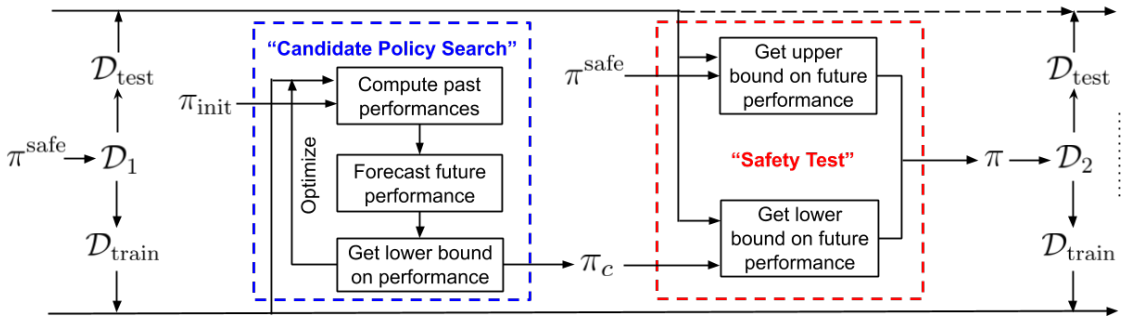


Figure 4.2. The proposed algorithm first partitions the initial data $\mathcal{D}_1$ into two sets, namely $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$. Subsequently, $\mathcal{D}_{\text{train}}$ is used to search for a possible *candidate policy* π_c that might improve the future performance, and $\mathcal{D}_{\text{test}}$ is used to perform a safety test on the proposed candidate policy π_c. The existing safe policy π^{safe} is only updated if the proposed policy π_c passes the safety test.

consists of continually (1) taking an existing safe policy; (2) finding a candidate policy that has (reasonably high) potential to be a strict improvement on the safe policy; (3) testing if this candidate policy is still safe and is an improvement with high confidence; (4) updating the policy to be the candidate policy only if it passes the test; and (5) gathering more data with the current safe policy to get data for the next candidate policy search. This procedure consists of four key technical steps: *performance estimation*, *safety test*, *candidate policy search*, and *data-splitting*. A schematic diagram of the overall procedure is provided in Figure 4.2.

4.5.1 Performance Estimation

To develop an algorithm that ensures the safety constraint in (4.1), we first require an estimate $\hat{J}(\pi, k+\delta)$ of the future performance $J(\pi, k+\delta)$ and the uncertainty of this estimate, namely a function $\mathscr{C}$ for obtaining a confidence interval (CI) on future performances. The function $\mathscr{C}$ should take as input a dataset, number of future episodes δ, and a confidence parameter α. Ideally, when a two-sided high-confidence bound is required then the function $\mathscr{C}$ should output $\hat{J}^{\text{lb}}(\pi)$ and $\hat{J}^{\text{ub}}(\pi)$ such that $\Pr(J(\pi, k+\delta) \in [\hat{J}^{\text{lb}}(\pi), \hat{J}^{\text{ub}}(\pi)]) \geq 1-\alpha$. Sometimes, only a high confidence

lower bound would be needed, in which case $\mathscr{C}$ outputs $\hat{J}^{\text{lb}}(\pi)$ and ∞ such that $\Pr(J(\pi, k+\delta) \in [\hat{J}^{\text{lb}}(\pi), \infty)) \geq 1-\alpha$.

Under Assumption 3, estimating $J(\pi, k+\delta)$ (the performance of a policy δ episodes into the future) can be seen as a *time-series forecasting* problem given the *performance trend* $(J(\pi, i))_{i=1}^{k}$. We build upon the work by Chandak et al. (2020c) to estimate $J(\pi, k+\delta)$. However, to the best of our knowledge, no method yet exists to obtain $\mathscr{C}$. A primary contribution of this work is to provide a solution to this technical problem, developed in Section 4.6.

4.5.2 Safety Test

To satisfy the required safety constraint in (4.1), an algorithm `alg` needs to ensure with high-confidence that a given π_c, which is a *candidate policy* for updating the existing safe policy π^{safe}, will have a higher future performance than that of π^{safe}. Importantly, just as the future performance, $J(\pi_c, k+\delta)$, of π_c is not known *a priori* for a non-stationary domain, the future performance of the baseline policy π^{safe} is also not known *a priori*. Therefore, to ensure that the constraint in (4.1) is satisfied, we use $\mathscr{C}$ to obtain a *one-sided* high-confidence lower and upper bound for $J(\pi_c, k+\delta)$ and $J(\pi^{\text{safe}}, k+\delta)$, respectively, each with confidence level $\alpha/2$. The confidence level is set to $\alpha/2$ so that the total failure rate (i.e., either $J(\pi_c, k+\delta)$ or $J(\pi^{\text{safe}}, k+\delta)$ is outside their respective high-confidence bounds) is no more than α. Subsequently, `alg` only updates π^{safe} if the high-confidence lower bound of $J(\pi_c, k+\delta)$ is higher than the high-confidence upper bound of $J(\pi^{\text{safe}}, k+\delta)$; otherwise, no update is made and π^{safe} is chosen to be executed again.

4.5.3 Candidate Policy Search

An ideal candidate policy π_c would be one that has high future performance $J(\pi_c, k+\delta)$, along with a higher high-confidence lower bound on its performance, so that it can pass the safety test. However, in practice, there could often be conflicts

between policies that might have higher estimated future performance but with a low high-confidence lower bound, and policies with lower estimates of future performance but with higher high-confidence lower bound. As the primary objective of our method is to ensure safety, we draw inspiration from prior methods for conservative/safe learning in stationary domains (Garcıa and Fernández, 2015; Thomas et al., 2015a; Kazerouni et al., 2017; Chow et al., 2018) and propose searching for a policy that has the *greatest* high-confidence lower bound. That is, let the one-sided CI for the future performance $J(\pi, k+\delta)$ obtained using $\mathscr{C}$ be $[\hat{J}^{\text{lb}}(\pi), \infty)$, then $\pi_c \in \text{argmax}_\pi \hat{J}^{\text{lb}}(\pi)$.

4.5.4 Data-Splitting:

Conventionally, in the time-series literature, there is only a single trend that needs to be analyzed. In our problem setup, however, the time series forecasting function is used to analyze trends of multiple policies during the candidate policy search. If all of the available data $\mathcal{D}$ is used to estimate the high-confidence lower bound $\hat{J}^{\text{lb}}(\pi)$ for $J(\pi, k+\delta)$ and if π is chosen by maximizing $\hat{J}^{\text{lb}}(\pi)$, then due to the *multiple comparisons problem* (Benjamini and Hochberg, 1995) we are likely to find a π that over-fits to the data and achieves a higher value of $\hat{J}^{\text{lb}}(\pi)$. A safety test based on such a $\hat{J}^{\text{lb}}(\pi)$ would thus be unreliable. To address this problem, we partition $\mathcal{D}$ into two mutually exclusive sets, namely $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$, such that only $\mathcal{D}_{\text{train}}$ is used to search for a candidate policy π_c and only $\mathcal{D}_{\text{test}}$ is used during the safety test.

4.6 Estimating Confidence Intervals for Future Performance

To complete the SPIN framework discussed in Section 4.5, we need to obtain an estimate $\hat{J}(\pi, k+\delta)$ of $J(\pi, k+\delta)$ and its confidence interval using the function $\mathscr{C}$. This requires answering two questions: (1) Given that in the past, policies $(\beta_i)_{i=1}^k$ were used to generate the observed returns, how do we estimate $\hat{J}(\pi, k+\delta)$ for a *different* policy π? (2) Given that the trajectories are obtained only from a *single* sample of

the sequence $(M_i)_{i=1}^k$, how do we obtain a confidence interval around $\hat{J}(\pi, k+\delta)$? We answer these two questions in this section.

4.6.1 Point Estimate of Future Performance

To answer the first question, we build upon the following observation used by Chandak et al. (2020c): While in the past, returns were observed by executing policies $(\beta_i)_{i=1}^k$, *what if* policy π was executed instead?

Formally, we use per-decision importance sampling (Precup, 2000) for H_i, to obtain a *counterfactual* estimate $\hat{J}(\pi, i) \coloneqq \sum_{t=0}^{\infty} \left(\prod_{l=0}^{t} \frac{\pi(A_i^l|S_i^l)}{\beta_i(A_i^l|S_i^l)}\right) \gamma^t R_i^t$, of π's performance in the past episodes $i \in [1, k]$. This estimate $\hat{J}(\pi, i)$ is an unbiased estimator of $J(\pi, i)$, i.e., $\mathbb{E}[\hat{J}(\pi, i)] = J(\pi, i)$, under the the following assumption (Thomas, 2015), which can typically be satisfied using an entropy-regularized policy β_i.

Assumption 4 (Full Support)**.** *$\forall a \in \mathcal{A}$ and $\forall s \in \mathcal{S}$ there exists a $c > 0$ such that $\forall i, \beta_i(a|s) > c$.*

Having obtained counterfactual estimates $(\hat{J}(\pi, i))_{i=1}^k$, we can then estimate $J(\pi, k+\delta)$ by analysing the performance trend of $(\hat{J}(\pi, i))_{i=1}^k$ and forecasting the future performance $\hat{J}(\pi, k+\delta)$. That is, let $X \coloneqq [1, 2, ..., k]^\top \in \mathbb{R}^{k\times 1}$, let $\Phi \in \mathbb{R}^{k\times d}$ be the corresponding basis matrix for X such that i^{th} row of Φ, $\forall i \in [1, k]$, is $\Phi_i \coloneqq \phi(X_i)$, and let $Y \coloneqq [\hat{J}(\pi, 1), \hat{J}(\pi, 2), ..., \hat{J}(\pi, k)]^\top \in \mathbb{R}^{k\times 1}$. Then under Assumptions 3 and 4, an estimate $\hat{J}(\pi, k+\delta)$ of the future performance can be computed using least-squares (LS) regression, i.e., $\hat{J}(\pi, k+\delta) = \phi(k+\delta)\hat{w} = \phi(k+\delta)(\Phi^\top\Phi)^{-1}\Phi^\top Y$.

4.6.2 Confidence Intervals for Future Performance

We now aim to quantify the uncertainty of $\hat{J}(\pi, k+\delta)$ using a confidence interval (CI), such that the true future performance $J(\pi, k+\delta)$ will be contained within the CI with the desired confidence level. To obtain a CI for $J(\pi, k+\delta)$, we make use of $\mathtt{t}$-statistics (Wasserman, 2013) and use the following notation. Let the sample standard

deviation for $\hat{J}(\pi, k+\delta)$ be $\hat{s}$, where $\hat{s}^2 \coloneqq \phi(k+\delta)(\Phi^\top\Phi)^{-1}\Phi^\top\hat{\Omega}\Phi(\Phi^\top\Phi)^{-1}\phi(k+\delta)^\top$, where $\hat{\Omega}$ is a diagonal matrix containing the square of the regression errors $\hat{\xi}$ (see Section 4.11.2.0.1 for more details), and let the $\mathtt{t}$-statistic be $\mathtt{t} \coloneqq (\hat{J}(\pi, k+\delta) - J(\pi, k+\delta))/\hat{s}$.

If the distribution of $\mathtt{t}$ was known, then a $(1-\alpha)100\%$ CI could be obtained as $[\hat{J}(\pi, k+\delta) - \hat{s}\mathtt{t}_{1-\alpha/2},\ \hat{J}(\pi, k+\delta) - \hat{s}\mathtt{t}_{\alpha/2}]$, where for any $\alpha \in [0,1], \mathtt{t}_\alpha$ represents α-*percentile* of the $\mathtt{t}$ distribution. Unfortunately, the distribution of $\mathtt{t}$ is not known. One alternative could be to assume that $\mathtt{t}$ follows the *student*-$\mathtt{t}$ distribution (Student, 1908). However, that would only be valid if *all* the error terms in regression are *homoscedastic* and *normally* distributed. Such an assumption could be severely violated in our setting due to the heteroscedastic nature of the estimates of the past performances resulting from the use of potentially different behavior policies $(\beta_i)_{i=1}^k$ and due to the unknown form of stochasticity in $(M_i)_{i=1}^k$. Further, due to the use of importance sampling, the performance estimates $(\hat{J}(\pi, i))_{i=1}^k$ can often be skewed and have a heavy-tailed distribution with high-variance (Thomas et al., 2015c). We provide more discussion on these issues in Section 4.6.3.2.

To resolve the above challenges, we make use of *wild bootstrap*, a semi-parametric bootstrap procedure that is popular in time series analysis and econometrics (Wu et al., 1986; Liu et al., 1988; Mammen, 1993; Davidson and Flachaire, 1999; 2008). The idea is to generate multiple pseudo-samples of performance for each M_i, using the single performance estimate that was sampled. These multiple pseudo-samples can then be used to obtain an empirical distribution and thus characterize the range of possible performances. As we elaborate later, $\mathtt{t}$-statistic $\mathtt{t}^*$ corresponding to the pseudo samples can be constructed, and subsequently the α-percentile for these $\mathtt{t}^*$ can be used to estimate the α-percentile of the distribution of $\mathtt{t}$. Below, we discuss how to get these multiple pseudo-samples.

Recall that trajectories $(H_i)_{i=1}^k$ are obtained only from a *single* sample of the sequence $(M_i)_{i=1}^k$. Due to this, only a single point estimate $\hat{J}(\pi, k+\delta)$, devoid of

any estimate of uncertainty, of the future performance $J(\pi, k+\delta)$ can be obtained. Therefore, we aim to create *pseudo-samples* of $(\hat{J}(\pi, i))_{i=1}^{k}$ that *resemble* the estimates of past performances that would have been obtained using trajectories from an alternate sample of the sequence $(M_i)_{i=1}^{k}$. The wild bootstrap procedure provides just such an approach, with the following steps.

1. Let $Y^+ \coloneqq [J(\pi,1), ..., \ J(\pi,k)]^\top \in \mathbb{R}^{k\times 1}$ correspond to the true performances of π. Create $\hat{Y} = \Phi(\Phi^\top\Phi)^{-1}\Phi^\top Y$, an LS estimate of Y^+, using the counterfactual performance estimates Y and obtain the regression errors $\hat{\xi} \coloneqq \hat{Y} - Y$.

2. Create pseudo-noises $\xi^* \coloneqq \hat{\xi} \odot \sigma^*$, where $\odot$ represents the Hadamard product and $\sigma^* \in \mathbb{R}^{k\times 1}$ is a Rademacher random variable (i.e., $\forall i \in [1,k],\ \Pr(\sigma_i^* = +1) = \Pr(\sigma_i^* = -1) = 0.5$).[1]

3. Create pseudo-performances $Y^* \coloneqq \hat{Y} + \xi^*$, to obtain pseudo-samples for $\hat{Y}$ and $\hat{J}(\pi, k+\delta)$ as $\hat{Y}^* = \Phi(\Phi^\top\Phi)^{-1}\Phi^\top Y^*$ and $\hat{J}(\pi,k+\delta)^* = \phi(k+\delta)(\Phi^\top\Phi)^{-1}\Phi^\top Y^*$.

Steps 2 and 3 can be repeated to re-sample up to $B \leq 2^k$ *similar* sequences of past performance Y^*, from a *single* observed sequence Y of length k, while also preserving the time-series structure. This unreasonable property led Mammen (1993) to coin the term 'wild bootstrap'. For a brief discussion on *why* wild bootstrap works, see Section 4.3.2.

Given these multiple pseudo-samples, we can now obtain an empirical distribution for pseudo $\mathtt{t}$-statistic, $\mathtt{t}^*$. Let the pseudo-sample standard deviation be $\hat{s}^*$, where $\hat{s}^{*2} \coloneqq \phi(k+\delta)(\Phi^\top\Phi)^{-1}\Phi^\top\hat{\Omega}^*\Phi(\Phi^\top\Phi)^{-1}\phi(k+\delta)^\top$, where $\hat{\Omega}^*$ is a diagonal matrix containing the square of the pseudo-errors $\hat{\xi}^* \coloneqq \hat{Y}^* - Y^*$. Let $\mathtt{t}^* \coloneqq (\hat{J}(\pi,k+\delta)^* - \hat{J}(\pi,k+\delta))/\hat{s}^*$. Then an α-percentile $\mathtt{t}_\alpha^*$ of the empirical distribution of $\mathtt{t}^*$ is used to estimate the α-percentile of $\mathtt{t}$'s distribution.

[1] While in machine learning the '*' symbol is often used to denote optimal variables, to be consistent with the bootstrap literature our usage of this symbol denotes pseudo-variables.

Finally, we can define $\mathscr{C}$ to use the wild bootstrap to produce CIs. To ensure this is principled, we leverage a property proven by Djogbenou et al. (2019) and show in the following theorem that the CI for $J(\pi, k+\delta)$ obtained using pseudo-samples from wild bootstrap is *consistent*. For simplicity, we restrict our focus to settings where ϕ is the Fourier basis (see Section 4.11.2.0.2 for more discussion).

Theorem 10 (Consistent Coverage)**.** *Under Assumptions 3 and 4, if the trajectories* $(H_i)_{i=1}^k$ *are independent and if* $\phi(x)$ *is a Fourier basis, then as* $k \to \infty$*,*

$$\Pr\left(J(\pi, k+\delta) \in \left[\hat{J}(\pi, k+\delta) - \hat{s}t^*_{1-\alpha/2},\ \hat{J}(\pi, k+\delta) - \hat{s}t^*_{\alpha/2}\right]\right) \to 1-\alpha.$$

Remark: We considered several factors when choosing the wild bootstrap to create pseudo-samples of $\hat{J}(\pi, k+\delta)$:

(a) Because of the time-series structure, there exists no joint distribution between the deterministic sequence of time indices, X, and the stochastic performance estimates, Y.

(b) Trajectories from only a *single* sequence of $(M_i)_{i=1}^k$ are observed.

(c) Trajectories could have been generated using different β_i's leading to *heteroscedasticity* in the performance estimates $(\hat{J}(\pi, i))_{i=1}^k$.

(d) Different policies π can lead to different distributions of performance estimates, even for the same behavior policy β.

(e) Even for a fixed π and β, performance estimates $(\hat{J}(\pi, i))_{i=1}^k$ can exhibit heteroskedasticty due to inherent stochasticity in $(M_i)_{i=1}^k$ as mentioned in Assumption 3.

These factors make popular approaches like pairs bootstrap, residual bootstrap, and block bootstrap not suitable for our purpose. In contrast, the wild bootstrap can take all these factors into account.

4.6.3 Extended Discussion on Bootstrap

4.6.3.1 Why Not Use Other Bootstrap Methods?

One popular non-parametric technique for bootstrapping in regression is to re-sample, with replacement, (x, y) pairs from the set of observed samples (X, Y) (Carpenter and Bithell, 2000). However, in our setup, X variable corresponds to the (deterministic) time index and thus there exists no joint distribution between the X and the Y variables from where time can be sampled stochastically. Therefore, paired re-sampling will not mimic the underlying data generative process in our setting.

A semi-parametric technique overcomes the above problem by only re-sampling the Y variable as follows. First, a model is fit to the observed data (X, Y) and predictions $\hat{Y}$ are obtained. Then an empirical cumulative distribution function, $\Psi(e)$ of all the errors, $e \coloneqq Y - \hat{Y}$, is obtained. Subsequently, new bootstrapped variables are created as $Y^* \coloneqq \hat{Y} + \xi^*$, where ξ^* is the re-sampled noise from $\Psi(e)$ (Efron and Tibshirani, 1994). However, such a process assumes that noises are homoscedastic, which will be severely violated for our purpose.

Another popular technique for *auto-correlated* data uses the idea of *block re-sampling* (Efron and Tibshirani, 1994). However, this assumes that the underlying process is stationary, and hence is not suitable for our purpose.

4.6.3.2 Why Not Use Standard `t`-test?

Standard `t`-test assumes that the predictions will follow the student-`t` distribution. Such an assumption can be severely violated, especially in the presence of heteroscedasticity, and heavy tailed noises, when the sample size is not sufficiently large. Unfortunately, in our setting the use of multiple behavior policies results in heteroscedasticity, and importance sampling results in a heavy tailed distribution (Thomas et al., 2015c) for counterfactual estimates of past performances.

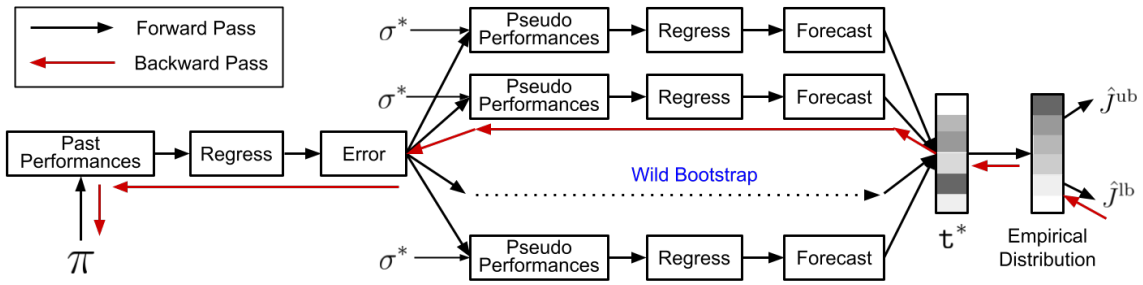


Figure 4.3. To search for a candidate policy π_c, regression is first used to analyze the trend of a given policy's past performances. Wild bootstrap then leverages Rademacher variables σ^* and the errors from regression to create pseudo-performances. Based on these pseudo-performances, an empirical distribution of the pseudo $\mathtt{t}$-statistic, $\mathtt{t}^*$, of the estimate of future performance, is obtained. The candidate policy π_c is found using a differentiation based optimization procedure that maximizes the high-confidence lower bound, $\hat{J}^{\mathrm{lb}}$, computed using the empirical distribution of $\mathtt{t}^*$.

It can also be shown that for a finite sample of size n, the coverage error of CIs obtained using the standard $\mathtt{t}$-statistic is of order $O(n^{-1/2})$ (Wasserman, 2013; Hall, 2013). In comparison, it can be shown using Edgeworth expansions (Hall, 2013) that the coverage error rate of CIs obtained using bootstrap methods typically provide higher-order refinement by providing error rates up to $O(n^{-p/2})$, where $p \in [1, 3]$ (Hall, 1989; DiCiccio and Efron, 1996; Hall, 2013). For more elaborate discussions in the context of wild bootstrap, see the work by Kline and Santos (2012) and by Djogbenou et al. (2019). Also, see the work by Mammen (1993) for detailed empirical comparison of standard t-test against wild-bootstrap.

4.7 Algorithm

Notice that as the CI $[\hat{J}^{\mathrm{lb}}(\pi), \hat{J}^{\mathrm{ub}}(\pi)]$ obtained from $\mathscr{C}$ is based on the wild bootstrap procedure, a gradient based optimization procedure for maximizing the high-confidence lower bound $\hat{J}^{\mathrm{lb}}(\pi)$ would require differentiating through the entire bootstrap process. Figure 4.3 illustrates the high-level steps in this optimization process. More elaborate details and complete algorithms are deferred to Section 4.7.

Further, notice that a smaller amount of data results in greater uncertainty and thus wider CIs. While a tighter CI during candidate policy search can be obtained by combining all the past $\mathcal{D}_{\text{train}}$ to increase the amount of data, each safety test should ideally be independent of all the previous tests, and should therefore use data that has never been used before. While it is possible to do so, using only new data for each safety test would be data-inefficient.

To make our algorithm more data efficient, similar to the approach of Thomas et al. (2019b), we re-use the test data in subsequent tests. As illustrated by the black dashed arrows in Figure 4.2, this modification introduces a subtle source of error because the data used in consecutive tests are not completely independent. However, the practical advantage of this approach in terms of tighter confidence intervals can be significant. Further, as we demonstrate empirically, the error introduced by re-using test data can be negligible in comparison to the error due to the false assumption of stationarity. In Algorithms 2-4,[2] we provide the steps for our method: SPIN. In Algorithm 3, PDIS is shorthand for per-decision importance sampling discussed in Section 4.6. In the following, we discuss certain aspects of SPIN, especially pertaining to the search of a candidate policy π_c.

[2]When $(\alpha/2)B$ or $(1-\alpha/2)B$ is not an integer, then `floor` or `ceil` operation should be used, respectively.

Algorithm 2: Forecast

Input Predicates Φ, Targets Y, Forecast time(s) τ.

$H \leftarrow (\Phi^\top \Phi)^{-1} \Phi^\top$

$\varphi \leftarrow [\phi(\tau_1), ..., \phi(\tau_\delta)]$

$\hat{Y} \leftarrow \Phi H Y$

$\hat{J} \leftarrow \texttt{mean}(\varphi H Y)$

$\hat{\xi} \leftarrow Y - \hat{Y}$

$\hat{\Omega} \leftarrow \texttt{diag}(\hat{\xi}^2)$

$\hat{V} \leftarrow \texttt{mean}(\varphi H \hat{\Omega} H^\top \varphi^\top)$

Return $\hat{J}, \hat{V}, \hat{\xi}$

Algorithm 3: PI: Prediction Interval

Input Data $\mathcal{D}$, Policy π, Safety-violation rate α, Forecast time(s) τ.

$\Phi \leftarrow \emptyset, Y \leftarrow \emptyset$

Create regression variables

for $(k, h) \in \mathcal{D}$ **do**

  $\hat{J}(\pi, k) \leftarrow \text{PDIS}(\pi, h)$

  $\Phi.\texttt{append}(\phi(k))$

  $Y.\texttt{append}(\hat{J}(\pi, k))$

$\hat{J}, \hat{V}, \hat{\xi} \leftarrow \text{Forecast}(\Phi, Y, \tau)$

Wild Bootstrap (in parallel)

$\texttt{t}^* \leftarrow \emptyset, \texttt{t}^{**} \leftarrow \emptyset$

for $i \in [1, ..., B]$ **do**

  $\sigma^* \leftarrow [\pm 1, \pm 1, ..., \pm 1]$

  $\xi^* \leftarrow \hat{\xi} \odot \sigma^*$

  $Y^* \leftarrow \hat{Y} + \xi^*$

  $\hat{J}^*, \hat{V}^*, \_ \leftarrow \text{Forecast}(\Phi, Y^*, \tau)$

  $\texttt{t}^*[i] \leftarrow (\hat{J}^* - \hat{J})/\sqrt{\hat{V}^*}$

Get prediction interval

$\texttt{t}^{**} \leftarrow \texttt{sort}(\texttt{t}^*)$

$\hat{J}^{\text{lb}} \leftarrow \hat{J} - \texttt{t}^{**}[(1 - \alpha/2)B]\sqrt{\hat{V}}$

$\hat{J}^{\text{ub}} \leftarrow \hat{J} - \texttt{t}^{**}[(\alpha/2)B]\sqrt{\hat{V}}$

Return $(\hat{J}^{\text{lb}}, \hat{J}^{\text{ub}})$

Algorithm 4: SPIN: Safe Policy Improvement for Non-stationary settings

Input Safety-violation rate α, Initial safe policy π^{safe}, Entropy-regularizer λ, Batch-size δ.

Initialize $\mathcal{D}_{\text{train}} \leftarrow \emptyset$, $\mathcal{D}_{\text{test}} \leftarrow \emptyset$, $\pi \leftarrow \pi_1^{\text{safe}}$, $k \leftarrow 0$.

while *True* **do**

  # Collect new trajectories using π

  $\mathcal{D} \leftarrow \emptyset$

  for `episode` $\in [1, 2, ..., \delta]$ **do**

    $k \leftarrow k + 1$

    $h \leftarrow ((s^t, a^t, \Pr(a^t|s^t), r^t))_{t=0}^{T}$

    $\mathcal{D} \leftarrow \mathcal{D} \cup (k, h)$

  # Split data

  $\mathcal{D}_1, \mathcal{D}_2 \leftarrow \texttt{split}(\mathcal{D})$

  $\mathcal{D}_{\text{train}} \leftarrow \mathcal{D}_1 \cup \mathcal{D}_{\text{train}}$

  $\mathcal{D}_{\text{test}} \leftarrow \mathcal{D}_2 \cup D_{\text{test}}$

  # Candidate search

  $\tau \leftarrow [k + 1, ..., k + \delta]$

  $\hat{J}^{\text{lb}}(\pi), \_ \leftarrow \text{PI}(\mathcal{D}_{\text{train}}, \pi, \alpha/2, \tau)$

  $\pi_c \leftarrow \text{argmax}_\pi[\hat{J}^{\text{lb}}(\pi) + \lambda \mathcal{H}(\pi, \mathcal{D}_{\text{train}})]$

  # Safety test

  $\hat{J}^{\text{lb}}, \_ \leftarrow \text{PI}(\mathcal{D}_{\text{test}}, \pi_c, \alpha/2, \tau)$

  $\_, \hat{J}^{\text{ub}} \leftarrow \text{PI}(\mathcal{D}_{\text{test}}, \pi^{\text{safe}}, \alpha/2, \tau)$

  if $\hat{J}^{lb} > \hat{J}^{ub}$ **then**

    $\pi \leftarrow \pi_c$

  else

    $\pi \leftarrow \pi^{\text{safe}}$

Mean future performance: In many practical applications, it is often desirable to reduce computational costs by executing a given policy π for multiple episodes before an update, i.e., $\delta > 1$. This raises the question regarding which episode, among the future δ episodes, should a policy π be optimized for before execution? To address this question, in settings where $\delta > 1$, instead of choosing a single future episode's performance for optimization and safety check, we propose using the average performance across all the δ future episodes, i.e., $(1/\delta)\sum_{i=1}^{\delta} J(\pi, k+i)$.

Differentiating the high-confidence lower bound: SPIN proposes a candidate policy π_c by finding a policy π that maximizes the high-confidence lower bound $\hat{J}^{\text{lb}}$ of the future performance (Line 14 in Algorithm 4). To find π_c efficiently, we propose using a differentiable optimization procedure. A visual illustration of the process is given in Figure 4.3.

Derivatives of most of the steps in Algorithms 2 and 3 can be efficiently solved for using modern automatic differentiable programming libraries. Hence, in the following, we restrict the focus of our discussion for describing a *straight-through* gradient estimator for sorting performed in Line 15 in Algorithm 3. Note that sorting is required to obtain the *ordered-statistics* to create an empirical distribution of $\mathtt{t}^*$ such that in Line 16 and 17 of Algorithm 3 the desired percentiles of $\mathtt{t}^*$ can be obtained.

We first introduce some notations. Let $\mathtt{t}^* \in \mathbb{R}^{B\times 1}$ be the unsorted array and $\mathtt{t}^{**} \in \mathbb{R}^{B\times 1}$ be its sorted counterpart. To avoid breaking ties when sorting, we assume that there exists $C_3 > 0$ such that all the values of $\mathtt{t}^*$ are separated by at least C_3. Let $\Gamma \in (0,1)^{B\times B}$ be a *permutation matrix* (i.e., $\forall(i,j),\ \Gamma(i,j) \in (0,1)$, and each row and each column of Γ sums to 1) obtained using any sorting function such that $\mathtt{t}^{**} = \Gamma\mathtt{t}^*$. This operation has a computational graph as shown in Figure 4.4.

Notice that when the values to be sorted are perturbed by a very small amount, the order of the sorted array remains the same (e.g., sorting both the array $[30, 10, 20]$ and its perturbed version results in $[10, 20, 30]$). That is, if $\mathtt{t}^*$ is perturbed by an

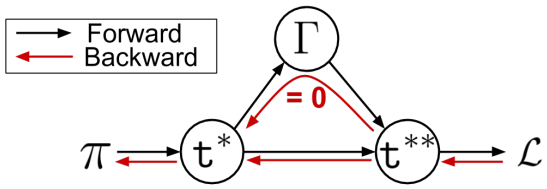


Figure 4.4. Computational graph for obtaining ordered-statistics $\mathtt{t}^{**}$.

$\epsilon \to 0$, then the Γ obtained using the sorting function will not change at all. Therefore, the derivative of Γ with respect to $\mathtt{t}^*$ is 0 and derivative of a desired loss function $\mathcal{L}$ with respect to $\mathtt{t}^*$ is

$$\frac{\partial \mathcal{L}}{\partial \mathtt{t}^*} = \Gamma^\top \frac{\partial \mathcal{L}}{\partial \mathtt{t}^{**}} = \Gamma^{-1} \frac{\partial \mathcal{L}}{\partial \mathtt{t}^{**}},$$

as for any permutation matrix, $\Gamma^\top = \Gamma^{-1}$. Therefore, derivatives are back-propagated through the sorting operation in a straight-through manner by directly performing *un*-sorting.

More advanced techniques for differentiable sorting have been proposed by Cuturi et al. (2019) and Blondel et al. (2020). These methods can be leveraged to further improve our algorithm. We leave these for future work.

Entropy regularization:

As we perform iterative safe policy improvement, the current policy π becomes the behavior policy β for future updates. Therefore, if the current policy π becomes nearly deterministic then the *past performance estimates for a future policy*, which is computed using importance sampling, can suffer from high-variance. To mitigate this issue, we add a λ regularized entropy bonus $\mathcal{H}$ in the optimization objective. This is only done during candidate policy search and hence does not impact the safety check procedure.

Percentile CIs: Notice that each step of the inner optimization process to search for a candidate policy π_c requires computing multiple estimates of the pseudo standard deviation $\hat{s}^*$, one for each sample of $\mathtt{t}^*$, using wild-bootstrap to obtain the CIs. This can be computationally expensive for real-world applications that run on low-powered

devices. As an alternative, we propose using the *percentile* method (Carpenter and Bithell, 2000; Efron and Tibshirani, 1994) during the candidate policy search, which unlike the t-statistic method, does not require computation of $\hat{s}^*$.

While the percentile method can provide a significant computational speed-up, the CIs obtained from it are typically less accurate than those obtained from the method that uses the t-statistic (Carpenter and Bithell, 2000; Efron and Tibshirani, 1994). To get the best of both, (i) as searching for π_c requires an inner optimization routine and accuracy of CIs are less important, we use the percentile method to when computing π_c, and (ii) as the safety test requires no inner optimization and the coverage[3] of CIs are more important to ensure safety, we use the t-statistic method.

To obtain the CIs on $J(\pi, k+\delta)$ using the percentile method, let Ψ denote the empirical cumulative distribution function (CDF) of the pseudo performance forecasts $\hat{J}(\pi, k+\delta)^*$. Then a $(1-\alpha)100\%$ CI, $[\hat{J}^{\text{lb}}, \hat{J}^{\text{ub}}]$, can be estimated as $[\Psi^{-1}(\alpha/2), \Psi^{-1}(1-\alpha/2)]$, where Ψ^{-1} denotes the inverse CDF distribution. That is, if J^* is an array of B pseudo samples of $\hat{J}(\pi, k+\delta)$, and J^{**} contains its sorted ordered-statistics, then a $(1-\alpha)100\%$ CI for $J(\pi, k+\delta)$ is $[J^{**}[(\alpha/2)B], J^{**}[(1-\alpha/2)B]]$. Gradients of the high-confidence lower bound from the percentile method can be computed using the same straight-through gradient estimator discussed earlier.

Complexity analysis (space, time, and sample size): The memory requirement for SPIN is linear in the number of past episodes because it stores all the past data to analyze the performance trends of policies. As both SPIN and Baseline (Thomas et al., 2015a; 2019b) incorporate an inner optimization loop, the computational cost to search for a candidate policy π_c before performing a safety test is similar. Additional computational cost is incurred by our method as it requires computing $(\Phi^\top \Phi)^{-1}$ and $\hat{V}$ in Algorithm 2 for time series analysis. However, note

[3]The *coverage probability* of a CI is the probability with which the target statistic falls within the CI.

that as $\Phi^\top\Phi \in \mathbb{R}^{d\times d}$, where d is the dimension of basis function and $d << k$, the cost of inverting $\Phi^\top\Phi$ is negligible. To avoid the computational cost of computing $\hat{V}$, the percentile method can be used during candidate policy search (as discussed earlier), and the $\mathtt{t}$-statistic method can be used only during the safety test to avoid compromising on safety. An empirical comparison of the sample efficiency of SPIN and Baseline is presented in Figure 4.5.

4.8 Empirical Analysis

In this section, we provide an empirical analysis on two domains inspired by safety-critical real-world problems that exhibit non-stationarity. In the following, we first briefly discuss these domains, and in Figure 4.5 we present a summary of results for eight settings (four for each domain).

4.8.1 Domains

Non-stationary Recommender System (RecoSys): Online recommendation systems for tutorials, movies, advertisements and other products are ubiquitous (Theocharous et al., 2015; 2020). Personalizing for each user is challenging in such settings as interests of an user for different items among the products that can be recommended fluctuate over time. For an example, in the context of online shopping, interests of customers can vary based on seasonality or other unknown factors.

To abstract such settings, in this domain, a synthetic recommender system interacts with a user whose interests in different products change over time. Specifically, the reward for recommending each product varies in a seasonal cycle. Such a scenario is ubiquitous in industrial applications, and updates to an existing system should be made responsibly; if it is not ensured that the new system is better than the existing one, then it might result in a loss of revenue.

For π^{safe}, we set the probability of choosing each item proportional to the reward associated with each item in M_1. This resembles how recommendations would have been set by an expert system initially, such that most relevant recommendation is prioritized while some exploration for other items is also ensured.

Non-stationary Diabetes Treatment: This domain is modeled using an open-source implementation (Xie, 2019) of the U.S. Food and Drug Administration (FDA) approved type-1 Diabetes Mellitus simulator (T1DMS) (Man et al., 2014) for the treatment of type-1 diabetes, where we induced non-stationarity by oscillating the body parameters (e.g., rate of glucose absorption, insulin sensitivity, etc.) between two known configurations available in the simulator. Each step of an episode corresponds to a minute (1440 timesteps–one for each minute in a day) in an *in-silico* patient's body and is governed by a continuous time non-linear ordinary differential equation (ODE) (Man et al., 2014). The goal of the system is to responsibly update the doctor's initial prescription, ensuring that the treatment is only made better. More description on this domain can be found in Chapter 3.6.1.

The diabetes treatment problem is particularly challenging as the performance trend of policies in this domain can violate Assumption 3. Notice that as the parameters that are being oscillated are inputs to a non-linear ODE system, the exact trend of performance for any policy in this NS-MDP is unknown. This more closely reflects a real-world setting where Assumption 3 might not hold, as every policy's performance trend in real-world problems cannot be expected to follow *any* specific trend *exactly*–one can only hope to obtain a coarse approximation of the trend.

4.8.2 Baseline

For a fair comparison, the baseline algorithm, which we call *Baseline*, used for our experiments corresponds to the algorithm presented by Thomas et al. (2015a), which is also a type of Seldonian algorithm (Thomas et al., 2019b). While this algorithm is also

designed to ensure safe policy improvement, it assumes that the domain is stationary. Specifically, during the safety test it ensures that a candidate policy's performance is higher than that of π^{safe}'s by computing CIs on the *average* performance over the past episodes.

4.8.3 Hyper-parameters

In Table 4.3, we provide hyper-parameter (HP) ranges that were used for SPIN and Baseline for both the domains. As obtaining optimal HPs is often not feasible in practical scenarios, algorithms that ensure safety should be robust to how an end-user sets the HPs. Therefore, we set the hyper-parameters within reasonable ranges and report the results in Figure 4.5. These results are aggregated over the *entire* distribution of hyper-parameters, and *not* just for the best hyper-parameter setting. This choice is motivated by the fact that best performances can often be misleading as it only shows what an algorithm *can* achieve and not what it is *likely* to achieve (Jordan et al., 2018; 2020).

For both RecoSys and Diabetes, we ran 1000 HPs per algorithm, per speed, per domain. For RecoSys, we ran 10 trials per HP and 1 trial per HP for diabetes treatment as it involves solving a continuous time ODE and hence is relatively computationally expensive. For experiments, the authors had shared access to a computing cluster, consisting of 50 compute nodes with 28 cores each.

For both the domains, (a) we set π^{safe} to a near-optimal policy for the starting POMDP M_1, representing how a doctor would have set the treatment initially, or how an expert would have set the recommendations, (b) we set the safety level $(1-\alpha)$ to 95%, (c) we modulate the "speed" of non-stationarity, such that higher speeds represent a faster rate of non-stationarity and a speed of zero represents a stationary domain, and (d) we consider the following two algorithms for comparison: (i) **SPIN:** The proposed algorithm that ensures safety while taking into account the impact of

Algorithm	Hyper-parameter	Range
SPIN & Baseline	α	0.05
SPIN & Baseline	δ	$(2, 4, 6, 8)$
SPIN & Baseline	N	$\delta \times$ `uniform`$((2, 5))$
SPIN & Baseline	η	10^{-1}
SPIN & Baseline	λ (RecoSys)	`loguniform`$(5 \times 10^{-5}, 10^{0})$
SPIN & Baseline	λ (Diabetes)	`loguniform`$(10^{-2}, 10^{0})$
SPIN & Baseline	B (candidate policy search)	200
SPIN & Baseline	B (safety test)	500
SPIN	d	`uniform`$((2, 3, 4, 5))$

Table 4.3. Here, N and η represents the number of gradient steps, and the learning rate used while performing Line 14 of Algorithm 4. The dimension of Fourier basis is given by d. Notice that d is set to different values to provide results for different settings where SPIN is *incapable* of modeling the performance trend of policies exactly, and thus Assumption 3 is violated. This resembles practical settings, where it is not possible to exactly know the true underlying trend–it can only be coarsely approximated.

non-stationarity, and (ii) **Baseline:** An algorithm similar to those used in prior works (Thomas et al., 2015a; 2019b; Metevier et al., 2019), which is aimed at ensuring safety but ignores the impact of non-stationarity (see Section 4.8.2 for details).

4.8.4 Results

We present these results for SPIN and Baseline on both the domains in Figure 4.5 (bottom). In Figure 4.5, the plot on the bottom-left corresponds to how often unsafe policies were executed during the process whose learning curves were plotted in Figure 4.5 (top-left). It can be seen that SPIN remains safe almost always. The middle and the right plots in the top row of Figure 4.5 show the normalized performance improvement over the known safe policy π^{safe}. The middle and the right plots in the bottom row of Figure 4.5 show how often unsafe policies were executed.

Note that the performance for any policy π is defined in terms of the expected return. However, for the diabetes domain, we do not know the exact performances of any policy–we can only observe the returns obtained. Therefore, even when an `alg`

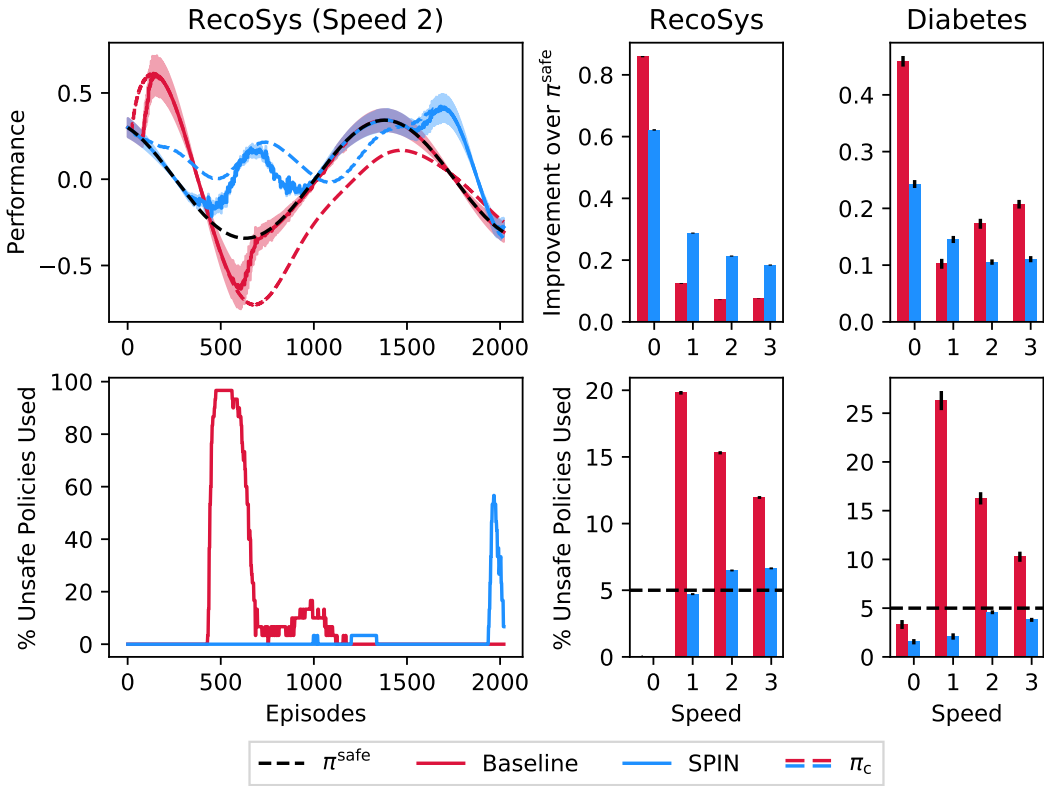


Figure 4.5. (Top-left) An illustration of a typical learning curve. Notice that SPIN updates a policy whenever there is room for a significant improvement. (Middle and Right) As our main goal is to ensure safety, *while being robust to how a user of our algorithm sets the hyper-parameters (HPs)*, we do *not* show results from the best HP. This choice is motivated by the fact that best performances can often be misleading as it only shows what an algorithm *can* achieve and not what it is *likely* to achieve (Jordan et al., 2018; 2020). Therefore, we present the aggregated results averaged over the *entire sweep* of 1000 HPs per algorithm, per speed, per domain. Shaded regions and intervals correspond to the standard error.

selects π^{safe}, it is not possible to get an accurate estimate of its safety violation rate by directly averaging returns observed using a finite number of trials. To make the evaluation process more accurate, we use the following evaluation procedure.

Let a policy π be 'unsafe' when $J(\pi, k+\delta) < J(\pi^{\text{safe}}, k+\delta)$, and let π_c denote policies not equal to π^{safe}, then,

$$\begin{aligned}\Pr(\texttt{alg}(\mathcal{D}) = \text{unsafe}) &= \Pr(\pi_c = \text{unsafe} | \texttt{alg}(\mathcal{D}) = \pi_c) \Pr(\texttt{alg}(\mathcal{D}) = \pi_c) \\ &\quad + \Pr(\pi^{\text{safe}} = \text{unsafe} | \texttt{alg}(\mathcal{D}) = \pi^{\text{safe}}) \Pr(\texttt{alg}(\mathcal{D}) = \pi^{\text{safe}}) \\ &\overset{\textbf{(a)}}{=} \Pr(\pi_c = \text{unsafe} | \texttt{alg}(\mathcal{D}) = \pi_c) \Pr(\texttt{alg}(\mathcal{D}) = \pi_c),\end{aligned}$$

where **(a)** holds because $\Pr(\pi^{\text{safe}} = \text{unsafe}) = 0$. Therefore, to evaluate whether $\texttt{alg}(\mathcal{D})$ is unsafe, for each episode we compare the sample average of returns obtained whenever $\texttt{alg}(\mathcal{D}) \neq \pi^{\text{safe}}$ to the sample average of returns observed using π^{safe}, multiplied by the probability of how often $\texttt{alg}(\mathcal{D}) \neq \pi^{\text{safe}}$.

4.8.5 Discussion on Results

An ideal algorithm should adhere to the safety constraint in (4.1), maximize future performance, and also be robust to hyper-parameters even in the presence of non-stationarity. Therefore, to analyse an algorithm's behavior, we aim to investigate the following three questions:

Q1: *How often does an algorithm violate the safety constraint* $J(\texttt{alg}(\mathcal{D}), k+\delta) \geq J(\pi^{safe}, k+\delta)$*?*

Baseline ensures safety for the stationary setting (speed = 0) but has a severe failure rate otherwise. Perhaps counter-intuitively, the failure rate for Baseline is much *higher* than 5% for *slower* speeds. This can be attributed to the fact that at higher speeds, greater reward fluctuations result in more variance in the performance estimates, causing the CIs within Baseline to be looser, and thereby causing Baseline to have insufficient confidence of policy improvement to make a policy update. Thus,

at higher speeds Baseline becomes safer as it reverts to π^{safe} more often. This calls into question the popular misconception that the stationarity assumption is not severe when changes are slow, as in practice slower changes might be harder for an algorithm to identify, and thus might jeopardize safety. By comparison, even though bootstrap CIs do not have guaranteed coverage when using a finite number of samples (Efron and Tibshirani, 1994), it still allows SPIN to maintain a failure rate near the 5% target.

Q2: *What is the performance gain of an algorithm over the existing known safe policy π^{safe}?*

Notice that any algorithm `alg` can satisfy the safety constraint in (4.1) by *never* updating the existing policy π^{safe}. Such an `alg` is not ideal as it will provide no performance gain over π^{safe}. In the stationary settings, Baseline provides better performance gain than SPIN while maintaining the desired failure rate. However, in the non-stationary setting, the performance gain of SPIN is higher for the recommender system. For diabetes treatment, both the methods provide similar performance gain but only SPIN does so while being safe (see the bottom-right of Figure 4.5). The similar performance of Baseline to SPIN despite being unsafe can be attributed to occasionally deploying better policies than SPIN, but having this improvement negated by deploying policies worse than the safety policy (e.g., see the top-left of Figure 4.5).

Q3: *How robust is SPIN to hyper-parameter choices?*

To analyze the robustness of our method to the choice of relative train-test data set sizes, the objective for the candidate policy search, and to quantify the benefits of the proposed safety test, we provide an ablation study on the RecoSys domain, for all speeds $(0, 1, 2, 3)$ in Table 4.4. All other experimental details are the same, except for (iv), where mean performance, as opposed to the high-confidence lower bound, is optimized during the candidate search. Table 4.4 shows that the safety violation rate of SPIN is robust to such hyper-parameter changes. However, it is worth noting that

		train-test	0	1	2	3	0	1	2	3
(i)	SPIN	75%–25%	.56	.22	.17	.14	0.0	3.6	5.1	5.4
(ii)	SPIN	25%–75%	.48	.29	.21	.19	0.0	4.6	6.5	7.0
(iii)	SPIN	50%–50%	.62	.28	.21	.18	0.0	4.7	6.4	6.6
(iv)	SPIN-mean	50%–50%	.70	.28	.24	.19	0.2	4.9	6.3	7.1
(v)	NS + No safety	100%–0%	.73	.22	.16	.19	9.4	37.6	40.2	38.6
(vi)	Stationary + Safety	50%–50%	.85	.12	.07	.07	0.0	19.8	15.3	11.9

Table 4.4. Ablation study on the RecoSys domain. Top row corresponds to different speeds. (Left) Algorithm and the train-test split ratios. (Middle) Amount of performance improvement over π^{safe}. (Right) Safety violation percentage. Rows (iii) and (vi) correspond to results in Figure 4.5.

too small a test set can make it harder to pass the safety-test, and so performance improvement is small in (i). In contrast, if the proposed safety check procedure for a policy's performance on a non-stationary domain is removed, then the results can be catastrophic, as can be seen in (v).

4.9 Conclusion

In this paper, we took several first steps towards ensuring safe policy improvement for NS-MDPs. We discussed the difficulty of this problem and presented an algorithm for ensuring the safety constraint in (4.1) under the assumption of a smooth performance trend. Further, our experimental results call into question the popular misconception that the stationarity assumption is not severe when changes are slow. In fact, it can be quite the opposite: Slow changes can be more *deceptive* and can make existing algorithms, which do not account for non-stationarity, more susceptible to deploying unsafe policies.

4.10 Limitations and Future Work

The method that we propose is limited to settings where both (a) non-stationarity is governed by an exogenous process (that is, past actions do not impact the underlying non-stationarity), and (b) the performance of every policy changes smoothly over

time and has no discontinuities (abrupt breaks or jumps). Such assumptions need not be applicable to all problems of interests. For example, when there are jumps or breaks in the time series, then the behavior of the proposed method is not ensured to be safe. Further, our method also makes use of importance sampling which requires access to the probabilities of the past actions taken under the behavior policy β. If these probabilities are not available and are instead estimated from data then it may introduce bias and may result in a greater violation of the safety constraint. Finally, all of our experiments were conducted on simulated domains, where the exact nature of non-stationarity may *not* reflect the non-stationarity observed during actual interactions in the physical world. Developing simulators that closely mimic the physical world, without incorporating systematic and racial bias, remains an open problem and is complementary to our research.

There are several exciting directions for future research. We used the ordinary importance sampling procedure to estimate past performances of a policy. However, it suffers from high variance and leveraging better importance sampling procedures (Jiang and Li, 2015; Thomas and Brunskill, 2016) can be directly beneficial to obtain better estimates of past performances. Leveraging time-series models like ARIMA (Chen et al., 2009) and their associated wild-bootstrap methods (Godfrey and Tremayne, 2005; Djogbenou et al., 2015; Friedrich et al., 2020) can be a fruitful direction for extending our algorithm to more general settings that have correlated noises or where the performance trend, both locally and globally, can be better modeled using auto-regressive functions. In Chapter 5 we build upon this direction to develop autoregressive methods for forecasting performances. Further, goodness-of-fit tests (Chen et al., 2003) could be used to search for a time-series model that best fits the application.

4.11 Proofs

4.11.1 Hardness Results

Several works in the past have presented performance bounds for a policy when executed on an approximated stationary MDP (Whitt, 1978; Kakade and Langford, 2002; Kearns and Singh, 2002; Ravindran and Barto, 2004; Pirotta et al., 2013; Achiam et al., 2017). See Section 6 of the work by Bertsekas and Tsitsiklis (1996) for a textbook reference. The technique of our proof for Theorem 9 regarding non-stationary domains is based on these earlier results.

Theorem 11 (Lipschitz smooth performance)**.** *If* $\exists \epsilon_P \in \mathbb{R}$ *and* $\exists \epsilon_R \in \mathbb{R}$ *such that for any* M_k *and* M_{k+1}, $\forall s \in \mathcal{S}, \forall a \in \mathcal{A}$, $\|\mathscr{P}_k(\cdot|s,a) - \mathcal{P}_{k+1}(\cdot|s,a)\|_1 \leq \epsilon_P$ *and* $|\mathbb{E}[\mathscr{R}_k(s,a)] - \mathbb{E}[\mathcal{R}_{k+1}(s,a)]| \leq \epsilon_R$*, then the performance of any policy* π *is Lipschitz smooth over time, with Lipschitz constant* $L := \left(\frac{\gamma R_{max}}{(1-\gamma)^2}\epsilon_P + \frac{1}{1-\gamma}\epsilon_R\right)$*. That is,*

$$\forall k \in \mathbb{N}_{>0}, \forall \delta \in \mathbb{N}_{>0}, \quad |J(\pi, k) - J(\pi, k+\delta)| \leq L\delta. \tag{4.2}$$

Proof. We begin by noting that,

$$|J(\pi, k) - J(\pi, k+\delta)| \leq \sup_{M_k \in \mathcal{M}, M_{k+\delta} \in \mathcal{M}} |J(\pi, M_k) - J(\pi, M_{k+\delta})|. \tag{4.3}$$

We now aim at bounding $|J(\pi, M_k) - J(\pi, M_{k+\delta})|$ in (4.3). Let $R_k(s,a) = \mathbb{E}[\mathscr{R}_k(s,a)]$. Notice that the on-policy distribution and the performance of a policy π in the episode k can be written as,

$$\begin{aligned} d^\pi(s, M_k) =& (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \Pr(S_t = s|\pi, M_k), \\ J(\pi, M_k) =& (1-\gamma)^{-1} \sum_{s \in \mathcal{S}} d^\pi(s, M_k) \sum_{a \in \mathcal{A}} \pi(a|s) R_k(s,a). \end{aligned}$$

We begin the proof by expanding the absolute difference between the two performances as follows:

$$
\begin{aligned}
&|J(\pi, M_k) - J(\pi, M_{k+\delta})| \\
=&|J(\pi, M_k) - J(\pi, M_{k+1}) + J(\pi, M_{k+1}) - ... - J(\pi, M_{k+\delta-1}) + J(\pi, M_{k+\delta-1}) - J(\pi, M_{k+\delta})| \\
\leq& \sum_{i=k}^{k+\delta-1} |J(\pi, M_i) - J(\pi, M_{i+1})|. \qquad (4.4)
\end{aligned}
$$

To simplify further, we introduce a temporary notation $\Delta(s,a) \coloneqq R_i(s,a) - R_{i+1}(s,a)$. Now on expanding each of the consecutive differences in (4.4) and multiplying by $(1-\gamma)$ on both sides:

$$
\begin{aligned}
&(1-\gamma)|J(\pi, M_i) - J(\pi, M_{i+1})| \\
=&\left|\sum_{s\in\mathcal{S}} d^\pi(s, M_i) \sum_{a\in\mathcal{A}} \pi(a|s) R_i(s,a) - \sum_{s\in\mathcal{S}} d^\pi(s, M_{i+1}) \sum_{a\in\mathcal{A}} \pi(a|s) R_{i+1}(s,a)\right| \\
=&\left|\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \pi(a|s)\Big(d^\pi(s, M_i) R_i(s,a) - d^\pi(s, M_{i+1}) R_{i+1}(s,a)\Big)\right| \\
=&\left|\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \pi(a|s)\Big(d^\pi(s, M_i)(R_{i+1}(s,a) + \Delta(s,a)) - d^\pi(s, M_{i+1}) R_{i+1}(s,a)\Big)\right| \\
=&\left|\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \pi(a|s)\Big(d^\pi(s, M_i) - d^\pi(s, M_{i+1})\Big) R_{i+1}(s,a) + \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \pi(a|s) d^\pi(s, M_i)\Delta(s,a)\right|. \qquad (4.5)
\end{aligned}
$$

In the following, we bound the terms in (4.5) using the following three steps, (a) use the Cauchy Schwartz inequality and bound each possible negative term with its absolute value, (b) bound each reward $R_{i+1}(s,a)$ using $R_{\max}$ and use the Lipschitz smoothness assumption to bound each $\Delta(s,a)$ using ϵ_R, and (c) equate sum of probabilities to one. Formally,

$$
\begin{aligned}
&(1-\gamma)|J(\pi, M_i) - J(\pi, M_{i+1})| \\
&\overset{(a)}{\leq} \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \pi(a|s)\,|d^\pi(s, M_i) - d^\pi(s, M_{i+1})|\,|R_{i+1}(s,a)| + \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \pi(a|s) d^\pi(s, M_i)\,|\Delta(s,a)| \\
&\overset{(b)}{\leq} R_{\max} \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \pi(a|s)\,|d^\pi(s, M_i) - d^\pi(s, M_{i+1})| + \epsilon_R \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \pi(a|s) d^\pi(s, M_i) \\
&\overset{(c)}{=} R_{\max} \sum_{s\in\mathcal{S}} |d^\pi(s, M_i) - d^\pi(s, M_{i+1})| + \epsilon_R. \qquad (4.6)
\end{aligned}
$$

To simplify (4.6) further, we make use of the following property,

Property 2 (Achiam et al. (2017))**.** *Let $P_i^\pi \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ be the transition matrix (s' in rows and s in columns) resulting due to π and P_i, i.e., $\forall t,\ P_i^\pi(s', s) \coloneqq \Pr(S_{t+1} = s' | S_t = s, \pi, M_i)$, and let $d^\pi(\cdot, M_i) \in \mathbb{R}^{|\mathcal{S}|}$ denote the vector of probabilities for each state, then*[4]

$$
\sum_{s\in\mathcal{S}} |d^\pi(s, M_i) - d^\pi(s, M_{i+1})| \leq \gamma(1-\gamma)^{-1} \left\| (P_i^\pi - P_{i+1}^\pi) d^\pi(\cdot, M_i) \right\|_1 .
$$

Using Property 2,

[4]Note that the original result by Achiam et al. (2017) bounds the change in distribution between two different policies under the same dynamics. Here, we have modified the property for our case, where the policy is fixed but the dynamics are different.

$$
\begin{aligned}
&\sum_{s\in\mathcal{S}} |d^\pi(s, M_i) - d^\pi(s, M_{i+1})| \\
\overset{(d)}{\leq}&\gamma(1-\gamma)^{-1} \sum_{s'\in\mathcal{S}} \left| \sum_{s\in\mathcal{S}} \left(P_i^\pi(s', s) - P_{i+1}^\pi(s', s)\right) d^\pi(s, M_i) \right| \\
\leq&\gamma(1-\gamma)^{-1} \sum_{s'\in\mathcal{S}} \sum_{s\in\mathcal{S}} \left|P_i^\pi(s', s) - P_{i+1}^\pi(s', s)\right| d^\pi(s, M_i) \\
=&\gamma(1-\gamma)^{-1} \sum_{s'\in\mathcal{S}} \sum_{s\in\mathcal{S}} \left| \sum_{a\in\mathcal{A}} \pi(a|s) \Big(\Pr(s'|s, a, M_i) - \Pr(s'|s, a, M_{i+1}) \Big) \right| d^\pi(s, M_i) \\
\leq&\gamma(1-\gamma)^{-1} \sum_{s'\in\mathcal{S}} \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \pi(a|s) \left|\Pr(s'|s, a, M_i) - \Pr(s'|s, a, M_{i+1})\right| d^\pi(s, M_i) \\
=&\gamma(1-\gamma)^{-1} \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \pi(a|s) d^\pi(s, M_i) \sum_{s'\in\mathcal{S}} |\Pr(s'|s, a, M_i) - \Pr(s'|s, a, M_{i+1})| \\
\overset{(e)}{\leq}&\gamma(1-\gamma)^{-1} \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \pi(a|s) d^\pi(s, M_i) \epsilon_P \\
=&\gamma(1-\gamma)^{-1} \epsilon_P,
\end{aligned} \tag{4.7}
$$

where (d) follows from expanding the L1 norm of a matrix-vector product, and (e) follows from using the Lipschitz smoothness to bound the difference between successive transition matrices. Combining (4.6) and (4.7),

$$
\begin{aligned}
|J(\pi, M_i) - J(\pi, M_{i+1})| \leq&(1-\gamma)^{-1} \left(R_{\max}\gamma(1-\gamma)^{-1}\epsilon_P + \epsilon_R\right). \\
=&\frac{\gamma R_{\max}}{(1-\gamma)^2}\epsilon_P + \frac{1}{1-\gamma}\epsilon_R.
\end{aligned} \tag{4.8}
$$

Finally, combining (4.4) and (4.8),

$$
\begin{aligned}
|J(\pi, M_i) - J(\pi, M_{i+\delta})| \leq& \sum_{i=k}^{k+\delta-1} \left(\frac{\gamma R_{\max}}{(1-\gamma)^2}\epsilon_P + \frac{1}{1-\gamma}\epsilon_R \right) \\
=&\delta \left(\frac{\gamma R_{\max}}{(1-\gamma)^2}\epsilon_P + \frac{1}{1-\gamma}\epsilon_R \right).
\end{aligned}
$$

Tightness of The Bound: In this paragraph, we present a non-stationary decision process where (4.2) holds with exact equality, illustrating that the bound given by Theorem 9 is tight.

Consider the domain given in Figure 4.6. Let $\gamma = 0$ and let $\mathcal{A} = \{a\}$ such that the size of action set $|\mathcal{A}| = 1$. Let the state set $\mathcal{S} = (s_1, s_2)$ and let the initial state for an episode always be state s_1. Let rewards be in the range $[-1, +1]$ such that $R_{\max} = 1$.

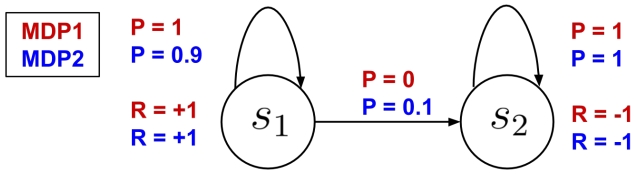


Figure 4.6. Example NS-MDP.

Notice that for NS-MDP in Figure 4.6, $\epsilon_R = |\mathbb{E}[\mathscr{R}_1(s_1, a)] - \mathbb{E}[\mathscr{R}_2(s_1, a)]| = 0.2$ as

$$R_1(s_1, a) = \mathbb{E}[\mathscr{R}_1(s_1, a)] = 1(+1) + 0(-1) = 1,$$
$$R_2(s_1, a) = \mathbb{E}[\mathscr{R}_2(s_1, a)] = 0.9(+1) + 0.1(-1) = 0.8.$$

Similarly,

$$\epsilon_P = |\mathscr{P}_1(s_1|s_1, a) - \mathscr{P}_2(s_1|s_1, a)| + |\mathscr{P}_1(s_2|s_1, a) - \mathscr{P}_2(s_2|s_1, a)| = 0.2.$$

Therefore, substituting the values $\gamma = 0, R_{\max} = 1, \epsilon_P = \epsilon_R = 0.2$, and $\delta = 1$ in (4.2), we get

$$|J(\pi, M_1) - J(\pi, M_2)| \leq 0.2. \tag{4.9}$$

Now to illustrate that the bound is tight, we compute the true difference in performances of a policy π for the domains given in Figure 4.6, i.e., the LHS of (4.9).

Notice that

$$J(\pi, M_1) = (1-\gamma)^{-1} \sum_{s \in \mathcal{S}} d^{\pi}(s, M_1) \sum_{a \in \mathcal{A}} \pi(a|s) R_1(s,a) \overset{\textbf{(a)}}{=} R_1(s_1, a) = 1,$$

where **(a)** follows because (i) $\gamma = 0$, (ii) as there is only a single action, $\pi(a|s) = 1$, and (iii) since s_1 is the starting state and $\gamma = 0$, therefore, $d^{\pi}(s_1, M_1) = 1$ and $d^{\pi}(s_2, M_1) = 0$. Similarly, $J(\pi, M_2) = R_2(s_1, a) = 0.8$. Therefore, $|J(\pi, M_1) - J(\pi, M_2)| = 0.2$, which is exactly equal to the value of bound in (4.9).

□

4.11.2 Uncertainty Estimation

Our Theorem 10 makes use of a property proven by Djogbenou et al. (2019). This property by Djogbenou et al. (2019) was established for inference about the *parameters* of a regression model. We leverage this property to obtain confidence intervals for *predictions* of future performance. In the following section, we first review their results and then in the section thereafter we present the proof of Theorem 10.

4.11.2.0.1 Preliminary Before moving forward, we first revisit all the necessary notations and review the result by Djogbenou et al. (2019). For a regression problem, let $Y \in \mathbb{R}^{k \times 1}$ be the stochastic observations, let $\Phi \in \mathbb{R}^{k \times d}$ be the deterministic predicates, and let $w \in \mathbb{R}^{d \times 1}$ be the regression parameters. Let $\xi \in \mathbb{R}^{k \times 1}$ be a vector of k independent noises. The linear system of equations for regression is then given by,

$$Y = \Phi w + \xi. \tag{4.10}$$

The least-squares estimate $\hat{w}$ of w is given by $\hat{w} := (\Phi^{\top}\Phi)^{-1}\Phi^{\top}Y$ and the estimate $\hat{Y} := \Phi\hat{w}$. Subsequently, the covariance of the estimate $\hat{w}$ can be computed as follows.

$$
\begin{aligned}
V \coloneqq \mathbb{V}(\hat{w}) =& \mathbb{E}\left[(\hat{w} - \mathbb{E}[\hat{w}])\,(\hat{w} - \mathbb{E}[\hat{w}])^\top\right] \\
=& \mathbb{E}\left[\left((\Phi^\top\Phi)^{-1}\Phi^\top(Y - \mathbb{E}[Y])\right)\left((\Phi^\top\Phi)^{-1}\Phi^\top(Y - \mathbb{E}[Y])\right)^\top\right] \\
=& \mathbb{E}\left[\left((\Phi^\top\Phi)^{-1}\Phi^\top\xi\right)\left((\Phi^\top\Phi)^{-1}\Phi^\top\xi\right)^\top\right] \\
=& \mathbb{E}\left[(\Phi^\top\Phi)^{-1}\Phi^\top\xi\xi^\top\Phi(\Phi^\top\Phi)^{-1}\right] \\
\overset{\textbf{(a)}}{=}& (\Phi^\top\Phi)^{-1}\Phi^\top\mathbb{E}\left[\xi\xi^\top\right]\Phi(\Phi^\top\Phi)^{-1} \\
=& (\Phi^\top\Phi)^{-1}\Phi^\top\Omega\Phi(\Phi^\top\Phi)^{-1},
\end{aligned}
\tag{4.11}
$$

where **(a)** follows from the fact that Φ is deterministic, and Ω is the covariance matrix of the mean-zero and heteroscedastic noises ξ. Notice that as the noises are independent, the off-diagonal terms in Ω are zero. However, since the true Ω is not known, it can be estimated using $\hat{\Omega}$ which contains the squared errors from the OLS estimate (MacKinnon, 2012). That is, letting $\hat{\xi} \coloneqq \hat{Y} - Y$, we have that $\hat{\Omega}$ is a diagonal matrix with $\hat{\xi}^2$ in the diagonal. Let such an estimator of $\mathbb{V}(w)$ be,

$$
\hat{V} \coloneqq (\Phi^\top\Phi)^{-1}\Phi^\top\hat{\Omega}\Phi(\Phi^\top\Phi)^{-1}. \tag{4.12}
$$

Let $b^\top w$ be a desired null hypothesis with $b^\top b = 1$. Let $\mathtt{t}_b$, the $\mathtt{t}$-statistic for testing this hypothesis, and its pseudo-sample $\mathtt{t}_b^*$ obtained using the wild bootstrap procedure with Rademacher variables σ^* be (see Section 4.6 in the main body for exact steps),

$$
\mathtt{t}_b = \frac{b^\top(\hat{w} - w)}{\sqrt{b^\top\hat{V}b}}, \qquad \mathtt{t}_b^* \coloneqq \frac{b^\top(\hat{w}^* - \hat{w})}{\sqrt{b^\top\hat{V}^*b}}. \tag{4.13}
$$

Note that in (4.13), the subscript of b is *not* related to percentile of the previously defined $\mathtt{t}$-statistic: $\mathtt{t}$. That is, $\mathtt{t}_b$ and $\mathtt{t}_b^*$ are new variables.

Now we state the result we use from the work by Djogbenou et al. (2019). This result requires two main assumptions. Our presentations of these assumptions are

slightly different from the exact statements given by Djogbenou et al. (2019). The differences are (a) we make the assumptions stronger than what is required for their results to hold, and (b) we ignore a third assumption that is related to cluster sizes, as our setting is a special case where the cluster size is equal to 1. We call these assumptions *requirements* to distinguish them from our assumptions.

Requirement 1 (Independence)**.** *$\forall i \in [1, k]$, the noise terms ξ_i are mean-zero, bounded, and independent random variables.*

Requirement 2 (Positive Definite)**.** *$(\Phi^\top \Phi)^{-1}$ is positive-definite and $\exists C_2 > 0$ such that $\|\Phi\|_\infty < C_2$.*

Lemma 1 (Theorem 3.2 Djogbenou et al. (2019))**.** *Under Requirements 1 and 2, if $\mathbb{E}[\sigma^{*3}] < \infty$ and if the true value of w is given by (4.10), then as $k \to \infty$,*

$$\Pr\left(\sup_{x \in \mathbb{R}} |\Pr(t_b^* < x) - \Pr(t_b < x)| > \alpha\right) \to 0.$$

4.11.2.0.2 Proof of Coverage Error First, we recall the notations established in the main body, which are required for the proof. Using similar steps to those in (4.11), it can be seen that the variance V_f of the estimator $\hat{J}(\pi, k+\delta) \coloneqq \phi(k+\delta)\hat{w}$ of future performance is

$$V_f = \phi(k+\delta)(\Phi^\top \Phi)^{-1} \Phi^\top \Omega \Phi (\Phi^\top \Phi)^{-1} \phi(k+\delta)^\top.$$

Similar to before, let an estimate $\hat{V}_f$ of V_f be defined as,

$$\hat{V}_f = \phi(k+\delta)(\Phi^\top \Phi)^{-1} \Phi^\top \hat{\Omega} \Phi (\Phi^\top \Phi)^{-1} \phi(k+\delta)^\top, \tag{4.14}$$

where $\hat{\Omega}$ is the same as in (4.12). Recall from Section 4.6 that the sample standard deviation of $\phi(k+\delta)\hat{w}$ is $\hat{s} = \sqrt{\hat{V}_f}$ and the pseudo standard deviation is $\hat{s}^* \coloneqq \sqrt{\hat{V}_f^*}$,

where the pseudo variables are created using the wild bootstrap procedure outlined in Section 4.6. Similarly, recall that the $\mathtt{t}$-statistic and the pseudo $\mathtt{t}$-statistic for estimating future performance are given by

$$\mathtt{t} := \frac{\hat{J}(\pi, k+\delta) - J(\pi, k+\delta)}{\hat{s}}, \qquad \mathtt{t}^* := \frac{\hat{J}(\pi, k+\delta)^* - \hat{J}(\pi, k+\delta)}{\hat{s}^*}.$$

For the purpose of Theorem 10, we use a Fourier basis of order d, which is given by (Bloomfield, 2004):

$$\phi(x) := \left(\frac{\sin(2\pi n x)}{C} \middle| n \in [1, d] \right) \cup \left(\frac{\cos(2\pi n x)}{C} \middle| n \in [1, d] \right) \cup \left(\frac{1}{C} \right), \quad (4.15)$$

where $C := \sqrt{d+1}$.

Theorem 12 (Consistent Coverage)**.** *Under Assumptions 3 and 4, if the set of trajectories $(H_i)_{i=1}^k$ are independent and if $\phi(x)$ is a Fourier basis of order d, then as $k \to \infty$,*

$$\Pr\left(J(\pi, k+\delta) \in \left[\hat{J}(\pi, k+\delta) - \hat{s}\mathtt{t}^*_{1-\alpha/2},\ \hat{J}(\pi, k+\delta) - \hat{s}\mathtt{t}^*_{\alpha/2} \right] \right) \to 1 - \alpha. \quad (4.16)$$

Proof. For the purpose of this proof, we will make use Lemma 1. Therefore, we first discuss how our method satisfies the requirements for Lemma 1.

To satisfy Requirement 1, recall that in the proposed method the estimates $(\hat{J}(\pi, i))_{i=1}^k$ of past performances are obtained using counter-factual reasoning. Therefore, satisfying Requirement 1 in our method requires consideration of two sources of noise: (a) the noise resulting from the inherent stochasticity in the non-stationary POMDP sequence, as given in Assumption 3, and (b) the other noise resulting due to our use of importance sampling to estimate past performances $(J(\pi, i))_{i=1}^k$, which are subsequently used to obtain the forecast for $J(\pi, k+\delta)$.

Notice that the noises $(\xi_i)_{i=1}^k$ inherent to the non-stationary POMDP are both mean-zero and independent because of Assumption 3. Further, as importance sampling is unbiased and uses independent draws of trajectories $(H_i)_{i=1}^k$, the additional noises in the estimates $(\hat{J}(\pi, i))_{i=1}^k$ are also mean-zero and independent. The boundedness condition of each ξ_i also holds as (a) all episodic returns are bounded, which is because every reward is bounded between $[-R_{\max}, R_{\max}]$ and $\gamma < 1$, and (b) following Assumption 4, the denominators of importance sampling ratios are lower bounded by C. Therefore, importance weighted returns are upper bounded by a finite constant. This makes the noise from importance sampling estimates also bounded. Hence, all the noises in our performance estimates are independent, bounded, and mean zero.

To satisfy Requirement 2, note that as $\Phi^\top \Phi$ is an inner product matrix, it must be positive semi-definite. Further, as the Fourier basis creates linearly independent features, when $k > d$ (i.e., it has more samples than number of parameters) the matrix will have full column-rank. Combining these two points it can be seen that $\Phi^\top \Phi$ is a positive-definite matrix and as the eigenvalues of $(\Phi^\top \Phi)^{-1}$ are just the reciprocals of the eigenvalues of $\Phi^\top \Phi$, the matrix $(\Phi^\top \Phi)^{-1}$ is also positive-definite. The second half of Requirement 2 is trivially satisfied as all the values of $\phi(x)$ are in $[-1/C, 1/C]$.

Finally, note that when $\phi : \mathbb{N} \to \mathbb{R}^{1\times d}$ is a Fourier basis then $\forall x \in \mathbb{R},\ \phi(x)\phi(x)^\top = 1$. To see why, notice from (4.15) that

$$\begin{aligned}\phi(x)\phi(x)^\top &= \sum_{n=1}^{d} \left(\frac{\sin(2\pi n x)}{C}\right)^2 + \sum_{n=1}^{d} \left(\frac{\cos(2\pi n x)}{C}\right)^2 + \left(\frac{1}{C}\right)^2 \\ &= \frac{\sum_{n=1}^{d} \left(\sin^2(2\pi n x) + \cos^2(2\pi n x)\right) + 1}{C^2} \overset{\textbf{(a)}}{=} \frac{d+1}{C^2} = 1, \end{aligned} \tag{4.17}$$

where **(a)** follows from the trignometric inequality that $\forall x \in \mathbb{R}\ \sin^2(x) + \cos^2(x) = 1$.

Now we are ready for the complete proof. For brevity, we define

$$\mathcal{C} := \left[\hat{J}(\pi, k+\delta) - \hat{\mathtt{st}}^*_{1-\alpha/2},\ \hat{J}(\pi, k+\delta) - \hat{\mathtt{st}}^*_{\alpha/2}\right],$$

$J \coloneqq J(\pi, k+\delta)$, and $\hat{J} \coloneqq \hat{J}(\pi, k+\delta)$, and expand the LHS of (4.16):

$$\begin{aligned}
\Pr\left(J \in \mathcal{C}\right) &= \Pr\left(\hat{J} - \hat{s}\mathtt{t}^*_{1-\alpha/2} \leq J \leq \hat{J} - \hat{s}\mathtt{t}^*_{\alpha/2}\right) \\
&= \Pr\left(-\hat{s}\mathtt{t}^*_{1-\alpha/2} \leq J - \hat{J} \leq -\hat{s}\mathtt{t}^*_{\alpha/2}\right) \\
&= \Pr\left(\hat{s}\mathtt{t}^*_{1-\alpha/2} \geq \hat{J} - J \geq \hat{s}\mathtt{t}^*_{\alpha/2}\right) \\
&= \Pr\left(\mathtt{t}^*_{1-\alpha/2} \geq \frac{\hat{J} - J}{\hat{s}} \geq \mathtt{t}^*_{\alpha/2}\right). \\
&= \Pr\left(\mathtt{t}^*_{1-\alpha/2} \geq \mathtt{t} \geq \mathtt{t}^*_{\alpha/2}\right) \\
&= \Pr\left(\mathtt{t} \leq \mathtt{t}^*_{1-\alpha/2}\right) - \Pr\left(\mathtt{t} \leq \mathtt{t}^*_{\alpha/2}\right). \qquad (4.18)
\end{aligned}$$

To simplify (4.18), let $b = \phi(k+\delta)^\top$. Under this instantiation of b, the null hypothesis $b^\top w$ in Section 4.11.2.0.1 for our setting corresponds to $\phi(k+\delta)w$, which is the true future performance under Assumption 3. Further, for this instantiation of b, note from (4.17) that $b^\top b = 1$. Now, it can be seen from (4.14) that $\hat{V}_f = b^\top \hat{V} b$, and $\hat{V}^*_f = b^\top \hat{V}^* b$. Thus, $\mathtt{t} = \mathtt{t}_b$ and $\mathtt{t}^* = \mathtt{t}^*_b$. Finally, note that as σ^* corresponds to the Rademacher random variable, $\mathbb{E}[\sigma^{*3}] = 0$.

Therefore, leveraging Lemma 1, in the limit, for any x, we can substitute $\Pr(\mathtt{t} < x)$ with $\Pr(\mathtt{t}^* < x)$ in (4.18). This substitution yields

$$\begin{aligned}
\Pr\left(J \in \mathcal{C}\right) &\to \Pr\left(\mathtt{t}^* \leq \mathtt{t}^*_{1-\alpha/2}\right) - \Pr\left(\mathtt{t}^* \leq \mathtt{t}^*_{\alpha/2}\right) \\
&= (1-\alpha/2) - (\alpha/2) \\
&= 1 - \alpha. \qquad \square
\end{aligned}$$

Notice that using the Fourier basis, we were able to satisfy the condition that $b^\top b = 1$ directly. This allowed us to leverage Lemma 1 without much modification. However, as noted by Djogbenou et al. (2019), the constraint on $b^\top b$ is not necessary and was used to simplify the proof.

CHAPTER 5

ACTION-DEPENDENT NON-STATIONARITY

Methods for sequential decision making are often built upon a foundational assumption that the underlying decision process is stationary (Sutton and Barto, 2018a). While this assumption was a cornerstone when laying the theoretical foundations of the field, it is seldom true for real-world problems. Using the taxonomy proposed by Khetarpal et al. (2020), such non-stationarity can be broadly classified as (a) *Passive:* where the changes to the system are induced only by external (exogenous) factors, (b) *Active:* where the changes result due to the agent's past interactions with the system, or (c) *Hybrid:* where both passive and active changes can occur together.

When the transition dynamics and the reward function of the decision process are changing, even the fundamental problem of *policy evaluation* is challenging. If changes can be abrupt and arbitrary, then there is not much hope of estimating what a policy's future performance will be. However, when the underlying changes are structured, can their affect on a policy's performance be extracted without requiring estimation of the true underlying non-stationary environment? This raises the main question of interest:

How do we provide a unified procedure for (off) policy evaluation amidst active, passive, or hybrid non-stationarity, when there is a structure in the underlying changes?

Motivation: Both active and passive non-stationary are ubiquitous in real-world problems. For example, prior work has proposed using reinforcement learning (RL)

for automated healthcare for patients diagnosed with type-1 diabetes (Bastani, 2014), sepsis (Saria, 2018), HIV (Ernst et al., 2006), etc. These methods consider optimizing treatments either individually for each patient, or at a population level. (a) When considering patients *individually*, often each *day* of interaction is considered to be an *independent* episode (Bastani, 2014; Thomas et al., 2019b). These methods manifest stationarity by assuming that the interaction pattern of the patient is the same each day. However, notice that independence across episodes is clearly violated as the state of the patient at the start of each day is *dependent* on the decisions taken at the end of the previous day. This results in active non-stationarity. Additionally, the physiology and behavior of a patient varies with age, and therefore age is an important feature of the state that changes across episodes, thereby also resulting in passive non-stationarity. (b) At the *population* level, interactions with each *patient* are considered to be an independent episode. When considering data collected over extended periods, not only do the healthcare facilities change over time, but public health also continuously evolves based on the treatments made available in the past, thereby resulting in hybrid non-stationarity.

Similar to automated healthcare, other applications like online education, product recommendations, and in fact almost all human-computer interaction systems need to not only account for the continually drifting behavior of the user demographic but also how the preferences of users may change due to interactions with the system (Theocharous et al., 2020). Even social media platforms need to account for the partisan bias of their users that changes due to both external political developments and increased self-validation resulting from previous posts/ads suggested by the recommender system itself (Cinelli et al., 2021; Gillani et al., 2018). Similarly, motors in a robot suffer wear and tear over time not only based on natural corrosion but also on how vigorous the past actions were. These present a range of applications that are subject to hybrid non-stationarity.

However, conventional off-policy evaluation methods (Precup, 2000; Jiang and Li, 2015; Xie et al., 2019) predominantly focus on the stationary setting. These methods assume availability of either (a) *resetting assumption* to sample multiple sequences of interactions from a stationary environment with a fixed starting state distribution (i.e., episodic setting), or (b) *ergodicity assumption* such that interactions can be sampled from a steady-state/stationary distribution (i.e., continuing setting). For the problems of our interest, methods based on these assumptions may not be viable. For e.g., in automated healthcare, we have a single long history for the evolution of public health, which is neither in a steady state distribution nor can we reset and go back in time to sample another history of interactions.

Contributions: In this work, we focus on the fundamental challenge of policy evaluation amidst structured changes due to active, passive, or hybrid non-stationarity. To the best of our knowledge, our work presents the first steps towards addressing this in both the off-policy and the on-policy settings. We provide:

- A unified formulation of different forms of nonstationarity, and discuss the assumptions necessary for tractability.
- A procedure for policy evaluation that avoids the complexities of directly modeling the nonstationary environment.
- A variance reduction procedure for the non-stationary setting that can help mitigate the high-variance resulting from the use of off-policy data.

We call the proposed method OPEN: off-policy evaluation for non-stationary domains. OPEN primarily relies upon two key insights: **(a)** For active/hybrid non-stationarity, as the underlying changes may dependend on past interactions, the structure in the changes observed when executing the data collection policy can be different than if one were to execute the evaluation policy. To address this challenge, OPEN makes uses counterfactual reasoning twice and permits reduction of this off-

policy evaluation problem to an auto-regression based forecasting problem. **(b)** Despite reduction to a more familiar auto-regression problem, in this setting naive least-squares based estimates of parameters for auto-regression suffers from high variance and can even be asymptotically biased. Finally, to address this challenge, OPEN uses a novel importance-weighted instrument-variable (auto-)regression technique to obtain asymptotically consistent and lower variance parameter estimates.

Importantly, instead of assuming that all changes are due to external factors, our proposed methods can account for the changes in the environment caused by the agent's past decisions. As we discuss later in Figure 5.2, this can not only help in the identification of policies that may be actively causing harm or damage, but may also enable *control* of non-stationary processes.

5.1 Notation

We build upon the formulation used by past work (Xie et al., 2020a; Chandak et al., 2020b) and consider the setting wherein an agent interacts with a lifelong sequence of partially observable Markov decision processes (POMDPs), $(M_i)_{i=1}^{\infty}$. However, unlike prior problem formulations, we account for active and hybrid non-stationarity by considering a Markov structure where the POMDP M_{i+1} is dependent *both* on the previous POMDP M_i and the decisions made by the agent while interacting with M_i. For simplicity of presentation, we will often ignore the dependency of M_{i+1} on M_{i-k} for $k > 0$, although our results can be extended to settings with $k > 0$.

Notation: Let $\mathcal{M}$ be a finite set of POMDPs. Each POMDP $M_i \in \mathcal{M}$ is a tuple $(\mathcal{O}, \mathcal{S}, \mathcal{A}, \Omega_i, P_i, R_i, \mu_i)$, where $\mathcal{O}$ is the set of observations, $\mathcal{S}$ is the set of states, and $\mathcal{A}$ is the set of actions, which are the same for all the POMDPs in $\mathcal{M}$. For simplicity of notation, we assume $\mathcal{M}, \mathcal{S}, \mathcal{O}, \mathcal{A}$ are finite sets, although our results can be extended to settings where these sets are infinite or continuous. Let $\Omega_i : \mathcal{S} \times \mathcal{O} \to [0, 1]$ be the observation function, $P_i : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ be the transition function, $\mu_i : \mathcal{S} \to [0, 1]$

be the starting state distribution, and $R_i : \mathcal{S} \times \mathcal{A} \to [-R_{\max}, R_{\max}]$ be the reward function with $0 \leq R_{\max} < \infty$.

Let $\pi : \mathcal{O} \times \mathcal{A} \to [0, 1]$ be any policy and Π be the set of all policies. Let $H_i \coloneqq (S_i^t, O_i^t, A_i^t, R_i^t)_{i=1}^T$ be a sequence of at most T interactions in M_i, where O_i^t, A_i^t, R_i^t are the random variables corresponding to the observation, action, and reward at the step t. Let $G_i \coloneqq \sum_{t=1}^T R_i^t$ be an observed return and $J_i(\pi) \coloneqq \mathbb{E}_\pi[G_i|M_i]$ be the performance of π on M_i. Let $\mathcal{H}$ be the set of possible interaction sequences, and finally let $\mathcal{T} : \mathcal{M} \times \mathcal{H} \times \mathcal{M} \to [0, 1]$ be the 'meta-transition' function that governs the non-stationarity in the POMDPs. That is, $\mathcal{T}(m, h, m') = \Pr(M_{i+1}{=}m'|M_i{=}m, H_i{=}h)$. We provide an illustration of the control process in Figure 5.1.

5.2 Problem Statement:

We look at the fundamental problem of evaluating the performance of a policy π in the presence of non-stationarity. Let $(H_i)_{i=1}^n$ be the data collected in the past by interacting using policies $(\beta_i)_{i=1}^n$. Let D_n be the dataset consisting of $(H_i)_{i=1}^n$ and the probabilities of the actions taken by $(\beta_i)_{i=1}^n$. With a slight abuse of notation we define Dataset D_n using trajectory variable $H_i \coloneqq (O_i^t, A_i^t, R_i^t)_{i=1}^T$ as in practice we do not have access to the true underlying state variable. Given D_n, we aim to evaluate the performance of π if it is deployed for the *next* L future episodes (each a different POMDP), that is

$$\mathscr{J}(\pi) \coloneqq \mathbb{E}_\pi \left[\sum_{k=n+1}^{n+L} J_k(\pi) \middle| (H_i)_{i=1}^n \right]. \tag{5.1}$$

We call it the *on-policy* setting if $\forall i, \beta_i = \pi$, and the *off-policy* setting otherwise. Notice that even in the on-policy setting, naively aggregating observed performances from $(H_i)_{i=1}^n$ may not be indicative of $\mathscr{J}(\pi)$ as M_k for $k > n$ may be different than $M \in (M_i)_{i=1}^n$ due to non-stationarity.

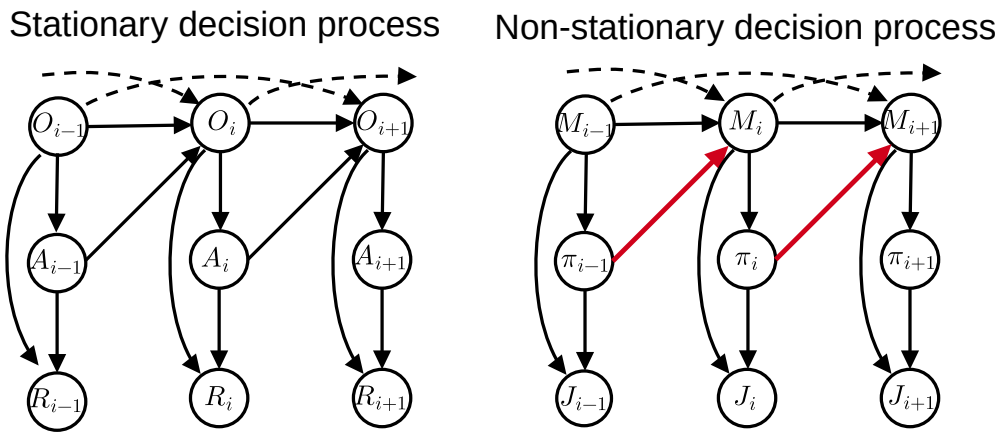


Figure 5.1. **(Left)** Control graph for interaction in a stationary POMDP, where each column corresponds to one time step. Here, *independent* episodes from the *same* POMDP can be resampled. **(Right)** Control graph that we consider for a non-stationary decision process, where each column corresponds to one episode. Here, the agent interacts with a sequence of related POMDPs $(M_i)_{i=1}^n$. In the absence of red arrows, the change from M_i to M_{i+1} is independent of the past decisions and is governed only by external factors (passive non-stationarity). The presence of red arrows indicated that M_{i+1} can *also* be dependent on the past decisions made in M_i (active non-stationarity).

5.3 Related Work

Recent methods that tackle non-stationarity only consider passive changes that are due to some external factor (Doshi-Velez and Konidaris, 2016; Chandak et al., 2020b; Xie et al., 2020a; Poiani et al., 2021). While these methods present an important stepping stone, such methods may result in catastrophic outcomes if used as-is in real-world settings that are subject to active (or hybrid) non-stationarity. We provide a simple illustrative example of this type of failure in Figure 5.2. Additionally, as we discuss in Section 5.4, alternative approaches such as modeling the problem as a large stationary POMDP or as a continuing average-reward MDP are not viable.

Non-stationarity can also be observed in multi-agent systems and games due to different agents/players interacting with the system. However, often the goal in these other areas is to search for (Nash) equilibria, which may not even exist under hybrid non-stationarity. Non-stationarity may also result due to artifacts of the learning

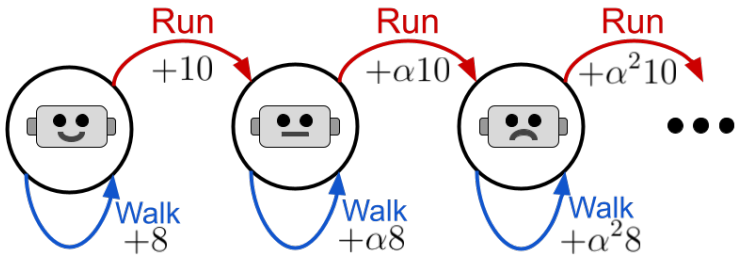


Figure 5.2. Consider a robot that can perform a task each day either by 'walking' or 'running'. A reward of 8 is obtained upon completion using 'walking', but 'running' finishes the task quickly and results in a reward of 10. However, 'running' wears out the motors, thereby increasing the time to finish the task the next day and reduces the returns for *both* 'walking' and 'running' by a small factor, $\alpha \in (0, 1)$. Here, methods for tackling *passive* non-stationarity will track the best policy under the assumption that the changes due to damages are because of external factors and would fail to attribute the cause of damage to the agent's decisions. Therefore, as on any given day 'running' will always be better, every day these methods will prefer 'running' over 'walking' and thus aggravate the damage. Since the outcome on each day is dependent on decisions made during previous days (active non-stationarity) this is effectively a task with a single lifelong episode, where 'walking' might be better in the long run. Finding a better policy first requires a method to evaluate a policy's (future) performance, which is the focus of this work.

algorithm even when the problem is stationary. While relevant, these other research areas are distinct from our setting of interest and we discuss them and other related work in more detail in Section 2.4.

One may also wonder if the average-reward/continuing setting (Sutton and Barto, 2018a) could be useful for our problem of interest. Unfortunately, the ergodicity assumption necessary for the average-reward setting is often violated because of the non-stationarity in the underlying decision process. For instance, in the healthcare example discussed in the introduction, a state with an earlier age cannot be revisited.

In multi-agent systems, non-stationarities may be induced by other agents trying to influence others (Xie et al., 2020b; Wang et al., 2021). To understand these methods in the context of our work, consider that there is a single lifelong episode in a 'mega' *stationary* POMDP comprised of all possible $M \in \mathcal{M}$. From this point of view, their method can be seen to be performing (soft) Q-learning/actor-critic in the continuing setting, where the state is the concatenation of the observation and the estimate of the unobserved component z, estimated from a fixed number of past interactions. As this is analogous to maximizing discounted returns in the continuing setting, estimating the Q-values here would ideally require the ergodicity assumption to revisit different o and z values multiple times, which may not be feasible in many settings.

5.4 Understanding Structural Assumptions

A careful reader would have observed that instead of considering interactions with a sequence of POMDPs $(M_i)_{i=1}^n$ that are each dependent on the past POMDPs and decisions, an equivalent setup might have been to consider a 'chained' sequence of interactions $(H_1, H_2, ..., H_n)$ in a *single* episode of some 'mega' POMDP capturing all $M \in \mathcal{M}$. Consequently, $\mathscr{J}(\pi)$ would correspond to the expected future return given the history $(H_i)_{i=1}^n$. Thus, $\mathscr{J}(\pi)$ could have been approximated using existing

methods *if* one could re-sample *multiple, independent,* sequences of interaction starting from μ_1.

However, 'chaining' $(H_i)_{i=1}^n$ results in only a *single* lifelong sequence of interactions. Without the ability to resample another sequence, it may be infeasible to estimate $\mathscr{J}(\pi)$ as future outcomes can be arbitrarily different from the past. Notice that the continuing/average-reward setting is not viable either because it relies on an ergodicity assumption that does not necessarily hold in the presence of non-stationarity. For instance, in the earlier example for automated healthcare personalized for individuals, it may not be possible to revisit the state of a patient at an earlier age.

In the following, we show how considering a sequence of POMDPs instead permits splitting this single interaction sequence into multiple (dependent) fragments, with additional structure linking together the fragments, thereby making the problem feasible. Let $\phi_i \in \Phi$ be some statistic associated with POMDP M_i for all $i > 0$ such that there is no uncertainty in $J_i(\pi)$ once ϕ_i is known, i.e., there exists a deterministic mapping from ϕ_i to $J_i(\pi)$. Therefore, if ϕ_k for $k > n$ can be obtained then these estimates of ϕ_k can help towards estimating $\mathscr{J}(\pi)$. We present some examples of ϕ_i in the following.

Complete model: If $\phi_i \coloneqq M_i$ then $J_i(\pi)$ can be obtained directly by evaluating π on M_i. However, obtaining M_i can be impractical and thus we do not consider this ϕ_i any further in this work.

Partial model (Hidden parameter): Analogous to HiP-MDPs (Doshi-Velez and Konidaris, 2016; Xie et al., 2020a), if z_i is an unobserved parameter of M_i that induces non-stationarity then we can define $\phi_i \coloneqq z_i$. For instance, the motor condition in the example in Figure 5.2.

Model-free (Policy performance): Instead of relying upon some intermediate statistics of the model, we can also consider a model-free approach wherein $\phi_i \coloneqq (J_i(\pi), \pi)$.

In all three cases, observe that for any statistic ϕ, and $\forall i > 0$, when executing a policy $\pi' \in \Pi$ throughout, there exists a *sequence* of $\mathcal{F}_i : \Phi \times \Pi \to \Delta(\Phi)$, where $\Delta(\Phi)$ is a distribution over Φ, such that,

$$\phi_{i+1} \sim \mathcal{F}_i(\phi_i, \pi'). \tag{5.2}$$

Implicitly, $\mathcal{F}_i$ captures the effect of both the underlying passive and active non-stationarity by modeling the conditional distribution of a statistic ϕ_{i+1} given ϕ_i, when π' is executed. For $\phi_i = (J_i(\pi), \pi)$, $\mathcal{F}_i$ models how $J(\pi)$ changes for a policy π when executing a different policy π'. As we will discuss later, this model-free free setting is particularly appealing as it directly captures the impact of non-stationarity on a policy's performance.

Notice that passive non-stationarity is a special case of (5.2) wherein π' does not influence the outcome of $\mathcal{F}_i$. That is,

$$\forall \pi, \pi' \in \Pi^2, \forall i > 0, \quad \mathcal{F}_i(\phi, \pi) \overset{D}{=} \mathcal{F}_i(\phi, \pi'),$$

where $\overset{D}{=}$ represents equality in distribution. Similarly, depending on ϕ, one may obtain stationarity if

$$\forall i > 0, \qquad \phi_{i+1} = \phi_i.$$

While a sequence of functions $(\mathcal{F}_i)_{i=1}^n$ in (5.2) provides complete generality, it does not yet enforce any useful structure as for $i > n$, $\mathcal{F}_i$ could still be arbitrary. Therefore, to make the problem tractable, we assume that the effect of non-stationarity on ϕ can be modeled using a fixed $\mathcal{F} \coloneqq \mathcal{F}_k = \mathcal{F}_j$ for all k, j.

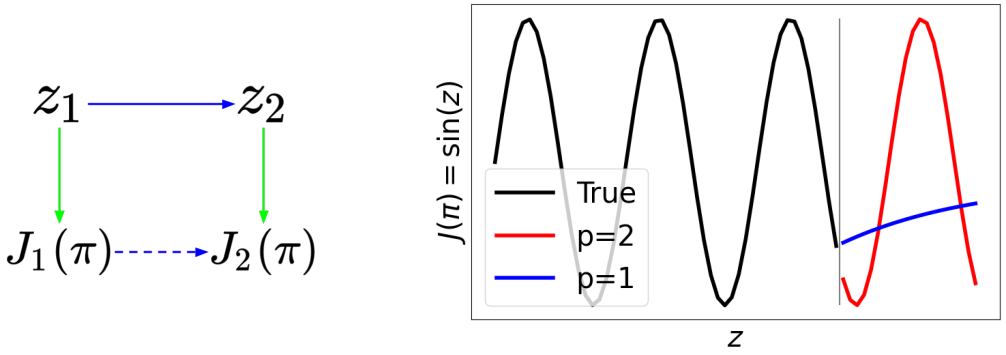


Figure 5.3. **(Left)** Considering structured changes in z (blue arrow) might often be more intuitive. However, as $J(\pi)$ estimation is ultimately required, unless performance of a policy also has some structure (green arrows) given z, generalizing across (potentially unseen) z's may not be possible. Structured changes for blue and green arrows consequently results in structured changes in $J(\pi)$ (dashed-blue arrows). For example, if the performance $J(\pi)$ of a policy changes (Lipschitz) smoothly with z, then smooth changes between z values automatically also imply smooth changes between $J(\pi)$ values. **(Right)** When executing a policy π, say z changes as $z_i = i$, and $J_i(\pi)$ changes periodically as $\sin(z_i)$. Here, even though both z and J change smoothly, changes in z_{i+1} can be modeled using one past term (i.e, z_i), but changes in $J_{i+1}(\pi)$ cannot be modeled only using $J_i(\pi)$ (which we denote as $p = 1$). Making $\mathcal{F}$ a function of the past $J(\pi)$ *sequence* (here, $J_i(\pi)$ and $J_{i-1}(\pi)$, denoted as $p = 2$) can alleviate such issues.

Assumption 5. *$\forall m \in \mathcal{M}$ such that the statistic associated with m is ϕ_i, there exists $\mathcal{F} : \Phi \times \Pi \to \Delta(\Phi)$, where*

$$\forall i, \quad \mathcal{F}(\phi_i, \pi')(\phi_{i+1}) = \Pr(\phi_{i+1} | M_i = m; \pi').$$

Intuitively, Assumption 5 imposes a *higher-order stationarity* condition under which non-stationarity can result in changes over time, but *the way the changes happen is fixed*. For example, in the healthcare setting where the physiology of a patient might change over days $(M_i)_{i=1}^{\infty}$, Assumption 5 states that how the unobserved factor ($\phi_i = z_i$) governing the non-stationarity changes, subject to treatment via π, is the same. Or alternatively, (the distribution of) the outcome $\phi_{i+1} = J_{i+1}(\pi)$ of a treatment on the $i + 1^{\text{th}}$ day given that the outcome of that treatment on the i^{th} day was $\phi_i = J_i(\pi)$ is

the same $\forall i > 0$. When $\phi_i = M_i$, then $\mathcal{F}$ is related to the 'meta-transition' function $\mathcal{T}$. Roughly, the tuple (ϕ, π, ϕ') for the non-stationary process can be viewed analogous to the (s, a, s') tuple in the stationary setting.

In some cases the definition of $\mathcal{F}$ may be restrictive, e.g., see Figure 5.3. Therefore, instead of making $\mathcal{F}$ dependent only on ϕ_i, one can redefine $\mathcal{F} : \Phi^p \times \Pi \to \Delta(\Phi)$. This permits more complex functions that model changes in ϕ conditioned on a *sequence* of the past p values of ϕ. However, for simplicity, to present the key ideas we will consider $\mathcal{F} : \Phi \times \Pi \to \Delta(\Phi)$.

We provide some examples in Figure 5.4 to demonstrate few settings to discuss the applicability of this assumption.

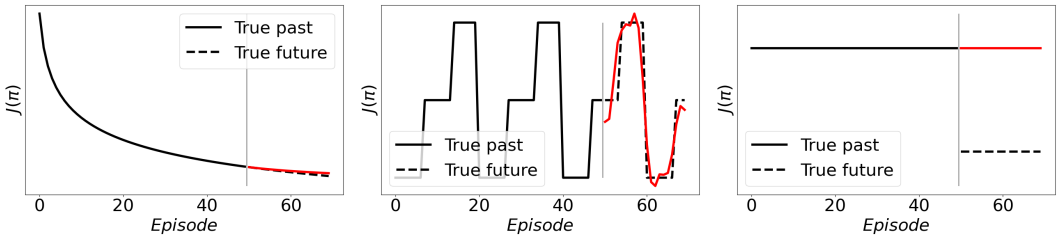


Figure 5.4. In this figure we plot different kinds of performance trends and discuss the applicability of Assumption 5 for each. The red curve corresponds to the forecast obtained using an auto-regressive model. **(Left)** In many cases where the performance of a policy is smoothly changing over time (for e.g., drifts in interests of an user that a recommender system needs to account for), looking at the past performances can often provide indication of how the performance would evolve in the future. **(Middle)** Changes in performances does not necessarily have to be smooth. What Assumption 5 enforces is that the changes have some structure which can be generalized to make predictions about how the performance would change in the future. Here, the performance jumps between different values (for e.g., if there is discontinuous change in the underlying system), but till their is some structure in the changes, it can be leveraged to make predictions about the future performances as well. **(Right)** While Assumption 5 can be applicable in many setting, there can be settings where this assumption does not hold. For example, if a motor of an industrial system is degrading over time but this degradation has no effect on the observable performance, until the point when the motor breaks down and the performance drops completely. In such cases, just looking at past performances may not be sufficient to infer how performance will change in the future.

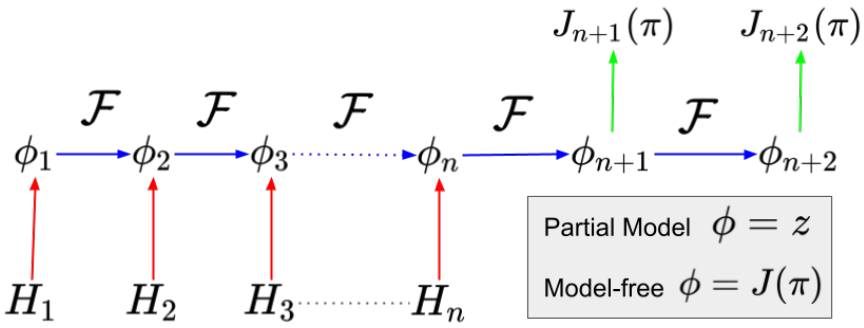


Figure 5.5. A high-level illustration of the proposed approach for estimating $\mathscr{J}(\pi)$. As we are only evaluating a particular policy π, we have removed the explicit dependence of π on both $\mathcal{F}$ and ϕ for a cleaner illustration.

5.5 Idea in a Nutshell

To evaluate $\mathscr{J}(\pi)$ we do not have any interaction data with the future POMDPs $(M_i)_{i=n+1}^{n+L}$. Therefore, we use D_n to extract the structure in how ϕ changes due to the hybrid non-stationarity, and use it to predict the values of $(J_i(\pi))_{i=n+1}^{n+L}$. The proposed approach can be divided into three broad steps, as illustrated using three arrow colors in Figure 5.5.

Red: Use $(H_i)_{i=1}^{n}$ to infer the values of $(\phi_i)_{i=1}^{n}$ associated with POMDPs $(M_i)_{i=1}^{n}$ (and the evaluation policy π). That is, infer the unobserved parameter when $\phi_i = z_i$, or infer the policy's past performances when $\phi_i = (J_i(\pi), \pi)$.

Blue: Leveraging the structure that there is a fixed $\mathcal{F}$ (Assumption 5), use auto-regression on the inferred values of $(\phi_i)_{i=1}^{n}$ to estimate $\mathcal{F}$, i.e., how ϕ changes due to external factors (passive non-stationarity) and due to the execution of π (active non-stationarity). Using this estimate for $\mathcal{F}$, predict what $(\phi_i)_{i=n+1}^{n+L}$ will be when π is deployed in the future.

Green: Infer the performances $(J_i(\pi))_{i=n+1}^{n+L}$ using the predicted values of $(\phi_i)_{i=n+1}^{n+L}$. In the model-free setting, where $J_i(\pi)$ is already a part of ϕ_i, this step is trivial. When $\phi_i = z_i$ this step corresponds to learning a function that can map z_i to $J_i(\pi)$.

5.6 Model-Free Policy Evaluation

In the following, we discuss the details for the model-free setting where $\phi_i = (J_i(\pi), \pi)$, with an emphasis on addressing the challenges of learning $\mathcal{F}$. As we are only considering evaluating a given π, we will make the dependency of π implicit and let $\phi_i \coloneqq J_i(\pi)$ from here on. Using the developed insights as guiding principles, we will later also provide algorithms for the partial-model setting ($\phi_i \coloneqq z_i$).

When evaluating $\mathscr{J}(\pi)$ for a given policy π, the model-free setting can be advantageous as the idea in Figure 5.5 requires accounting for changes in a *univariate statistic* $\phi_i = J_i(\pi)$, as opposed to the the partial-model setting where $\phi_i = z_i$ can be a *multivariate* statistic with an unknown dimension. However, there are three immediate challenges that can be observed in the model-free setting:

1. $(J_i(\pi))_{i=1}^n$ could have been directly estimated if we had access to $(M_i)_{i=1}^n$. However, we only have past interactions $(H_i)_{i=1}^n$ by possibly different policies $(\beta_i)_{i=1}^n$.

2. While Assumption 5 provides a structure $\mathcal{F}(J_i(\pi), \pi')$ for a *fixed* π', the underlying changes that occurred so far may be dependent on *multiple, different,* $(\beta_i)_{i=1}^n$ through $(\mathcal{F}(J_i(\pi), \beta_i))_{i=1}^n$. Therefore, the effect of underlying changes on $(J_i(\pi))_{i=1}^n$ so far may not directly reflect the changes that might occur when deploying π in the future.

3. How can we recover $\mathcal{F}(\cdot, \pi)$ such that it can be used to estimate $(J_i(\pi))_{i=n+1}^{n+L}$, and thus $\mathscr{J}(\pi)$, when considering executing π in the future?

5.6.1 Counterfactual Reasoning

To estimate $(J_i(\pi))_{i=1}^n$ when POMDPs $(M_i)_{i=1}^n$ are not available, we propose using the collected data D_n. Particularly, we aim to counterfactually predict *what the performance of* π *would have been, had* π *been executed on* M_i. To do so, we make the following standard support assumption that says that any action that is likely under π is also sufficiently likely under the policy β_i for all i.

Assumption 6. *$\forall o \in \mathcal{O}, \forall a \in \mathcal{A}$, and $\forall i \leq n$, $\frac{\pi(o,a)}{\beta_i(o,a)}$ exists and is bounded above by a (possibly unknown) constant c.*

Under Assumption 6, an unbiased estimate of $J_i(\pi)$ can be obtained using common off-policy evaluation methods like importance sampling (IS) or per-decision importance sampling (PDIS) (Precup, 2000),

$$\forall i, \widehat{J}_i(\pi) := \sum_{t=1}^{T} \rho_i^t R_i^t, \text{ where, } \rho_i^t := \prod_{j=1}^{t} \frac{\pi(O_i^j, A_i^j)}{\beta_i(O_i^j, A_i^j)}. \tag{5.3}$$

$\widehat{J}_i(\pi)$ provides us with an estimate of $\phi_i = J_i(\pi)$ associated with each M_i and policy π, as required for the red arrows in Figure 5.5.

5.6.2 Double Counterfactual Reasoning

Having obtained the estimates for $(J_i(\pi))_{i=1}^n$, we now aim to estimate how the performance of π changes due to the underlying hybrid non-stationarity. Recall from Assumption 5 that the changes in the performance of π, when executing π, can be modeled as

$$\forall i, \quad J_{i+1}(\pi) \sim \mathcal{F}(J_i(\pi), \pi). \tag{5.4}$$

Equivalently, $J_{i+1}(\pi)$ in (5.4) can be expressed as follows without loss of generality:

$$\forall i, \quad J_{i+1}(\pi) = f(J_i(\pi); \theta_\pi) + \xi(J_i(\pi); \omega_\pi), \tag{5.5}$$

where $f(J_i(\pi); \theta_\pi) = \mathbb{E}_\pi \left[J_{i+1}(\pi) | J_i(\pi) \right]$ and $\xi(J_i(\pi); \omega_\pi)$ is a mean zero random variable. Parameters $\theta_\pi \in \Theta$ and ω_π depend on π, and can thus model different types of changes when executing different policies.

If pairs of $(J_i(\pi), J_{i+1}(\pi))$ are available when the transition between M_i and M_{i+1} occurs due to execution of π, then one could auto-regress $J_{i+1}(\pi)$ on $J_i(\pi)$ to estimate

$f(\cdot;\theta_\pi)$ and model the changes in the performance of π. However, the sequence $(\widehat{J}_i(\pi))_{i=1}^n$ obtained from (5.3) cannot be used as-is for auto-regression. This is because the changes that occurred between M_i and M_{i+1} are associated with the execution of β_i, not π.

For example, recall the toy robot example in Figure 5.2. If data was collected by mostly 'running', then the performance of 'walking' would decay as well. Directly auto-regressing on the past performances of 'walking' would result in how the performance of 'walking' would change *when actually executing 'running'*. However, if we want to predict performances of 'walking' in the future, what we actually want to estimate is how the performance of 'walking' changes *if 'walking' is actually performed.*

To resolve the above issue, we ask another counter-factual question: *What would the performance of π in M_{i+1} have been had we executed π, instead of β_i, in M_i?*

To resolve the above issue, we ask another counter-factual question: *What would the performance of π in M_{i+1} have been had we executed π, instead of β_i, in M_i?* In the following theorem we show how this question can be answered with a second application of the importance ratio $\rho_i \coloneqq \rho_i^T$.

Theorem 1. *Under Assumptions 5 and 6, $\forall m \in \mathcal{M}$ such that the performance $J(\pi)$ associated with m is j,*

$$\mathbb{E}_\pi \left[J_{i+1}(\pi) | J_i(\pi) = j \right] = \mathbb{E}_{\beta_i, \beta_{i+1}} \left[\rho_i \widehat{J}_{i+1}(\pi) \middle| M_i = m \right].$$

See Section 5.10.1 for the proof. Intuitively, Theorem 1 uses ρ_i to first correct for the mismatch between π and β_i that influences how M_i changes to M_{i+1} due to interactions H_i. Secondly, $\widehat{J}_{i+1}$ corrects for the mismatch between π and β_{i+1} for the sequence of interactions H_{i+1} in M_{i+1}.

5.6.3 Importance Weighted IV-Regression

An important advantage of Theorem 1 is that given $J_i(\pi)$, $\rho_i \widehat{J}_{i+1}(\pi)$ provides an unbiased estimate of $\mathbb{E}_\pi\left[J_{i+1}(\pi)|J_i(\pi)\right]$, even though π may not have been used for data collection. This permits using $\rho_i \widehat{J}_{i+1}(\pi)$ as a target for the prediction of the next performance given $J_i(\pi)$, i.e., to estimate $f(J_i(\pi);\theta_\pi)$ in (5.5).

However, notice that performing regression on the pairs $(X_i = J_i(\pi), Y_i = \rho \widehat{J}_{i+1}(\pi))_{i=1}^{n-1}$ may not be directly possible as we do not have $J_i(\pi)$; only unbiased *estimates* $\widehat{J}_i(\pi)$ of $J_i(\pi)$. This is problematic because in least-squares regression, while noisy estimates of the *target* variable Y_i are fine, noisy estimates of the *input* variable X_i may result in estimates of θ_π that are not asymptotically consistent even when the underlying f in (5.5) is a linear function of its inputs. To see this clearly, consider the following naive estimator,

$$\hat{\theta}_{\texttt{naive}} \in \operatorname*{argmin}_{\theta\in\Theta} \sum_{i=1}^{n-1} \left(f\left(\widehat{J}_i(\pi);\theta\right) - \rho_i \widehat{J}_{i+1}(\pi)\right)^2.$$

Because $\widehat{J}_i(\pi)$ is an unbiased estimate of J_π, let $\widehat{J}_i(\pi) = J_i(\pi) + \eta_i$, where η_i is mean zero noise. Let $\mathbb{N} \coloneqq [\eta_1, \eta_2, ..., \eta_{n-1}]^\top$ and $\mathbb{J} \coloneqq [J_1(\pi), J_2(\pi), ..., J_{n-1}(\pi)]^\top$. When f is a linear function of its inputs, the expected value $\mathbb{E}_\pi[J_{i+1}(\pi)|J_i(\pi)] = J_i\theta_\pi$. Also, as $\rho_i \hat{J}_{i+1}(\pi)$ is an unbiased estimator for $J_i(\pi)\theta_\pi$ given $J_i(\pi)$, let $\rho_i \hat{J}_{i+1}(\pi) = J_i(\pi)\theta_\pi + \zeta_i$, where ζ_i is mean zero noise. Let $\mathbb{N}_2 \coloneqq [\zeta_1, \zeta_2, ..., \zeta_{n-1}]^\top$ then $\theta_{\texttt{naive}}$ can be expressed as,

$$\begin{aligned}
\hat{\theta}_{\text{naive}} &= \Big((\mathbb{J}+\mathbb{N})^\top(\mathbb{J}+\mathbb{N})\Big)^{-1}(\mathbb{J}+\mathbb{N})^\top(\mathbb{J}\theta_\pi + \mathbb{N}_2) \\
&= \big(\mathbb{J}^\top\mathbb{J} + 2\mathbb{J}^\top\mathbb{N} + \mathbb{N}^\top\mathbb{N}\big)^{-1}\big(\mathbb{J}^\top\mathbb{J}\theta_\pi + \mathbb{N}^\top\mathbb{J}\theta_\pi + \mathbb{J}^\top\mathbb{N}_2 + \mathbb{N}^\top\mathbb{N}_2\big) \\
&= \left(\frac{1}{n}\big(\mathbb{J}^\top\mathbb{J} + 2\mathbb{J}^\top\mathbb{N} + \mathbb{N}^\top\mathbb{N}\big)\right)^{-1}\left(\frac{1}{n}\big(\mathbb{J}^\top\mathbb{J}\theta_\pi + \mathbb{N}^\top\mathbb{J}\theta_\pi + \mathbb{J}^\top\mathbb{N}_2 + \mathbb{N}^\top\mathbb{N}_2\big)\right) \quad (5.6)
\end{aligned}$$

In the limit, using the continuous mapping theorem when the inverse in (5.6) exists,

$$\lim_{n\to\infty} \hat{\theta}_{\text{naive}} = \left(\lim_{n\to\infty} \frac{1}{n}\left(\mathbb{J}^\top\mathbb{J} + 2\mathbb{J}^\top\mathbb{N} + \mathbb{N}^\top\mathbb{N}\right)\right)^{-1} \left(\lim_{n\to\infty} \frac{1}{n}\left(\mathbb{J}^\top\mathbb{J}\theta_\pi + \mathbb{N}^\top\mathbb{J}\theta_\pi + \mathbb{J}^\top\mathbb{N}_2 + \mathbb{N}^\top\mathbb{N}_2\right)\right). \tag{5.7}$$

Observe that both $\mathbb{N}$ and $\mathbb{N}_2$ are mean zero and uncorrelated with each other and also with $\mathbb{J}$. Therefore, the terms corresponding to $\mathbb{J}^\top\mathbb{N}$, $\mathbb{J}^\top\mathbb{N}_2$, and $\mathbb{N}^\top\mathbb{N}_2$ in (5.7) will be zero almost surely due to Rajchaman's strong law of large numbers for uncorrelated random variables (Rajchman, 1932; Chandra, 1991). However, the term corresponding to $\mathbb{N}^\top\mathbb{N}$ will not be zero in the limit, and instead roughly result in (average of the) variances of η_i. Consequently, this results in,

$$\hat{\theta}_{\texttt{naive}} \xrightarrow{a.s.} \left(\mathbb{J}^\top\mathbb{J} + \mathbb{N}^\top\mathbb{N}\right)^{-1} \mathbb{J}^\top\mathbb{J}\theta_\pi.$$

Observe that $\mathbb{N}^\top\mathbb{N}$ in (5.7) relates to the variances of the mean zero noise variables η_i and this would bias $\hat{\theta}_{\texttt{naive}}$ towards zero (if $\forall i$, $\eta_i = 0$, then the true θ_π is trivially recovered). Intuitively, when the variance of η_i is high, noise dominates and the structure in the data gets suppressed even in the large-sample regime.

Unfortunately, the importance sampling based estimator $\widehat{J}_i(\pi)$ is known to suffer from high variance (Thomas et al., 2015b). Therefore, $\hat{\theta}_{\texttt{naive}}$ can be very biased and we will not be able to capture the trend in how performances are changing, even in the limit of infinite data and linear f. The problem may be exacerbated when f is non-linear.

Bias Reduction: To mitigate the bias stemming from noise in input variables, we introduce a novel instrument variable (IV) (Pearl et al., 2000) regression method for tackling non-stationarity. Instrument variables represent some side-information and were originally used in the causal literature to mitigate any bias resulting due to spurious correlation, caused by unobserved confounders, between the input and the target variables. For mitigating bias in our setting, IVs can intuitively be considered as some side-information to 'denoise' the input variable before performing regression.

For this IV-regression, an ideal IV is *correlated* with the input variables (e.g., $\widehat{J}_i(\pi)$) but *uncorrelated* with the noises in the input variable (e.g., η_i).

We propose leveraging statistics based on past performances as an IV for $\widehat{J}_i(\pi)$. For instance, using $\widehat{J}_{i-1}(\pi)$ as an IV for $\widehat{J}_i(\pi)$. Notice that while correlation between $J_{i-1}(\pi)$ and $J_i(\pi)$ can directly imply correlation between $\widehat{J}_{i-1}(\pi)$ and $\widehat{J}_i(\pi)$, values of $J_{i-1}(\pi)$ and $J_i(\pi)$ are dependent on non-stationarity in the past. Therefore, we make the following assumption, which may easily be satisfied when the consecutive performances do not change arbitrarily.

Assumption 7. $\forall i, \quad \mathrm{Cov}\left(\widehat{J}_{i-1}(\pi), \widehat{J}_i(\pi)\right) \neq 0.$

However, notice that the noise in $\widehat{J}_i(\pi)$ can be *dependent* on $\widehat{J}_{i-1}(\pi)$. This is because non-stationarity can make H_{i-1} and H_i dependent, which are in turn used to estimate $\widehat{J}_{i-1}(\pi)$ and $\widehat{J}_i(\pi)$, respectively. Nevertheless, perhaps interestingly, we show that despite not being independent, $\widehat{J}_{i-1}(\pi)$ is *uncorrelated* with the noise in $\widehat{J}_i(\pi)$.

Theorem 2. *Under Assumptions 5 and 6,*

$$\forall i, \quad \mathrm{Cov}\left(\widehat{J}_{i-1}(\pi), \widehat{J}_i(\pi) - J_i(\pi)\right) = 0.$$

See Section 5.10.2 for the proof. Now using $\widehat{J}_{i-1}(\pi)$ as an IV for $\widehat{J}_i(\pi)$, IV regression requires learning an additional function $g \coloneqq \mathbb{R} \to \mathbb{R}$ parameterized by $\varphi \in \Omega$, and propose the following IV-regression based estimator,

$$\hat{\varphi}_n \in \operatorname*{argmin}_{\varphi \in \Omega} \sum_{i=2}^{n} \left(g\left(\widehat{J}_{i-1}(\pi); \varphi\right) - \widehat{J}_i(\pi)\right)^2 \tag{5.8}$$

$$\hat{\theta}_n \in \operatorname*{argmin}_{\theta \in \Theta} \sum_{i=2}^{n-1} \left(f\left(g\left(\widehat{J}_{i-1}(\pi); \hat{\varphi}_n\right); \theta\right) - \rho_i \widehat{J}_{i+1}(\pi)\right)^2.. \tag{5.9}$$

Theorem 3. *Under Assumptions 5, 6, and 7, if f and g are linear functions of their inputs, then $\hat{\theta}_n$ is a strongly consistent estimator of θ_π, i.e.,*

$$\hat{\theta}_n \xrightarrow{a.s.} \theta_\pi.$$

See Section 5.10.2 for the proof.

Remark 2. *Other choices of instrument variables (apart from $\widehat{J}_{i-1}(\pi)$) are also viable. We discuss some alternate choices in Section 5.7.3. These other IVs can be used in (5.8) and (5.9) by replacing $\widehat{J}_{i-1}(\pi)$ with the alternative instrument variable.*

Remark 3. *As discussed earlier, it may be beneficial to model $J_{i+1}(\pi)$ using $(J_k(\pi))_{k=i-p+1}^{i}$ with $p > 1$. The proposed estimator can be easily extended by making f dependent on multiple past terms $(X_k)_{k=i-p+1}^{i}$, where $\forall k$, $X_k \coloneqq g((\widehat{J}_l(\pi))_{l=k-p}^{k-1}; \hat{\phi})$. We discuss this in more detail in Section 5.7.3. The proposed procedure is also related to methods that use lags of the time series as instrument variables (Bellemare et al., 2017; Wilkins, 2018; Wang and Bellemare, 2019).*

Remark 4. *An advantage of the model-free setting is that we only need to consider changes in $J(\pi)$, which is a* **scalar** *statistic. In such a setting,* linear *auto-regressive models have been known to be useful in modeling a wide variety of time-series trends, e.g., in Figure 5.3, the forecast for $p = 2$ was obtained using a linear model. Further,* non-linear *functions like recurrent neural networks and LSTMs (Hochreiter and Schmidhuber, 1997) can also be leveraged using deep instrument variable methods (Hartford et al., 2017; Bennett et al., 2019; Liu et al., 2020; Xu et al., 2020).*

As required for the blue arrows in Figure 5.5, $f(\cdot; \hat{\theta}_n)$ can now be used to estimate the expected value $\mathbb{E}_\pi\left[J_{i+1}(\pi) | J_i(\pi)\right]$ under hybrid non-stationarity. Therefore, using $f(\cdot; \hat{\theta}_n)$ we can now auto-regressively forecast the future values of $(J_i(\pi))_{i=n+1}^{n+L}$ and obtain an estimate for $\mathscr{J}(\pi)$. Note that when $\phi_i = J_i(\pi)$, the green arrows in Figure 5.5 correspond to identity functions.

Variance Reduction: As discussed earlier, importance sampling results in noisy estimates of $J_i(\pi)$. During regression, while high noise in the input variable leads to high bias, high noise in the target variables leads to high variance parameter estimates. As discussed earlier, the instrument variable technique helps to mitigate bias. To mitigate variance, we draw inspiration from the reformulation of weighted-importance sampling presented for the *stationary* setting by Mahmood et al. (2014), and propose the following estimator,

$$\tilde{\varphi}_n \in \operatorname*{argmin}_{\varphi \in \Omega} \sum_{i=2}^{n} \bar{\rho}_i \left(g\left(\widehat{J}_{i-1}(\pi); \varphi\right) - G_i(\pi) \right)^2, \tag{5.10}$$

$$\tilde{\theta}_n \in \operatorname*{argmin}_{\theta \in \Theta} \sum_{i=2}^{n-1} \rho_i^\dagger \left(f\left(g\left(\widehat{J}_{i-1}(\pi); \tilde{\varphi}_n\right); \theta\right) - G_{i+1}(\pi) \right)^2, \tag{5.11}$$

$$\text{where,} \quad \bar{\rho}_i \coloneqq \frac{\rho_i}{\left(\sum_{j=2}^{n} \rho_j\right)} \quad \text{and} \quad \rho_i^\dagger \coloneqq \frac{\rho_i \rho_{i+1}}{\left(\sum_{j=2}^{n-1} \rho_j \rho_{j+1}\right)},$$

where g and $\hat{\varphi}_n$ are the same as defined in (5.8), and G_{i+1} is the return observed for M_{i+1}. Intuitively, instead of importance weighting the *target*, to obtain $\tilde{\theta}_n$ we importance weight the squared error, proportional to how likely that *error* would be if π was used to collect the data. Since dividing by any constant does not affect $\tilde{\theta}_n$, the choice of $\bar{\rho}_i$ ensures that $\bar{\rho}_i \leq 1$ always, thereby mitigating the variance but still providing consistency.

Theorem 4. *Under Assumptions 5, 6, and 7, if f and g are linear functions of their inputs, then $\tilde{\theta}_n$ is a strongly consistent estimator of θ_π, i.e.,*

$$\tilde{\theta}_n \xrightarrow{a.s.} \theta_\pi.$$

See Section 5.10.2 for the proof.

5.7 Empirical Analysis

This section presents empirical evaluations using several environments inspired by real-world applications that exhibit non-stationarity. In the following paragraphs, we briefly discuss each environment.

5.7.1 Environments

We provide empirical results on four non-stationary environments: a toy robot environment, non-stationary mountain car, diabetes treatment, and MEDEVAC domain for routing air ambulances. Details for each of these environments are provided in this section. For all of the above environments, we regulate the 'speed' of non-stationarity to characterize an algorithms' ability to adapt. Higher speed corresponds to a faster rate of non-stationarity; A speed of zero indicates that the environment is stationary.

5.7.1.0.1 RoboToy: This domain corresponds to the toy robot scenario depicted in Figure 5.2. Here, a robot can accomplish a task by either 'walking' or 'running'. 'Running' performs the task faster than 'walking' and thus the reward received at the end of executing 'running' is higher. However, 'running' cause more wear and tear on the robot, degrading the performance of both the options. Since the past interactions influence the non-stationarity, this is an instance of active non-stationarity. We call this domain **RoboToy-Active**.

To test our algorithms, we also simulated a **RoboToy-Passive** domain, where there is no active non-stationarity as above. Instead, the reward obtained at the end of executing the options fluctuate across episodes. Therefore, the changes to the underlying system are independent of the actions taken by the agent in the past.

For both the active and passive version of this domain, we collect data using a behavior policy that chooses 'walking' more frequently, and the evaluation policy is designed such that it chooses 'running' more frequently.

5.7.1.0.2 Non-stationary Mountain Car: In real-world mechanical systems, motors undergo wear and tear over time based on how vigorously they have been used in the past. To simulate similar performance degradation, we adapt the classic (stationary) mountain car domain (Moore, 1990). We modify the domain such that at every episode the effective acceleration force is decayed proportional to the average velocity of the car in the previous episode. This results in active non-stationarity as the change in the system is based on the actions taken by the agent in the past. Similar to the works by (Thomas, 2015; Jiang and Li, 2015), we make use of macro-actions to repeat an action 10 times, which helps in reducing the effective horizon length of each episode. The maximum number of step per episode using these macros is 30.

For our experiments, using an actor-critic algorithm (Sutton and Barto, 2018a) we find a near-optimal policy π on the stationary version of the mountain car domain, which we use as the evaluation policy. Let $\pi^{\texttt{rand}}$ be a random policy with uniform distribution over the actions. Then we define the behavior policy $\beta(o, a) \coloneqq 0.5\pi(o, a) + 0.5\pi^{\texttt{rand}}(o, a)$ for all states and actions.

5.7.1.0.3 Type-1 Diabetes Management: Automated healthcare systems that aim to personalise for individual patients should account for the physiological changes of the patient over time. To simulate such a scenario we use an open-source implementation (Xie, 2019) of the U.S. Food and Drug Administration (FDA) approved Type-1 Diabetes Mellitus simulator (T1DMS) (Man et al., 2014) for the treatment of Type-1 diabetes, where we induced non-stationarity by oscillating the body parameters (e.g., rate of glucose absorption, insulin sensitivity, etc.) between two known configurations available in the simulator. This induces passive non-stationarity, that is, changes are not dependent on past actions.

Each step of an episode corresponds to a minute (1440 timesteps–one for each minute in a day) in an *in-silico* patient's body and state transitions are governed by a continuous time non-linear ordinary differential equation (ODE) (Man et al., 2014).

This makes the problem particularly challenging as it is unclear how the performance trends of policies vary in this domain when the physiological parameters of the patient are changed. Notice that as the parameters that are being oscillated are inputs to a non-linear ODE system, the exact trend of performance for any policy is unknown. This more closely reflects a real-world setting where Assumption 3 might not hold, as every policy's performance trend in real-world problems cannot be expected to follow *any* specific trend *exactly*–one can only hope to obtain a coarse approximation of the trend.

For our experiments, using an actor-critic algorithm (Sutton and Barto, 2018a) we find a near-optimal policy π on the stationary version of this domain, which we use as the evaluation policy. The policy learns the CR and CF parameters of the basal-bolus controller discussed in Chapter 3.6.1. Let $\pi^{\texttt{rand}}$ be a random policy with uniform distribution over actions. Then we define the behavior policy $\beta(o, a) \coloneqq 0.5\pi(o, a) + 0.5\pi^{\texttt{rand}}(o, a)$ for all states and actions.

5.7.1.0.4 MEDEVAC: This domain stands for *med*ical *evac*uation using air ambulances. This domain was developed by Robbins et al. (2020) for optimally routing air ambulances to provide medical assistance in regions of conflict. This domain divides the region of conflict into 34 mutually exclusive zones, and has 4 air ambulances to serve all zones when an event occurs. Based on real-data, this domain simulates the arrival of different events, from different zones, where each event can have 3 different priority levels. Serving higher priority events yields higher rewards. If an ambulance is assigned to an event, it will finish the assignment in a time dependent on the distance between the base of the ambulance and the zone of the corresponding event. While engaged in an assignment, that ambulance is no longer available to serve other events. A good controller decides whether to deploy, and which MEDEVAC to deploy, to serve any event (at the risk of not being able to serve a new high-priority event if all ambulances become occupied).

The original implementation of the domain assumes that the arrival rates of the events and the time taken by an ambulance to complete an event follow a Poisson process with a fixed rate. However, in reality, the arrival rates of different events can change based on external incidents during conflict. Similarly, the completion rate can also change based on how frequently an ambulance is deployed. To simulate such non-stationarity, we oscillate the arrival rate of the incoming high-priority events, which induces passive non-stationarity. Further, to induce wear and tear, we slowly decay the rate at which an ambulance can finish an assignment. This decay is proportional to how frequently the ambulance was used in the past. This induces active non-stationarity. The presence of both active and passive changes makes this domain subject to hybrid non-stationarity.

Similar to other domains, we used an actor-critic algorithm (Sutton and Barto, 2018a) we find a near-optimal policy π on the stationary version of this domain, which we use as the evaluation policy. Let $\pi^{\texttt{rand}}$ be a random policy with uniform distribution over the actions. Then we define the behavior policy $\beta(o, a) \coloneqq 0.5\pi(o, a) + 0.5\pi^{\texttt{rand}}(o, a)$ for all states and actions.

5.7.2 Algorithms Compared

We consider the following algorithms for comparison:

5.7.2.0.1 OPEN: We call our proposed method OPEN, which stands for '**o**ff-**p**olicy **e**valuation for **n**on-stationary domains'. This method is based on our bias and variance reduced estimator developed in (5.10) and (5.11) and is developed to handle structured passive, active, and hybrid non-stationarity.

5.7.2.0.2 Pro-WLS: As a baseline, we use the algorithm developed in Chapter 3 for tackling passive non-stationarity. Particularly, we use Prognosticator with weighted least-squares (Pro-WLS) to obtain variance reduction. However, note that as Pro-WLS

is designed to provide policy improvement, we only use equation (3.4.5) to evaluate future performance of a policy.

5.7.2.0.3 WIS: This is the standard weighted importance sampling based estimator that ignores presence of non-stationarity completely.

5.7.3 Implementation and Hyper-parameters

We have established the key insight for how to forecast the next performance based on a single previous performance, when the true performance trend of a policy can be modeled auto-regressively using a single past term. However, as noted in Figure 5.3 using more terms can provide more flexibility in the the type of trends that can be modeled. Therefore, we leverage statistics based on multiple past terms to form the instrument variable Z_i.

One immediate choice for Z_i is $\widehat{J}_i(\pi)$. However, we found that the high variance of IS estimate makes $\widehat{J}_i(\pi)$ a weak instrument variable (Pearl et al., 2000), that is not strongly correlated with $J_{i+1}(\pi)$. Better choices of Z_i may be the ones that are strongly correlated with $J_{i+1}(\pi)$ but uncorrelated with the noise in the $\widehat{J}_{i+1}(\pi)$ estimate. We found that an alternate choice of Z_i composed of the unweighted return G_i and a WIS estimate for $J_i(\pi)$ (where the normalization is done only using the importance ratios from episodes before i) to be more useful. Specifically, we let $Z_i \coloneqq [G_i, \widetilde{J}_i(\pi)]$, where

$$\widetilde{J}_i(\pi) \coloneqq \frac{\rho_i G_i}{\sum_{k=1}^{i} \rho_k}.$$

It can be observed similar to Theorem 2 that this Z_i is uncorrelated with the noise in $\widehat{J}_{i+1}(\pi)$ as well. Further, the weighted version $\widetilde{J}_i(\pi)$ suffers less from variance and we found it to be more strongly correlated with $J_{i+1}(\pi)$. Further, often the performance of the behavior policy is positively/negatively correlated with the performance of the evaluation policy and thus G_i tends to be correlated with $J_{i+1}(\pi)$ as well. One could also explore other potential IVs; we leave this for future work.

Now using past p values of Z_i to form the complete instrument variable, where p is a hyper-parameter, we use the following importance weighted instrument-variable regression,

$$\tilde{\varphi}_n \in \underset{\varphi\in\Omega}{\operatorname{argmin}} \sum_{i=p+1}^{n} \bar{\rho}_i \left(g\left((Z_j(\pi))_{j=i-p}^{i-1} ; \varphi \right) - G_i(\pi) \right)^2,$$
$$\tilde{\theta}_n \in \underset{\theta\in\Theta}{\operatorname{argmin}} \sum_{i=2p}^{n-1} \rho_i^{\dagger} \left(f\left(\left(\bar{J}_j(\pi)\right)_{j=i-p+1}^{i} ; \theta \right) - G_{i+1}(\pi) \right)^2,$$

where,

$$\begin{aligned}
\bar{J}_i(\pi) =& g\left((Z_j(\pi))_{j=i-p}^{i-1} ; \tilde{\varphi}_n \right), \quad \forall p < i \leq n, \\
\bar{\rho}_i \coloneqq& \frac{\rho_i}{(\sum_{j=2}^{n} \rho_j)} \\
\rho_i^{\dagger} \coloneqq& \frac{\rho_i \rho_{i+1}}{(\sum_{j=2}^{n-1} \rho_j \rho_{j+1})}.
\end{aligned}$$

Once $\tilde{\theta}_n$ is obtained, we use it to auto-regressively forecast the future performances. Particularly, we use $(\bar{J}_k)_{k=n+1}^{n+L}$ as the predicted performances for the next L episodes, where

$$\forall i > n,\ \bar{J}_i \coloneqq f\left(\left(\bar{J}_{i-k}(\pi)\right)_{k=1}^{p} ; \tilde{\theta}_n \right).$$

While our theoretical results were established for the setting where there is only a single regressor ($p = 1$), a more generalized theoretical result for $p > 1$ may be possible using the concepts of endogenous and exogenous regressors. Particularly, let $[..., X_i, X_{i+1}, X_{i+2}, X_{i+3}, ...]$, be observations from an $AR(2)$ time-series sequence where X_{i+3} depends on X_{i+1} and X_{i+2}. Here, using X_{i+1} as the only instrument variable for X_{i+2} is not possible as X_{i+3} is correlated with X_{i+1}. However, $Z = X_i$ or even $Z = [X_i, X_{i+1}]$ may form a valid instrument for X_{i+2} as neither the noise in

X_{i+3} nor the noise in X_{i+2} is correlated with at least one component of Z, i.e., X_i. For precise instrument relevance conditions and additional discussion, we refer the reader to the works by Abbott (2007); Cameron (2019); Parker (2020). We leave this theoretical extension for the future work.

5.7.3.0.1 Hyper-parameters: For the Pro-WLS baseline, we use the weighted least-squares procedure using the Fourier basis features (Chandak et al., 2020b). The hyper-parameter for this baseline is the number of Fourier terms d that should be used to estimate the performance trend. We found that setting d to be too high results in extremely high-variance and setting it to a lower value fails to capture the trend in performance. Therefore, based on ablation studies in Figure 5.11 we set $d = 5$ for all the experiments.

For OPEN, the hyper-parameter corresponds to the number of terms to condition on during auto-regression. Based on ablation studies in Figure 5.11 we set $p = 300$ (15% of the number of episodes in the data) for all the experiments.

For each environment, we collect data consisting of 2000 episodes of interaction using the behavior policy, and predict the expected future returns if executing the evaluation policy for the next 200 episodes. The behavior policy and the evaluation policy for each domain are described in Section 5.7.1.

Since the future outcomes are stochastic, to evaluate the true expected future performance in (5.1), we create digital-clones of the environment *after* data has been collected using the behavior policy. Using these clones, we compute the average of 30 possible futures when executing the evaluation policy. This estimate of the *expected* future returns are then used as the ground truth for comparison with the predictions made by the algorithms.

5.7.4 Results for Active/Hybrid Non-stationarity

For each environment, we collect data consisting of 2000 episodes of interaction using the behavior policy, and predict the expected future returns if executing the evaluation policy for the next 200 episodes. The behavior policy and the evaluation policy for each domain are described in Section 5.7.1.

Since the future outcomes are stochastic, to evaluate the true expected future performance in (5.1), we create digital-clones of the environment *after* data has been collected using the behavior policy. Using these clones, we compute the average of 30 possible futures when executing the evaluation policy. This estimate of the *expected* future returns are then used as the ground truth for comparison with the predictions made by the algorithms.

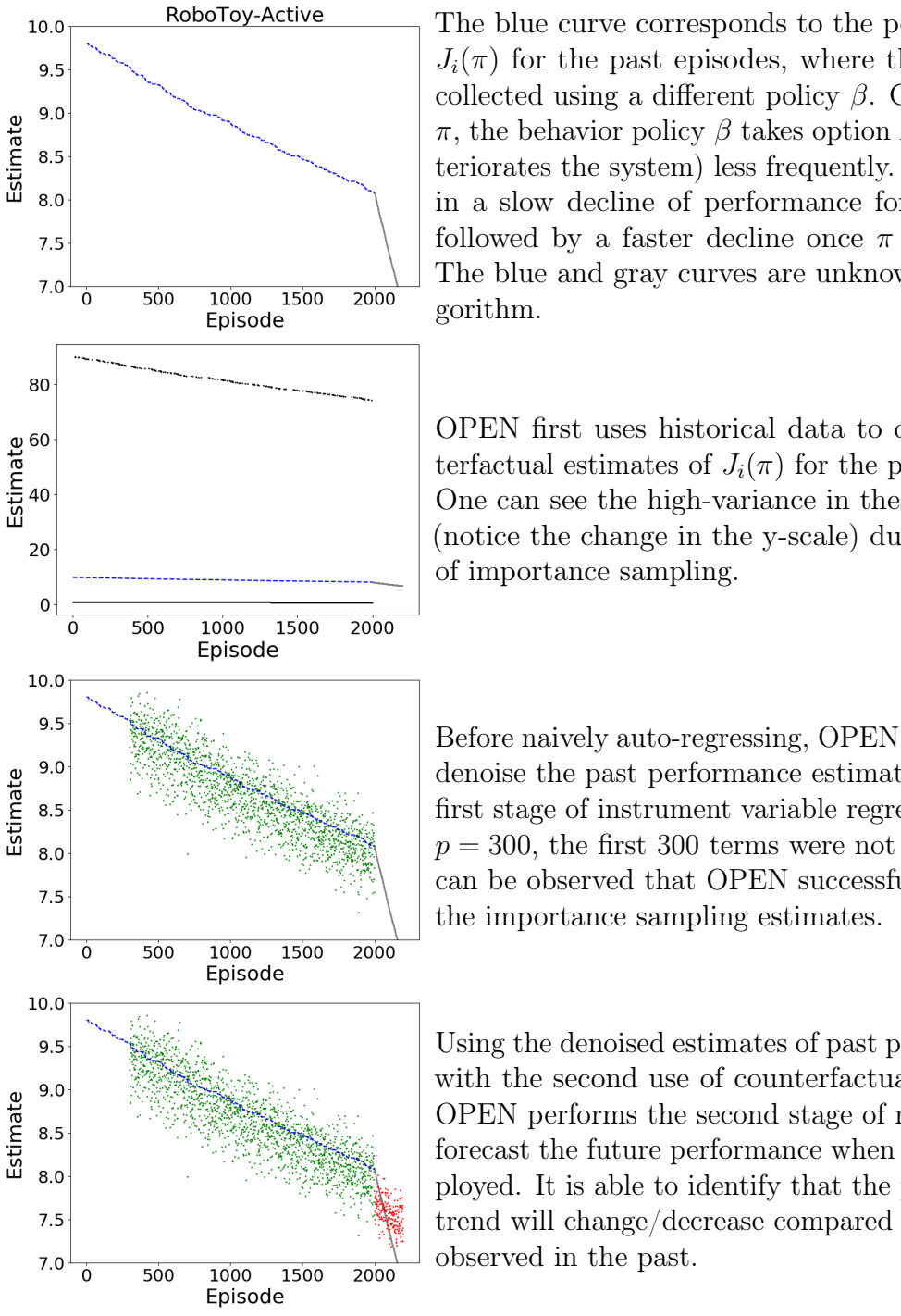
5.7.4.1 Single Run

In Figure 5.6 we present a step by step breakdown of the intermediate stages of a single run of OPEN on the RoboToy-Active domain. It can be observed how initially the data collecting policy was less frequently taking the option that damages the robot and hence the performance decline was slower. However, the performance of the evaluation policy, which more frequently takes the option that damages the robot, declines faster. OPEN is able to extract such information to detect if the evaluation policy will cause any active harm, if deployed in the future.

5.7.4.2 Summary Plots

In this section we present a summary of the results across all the domains with active non-stationarity.

5.7.4.2.1 Bias Analysis Figure 5.7 presents the (absolute) bias incurred by different algorithms for predicting the future performance of the evaluation policy π.



The blue curve corresponds to the performances $J_i(\pi)$ for the past episodes, where the data was collected using a different policy β. Compared to π, the behavior policy β takes option A (which deteriorates the system) less frequently. This results in a slow decline of performance for π initially, followed by a faster decline once π is deployed. The blue and gray curves are unknown to the algorithm.

OPEN first uses historical data to obtain counterfactual estimates of $J_i(\pi)$ for the past episodes. One can see the high-variance in these estimates (notice the change in the y-scale) due to the use of importance sampling.

Before naively auto-regressing, OPEN first aims to denoise the past performance estimates using the first stage of instrument variable regression. Since $p = 300$, the first 300 terms were not denoised. It can be observed that OPEN successfully denoises the importance sampling estimates.

Using the denoised estimates of past performances, with the second use of counterfactual reasoning, OPEN performs the second stage of regression to forecast the future performance when π will be deployed. It is able to identify that the performance trend will change/decrease compared to what was observed in the past.

Figure 5.6. An illustrative step by step breakdown of the stages in the proposed algorithm OPEN for the RoboToy-Active domain.

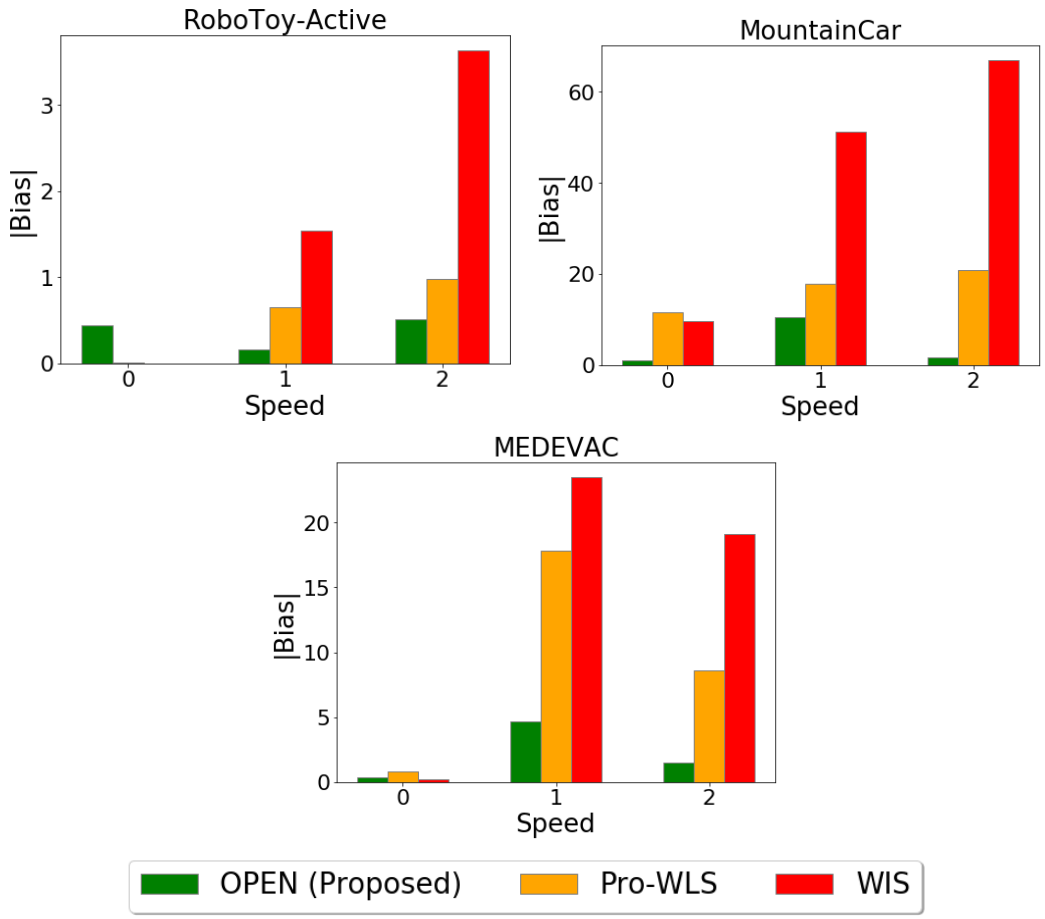


Figure 5.7. Comparison of different algorithms for predicting the future performance of evaluation policy π on domains that exhibit active/hybrid non-stationarity. On the x-axis is the speed which corresponds to the rate of non-stationarity; higher speed indicates faster rate of change and a speed of zero indicates stationary domain. On the y-axis is the absolute bias in the performance estimate **(lower is better)**. For each domain, for each speed, for each algorithm, 30 trials were executed. Discussions for these plots can be found in Section 5.7.4.2.1. Here, |bias| was computed using the absolute value of the difference between (a) the predicted future performance averaged across 30 trials and (b) the ground truth future performance. That is, for an estimator $\hat{J}$ of J, the bias is $|J - E[\hat{J}]|$. Because of this, 30 trials only gives us a point estimate for bias. (Notice that using the absolute value of the difference between (a) the predicted future performance for each trial and (b) the true future performance', averaged across 30 trials, will provide an estimate of $E[|J - \hat{J}|]$, which would not capture the bias but will be more like the variance (using L1/absolute distance instead of L2)).

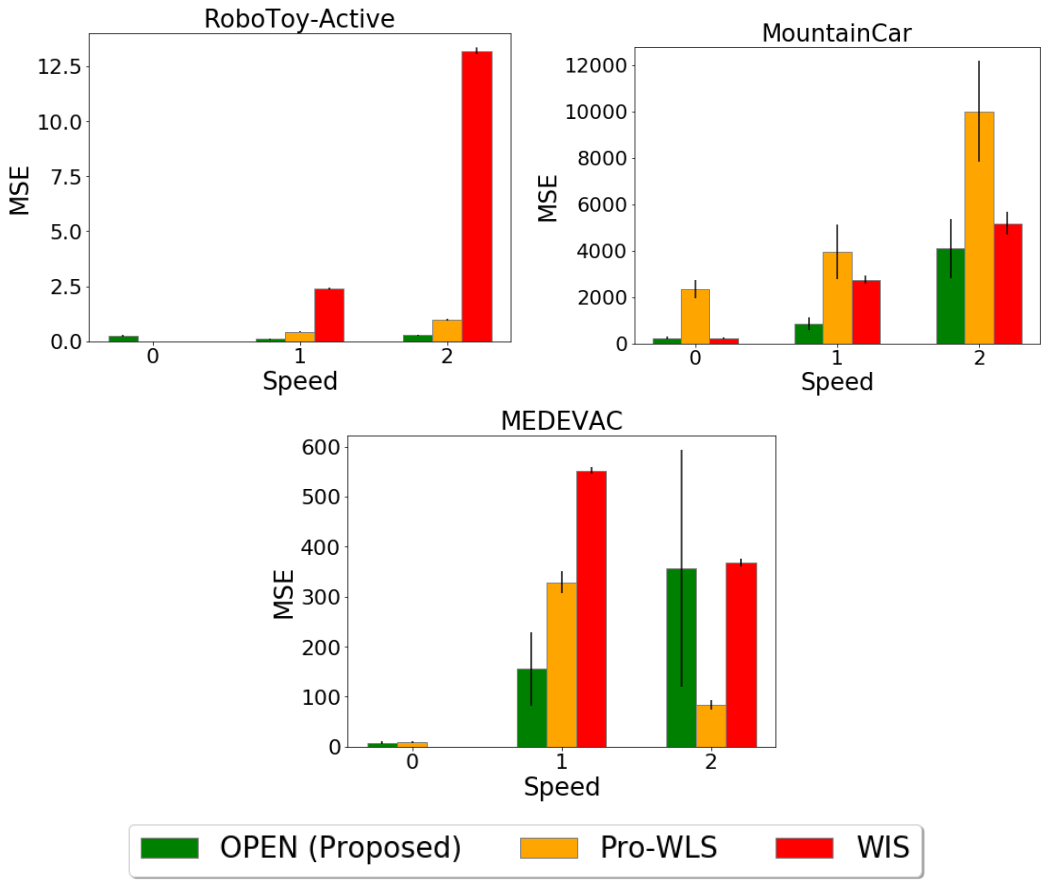


Figure 5.8. Comparison of different algorithms for predicting the future performance of evaluation policy π on domains that exhibit active/hybrid non-stationarity. On the x-axis is the speed which corresponds to the rate of non-stationarity; higher speed indicates faster rate of change and a speed of zero indicates stationary domain. On the y-axis is the mean squared error (MSE) in the performance estimate **(lower is better)**. For each domain, for each speed, for each algorithm, 30 trials were executed. Discussions for these plots can be found in Section 5.7.4.2.2.

As expected, the baseline method WIS that ignores the non-stationarity completely fails to capture the change in performances over time. Therefore, while WIS works well for the stationary setting, as the non-stationarity increase, the biased incurred by WIS grows.

In comparison, the baseline method Pro-WLS that can only account for passive non-stationarity captures the trend better than WIS, but still performs poorly in comparison to the proposed method OPEN that is explicitly designed to handle active/hybrid non-stationarity.

It is not surprising that when it is known that the domain is stationary, OPEN method may not be as reliable as WIS. We observe that for the RoboToy-Active domain OPEN performs the worst for the stationary setting. However, it performs the best for the MountainCar domain. We believe this good performance is just an artifact of this domain. WIS is known to be biased for finite samples, and in this particular setting, it happens to have higher bias than OPEN.

5.7.4.2.2 MSE Analysis Similarly, Figure 5.8 presents the mean-squared error (MSE) incurred by different algorithms when predicting the future performance of the evaluation policy π.

As MSE can be broken down in terms of bias and variance, these plots incorporate the impact of the variance of each estimator as well. For the RoboToy-Active domain, OPEN performs the best, followed by Pro-WLS, and WIS. While OPEN still performs well in terms of MSE for MountainCar, we observe that Pro-WLS does not. Since the bias of Pro-WLS was less than that of WIS for this domain, higher MSE for Pro-WLS can be attributed to high variance. To understand this, recall that Pro-WLS makes use of Fourier basis based parametric regression and uses extrapolation to forecast. Non-autoregressive extrapolation is not bound to follow any structure outside the support of observed data, and can thus result in unreliable estimates leading to high variance.

For the MEDEVAC domain, we observe that OPEN performs well for speeds of 0 and 1, but despite having the lowest bias for speed=2, OPEN has relatively higher MSE.

5.7.5 Results for Passive Non-stationarity

While the primary focus of this chapter was to develop methods to handle active/hybrid non-stationarity, we observed that the proposed method OPEN also provides benefits over the earlier algorithm Pro-WLS even when it is known that there is only passive non-stationarity in the environment.

5.7.5.1 Single Run

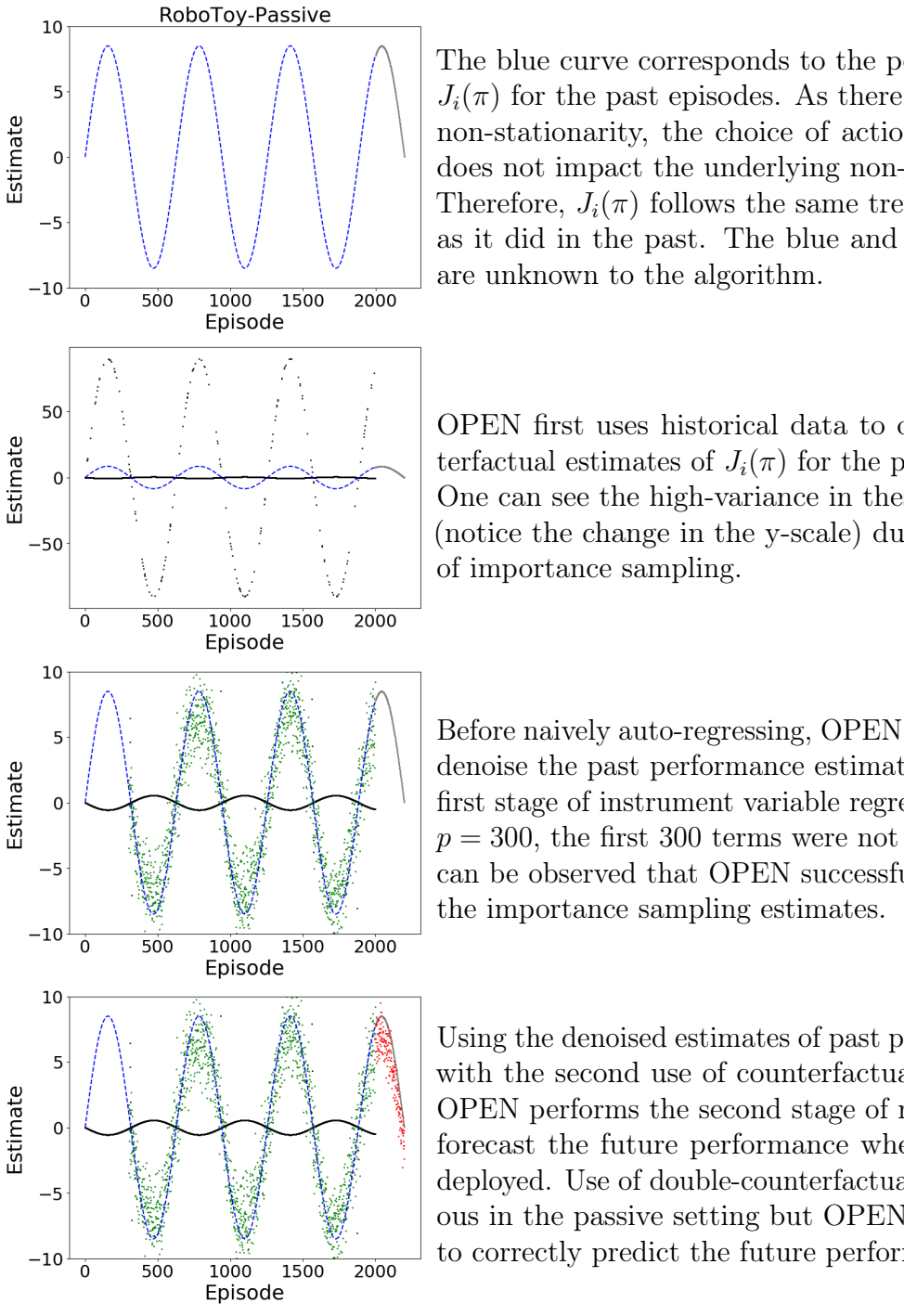
Similar to Figure 5.6, in Figure 5.9 we present a step by step breakdown of the intermediate stages of a single run of OPEN on the RoboToy-Passive domain. Here the trend in how the performance of the evaluation policy was changing in the past remains the same in the future. When only passive non-stationarity is present, the double counter-factual correction performed by OPEN is superfluous. However, it can be observed that OPEN can still correctly identify the trend and provide useful predictions of π's future performance.

5.7.5.2 Summary Plots

In Figure 5.10 we provide bias and MSE analysis of different algorithms on the domains that exhibit passive non-stationarity. Except for the stationary setting, where WIS has the best performance overall, we observe that for all other settings in the plot, OPEN performs better than both Pro-WLS and WIS consistently.

One thing that particularly stands out in these plots is the poor performance of Pro-WLS, despite being designed for the passive setting. We observed that because of the choice of parametric regression using the Fourier basis, Pro-WLS tends to suffer from high bias when the number of Fourier terms is not sufficient to model

the underlying trend. Also, if the number of Fourier terms is increased naively, then they overfit the data and extrapolate poorly, thereby resulting in high-variance. In contrast, our method is based on an auto-regressive based time-series forecast that is more robust to the model choice (we kept the number of lag terms for auto-regression as $p = 300$ for OPEN for all our experiments).



The blue curve corresponds to the performances $J_i(\pi)$ for the past episodes. As there is no active non-stationarity, the choice of actions executed does not impact the underlying non-stationarity. Therefore, $J_i(\pi)$ follows the same trend in future as it did in the past. The blue and gray curves are unknown to the algorithm.

OPEN first uses historical data to obtain counterfactual estimates of $J_i(\pi)$ for the past episodes. One can see the high-variance in these estimates (notice the change in the y-scale) due to the use of importance sampling.

Before naively auto-regressing, OPEN first aims to denoise the past performance estimates using the first stage of instrument variable regression. Since $p = 300$, the first 300 terms were not denoised. It can be observed that OPEN successfully denoises the importance sampling estimates.

Using the denoised estimates of past performances, with the second use of counterfactual reasoning, OPEN performs the second stage of regression to forecast the future performance when π will be deployed. Use of double-counterfactual is superfluous in the passive setting but OPEN is still able to correctly predict the future performance.

Figure 5.9. An illustrative step by step breakdown of the stages in the proposed algorithm OPEN for the RoboToy-Passive domain.

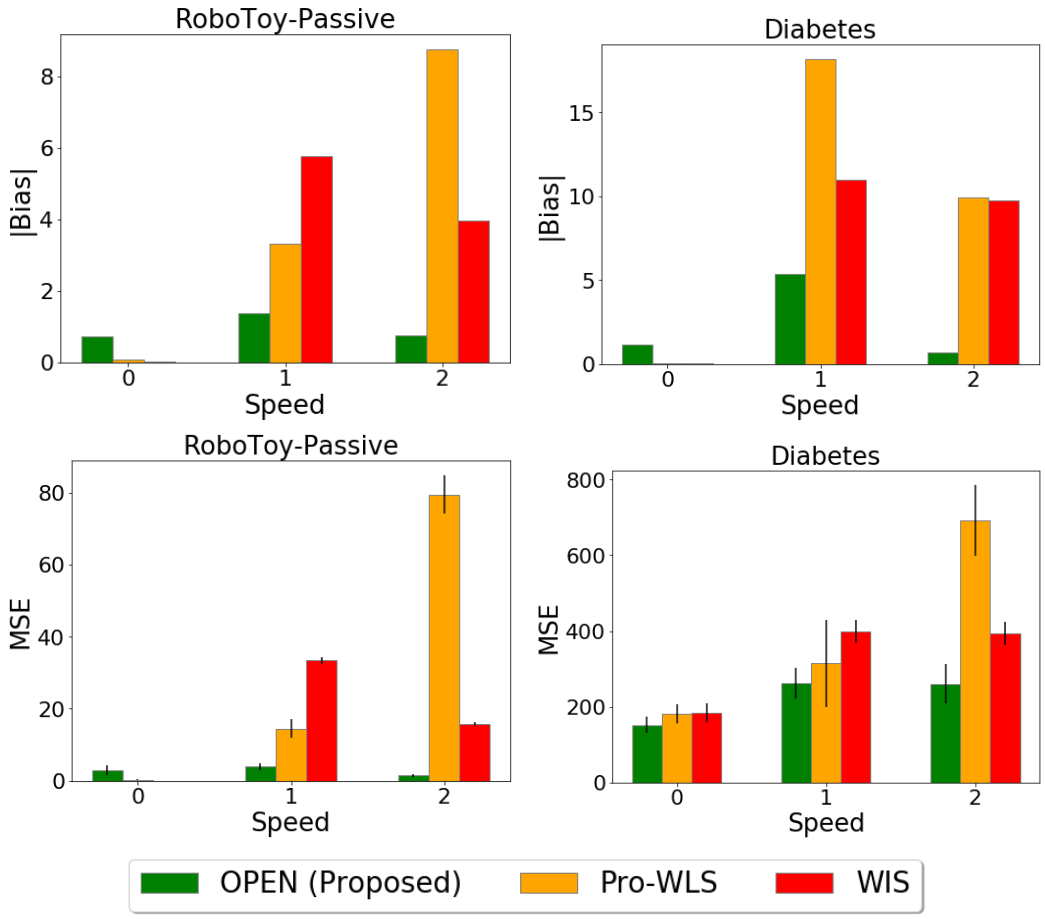


Figure 5.10. Comparison of different algorithms for predicting the future performance of evaluation policy π on domains that exhibit passive non-stationarity. On the x-axis is the speed, which corresponds to the rate of non-stationarity; higher speed indicates a faster rate of change and a speed of zero indicates a stationary domain. **(TOP)** On the y-axis is the absolute bias in the performance estimate. **(Bottom)** On the y-axis is the mean squared error (MSE) in the performance estimate. **Lower is better** for all of these plots. For each domain, for each speed, for each algorithm, 30 trials were executed. Discussion of these plots can be found in Section 5.7.5.

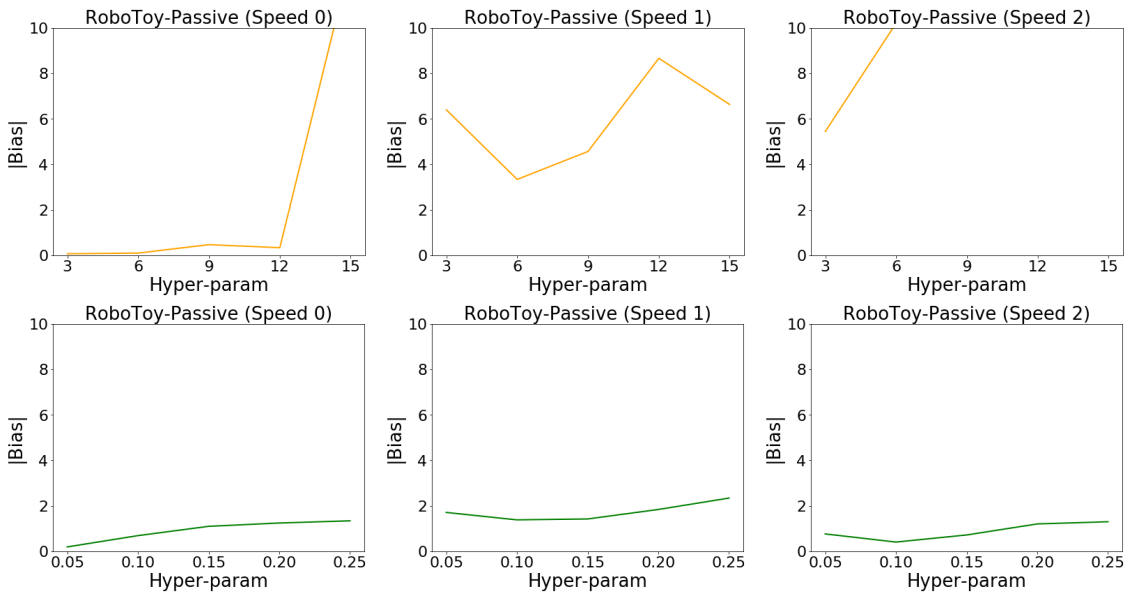


Figure 5.11. (Top) Absolute bias in prediction of Pro-WLS for different choices of its hyper-parameter. **(Bottom)** Absolute bias in prediction of OPEN for different choices of its hyper-parameter. For all the plots, lower value is better. Overall, we observe that OPEN being an auto-regressive method can extrapolate/forecast better and is thus more robust to hyper-parameters than Pro-WLS that uses Fourier bases for regression and is not as good for extrapolation.

5.7.6 Ablation Study

In this section we study the sensitivity to hyper-parameters for the proposed method OPEN and the baseline method Pro-WLS (Chandak et al., 2020b). The hyper-parameter for OPEN corresponds to the number of past terms to condition on for auto-regression, as discussed in Remark 3. The hyper-parameter for Pro-WLS corresponds to the order of Fourier bases required for parametric regression. In Figure 5.11 we present the results for how the performance of the methods vary for different choices of hyper-parameters.

5.8 Conclusion

We took the first steps towards addressing the fundamental question of off-policy policy evaluation under the presence of hybrid non-stationarity. Towards this goal we discussed the challenges associated with the availability of what is effectively a single lifelong sequence of interaction, and the need for structural assumptions to address this challenge. Finally, using a structural assumption, we developed a model-free based procedure that provides more accurate forecasts of future outcomes in the presence of active, passive, and hybrid, non-stationarity. We believe that our method may not only help in the identification of policies that may be actively causing harm or damage, but may also enable *control* of non-stationary processes in the future.

5.9 Limitations and Future Work

It is worth mentioning that our method builds upon techniques from RL, counterfactual reasoning, and auto-regressive time-series, and thus inherits their limitations,

- Use of trajectory based importance sampling results in high variance off-policy estimates of $J_i(\pi)$. While we developed WIS like methods for variance reduction, it is unclear how to theoretically characterize the amount of bias it induces in the finite samples regime. One future way to trade-off bias and variance might be to mix trajectory based importance sampling with marginalized importance sampling (Yuan et al., 2021).

- We made use of instrument variable regression to denoise performance estimates of the past. This relied on our choice of instrument variables, which need to be correlated with the performance we aim to denoise. While the instrument variable technique causes no bias asymptotically, if the instruments variables are not strongly correlated, they can significantly increase variance. An important step for future work will be to take multiple possible IVs and use correlation tests to detect which IVs might be better suited for the task at hand.

- Once the denoised estimates of past performances are available, we use an auto-regressive time-series model to forecast the expected future performance. Currently, we use a fixed number of past terms. In the future, we aim to explore heuristics to set this value adaptively and to investigate other deep models like LSTMs, which have a longer memory.

5.10 Proofs

5.10.1 Double Counterfactual Reasoning

Theorem 1. *Under Assumptions 5 and 6, $\forall m \in \mathcal{M}$ such that the performance $J(\pi)$ associated with m is j,*

$$\mathbb{E}_{\pi}\left[J_{i+1}(\pi) | J_i(\pi) = j\right] = \mathbb{E}_{\beta_i, \beta_{i+1}}\left[\rho_i \widehat{J}_{i+1}(\pi) \middle| M_i = m\right].$$

Proof. In the following, to make the dependence of trajectories explicit, we will additionally define $\rho(h)$ and $g(h)$ to be the importance ratios and the return associated with a trajectory h. Using this notation, it can be observed that,

$$
\begin{aligned}
\mathbb{E}_\pi\left[J_{i+1}(\pi)|M_i\right] &= \sum_{h_{i+1}} \Pr(h_{i+1}|M_i;\pi) g(h_{i+1}) \\
&\overset{(a)}{=} \sum_{h_{i+1}} \sum_{m_{i+1}} \sum_{h_i} \Pr(h_{i+1}, m_{i+1}, h_i|M_i;\pi) g(h_{i+1}) \\
&\overset{(b)}{=} \sum_{h_i} \Pr(h_i|M_i;\pi) \sum_{m_{i+1}} \Pr(m_{i+1}|h_i, M_i;\pi) \\
&\quad \sum_{h_{i+1}} \Pr(h_{i+1}|m_{i+1}, h_i, M_i;\pi) g(h_{i+1}) \\
&\overset{(c)}{=} \sum_{h_i} \Pr(h_i|M_i;\pi) \sum_{m_{i+1}} \Pr(m_{i+1}|h_i, M_i) \sum_{h_{i+1}} \Pr(h_{i+1}|m_{i+1};\pi) g(h_{i+1}) \\
&\overset{(d)}{=} \sum_{h_i} \rho(h_i) \Pr(h_i|M_i;\beta_k) \sum_{m_{i+1}} \Pr(m_{i+1}|h_i, M_i) \\
&\quad \sum_{h_{i+1}} \rho(h_{i+1}) \Pr(h_{i+1}|m_{i+1};\beta_{i+1}) g(h_{i+1}) \\
&\overset{(e)}{=} \sum_{h_i} \sum_{m_{i+1}} \sum_{h_{i+1}} \Pr(h_i|M_i;\beta_i) \Pr(m_{i+1}|h_i, M_i) \Pr(h_{i+1}|m_{i+1};\beta_{i+1}) \Big[\rho(h_i)\rho(h_{i+1}) g(h_{i+1}) \Big] \\
&= \mathbb{E}_{\beta_i\beta_{i+1}}\left[\rho_i \rho_{i+1} G_{i+1}|M_i\right] \\
&= \mathbb{E}_{\beta_i\beta_{i+1}}\left[\rho_i \widehat{J}_{i+1}(\pi)|M_i\right], \qquad (5.12)
\end{aligned}
$$

where (a) follows from the law of total probability, (b) follows from the chain rule of probability, (c) follows using conditional independence, (d) follows from the use of importance sampling to switch the sampling distribution under Assumption 6, and (e) follows from re-arrangement of terms. Finally, ρ_i and ρ_{i+1} are the random variables corresponding the importance ratios in episodes i and $i+1$. The random variable G_{i+1} corresponds to the return under β in episode $i+1$.

Now notice that

$$\begin{aligned}\mathbb{E}_\pi\left[J_{i+1}(\pi)|J_i(\pi)\right] &= \mathbb{E}_M[\mathbb{E}_\pi[J_{i+1}(\pi)|M]|J_i(\pi)] \\ &\overset{(f)}{=} \mathbb{E}_\pi[J_{i+1}(\pi)|M_i] \\ &\overset{(g)}{=} \mathbb{E}_{\beta_i\beta_{i+1}}\left[\rho_i\widehat{J}_{i+1}(\pi)|M_i\right]. \end{aligned} \tag{5.13}$$

To see why (f) holds, let M' be the subsequent POMDP observed after interacting with a POMDP M using π. Let $J(\pi)$ and $J'(\pi)$ be the performances of the policy π in POMDP M and M', respectively. Then $\forall m \in \mathcal{M}$ such that the performance of π in m is equal to $J(\pi)$, it follows from Assumption 5 (where ϕ is the policy performance) that $\mathbb{E}_\pi[J'(\pi)|M = m]$ is a fixed value. Now, M_i is a natural choice for such an m where the performance of π is $J_i(\pi)$. Finally, (g) follows from (5.12).

□

Similarly, under a more generalized Assumption 5, when $\forall m_k \in \mathcal{M}$ such that the statistics associated with $(m_k)_{k=i-p}^{i}$ are $(\phi_k)_{k=i-p}^{i}$, respectively, there exists $\mathcal{F} : \Phi^p \times \Pi \to \Delta(\Phi)$, such that

$$\forall i > p, \quad \mathcal{F}((\phi_k)_{i=i-p}^{i}, \pi')(\phi_{i+1}) = \Pr(\phi_{i+1}|M_i = m_i; \pi'),$$

then similar steps as earlier can be used to conclude that

$$\mathbb{E}_\pi\left[J_{i+1}(\pi)|(J_{i-k}(\pi))_{k=0}^{p}\right] = \mathbb{E}_{\beta_i\beta_{i+1}}\left[\rho_i\widehat{J}_{i+1}(\pi)|M_i\right]. \tag{5.14}$$

Note that no additional importance correction is needed in (5.14) compared to (5.13). The term ρ_i only shows up to correct for the transition between M_i and M_{i+1} due to the meta-transition function $\mathcal{T}(m, h, m') = \Pr(M_{i+1}{=}m'|M_i{=}m, H_i{=}h)$. This independence on the choice of p also holds if $\mathcal{T}$ is non-Markovian in the previous M_i values. Although, additional importance correction would be required if $\mathcal{T}$ is dependent on multiple past H_i terms.

5.10.2 Importance-Weighted IV-Regression

Theorem 2. *Under Assumptions 5 and 6,*

$$\forall i, \quad \text{Cov}\left(\widehat{J}_{i-1}(\pi), \widehat{J}_i(\pi) - J_i(\pi)\right) = 0.$$

Proof.

$$\forall i,\ \text{Cov}\left(\widehat{J}_i(\pi), \widehat{J}_{i+1}(\pi) - J_{i+1}(\pi)\right) = \underbrace{\mathbb{E}_\beta\left[\widehat{J}_i(\pi)\left(\widehat{J}_{i+1}(\pi) - J_{i+1}(\pi)\right)\right]}_{\text{(I)}} - \underbrace{\mathbb{E}_\beta\left[\widehat{J}_i(\pi)\right]\mathbb{E}_\beta\left[\widehat{J}_{i+1}(\pi) - J_{i+1}(\pi)\right]}_{\text{(II)}}. \quad (5.15)$$

Focusing on term (II),

$$\begin{aligned}\mathbb{E}_\beta\left[\widehat{J}_i(\pi)\right]\mathbb{E}_\beta\left[\widehat{J}_{i+1}(\pi) - J_{i+1}(\pi)\right] &= \mathbb{E}_\beta\left[\widehat{J}_i(\pi)\right]\left(\mathbb{E}_\beta\left[\widehat{J}_{i+1}(\pi)\right] - J_{i+1}(\pi)\right) \\ &\overset{(a)}{=} \mathbb{E}_\beta\left[\widehat{J}_i(\pi)\right](J_{i+1}(\pi) - J_{i+1}(\pi)) \\ &= 0,\end{aligned}$$

where (a) follows from the fact that under Assumption 6, $\widehat{J}_{i+1}(\pi)$ is an unbiased estimator for $J_{i+1}(\pi)$ (Thomas, 2015). Focusing on term (I) and using the law of total expectation,

$$\mathbb{E}_\beta\left[\widehat{J}_i(\pi)\left(\widehat{J}_{i+1}(\pi) - J_{i+1}(\pi)\right)\right] = \mathbb{E}_\beta\Big[\widehat{J}_i(\pi)\underbrace{\mathbb{E}_\beta\left[\widehat{J}_{i+1}(\pi) - J_{i+1}(\pi)\middle|\widehat{J}_i(\pi)\right]}_{\text{(III)}}\Big].$$

Expanding term (III) further using the law of total expectation,

$$\begin{aligned}\mathbb{E}_\beta\left[\widehat{J}_{i+1}(\pi) - J_{i+1}(\pi)\middle|\widehat{J}_i(\pi)\right] &\overset{(b)}{=} \mathbb{E}_\beta\left[\mathbb{E}_\beta\left[\widehat{J}_{i+1}(\pi) - J_{i+1}(\pi)\middle|M_{i+1}, \widehat{J}_i(\pi)\right]\middle|\widehat{J}_i(\pi)\right] \\ &\overset{(c)}{=} \mathbb{E}_\beta\left[\mathbb{E}_\beta\left[\widehat{J}_{i+1}(\pi) - J_{i+1}(\pi)\middle|M_{i+1}\right]\middle|\widehat{J}_i(\pi)\right] \\ &\overset{(d)}{=} 0,\end{aligned}$$

where in (b) the outer expectation is over the next environment M_{i+1} given that the current performance estimate is $\widehat{J}_i(\pi)$ and that β_i was used for interaction in episode i. The inner expectation is over $\widehat{J}_{i+1}(\pi)$ and the trajectory used for estimating $\widehat{J}_{i+1}(\pi)$ is collected using β in the environment M_{i+1}. Step (c) follows from the fact that conditioned on the environment M_{i+1}, interactions in M_{i+1} are independent of quantities observed in the episodes before $i+1$. Finally, step (d) follows from observing that

$$\begin{aligned}\mathbb{E}_\beta\left[\widehat{J}_{i+1}(\pi) - J_{i+1}(\pi)\middle|M_{i+1}\right] &= \mathbb{E}_\beta\left[\widehat{J}_{i+1}(\pi)\middle|M_{i+1}\right] - J_{i+1}(\pi) \\ &\overset{(e)}{=} J_{i+1}(\pi) - J_{i+1}(\pi) \\ &= 0,\end{aligned}$$

where (e) follows from the fact that under Assumption 6, $\widehat{J}_{i+1}(\pi)$ is an unbiased estimator of the performance of π for the given environment M_{i+1}. Therefore both (a) and (b) in (5.15) are zero, and we conclude the result. □

Theorem 3. *Under Assumptions 5, 6, and 7, if f and g are linear functions of their inputs, then $\hat{\theta}_n$ is a strongly consistent estimator of θ_π, i.e.,*

$$\hat{\theta}_n \xrightarrow{a.s.} \theta_\pi.$$

Proof. For the linear setting, $\hat{\theta}_n$ can be expressed as,

$$\hat{\phi}_n \in \underset{\phi \in \Phi}{\operatorname{argmin}} \sum_{i=2}^{n-1} \left(\widehat{J}_{i-1}(\pi)\phi - \widehat{J}_i(\pi) \right)^2 \tag{5.16}$$

$$\hat{\theta}_n \in \underset{\theta \in \Theta}{\operatorname{argmin}} \sum_{i=2}^{n-1} \left(\bar{J}_i(\pi)\theta - \rho_i \widehat{J}_{i+1}(\pi) \right)^2, \tag{5.17}$$

where $\bar{J}_i \coloneqq \widehat{J}_{i-1}\hat{\phi}_n$.

Before moving further, we introduce some additional notation.

$$\begin{aligned}
\mathbf{X_{n-2}} &\coloneqq \left[\widehat{J}_1(\pi), ..., \widehat{J}_{n-2}(\pi) \right]^\top, & \mathbf{\Lambda_{n-2}} &\coloneqq \texttt{diag}([\rho_1, ..., \rho_{n-2}]), \\
\mathbf{X_{n-1}} &\coloneqq \left[\widehat{J}_2(\pi), ..., \widehat{J}_{n-1}(\pi) \right]^\top, & \mathbf{\Lambda_{n-1}} &\coloneqq \texttt{diag}\left([\rho_2, ..., \rho_{n-1}]\right), \\
\mathbf{X_n} &\coloneqq \left[\widehat{J}_3(\pi), ..., \widehat{J}_n(\pi) \right]^\top, & \mathbf{\bar{X}_{n-1}} &\coloneqq \left[\bar{J}_2(\pi), ..., \bar{J}_{n-1}(\pi) \right]^\top,
\end{aligned}$$

where `diag` corresponds to a diagonal matrix with off-diagonals set to zero.

In the following, we split the proof into two parts: (a) we will first show that

$$\hat{\theta}_n = \left(\mathbf{X_{n-2}^\top X_{n-1}} \right)^{-1} \left(\mathbf{X_{n-2}^\top \Lambda_{n-1} X_n} \right),$$

and then (b) using this simplified form for $\hat{\theta}_n$ we will show that $\hat{\theta}_n \xrightarrow{\text{a.s.}} \theta_\pi$.

5.10.2.0.1 Part (a) Solving (5.16) in matrix form,

$$\hat{\phi}_n = \left(\mathbf{X_{n-2}^\top X_{n-2}} \right)^{-1} \mathbf{X_{n-2}^\top X_{n-1}}. \tag{5.18}$$

Similarly, solving (5.17) in matrix form,

$$\hat{\theta}_n = \left(\mathbf{\bar{X}_{n-1}^\top \bar{X}_{n-1}} \right)^{-1} \mathbf{\bar{X}_{n-1}^\top \Lambda_{n-1} X_n}. \tag{5.19}$$

Now substituting the value of $\bar{\mathbf{X}}_{\mathbf{n-1}}$ into (5.19),

$$\hat{\theta}_n = \left(\left(\underbrace{\mathbf{X_{n-2}}\hat{\phi}_{\mathbf{n}}}_{\bar{\mathbf{X}}_{\mathbf{n-1}}} \right)^{\top} \left(\underbrace{\mathbf{X_{n-2}}\hat{\phi}_{\mathbf{n}}}_{\bar{\mathbf{X}}_{\mathbf{n-1}}} \right) \right)^{-1} \left(\underbrace{\mathbf{X_{n-2}}\hat{\phi}_{\mathbf{n}}}_{\bar{\mathbf{X}}_{\mathbf{n-1}}} \right)^{\top} \mathbf{\Lambda_{n-1}X_n}. \tag{5.20}$$

Using (5.18) to substitute the value of $\hat{\phi}_n$ into (5.20),

$$\hat{\theta}_n = \left(\left(\mathbf{X_{n-2}} \underbrace{\left(\mathbf{X_{n-2}^{\top}X_{n-2}}\right)^{-1}\mathbf{X_{n-2}^{\top}X_{n-1}}}_{\hat{\phi}_{\mathbf{n}}} \right)^{\top} \left(\mathbf{X_{n-2}} \underbrace{\left(\mathbf{X_{n-2}^{\top}X_{n-2}}\right)^{-1}\mathbf{X_{n-2}^{\top}X_{n-1}}}_{\hat{\phi}_{\mathbf{n}}} \right) \right)^{-1}$$

$$\left(\mathbf{X_{n-2}} \underbrace{\left(\mathbf{X_{n-2}^{\top}X_{n-2}}\right)^{-1}\mathbf{X_{n-2}^{\top}X_{n-1}}}_{\hat{\phi}_{\mathbf{n}}} \right)^{\top} \mathbf{\Lambda_{n-1}X_n}. \tag{5.21}$$

Using matrix operations to expand the transposes in (5.21),

$$\hat{\theta}_n = \left(\left(\mathbf{X_{n-1}^{\top}X_{n-2}} \left(\mathbf{X_{n-2}^{\top}X_{n-2}}\right)^{-1} \mathbf{X_{n-2}^{\top}} \right) \left(\mathbf{X_{n-2}} \left(\mathbf{X_{n-2}^{\top}X_{n-2}}\right)^{-1} \mathbf{X_{n-2}^{\top}X_{n-1}} \right) \right)^{-1}$$
$$\left(\mathbf{X_{n-1}^{\top}X_{n-2}} \left(\mathbf{X_{n-2}^{\top}X_{n-2}}\right)^{-1} \mathbf{X_{n-2}^{\top}} \right) \mathbf{\Lambda_{n-1}X_n}. \tag{5.22}$$

Similarly, using matrix operations to expand inverses in (5.22) (colored underlines are used to match the terms before expansion in (5.22) and after expansion in (5.23)),

$$\hat{\theta}_n = \left(\mathbf{X_{n-2}^{\top}X_{n-1}}\right)^{-1} \left(\mathbf{X_{n-2}^{\top}X_{n-2}}\right) \left(\mathbf{X_{n-2}^{\top}X_{n-2}}\right)^{-1} \left(\mathbf{X_{n-2}^{\top}X_{n-2}}\right) \left(\mathbf{X_{n-1}^{\top}X_{n-2}}\right)^{-1}$$

$$\left(\mathbf{X_{n-1}^{\top}X_{n-2}}\right) \left(\mathbf{X_{n-2}^{\top}X_{n-2}}\right)^{-1} \left(\mathbf{X_{n-2}^{\top}\Lambda_{n-1}X_n}\right). \tag{5.23}$$

Notice that several terms in (5.23) cancel each other out, therefore,

$$\hat{\theta}_n = \left(\mathbf{X_{n-2}^{\top}X_{n-1}}\right)^{-1} \left(\mathbf{X_{n-2}^{\top}\Lambda_{n-1}X_n}\right). \tag{5.24}$$

5.10.2.0.2 Part (b) Now recall from (5.5) that when f is a linear function,

$$J_{i+1}(\pi) = J_i(\pi)\theta_\pi + U_{i+1}(H_i),$$

where U_{i+1} is bounded mean zero noise (which depends on the interaction H_i by π). Using Theorem 1, let $Y_{i+1} \coloneqq \mathbb{E}_\pi\left[J_{i+1}(\pi)|J_i(\pi)\right]$ and its unbiased estimate be

$$\widehat{Y}_{i+1} \coloneqq \rho_i \widehat{J}_{i+1}(\pi) = \rho_i \rho_{i+1} G_{i+1}. \tag{5.25}$$

For the regression, since $\widehat{J}_i(\pi)$ is an unbiased estimate of the input $J_i(\pi)$ and $\widehat{Y}_{i+1}$ is an unbiased estimate of the target $\mathbb{E}_\pi\left[J_{i+1}(\pi)|J_i(\pi)\right]$, these can be equivalently expressed as,

$$\begin{aligned}\widehat{J}_i(\pi) &= J_i(\pi) + V_i(H_i),\\ \widehat{Y}_{i+1} &= J_{i+1}(\pi) + W_{i+1}(H_i, H_{i+1}),\end{aligned}$$

where $V_i(H_i)$ is some bounded mean-zero noise (dependent on the unbiased estimate made using H_i) and $W_{i+1}(H_i, H_{i+1})$ is also a bounded mean-zero noise (dependent on the unbiased estimate made using H_i and H_{i+1}). Before moving further, we define some additional notation,

$$\begin{aligned}\mathbf{Y_n} &\coloneqq [Y_3, ..., Y_n]^\top & \mathbf{U_n} &\coloneqq [U_3(H_2), ..., U_n(H_{n-1})]^\top\\ \widehat{\mathbf{Y}}_\mathbf{n} &\coloneqq [\widehat{Y}_3, ..., \widehat{Y}_n]^\top & \mathbf{V_{n-1}} &\coloneqq [V_2(H_2), ..., V_{n-1}(H_{n-1})]^\top\\ \mathbb{J}_\mathbf{n-1} &\coloneqq [J_2(\pi), ..., J_{n-1}(\pi)]^\top & \mathbf{W_n} &\coloneqq [W_3(H_2, H_3), ..., W_n(H_{n-1}, H_n)]^\top.\end{aligned}$$

Using (5.25) note that $\widehat{\mathbf{Y}}_\mathbf{n} = \mathbf{\Lambda_{n-1}} \mathbf{X_n}$, therefore (5.24) can be expressed as,

$$\hat{\theta}_n = \left(\mathbf{X}_\mathbf{n-2}^\top \mathbf{X_{n-1}}\right)^{-1} \left(\mathbf{X}_\mathbf{n-2}^\top \widehat{\mathbf{Y}}_\mathbf{n}\right). \tag{5.26}$$

Unrolling the value of $\widehat{\mathbf{Y}}_\mathbf{n}$ in (5.26) using relations from (5.25),

$$\begin{aligned}\hat{\theta}_n &= \left(\mathbf{X_{n-2}^\top X_{n-1}}\right)^{-1}\left(\mathbf{X_{n-2}^\top}\left(\mathbf{Y_n}+\mathbf{W_n}\right)\right)\\ &= \left(\mathbf{X_{n-2}^\top X_{n-1}}\right)^{-1}\left(\mathbf{X_{n-2}^\top}\left(\mathbb{J}_{\mathbf{n-1}}\theta_\pi+\mathbf{U_n}+\mathbf{W_n}\right)\right)\\ &= \left(\mathbf{X_{n-2}^\top X_{n-1}}\right)^{-1}\left(\mathbf{X_{n-2}^\top}\left(\left(\mathbf{X_{n-1}}-\mathbf{V_{n-1}}\right)\theta_\pi+\mathbf{U_n}+\mathbf{W_n}\right)\right). \qquad (5.27)\end{aligned}$$

Expanding (5.27),

$$\hat{\theta}_n = \theta_\pi - \left(\mathbf{X_{n-2}^\top X_{n-1}}\right)^{-1}\mathbf{X_{n-2}^\top V_{n-1}}\theta_\pi + \left(\mathbf{X_{n-2}^\top X_{n-1}}\right)^{-1}\left(\mathbf{X_{n-2}^\top}\left(\mathbf{U_n}+\mathbf{W_n}\right)\right) \qquad (5.28)$$

Evaluating the value of (5.28) in the limit,

$$\lim_{n\to\infty}\hat{\theta}_n = \theta_\pi - \lim_{n\to\infty}\left(\underbrace{\left(\mathbf{X_{n-2}^\top X_{n-1}}\right)^{-1}\mathbf{X_{n-2}^\top V_{n-1}}\theta_\pi}_{(a)} + \underbrace{\left(\mathbf{X_{n-2}^\top X_{n-1}}\right)^{-1}\left(\mathbf{X_{n-2}^\top}\left(\mathbf{U_n}+\mathbf{W_n}\right)\right)}_{(b)}\right). \qquad (5.29)$$

It can be now seen from (5.29) that if in the limit the terms inside the paranthesis are zero, then we would obtain our desired result. Focusing on the term (a) and using the continuous mapping theorem,

$$\begin{aligned}\lim_{n\to\infty}\left(\mathbf{X_{n-2}^\top X_{n-1}}\right)^{-1}\mathbf{X_{n-2}^\top V_{n-1}}\theta_\pi &= \lim_{n\to\infty}\left(\frac{1}{n}\mathbf{X_{n-2}^\top X_{n-1}}\right)^{-1}\left(\frac{1}{n}\mathbf{X_{n-2}^\top V_{n-1}}\theta_\pi\right)\\ &= \left(\lim_{n\to\infty}\frac{1}{n}\mathbf{X_{n-2}^\top X_{n-1}}\right)^{-1}\left(\underbrace{\lim_{n\to\infty}\frac{1}{n}\mathbf{X_{n-2}^\top V_{n-1}}}_{(c)}\right)\theta_\pi. \qquad (5.30)\end{aligned}$$

Notice that term (c) (5.30) can be expressed as $\frac{1}{n}\sum_{i=2}^{n-1}X_{i-1}V_i$. Further, recall from Theorem 2 that V_i is a mean zero random variable uncorrelated with X_{i-1} for all i.

Further, V_i and X_{i-1} are also bounded for all i as both the rewards and importance ratios are bounded (Assumption 6), and T is finite. Now, for $\alpha_i \coloneqq X_{i-1}V_i$ observe that $\mathbb{E}[\alpha_i] = \mathbb{E}[X_{i-1}\mathbb{E}[V_i|X_{i-1}]] = \mathbb{E}[X_{i-1}0] = 0$ and thus α_i is a bounded and mean zero random variable $\forall i$. Therefore, as (c) is an average of α variables, it follows from Rajchaman's strong law of large numbers for uncorrelated random variables (Rajchman, 1932; Chandra, 1991) that term (c) is zero almost surely. Thus,

$$\left(\mathbf{X}_{\mathbf{n-2}}^{\top}\mathbf{X}_{\mathbf{n-1}}\right)^{-1}\mathbf{X}_{\mathbf{n-2}}^{\top}\mathbf{V}_{\mathbf{n-1}}\theta_{\pi} \xrightarrow{\text{a.s.}} \mathbf{0}.$$

Similarly, for term (b) in (5.29) observe that both U_n and W_n are zero mean random variables uncorrelated with X_{n-2}. Therefore, term (b) in (5.29) is also zero in the limit almost surely. It can now be concluded from (5.29) that

$$\hat{\theta}_n \xrightarrow{\text{a.s.}} \theta_{\pi}.$$

□

Theorem 4. *Under Assumptions 5, 6, and 7, if f and g are linear functions of their inputs, then $\tilde{\theta}_n$ is a strongly consistent estimator of θ_{π}, i.e.,*

$$\tilde{\theta}_n \xrightarrow{a.s.} \theta_{\pi}.$$

Proof. For the linear setting, $\tilde{\theta}_n$ can be expressed as,

$$\hat{\phi}_n \in \underset{\phi\in\Phi}{\operatorname{argmin}} \sum_{i=2}^{n-1} \rho_i \left(\widehat{J}_{i-1}(\pi)\phi - G_i(\pi)\right)^2. \tag{5.31}$$

$$\tilde{\theta}_n \in \underset{\theta\in\Theta}{\operatorname{argmin}} \sum_{i=2}^{n-1} \rho_i\rho_{i+1} \left(\bar{J}_i(\pi)\theta - G_{i+1}(\pi)\right)^2, \qquad \text{where} \quad \bar{J}_i \coloneqq \widehat{J}_{i-1}\hat{\phi}_n. \tag{5.32}$$

Notice that as dividing the objective by a positive constant does not change the optima, we drop the denominator terms in

$$\bar{\rho}_i \coloneqq \frac{\rho_i}{(\sum_{j=2}^{n-1} \rho_j)} \frac{\rho_{i+1}}{(\sum_{k=2}^{n-1} \rho_{k+1})}$$

for the purpose of the analysis. Before moving further, we introduce some additional notation that extends to notation introduced in the proof of Theorem 3:

$$\mathbf{G_n} \coloneqq [G_3, ..., G_n]^\top \qquad \bar{\mathbf{\Lambda}}_{\mathbf{n-1}} \coloneqq \mathtt{diag}([\rho_2\rho_3, \rho_3\rho_4 ..., \rho_{n-1}\rho_n]).$$

Solving (5.31) in matrix form,

$$\begin{aligned}\hat{\phi}_n &= \left(\mathbf{X}_{\mathbf{n-2}}^\top \mathbf{\Lambda_{n-1}} \mathbf{X_{n-2}}\right)^{-1} \mathbf{X}_{\mathbf{n-2}}^\top \mathbf{\Lambda_{n-1}} \mathbf{G_{n-1}}. \\ &= \left(\mathbf{X}_{\mathbf{n-2}}^\top \mathbf{\Lambda_{n-1}} \mathbf{X_{n-2}}\right)^{-1} \mathbf{X}_{\mathbf{n-2}}^\top \mathbf{X_{n-1}}.\end{aligned}$$

Similarly, solving (5.32) in matrix form,

$$\begin{aligned}\tilde{\theta}_n &= \left(\bar{\mathbf{X}}_{\mathbf{n-1}}^\top \bar{\mathbf{\Lambda}}_{\mathbf{n-1}} \bar{\mathbf{X}}_{\mathbf{n-1}}\right)^{-1} \bar{\mathbf{X}}_{\mathbf{n-1}}^\top \bar{\mathbf{\Lambda}}_{\mathbf{n-1}} \mathbf{G_n}. \\ &\overset{(a)}{=} \left(\bar{\mathbf{X}}_{\mathbf{n-1}}^\top \bar{\mathbf{\Lambda}}_{\mathbf{n-1}} \bar{\mathbf{X}}_{\mathbf{n-1}}\right)^{-1} \bar{\mathbf{X}}_{\mathbf{n-1}}^\top \mathbf{\Lambda_{n-1}} \mathbf{X_n}, \end{aligned} \tag{5.33}$$

where (a) follows from the fact that $\rho_i\rho_{i+1}G_{i+1} = \rho_i \widehat{J}_{i+1}(\pi)$. Now substituting the value of $\bar{\mathbf{X}}_{\mathbf{n-1}}$ into (5.33) similar to (5.20) and (5.21) in the proof of Theorem 3,

$$\tilde{\theta}_n = \left(\left(\mathbf{X}_{\mathbf{n-1}}^\top \mathbf{X_{n-2}} \left(\mathbf{X}_{\mathbf{n-2}}^\top \mathbf{\Lambda_{n-1}} \mathbf{X_{n-2}}\right)^{-1} \mathbf{X}_{\mathbf{n-2}}^\top\right) \bar{\mathbf{\Lambda}}_{\mathbf{n-1}} \left(\mathbf{X_{n-2}} \left(\mathbf{X}_{\mathbf{n-2}}^\top \mathbf{\Lambda_{n-1}} \mathbf{X_{n-2}}\right)^{-1} \mathbf{X}_{\mathbf{n-2}}^\top \mathbf{X_{n-1}}\right)\right)^{-1}$$

$$\left(\mathbf{X}_{\mathbf{n-1}}^\top \mathbf{X_{n-2}} \left(\mathbf{X}_{\mathbf{n-2}}^\top \mathbf{X_{n-2}}\right)^{-1} \mathbf{X}_{\mathbf{n-2}}^\top\right) \mathbf{\Lambda_{n-1}} \mathbf{X_n}. \tag{5.34}$$

Similarly, using matrix operations to expand inverses in (5.34) (colored underlines are used to match the terms before expansion in (5.34) and after expansion in (5.35)) and multiplying and dividing by n,

$$\tilde{\theta}_n = \left(\underline{\mathbf{X}_{\mathbf{n-2}}^\top \mathbf{X}_{\mathbf{n-1}}}\right)^{-1} \left(\frac{1}{n}\underline{\mathbf{X}_{\mathbf{n-2}}^\top \mathbf{\Lambda}_{\mathbf{n-1}} \mathbf{X}_{\mathbf{n-2}}}\right) \left(\frac{1}{n}\underline{\mathbf{X}_{\mathbf{n-2}}^\top \bar{\mathbf{\Lambda}}_{\mathbf{n-1}} \mathbf{X}_{\mathbf{n-2}}}\right)^{-1} \left(\frac{1}{n}\underline{\mathbf{X}_{\mathbf{n-2}}^\top \mathbf{\Lambda}_{\mathbf{n-1}} \mathbf{X}_{\mathbf{n-2}}}\right)$$

$$\left(\underline{\mathbf{X}_{\mathbf{n-1}}^\top \mathbf{X}_{\mathbf{n-2}}}\right)^{-1} \quad \left(\mathbf{X}_{\mathbf{n-1}}^\top \mathbf{X}_{\mathbf{n-2}}\right) \left(\frac{1}{n}\mathbf{X}_{\mathbf{n-2}}^\top \mathbf{\Lambda}_{\mathbf{n-1}} \mathbf{X}_{\mathbf{n-2}}\right)^{-1} \left(\mathbf{X}_{\mathbf{n-2}}^\top \mathbf{\Lambda}_{\mathbf{n-1}} \mathbf{X}_{\mathbf{n}}\right). \tag{5.35}$$

Now focusing on the term underlined in green, in the limit,

$$\begin{aligned}
\lim_{n\to\infty} \frac{1}{n}\mathbf{X}_{\mathbf{n-2}}^\top \bar{\mathbf{\Lambda}}_{\mathbf{n-1}} \mathbf{X}_{\mathbf{n-2}} &= \lim_{n\to\infty} \frac{1}{n} \sum_{i=2}^{n-1} \rho_i \rho_{i+1} \widehat{J}_{i-1}(\pi) \widehat{J}_{i-1}(\pi)^\top \\
&\overset{(a)}{=} \lim_{n\to\infty} \frac{1}{n} \sum_{i=2}^{n-1} \mathbb{E}_{\beta_i,\beta_{i+1}}[\rho_i \rho_{i+1}] \widehat{J}_{i-1}(\pi) \widehat{J}_{i-1}(\pi)^\top + \frac{1}{n} \sum_{i=2}^{n-1} \varepsilon_i \widehat{J}_{i-1}(\pi) \widehat{J}_{i-1}(\pi)^\top \\
&\overset{(b)}{=} \lim_{n\to\infty} \frac{1}{n} \sum_{i=2}^{n-1} \widehat{J}_{i-1}(\pi) \widehat{J}_{i-1}(\pi)^\top \\
&= \lim_{n\to\infty} \frac{1}{n} \mathbf{X}_{\mathbf{n-2}}^\top \mathbf{X}_{\mathbf{n-2}},
\end{aligned} \tag{5.36}$$

where in (a) we defined the random variable $\rho_i \rho_{i+1}$ as its expected value $E_{\beta_i,\beta_{i+1}}[\rho_i \rho_{i+1}]$ plus a mean zero noise ε_i. Step (b) follows from first observing that ρ_i and ρ_{i+1} are uncorrelated. Therefore $\mathbb{E}_{\beta_i,\beta_{i+1}}[\rho_i \rho_{i+1}] = \mathbb{E}_{\beta_i}[\rho_i]\mathbb{E}_{\beta_{i+1}}[\rho_{i+1}] = 1$ as the expected value of importance ratios is 1 (Thomas, 2015). Similarly, ε_i is uncorrelated with $\widehat{J}_{i-1}(\pi)$, i.e., the expected value $\mathbb{E}_{\beta_i,\beta_{i+1}}\left[\varepsilon_i | \widehat{J}_{i-1}(\pi)\right] = \mathbb{E}_{\beta_i,\beta_{i+1}}[\varepsilon_i] = 0$ for any given $J_{i-1}(\pi)$. (Intuitively, this step can be seen analogous to the derivation of PDIS, where the expected value of future IS ratios is always one, irrespective of the past events that it has been conditioned on). Now notice that the random variable $\zeta_i \coloneqq \varepsilon_i \widehat{J}_{i-1}(\pi) \widehat{J}_{i-1}(\pi)^\top$ is bounded and has mean zero for all i. Therefore, while ζ_i and ζ_j may be dependent, they

are uncorrelated for all $i \neq j$. Using the strong law of large number for uncorrelated random variables (Rajchman, 1932; Chandra, 1991) the second term in (a) is zero almost surely.

Similarly, it can be observed that $\frac{1}{n}\mathbf{X}_{\mathbf{n-2}}^{\top}\mathbf{\Lambda}_{\mathbf{n-1}}\mathbf{X}_{\mathbf{n-2}}$ converges to $\frac{1}{n}\mathbf{X}_{\mathbf{n-2}}^{\top}\mathbf{X}_{\mathbf{n-2}}$. Therefore using (5.36) in (5.35), and using the continuous mapping theorem,

$$\tilde{\theta}_n \xrightarrow{a.s.} \left(\mathbf{X}_{\mathbf{n-2}}^{\top}\mathbf{X}_{\mathbf{n-1}}\right)^{-1}\left(\frac{1}{n}\mathbf{X}_{\mathbf{n-2}}^{\top}\mathbf{X}_{\mathbf{n-2}}\right)\left(\frac{1}{n}\mathbf{X}_{\mathbf{n-2}}^{\top}\mathbf{X}_{\mathbf{n-2}}\right)^{-1}\left(\frac{1}{n}\mathbf{X}_{\mathbf{n-2}}^{\top}\mathbf{X}_{\mathbf{n-2}}\right)\left(\mathbf{X}_{\mathbf{n-1}}^{\top}\mathbf{X}_{\mathbf{n-2}}\right)^{-1}$$

$$\left(\mathbf{X}_{\mathbf{n-1}}^{\top}\mathbf{X}_{\mathbf{n-2}}\right)\left(\frac{1}{n}\mathbf{X}_{\mathbf{n-2}}^{\top}\mathbf{X}_{\mathbf{n-2}}\right)^{-1}\left(\mathbf{X}_{\mathbf{n-2}}^{\top}\mathbf{\Lambda}_{\mathbf{n-1}}\mathbf{X}_{\mathbf{n}}\right). \tag{5.37}$$

Notice that several terms in (5.37) cancel each other out, therefore,

$$\tilde{\theta}_n \xrightarrow{a.s.} \left(\mathbf{X}_{\mathbf{n-2}}^{\top}\mathbf{X}_{\mathbf{n-1}}\right)^{-1}\left(\mathbf{X}_{\mathbf{n-2}}^{\top}\mathbf{\Lambda}_{\mathbf{n-1}}\mathbf{X}_{\mathbf{n}}\right).$$

This proof can be completed similarly to the part (b) of the proof of Theorem 3. □

CHAPTER 6

CONCLUSION AND FUTURE WORK

Many real-world applications present problems that are not stationary. These applications motivated us to develop tools that do not require the stationarity assumption. Towards this goal, we discussed how this problem may be intractable unless some assumptions about the structure of the non-stationarity are imposed. Subsequently, we made several contributions, culminating with principled methods that can handle active and hybrid non-stationarity while remaining practical even in stationary settings.

In the presence of structured non-stationarity due to only external factors, we presented a procedure named *Prognosticator* that can (a) provide a model-free estimate of the performance of a policy if that policy is deployed in the future, and (b) proactively search for a good future policy through a gradient based procedure that maximizes the *future* performance. Perhaps surprisingly, we observe that *minimizing* performance over some of the data from the past can be beneficial when searching for a policy that *maximizes* future performance. We also show how *Prognosticator* is unbiased and strongly consistent in the stationary setting, thereby generalizing several existing methods for the stationary setting.

Many real-world applications are safety critical and thus require performance and safety guarantees. We formalized the conditions under which safety can be ensured in the presence of structured non-stationarity due to external factors. Under these conditions we propose *SPIN*, the first procedure for safe policy improvement under such non-stationarities. SPIN first constructed asymptotically valid confidence intervals of

a policy's future performance and then searched for a policy that maximized the lower bound obtained from this confidence interval. Empirically, we observed that SPIN provides safe policy improvement even in the finite sample setting and even when the structure resulting from non-stationarity is misspecified. In comparison, existing methods for ensuring safety that do not account for non-stationarity result in up to five times more unsafe behavior than desired.

Finally, we generalized to core idea underlying the previous contributions to account for a more general class of non-stationarity, where the changes may occur due to both external factors and due to the past decisions made by the agent. This setting was particularly challenging as it exposed a completely new feedback loop that allowed an agent to influence the non-stationarity. Under this setting, we formalized the fundamental problem of (off-policy) policy evaluation, established additional assumptions for tractability, and proposed a method, *OPEN*, to address this challenge. With OPEN, we took the first steps towards a unified procedure that can tackle general forms of structured non-stationarities (while remaining effective in the stationary setting).

6.1 Future Work

While our contributions provided some initial steps to address the challenges stemming from non-stationarity, we believe that we have only barely scratched the surface. There are numerous important questions that we have not yet been able to answer.

- **Re-exploration:** In the stationary setting, the de-facto strategy for searching for good policies is to first *explore* and then *exploit*. Unfortunately, this strategy may not be reasonable in the non-stationary setting. As the domain is changing, the rewards and transitions associated with the parts of the domain that an agent might have explored before can change later on. This necessitates careful

re-exploration to understand what aspects are changing and how to pro-actively adapt to those changes. Another interesting future avenue might be to explore such that the collected data enables accurate estimation of the time-series parameters for any policy's performance. If the time series model is well specified, this procedure could potentially mitigate the need for extensive re-exploration.

- **Time-series Model Selection:** An underlying theme across all the contributions was to extract the effect of the underlying non-stationarity on a policy's performance using a time-series model. In our work, we resorted to hyper-parameter tuning to choose the right time-series model. In practice, to achieve more reliability and automation, it would be ideal to have goodness-of-fit based cross-validation tests to choose the time series model.

- **Partial Model of Non-stationarity:** The contributions in this thesis were mostly focused on model-free approaches that do not require access to any known model of the environment, nor do they aim to estimate the environment from the data. While this is useful in the cases where accessing/developing good models of the environment is challenging, there may be better ways to leverage models of the environment when they are available. Particularly, doubly-robust methods (Jiang and Li, 2016) have become increasingly popular to combine (partial-)models with data under the stationarity assumption. One clear direction of future work would be to determine whether doubly robust methods can be extended to the non-stationary setting.

- **Bellman Recursions:** An important drawback of trajectory based importance sampling is that the resulting mean-squared-error can grow exponentially with respect to the horizon (Guo et al., 2017). Since our methods directly build upon these importance sampling methods, our methods inherit this limitation as well.

In the stationary setting, there has been recent work that leverages Bellman recursion for the state visitation distribution to construct importance sampling estimators that mitigate some of the above problems (Yuan et al., 2021). Is it possible to extend these ideas to the non-stationary setting?

- **Controlling Non-stationary Domains** In Chapter 5, we proposed a method to perform policy *evaluation* in the presence of structured passive, active, or hybrid non-stationarity. An important aspect of active/hybrid non-stationarity is the additional feedback loop that governs how the domain itself changes based on past interactions. Harnessing this feedback loop can allow policy *improvement* by *controlling* how the underlying non-stationarity evolves. Performing such policy improvement in a safe and reliable manner also remains an interesting avenue for future work.

BIBLIOGRAPHY

Y. Abbasi, P. L. Bartlett, V. Kanade, Y. Seldin, and C. Szepesvári. Online learning in Markov decision processes with adversarially chosen transition probability distributions. In *Advances in Neural Information Processing Systems*, pages 2508–2516, 2013.

M. Abbott. Instrumental variables (IV) estimation: An introduction, 2007. http://qed.econ.queensu.ca/pub/faculty/abbott/econ481/481note09_f07.pdf.

S. Abdallah and M. Kaisers. Addressing environment non-stationarity by repeating q-learning updates. *The Journal of Machine Learning Research*, 2016.

D. Abel, Y. Jinnai, S. Y. Guo, G. Konidaris, and M. Littman. Policy and value transfer in lifelong reinforcement learning. In *International Conference on Machine Learning*, pages 20–29, 2018.

J. Achiam, D. Held, A. Tamar, and P. Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 22–31. JMLR. org, 2017.

A. Agarwal, M. Henaff, S. Kakade, and W. Sun. Pc-pg: Policy cover directed exploration for provable policy gradient learning. *Advances in Neural Information Processing Systems*, 33:13399–13412, 2020.

M. Al-Shedivat, T. Bansal, Y. Burda, I. Sutskever, I. Mordatch, and P. Abbeel. Continuous adaptation via meta-learning in nonstationary and competitive environments. *arXiv preprint arXiv:1710.03641*, 2017.

L. N. Alegre, A. L. Bazzan, and B. C. da Silva. Minimum-delay adaptation in non-stationary reinforcement learning via online high-confidence change-point detection. *arXiv preprint arXiv:2105.09452*, 2021.

H. B. Ammar, R. Tutunov, and E. Eaton. Safe policy search for lifelong reinforcement learning with sublinear regret. In *International Conference on Machine Learning*, pages 2361–2369, 2015.

E. W. Basso and P. M. Engel. Reinforcement learning in non-stationary continuous time and space scenarios. In *Artificial Intelligence National Meeting*, volume 7, pages 1–8. Citeseer, 2009.

M. Bastani. Model-free intelligent diabetes management using machine learning. Master's thesis, University of Alberta, 2014.

M. F. Bellemare, T. Masaki, and T. B. Pepinsky. Lagged explanatory variables and the estimation of causal effect. *The Journal of Politics*, 79(3):949–963, 2017.

Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300, 1995.

A. Bennett, N. Kallus, and T. Schnabel. Deep generalized method of moments for instrumental variable analysis. *arXiv preprint arXiv:1905.12495*, 2019.

D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1st edition, 1996. ISBN 1886529108.

O. Besbes, Y. Gur, and A. Zeevi. Stochastic multi-armed-bandit problem with non-stationary rewards. In *Advances in Neural Information Processing Systems*, pages 199–207, 2014.

M. Blondel, O. Teboul, Q. Berthet, and J. Djolonga. Fast differentiable sorting and ranking. *arXiv preprint arXiv:2002.08871*, 2020.

P. Bloomfield. *Fourier analysis of time series: An introduction*. John Wiley & Sons, 2004.

M. Bowling. Convergence and no-regret in multiagent learning. In *Advances in Neural Information Processing Systems*, pages 209–216, 2005.

E. Brunskill and L. Li. PAC-inspired option discovery in lifelong reinforcement learning. In *International conference on machine learning*, pages 316–324, 2014.

J. Buckman, C. Gelada, and M. G. Bellemare. The importance of pessimism in fixed-dataset policy optimization. *arXiv preprint arXiv:2009.06799*, 2020.

A. C. Cameron. Instrument variables, 2019. http://cameron.econ.ucdavis.edu/e240a/ch04iv.pdf.

J. Carpenter and J. Bithell. Bootstrap confidence intervals: When, which, what? A practical guide for medical statisticians. *Statistics in Medicine*, 19(9):1141–1164, 2000.

E. Cetin and O. Celiktutan. Learning pessimism for robust and efficient off-policy reinforcement learning. *arXiv preprint arXiv:2110.03375*, 2021.

Y. Chandak, G. Theocharous, C. Nota, and P. S. Thomas. Lifelong learning with a changing action set. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pages 3373–3380, 2020a.

Y. Chandak, G. Theocharous, S. Shankar, S. Mahadevan, M. White, and P. S. Thomas. Optimizing for the future in non-stationary mdps. *International Conference on Machine Learning*, 2020b.

Y. Chandak, G. Theocharous, S. Shankar, M. White, S. Mahadevan, and P. S. Thomas. Optimizing for the future in non-stationary MDPs. *In Proceedings of the 37th International Conference on Machine Learning*, 2020c.

T. K. Chandra. Extensions of rajchman's strong law of large numbers. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 118–121, 1991.

T. Cheevaprawatdomrong, I. E. Schochetman, R. L. Smith, and A. Garcia. Solution and forecast horizons for infinite-horizon nonhomogeneous Markov decision processes. *Mathematics of Operations Research*, 32(1):51–72, 2007.

P. Chen, T. Pedersen, B. Bak-Jensen, and Z. Chen. ARIMA-based time series model of stochastic wind power generation. *IEEE Transactions on Power Systems*, 25(2): 667–676, 2009.

S. X. Chen, W. Härdle, and M. Li. An empirical likelihood goodness-of-fit test for time series. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(3):663–678, 2003.

W. C. Cheung, D. Simchi-Levi, and R. Zhu. Reinforcement learning under drift. *arXiv preprint arXiv:1906.02922*, 2019.

W. C. Cheung, D. Simchi-Levi, and R. Zhu. Drifting reinforcement learning: The blessing of (more) optimism in face of endogenous & exogenous dynamics. *Arxiv. 1906.02922v3*, 2020.

S. P. Choi, D.-Y. Yeung, and N. L. Zhang. An environment model for nonstationary reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 987–993, 2000.

Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh. A Lyapunov-based approach to safe reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 8092–8101, 2018.

M. Cinelli, G. D. F. Morales, A. Galeazzi, W. Quattrociocchi, and M. Starnini. The echo chamber effect on social media. *Proceedings of the National Academy of Sciences*, 118(9), 2021.

V. Conitzer and T. Sandholm. Awesome: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. *Machine Learning*, 67(1-2):23–43, 2007.

M. Cuturi, O. Teboul, and J.-P. Vert. Differentiable ranking and sorting using optimal transport. In *Advances in Neural Information Processing Systems*, pages 6858–6868, 2019.

B. C. Da Silva, E. W. Basso, A. L. Bazzan, and P. M. Engel. Dealing with non-stationary environments using context detection. In *Proceedings of the 23rd international conference on Machine learning*, pages 217–224, 2006.

R. Davidson and E. Flachaire. The wild bootstrap, tamed at last. *Citeseer*, 1999.

R. Davidson and E. Flachaire. The wild bootstrap, tamed at last. *Journal of Econometrics*, 146(1):162–169, 2008.

T. J. DiCiccio and B. Efron. Bootstrap confidence intervals. *Statistical Science*, pages 189–212, 1996.

A. Djogbenou, S. Gonçalves, and B. Perron. Bootstrap inference in regressions with estimated factors and serial correlation. *Journal of Time Series Analysis*, 36(3): 481–502, 2015.

A. A. Djogbenou, J. G. MacKinnon, and M. Ø. Nielsen. Asymptotic theory and wild bootstrap inference with clustered errors. *Journal of Econometrics*, 212(2):393–412, 2019.

F. Doshi-Velez and G. Konidaris. Hidden parameter markov decision processes: A semiparametric regression approach for discovering latent task parametrizations. In *IJCAI: proceedings of the conference*, volume 2016, page 1432. NIH Public Access, 2016.

B. Efron and R. J. Tibshirani. *An introduction to the Bootstrap*. CRC press, 1994.

D. Ernst, G.-B. Stan, J. Goncalves, and L. Wehenkel. Clinical data based optimal sti strategies for hiv: a reinforcement learning approach. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 667–672. IEEE, 2006.

E. Even-Dar, S. M. Kakade, and Y. Mansour. Experts in a Markov decision process. In *Advances in Neural Information Processing Systems*, pages 401–408, 2005.

C. Finn, A. Rajeswaran, S. Kakade, and S. Levine. Online meta-learning. *arXiv preprint arXiv:1902.08438*, 2019.

J. Foerster, R. Y. Chen, M. Al-Shedivat, S. Whiteson, P. Abbeel, and I. Mordatch. Learning with opponent-learning awareness. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 122–130. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

D. J. Foster, Z. Li, T. Lykouris, K. Sridharan, and E. Tardos. Learning in games: Robustness of fast convergence. In *Advances in Neural Information Processing Systems*, pages 4734–4742, 2016.

M. Friedrich, S. Smeekes, and J.-P. Urbain. Autoregressive wild bootstrap inference for nonparametric trends. *Journal of Econometrics*, 214(1):81–109, 2020.

P. Gajane, R. Ortner, and P. Auer. A sliding-window algorithm for Markov decision processes with arbitrarily changing rewards and transitions. *arXiv preprint arXiv:1805.10066*, 2018.

A. Garcia and R. L. Smith. Solving nonstationary infinite horizon dynamic optimization problems. *Journal of Mathematical Analysis and Applications*, 244(2):304–317, 2000.

J. Garcıa and F. Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

A. Ghate and R. L. Smith. A linear programming approach to nonstationary infinite-horizon Markov decision processes. *Operations Research*, 61(2):413–425, 2013.

M. Ghavamzadeh, Y. Engel, and M. Valko. Bayesian policy gradient and actor-critic algorithms. *The Journal of Machine Learning Research*, 17(1):2319–2371, 2016.

N. Gillani, A. Yuan, M. Saveski, S. Vosoughi, and D. Roy. Me, my echo chamber, and i: introspection on social media polarization. In *Proceedings of the 2018 World Wide Web Conference*, pages 823–831, 2018.

L. Godfrey and A. Tremayne. The wild bootstrap and heteroskedasticity-robust tests for serial correlation in dynamic regression models. *Computational Statistics & Data Analysis*, 49(2):377–395, 2005.

W. H. Greene. *Econometric analysis*. Pearson Education India, 2003.

E. Greensmith, P. L. Bartlett, and J. Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(Nov):1471–1530, 2004.

Z. Guo, P. S. Thomas, and E. Brunskill. Using options and covariance testing for long horizon off-policy policy evaluation. In *Advances in Neural Information Processing Systems*, pages 2492–2501, 2017.

T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

H. Hachiya, M. Sugiyama, and N. Ueda. Importance-weighted least-squares probabilistic classifier for covariate shift adaptation with application to human activity recognition. *Neurocomputing*, 80:93–101, 2012.

P. Hall. Unusual properties of bootstrap confidence intervals in regression problems. *Probability Theory and Related Fields*, 81(2):247–273, 1989.

P. Hall. *The bootstrap and Edgeworth expansion*. Springer Science & Business Media, 2013.

J. Hartford, G. Lewis, K. Leyton-Brown, and M. Taddy. Deep iv: A flexible approach for counterfactual prediction. In *International Conference on Machine Learning*, pages 1414–1423. PMLR, 2017.

D. Hennes, D. Morrill, S. Omidshafiei, R. Munos, J. Perolat, M. Lanctot, A. Gruslys, J.-B. Lespiau, P. Parmas, E. Duenez-Guzman, et al. Neural replicator dynamics. *arXiv preprint arXiv:1906.00190*, 2019.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9 (8):1735–1780, 1997.

W. J. Hopp, J. C. Bean, and R. L. Smith. A new optimality criterion for nonhomogeneous Markov decision processes. *Operations Research*, 35(6):875–883, 1987.

A. Jacobsen, M. Schlegel, C. Linke, T. Degris, A. White, and M. White. Meta-descent for online, continual prediction. In *AAAI Conference on Artificial Intelligence*, 2019.

R. Jagerman, I. Markov, and M. de Rijke. When people change their mind: Off-policy evaluation in non-stationary recommendation environments. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 447–455, 2019a.

R. Jagerman, I. Markov, and M. de Rijke. When people change their mind: Off-policy evaluation in non-stationary recommendation environments. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, Melbourne, VIC, Australia, February 11-15, 2019*, 2019b.

N. Jiang and L. Li. Doubly robust off-policy value evaluation for reinforcement learning. *arXiv preprint arXiv:1511.03722*, 2015.

N. Jiang and L. Li. Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 652–661. PMLR, 2016.

N. K. Jong and P. Stone. Bayesian models of nonstationary Markov decision processes. *Planning and Learning in A Priori Unknown or Dynamic Domains*, page 132, 2005.

S. Jordan, Y. Chandak, D. Cohen, M. Zhang, and P. Thomas. Evaluating the performance of reinforcement learning algorithms. In *International Conference on Machine Learning*, pages 4962–4973. PMLR, 2020.

S. M. Jordan, D. Cohen, and P. S. Thomas. Using cumulative distribution based performance analysis to benchmark models. In *NeurIPS 2018 Workshop on Critiquing and Correcting Trends in Machine Learning*, 2018.

L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

S. Kakade and J. Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2002.

A. Kazerouni, M. Ghavamzadeh, Y. A. Yadkori, and B. Van Roy. Conservative contextual linear bandits. In *Advances in Neural Information Processing Systems*, pages 3910–3919, 2017.

A. Kearney, V. Veeriah, J. B. Travnik, R. S. Sutton, and P. M. Pilarski. TIDBD: Adapting temporal-difference step-sizes through stochastic meta-descent. *arXiv preprint arXiv:1804.03334*, 2018.

M. Kearns and S. Singh. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 2002.

K. Khetarpal, M. Riemer, I. Rish, and D. Precup. Towards continual reinforcement learning: A review and perspectives. *arXiv preprint arXiv:2012.13490*, 2020.

P. Kline and A. Santos. Higher order properties of the wild bootstrap under misspecification. *Journal of Econometrics*, 171(1):54–70, 2012.

R. Laroche, P. Trichelair, and R. T. d. Combes. Safe policy improvement with baseline bootstrapping. *arXiv preprint arXiv:1712.06924*, 2017.

E. Lecarpentier and E. Rachelson. Non-stationary Markov decision processes, a worst-case approach using model-based reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 7214–7223, 2019.

E. Lecarpentier, D. Abel, K. Asadi, Y. Jinnai, E. Rachelson, and M. L. Littman. Lipschitz lifelong reinforcement learning. *arXiv preprint arXiv:2001.05411*, 2020.

N. Levine, K. Crammer, and S. Mannor. Rotting bandits. In *Advances in Neural Information Processing Systems*, pages 3074–3083, 2017.

C. Li and M. de Rijke. Cascading non-stationary bandits: Online learning to rank in the non-stationary cascade model. *arXiv preprint arXiv:1905.12370*, 2019.

Y. Li, A. Zhong, G. Qu, and N. Li. Online Markov decision processes with time-varying transition probabilities and rewards. In *Real-world Sequential Decision Making workshop at ICML 2019*, 2019.

R. Liu, Z. Shang, and G. Cheng. On deep instrumental variables estimate. *arXiv preprint arXiv:2004.14954*, 2020.

R. Y. Liu et al. Bootstrap procedures under some non-iid models. *The Annals of Statistics*, 16(4):1696–1708, 1988.

K. Lu, I. Mordatch, and P. Abbeel. Adaptive online planning for continual lifelong learning. *arXiv preprint arXiv:1912.01188*, 2019.

J. G. MacKinnon. Inference based on the wild bootstrap. In *Seminar presentation given to Carleton University in September*, 2012.

A. R. Mahmood, H. Van Hasselt, and R. S. Sutton. Weighted importance sampling for off-policy learning with linear function approximation. In *NIPS*, pages 3014–3022, 2014.

M. Mahmud and S. Ramamoorthy. Learning in non-stationary mdps as transfer learning. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 1259–1260. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

E. Mammen. Bootstrap and wild bootstrap for high dimensional linear models. *The Annals of Statistics*, pages 255–285, 1993.

C. D. Man, F. Micheletto, D. Lv, M. Breton, B. Kovatchev, and C. Cobelli. The UVA/PADOVA type 1 diabetes simulator: New features. *Journal of Diabetes Science and Technology*, 8(1):26–34, 2014.

R. Mealing and J. L. Shapiro. Opponent modelling by sequence prediction and lookahead in two-player games. In *International Conference on Artificial Intelligence and Soft Computing*, pages 385–396. Springer, 2013.

B. Metevier, S. Giguere, S. Brockman, A. Kobren, Y. Brun, E. Brunskill, and P. S. Thomas. Offline contextual bandits with high probability fairness guarantees. In *Advances in Neural Information Processing Systems*, pages 14893–14904, 2019.

M. Mohri and S. Yang. Accelerating online convex optimization via adaptive prediction. In *Artificial Intelligence and Statistics*, pages 848–856, 2016.

A. W. Moore. Efficient memory-based learning for robot control. 1990.

B. L. Moore, L. D. Pyeatt, V. Kulkarni, P. Panousis, K. Padrez, and A. G. Doufas. Reinforcement learning for closed-loop propofol anesthesia: A study in human volunteers. *The Journal of Machine Learning Research*, 15(1):655–696, 2014.

E. Moulines. On upper-confidence bound policies for non-stationary bandit problems. *arXiv preprint arXiv:0805.3415*, 2008.

A. Nagabandi, I. Clavera, S. Liu, R. S. Fearing, P. Abbeel, S. Levine, and C. Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. *arXiv preprint arXiv:1803.11347*, 2018a.

A. Nagabandi, C. Finn, and S. Levine. Deep online learning via meta-learning: Continual adaptation for model-based rl. *arXiv preprint arXiv:1812.07671*, 2018b.

M. Ornik and U. Topcu. Learning and planning for time-varying mdps using maximum likelihood estimation. *arXiv preprint arXiv:1911.12976*, 2019.

S. Padakandla. A survey of reinforcement learning algorithms for dynamically varying environments. *arXiv preprint arXiv:2005.10619*, 2020.

J. A. Parker. Endogenous regressors and instrumental variables, 2020. https://www.reed.edu/economics/parker/312/notes/Notes11.pdf.

J. Pearl et al. Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress*, 19, 2000.

J. Peters and S. Schaal. Natural actor-critic. *Neurocomputing*, 71(7-9):1180–1190, 2008.

M. Pirotta, M. Restelli, A. Pecorino, and D. Calandriello. Safe policy iteration. In *International Conference on Machine Learning*, pages 307–315, 2013.

R. Poiani, A. Tirinzoni, and M. Restelli. Meta-reinforcement learning by tracking task non-stationarity. *arXiv preprint arXiv:2105.08834*, 2021.

D. Precup. Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, page 80, 2000.

M. L. Puterman. Markov decision processes. *Handbooks in operations research and management science*, 2:331–434, 1990.

A. Rajchman. Zaostrzone prawo wielkich liczb. *Mathesis Polska*, 6:145–161, 1932.

A. Rakhlin and K. Sridharan. Online learning with predictable sequences. *arXiv preprint arXiv:1208.3728*, 2013.

B. Ravindran and A. G. Barto. Approximate homomorphisms: A framework for non-exact minimization in Markov decision processes. In *Proceedings of the Fifth International Conference on Knowledge Based Computer Systems*, 2004.

M. B. Ring. *Continual learning in reinforcement environments*. PhD thesis, University of Texas at Austin, Texas 78712, 1994.

M. J. Robbins, P. R. Jenkins, N. D. Bastian, and B. J. Lunday. Approximate dynamic programming for the aeromedical evacuation dispatching problem: Value function approximation utilizing multiple level aggregation. *Omega*, 91:102020, 2020.

P. R. Rosenbaum and D. B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

S. Saria. Individualized sepsis treatment using reinforcement learning. *Nature medicine*, 24(11):1641–1642, 2018.

J. Schmidhuber. A general method for incremental self-improvement and multi-agent learning. In *Evolutionary Computation: Theory and Applications*, pages 81–123. World Scientific, 1999.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

J. Seznec, A. Locatelli, A. Carpentier, A. Lazaric, and M. Valko. Rotting bandits are no harder than stochastic ones. *arXiv preprint arXiv:1811.11043*, 2018.

S. Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

S. Singh, M. Kearns, and Y. Mansour. Nash convergence of gradient dynamics in general-sum games. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 541–548. Morgan Kaufmann Publishers Inc., 2000.

S. Sinha and A. Ghate. Policy iteration for robust nonstationary Markov decision processes. *Optimization Letters*, 10(8):1613–1628, 2016.

G. Strang, G. Strang, G. Strang, and G. Strang. *Introduction to linear algebra*, volume 3. Wellesley-Cambridge Press Wellesley, MA, 1993.

Student. The probable error of a mean. *Biometrika*, pages 1–25, 1908.

R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT Press, Cambridge, MA, 2 edition, 2018a.

R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018b.

A. A. Taiga, W. Fedus, M. C. Machado, A. Courville, and M. G. Bellemare. On bonus-based exploration methods in the arcade learning environment. *arXiv preprint arXiv:2109.11052*, 2021.

G. Theocharous, P. S. Thomas, and M. Ghavamzadeh. Personalized ad recommendation systems for life-time value optimization with guarantees. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

G. Theocharous, Y. Chandak, P. S. Thomas, and F. de Nijs. Reinforcement learning for strategic recommendations. *arXiv preprint arXiv:2009.07346*, 2020.

P. Thomas and E. Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 2139–2148, 2016.

P. Thomas, G. Theocharous, and M. Ghavamzadeh. High confidence policy improvement. In *International Conference on Machine Learning*, pages 2380–2388, 2015a.

P. Thomas, G. Theocharous, and M. Ghavamzadeh. High-confidence off-policy evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015b.

P. S. Thomas. *Safe reinforcement learning*. PhD thesis, University of Massachusetts Libraries, 2015.

P. S. Thomas, G. Theocharous, and M. Ghavamzadeh. High-confidence off-policy evaluation. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015c.

P. S. Thomas, G. Theocharous, M. Ghavamzadeh, I. Durugkar, and E. Brunskill. Predictive off-policy policy evaluation for nonstationary decision problems, with applications to digital marketing. In *AAAI*, pages 4740–4745, 2017.

P. S. Thomas, B. Castro da Silva, A. G. Barto, S. Giguere, Y. Brun, and E. Brunskill. Preventing undesirable behavior of intelligent machines. *Science vol. 366*, pages 999–1004, 2019a.

P. S. Thomas, B. C. da Silva, A. G. Barto, S. Giguere, Y. Brun, and E. Brunskill. Preventing undesirable behavior of intelligent machines. *Science*, 366(6468):999–1004, 2019b.

S. Thrun. Lifelong learning algorithms. In *Learning to learn*, pages 181–209. Springer, 1998.

N. Wagener, C.-A. Cheng, J. Sacks, and B. Boots. An online learning approach to model predictive control. *arXiv preprint arXiv:1902.08967*, 2019.

J.-K. Wang, X. Li, and P. Li. Optimistic adaptive acceleration for optimization. *arXiv preprint arXiv:1903.01435*, 2019a.

L. Wang, H. Zhou, B. Li, L. R. Varshney, and Z. Zhao. Be aware of non-stationarity: Nearly optimal algorithms for piecewise-stationary cascading bandits. *arXiv preprint arXiv:1909.05886*, 2019b.

W. Z. Wang, A. Shih, A. Xie, and D. Sadigh. Influencing towards stable multi-agent interactions. *arXiv preprint arXiv:2110.08229*, 2021.

Y. Wang and M. F. Bellemare. Lagged variables as instruments, 2019.

L. Wasserman. *All of statistics: A concise course in statistical inference*. Springer Science & Business Media, 2013.

W. Whitt. Approximations of dynamic programs, i. *Mathematics of Operations Research*, 3(3):231–243, 1978.

V. Wieland and M. Wolters. Forecasting and policy making. In *Handbook of Economic Forecasting*, volume 2, pages 239–325. Elsevier, 2013.

A. S. Wilkins. To lag or not to lag?: Re-evaluating the use of lagged dependent variables in regression analysis. *Political Science Research and Methods*, 6(2):393–411, 2018.

C.-F. J. Wu et al. Jackknife, bootstrap and other resampling methods in regression analysis. *The Annals of Statistics*, 14(4):1261–1295, 1986.

D. Wu, X. Chen, X. Yang, H. Wang, Q. Tan, X. Zhang, J. Xu, and K. Gai. Budget constrained bidding by model-free reinforcement learning in display advertising. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1443–1451, 2018.

A. Xie, J. Harrison, and C. Finn. Deep reinforcement learning amidst lifelong non-stationarity. *arXiv preprint arXiv:2006.10701*, 2020a.

A. Xie, D. P. Losey, R. Tolsma, C. Finn, and D. Sadigh. Learning latent representations to influence multi-agent interaction. *arXiv preprint arXiv:2011.06619*, 2020b.

J. Xie. *Simglucose v0.2.1 (2018)*, 2019. URL https://github.com/jxx123/simglucose.

T. Xie, Y. Ma, and Y.-X. Wang. Towards optimal off-policy evaluation for reinforcement learning with marginalized importance sampling. *arXiv preprint arXiv:1906.03393*, 2019.

L. Xu, Y. Chen, S. Srinivasan, N. de Freitas, A. Doucet, and A. Gretton. Learning deep features in instrumental variable regression. *arXiv preprint arXiv:2010.07154*, 2020.

S. Yang and M. Mohri. Optimistic bandit convex optimization. In *Advances in Neural Information Processing Systems*, pages 2297–2305, 2016.

C. Yu, J. Liu, and S. Nemati. Reinforcement learning in healthcare: A survey. *arXiv:1908.08796*, 2019.

J. Y. Yu and S. Mannor. Online learning in Markov decision processes with arbitrarily changing rewards and transitions. In *2009 International Conference on Game Theory for Networks*, pages 314–322. IEEE, 2009.

C. Yuan, Y. Chandak, S. Giguere, P. S. Thomas, and S. Niekum. SOPE: Spectrum of off-policy estimators. *Advances in Neural Information Processing Systems*, 34, 2021.

C. Zhang and V. Lesser. Multi-agent learning with policy prediction. In *Twenty-fourth AAAI conference on artificial intelligence*, 2010.

J. Zhang and K. Cho. Query-efficient imitation learning for end-to-end autonomous driving. *arXiv preprint arXiv:1605.06450*, 2016.