Efficient methods in counterfactual policy learning and sequential decision making

Houssam Zenati

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

École doctorale : MSTII - Mathématiques, Sciences et technologies de l'information, Informatique
Spécialité : Mathématiques et Informatique
Unité de recherche : Laboratoire Jean Kuntzmann

Méthodes efficaces en apprentissage contrefactuel de politiques et prise de décisions séquentielles

Efficient methods in counterfactual policy learning and sequential decision making

Présentée par :

Houssam ZENATI

Direction de thèse :

Julien MAIRAL DIRECTEUR DE RECHERCHE HDR, INRIA CENTRE GRENOBLE-RHONE-ALPES	Directeur de thèse
Eustache DIEMERT CHERCHEUR, CRITEO	Co-encadrant de thèse
PIERRE GAILLARD CHARGE DE RECHERCHE, INRIA CENTRE GRENOBLE-RHONEALPES	Co-encadrant de thèse

Rapporteurs :

OLIVIER CAPPE
DIRECTEUR DE RECHERCHE HDR, CNRS PARIS CENTRE
NICOLO CESA-BIANCHI
PROFESSEUR DES UNIVERSITES, UNIVERSITE DE MILAN

Thèse soutenue publiquement le **21 septembre 2023**, devant le jury composé de :

JULIEN MAIRAL DIRECTEUR DE RECHERCHE HDR, INRIA CENTRE GRENOBLE-RHONE-ALPES	Directeur de thèse
OLIVIER CAPPE DIRECTEUR DE RECHERCHE HDR, CNRS PARIS CENTRE	Rapporteur
CLAIRE VERNADE CHERCHEUSE, UNIVERSITE DE TÜBINGEN	Examinatrice
MASSIH-REZA AMINI PROFESSEUR DES UNIVERSITES, UNIVERSITE GRENOBLE ALPES	Président
NICOLO CESA-BIANCHI PROFESSEUR DES UNIVERSITES, UNIVERSITE DE MILAN	Rapporteur

Invités :

EUSTACHE DIEMERT
CHERCHEUR, CRITEO
PIERRE GAILLARD
CHARGE DE RECHERCHE, INRIA CENTRE GRENOBLE-RHONEALPES

Abstract

Because logged data has become ubiquitous in wide-range applications and since online exploration may be sensitive, counterfactual methods have gained significant attention in the recent decade (Bottou et al., 2013). Such data come in the form of an observational dataset where partial feedback information is associated to context covariates and actions taken by a logging decision policy. The aim of counterfactual policy methods is then to learn a policy that improves upon that logging policy based on the observational data solely. While many applications require a discrete action setting, less attention has been given to continuous action spaces that are however widespread in online auction problems (Nedelec et al., 2022). In that sense, developing algorithms with guarantees that work well in these practical settings, as well as enlarging benchmark datasets represents an important research direction that has been a focus of this thesis. We introduce subsequently a method for continuous action policies along with a new CoCoA benchmark dataset. Moreover, we investigate the use of optimization approaches related to the counterfactual risk minimization learning objective function and propose a novel estimator that is more amenable to gradient based optimization.

Likewise, counterfactual learning methods typically use inverse propensity scoring estimators (Horvitz and Thompson, 1952) that are prone to variance issues (Owen, 2013). The latter is even more seen in cases where the past decisions (in the collected data) underexplored the action space. As such, an offline analysis may not suffice to undertake statistically plausible decisions; collecting additional data to increase the sample size may be necessary. In that sense, sequential designs of adaptively collected data should allow to improve the performance of counterfactual policy learning in terms of convergence guarantees and in practical settings. We investigate this direction in this thesis by proposing a novel estimator with improved variance-dependent convergence guarantees which in turn allow to obtain fast rates under an assumption that is similar to Holderian error bounds used in restart strategies for accelerated optimization (d'Aspremont et al., 2021).

Conversely, when online exploration is possible, a rich literature has been built (Lattimore and Szepesvári, 2020) to design effective online policies in contextual bandits. In that case, the Optimistim in the Face of Uncertainty Learning (OFUL) principle (Abbasi-yadkori et al., 2011) has been instrumental in obtaining algorithms with sublinear regret rates and especially practical performances. While seminal methods use linear assumptions on the form of the reward (Li et al., 2010; Chu et al., 2011), nonlinear embeddings of kernel methods (Shawe-Taylor and Cristianini, 2004) provide richer representations of the data that allow for controlled regret guarantees and improved performances in applications. However, such kernel methods suffer from scalability issues as they become computationally intensive when the number of decision steps increases. As such, we investigate in this thesis the use of kernel approximation methods (Smola and Schölkopf, 2000; Williams and Seeger,

2001) in the contextual bandit task to derive an efficient implementation of the Kernel UCB method (Valko et al., 2013). We analyze the regret and explicit in which kernel approximation regimes we manage to restore the original regret rate while obtaining faster computations.

Eventually, in sequential learning (Bubeck, 2011), an agent can be called to choose between arms in a set of alternatives and thereof develop a randomized strategy in adversarial settings (Cesa-Bianchi and Lugosi, 2006). However, in some applications the learner has to choose between a large number of alternatives of which many possess inherent similarities which may be implied by closely correlated losses. In that case, a naive learning agent may suffer unnecessary regret and conversely, an agent that would benefit from side information on a similarity structure may obtain improved performances. This thesis brings contributions with regards to a class of adversarial multi-armed bandit problems with novel algorithms on learning with expert advice and a nested exponential weights algorithms that performs a layered exploration of the learner nested set of alternatives.

Résumé

Étant donné que les données "loggées" sont devenues omniprésentes dans de nombreuses applications et que l'exploration en ligne peut être sensible, les méthodes contrefactuelles ont suscité un intérêt significatif au cours des dernières années (Bottou et al., 2013). Ces données se trouvent sous la forme d'un jeu de données observationnelles où des informations partielles de renforcement sont associées à des covariables contextuelles et aux actions prises par une politique de décision de "logging". Le but de ces méthodes d'apprentissage contrefactuel de politique est dès lors d'apprendre une politique qui améliore la politique initiale en utilisant seulement ces données observationnelles. Bien que de nombreuses applications nécessitent un espace d'action discret, un intérêt moindre a été accordé aux méthodes avec espaces d'action continus qui sont cependant présents dans des problèmes d'enchères en ligne (Nedelec et al., 2022). Aussi, le développement d'algorithmes avec des garanties théoriques qui fonctionnent dans des problèmes pratiques, ainsi que l'élargissement des données de référence en source ouverte représente une direction de recherche importante qui a été un objet de cette thèse. Nous présentons par la suite une méthode pour les politiques d'action continues ainsi qu'un nouvel ensemble de données de référence, le jeu de données CoCoA. De plus, nous étudions l'utilisation de méthodes d'optimisation liées à la nature de la fonction objective d'apprentissage en minimisation de risque contrefactuel et proposons un nouvel estimateur qui est plus adapté à l'optimisation basée sur des gradients.

Par ailleurs, les méthodes d'apprentissage contrefactuel utilisent généralement des estimateurs de pondération de propension inverse (Horvitz and Thompson, 1952) qui sont sujets à des problèmes de variance (Owen, 2013). Ce dernier est encore plus prononcé dans les cas où les décisions passées (dans les données collectées) ont sous-exploré l'espace d'action. Par conséquent, une analyse hors ligne peut ne pas suffire pour prendre des décisions statistiquement plausibles ; il peut être nécessaire de collecter des données supplémentaires pour augmenter la taille de l'échantillon. Ainsi, les conceptions séquentielles de collection de données de manière adaptative devraient permettre d'améliorer les performances de l'apprentissage contrefactuel de politique en termes de garanties de convergence mais également en pratique. Nous explorons cette direction dans cette thèse en proposant un nouvel estimateur avec des garanties de convergence améliorées qui permettent à leur tour d'obtenir des taux rapides sous une hypothèse similaire à celle des bornes d'erreur de Holder dans les stratégies de redémarrage dans les méthodes d'optimisation accélérée (d'Aspremont et al., 2021).

Inversement, lorsque l'exploration en ligne est possible, une littérature abondante a été élaborée (Lattimore and Szepesvári, 2020) pour concevoir des politiques en ligne efficaces dans les problèmes de bandits contextuels. Dans ce cas, le principe d'Optimisme Face à l'Incertitude de l'Apprentissage (Abbasi-yadkori et al., 2011) a été déterminant pour obtenir

des algorithmes avec des taux de regret sous-linéaires et des performances particulièrement satisfaisantes dans des problèmes pratiques. Alors que des premières méthodes pour ce problème requièrent des hypothèses de linéarité sur la forme de la fonction de renforcement (Li et al., 2010; Chu et al., 2011), les representations non linéaires des méthodes à noyau (Shawe-Taylor and Cristianini, 2004) permettent d'obtenir des représentations de données plus riches qui à leur tour fournissent des garanties de regret et des performances améliorées dans un grand nombre d'applications. Cependant, de telles méthodes à noyau souffrent de problèmes de scalabilité car elles deviennent couteuses en termes de ressources de calcul lorsque le nombre d'étapes de décision augmente. Nous étudions donc dans cette thèse l'utilisation de méthodes d'approximation à noyau (Smola and Schölkopf, 2000; Williams and Seeger, 2001) dans ce problème de bandit contextuel pour proposer une implémentation efficace de la méthode UCB à noyau (Valko et al., 2013). Nous analysons le regret et explicitons les régimes dans lesquels l'approximation des méthodes à noyau permet de restaurer le taux de regret original tout en obtenant des calculs plus rapides.

Enfin, en apprentissage séquentiel (Bubeck, 2011), un agent peut être appelé à choisir entre des actions dans un ensemble d'alternatives et développer une stratégie aléatoire dans des environnements adversariaux (Cesa-Bianchi and Lugosi, 2006). Cependant, dans certaines applications, l'apprenant doit choisir entre un grand nombre d'alternatives dont beaucoup présentent des similarités inhérentes qui peuvent être induites par des coûts étroitement corrélées. Dans ce cas, un agent d'apprentissage naïf peut souffrir d'un regret inutile et inversement, un agent qui bénéficierait d'informations annexes sur une structure de similarité devrait obtenir des performances améliorées. Cette thèse apporte des contributions sur des classes de problèmes de bandits multi-bras adversariaux avec un nouvel algorithme d'apprentissage avec conseils d'experts et un algorithme de poids exponentiel emboîté qui effectue une exploration en couches de l'ensemble emboîté d'alternatives de l'apprenant.

Acknowledgement

First, I would like to thank the jury members for willing to evaluate my work, and in particular Olivier Cappé and Nicolò Cesa-Bianchi for accepting to review this manuscript. I am sincerely honored and grateful to have received your insight and comments on my PhD research. Thank you Claire and Massih as well for your presence in my jury and your feedbacks.

Next, I would like to express my sincere gratitude to my PhD supervisors for their help and advice throughout this journey. Julien, your invaluable guidance and expertise have been instrumental in shaping my academic journey. Your high research standards and supervision have enriched my knowledge and inspired me to strive for excellence throughout my PhD. Working with you on all project aspects, from writing, coding, analysis to methodology positioning, has been an exceptional opportunity to me. I am deeply thankful for your dedication and mentorship. Eustache, I am incredibly grateful for your exceptional management at Criteo. Your unique skills and expertise in addressing meaningful problems, along with your significant contributions to my work, have been a constant source of motivation and inspiration. Thank you for your dedication during our last sprints for ICML23, our one-to-one discussions, and your sincere support as I approached the end of my PhD. It was truly a pleasure to be a part of your team and I could not wish for better managing than yours. Pierre, ever since you joined our projects my work has gained rigor, simplicity and impact. That you co-supervised me was of one the luckiest opportunity I had during my PhD. I sincerely appreciate the time we spent together, from our enlightening whiteboard sessions to your clear explanations and assistance with theoretical analysis. Your support in allowing me to explore and lead projects, such as co-supervising interns with you, was invaluable for my growth as a PhD student. Our engaging debates on life and society, along with the memorable chess games (where at least one of us used to blunder) with other teammates will always be cherished among my best memories of this PhD.

Furthermore, I had the privilege of collaborating with outstanding co-authors throughout my doctoral studies. To begin, Panagiotis, with whom I had the pleasure of starting to work with more than five years ago when I was in Singapore. From the outset, Pan, I was deeply impressed by your unwavering determination and commitment to producing exceptional and impactful research. Your dedication to the field and your kindness have been a tremendous source of inspiration to me. I would also like to express my largest gratitude to Thibaud for the meaningful moments we shared at Criteo and the collaborative working sessions on our projects. Your unwavering dedication and sense of ownership were truly remarkable, and I feel fortunate to have had a colleague and friend like you. Additionally, I sincerely appreciate the great moments we shared, from chess games to ski sessions. Lastly, I consider myself extremely fortunate to have had the opportunity to collaborate with you, Alberto. Your assistance and commitment to excellence in our joint projects were invaluable to my research. Thank you for your contributions and support. I genuinely liked working with

you and aspire to have such accomplished co-authors and friends in the future. To Matthieu, thank you for your help and participation in my projects, thank you for our discussions and the moments we shared.

I would also like to thank my colleagues at Criteo. Artem, it was a pleasure to have you in my team, I valued our exchanges and friendly chess matches. Julien, I've known you since HX1 and it has always been great to be around you. I wish you all the best for your PhD and was delighted to reconnect with you during this period. Masha, thank you for your energy and your vibes, it has been great to know you both in Criteo and at Thoth. Thank you Amelie, as well, for the times we had at the beginning of my PhD journey. To Camille who coordinates the Grenoble office, thank you again.

My experience at INRIA would definitely not have been the same without the people in my team at Thoth. Thank you Nathalie for coordinating our team with such kindness and supporting us the way you do. To my former teammates, Alberto, Nikita, Vlad, Mathilde, Gregoire, Dexiong, Matthieu, Minttu, Valentin, Florent, Khue, Hubert, Mert, thank you. To the best office mate ever, Margot, thank you so much. Special thanks as well for Bruno, my friend from long ago. To the current members of my team: thank you Julien, Juliette, Gaspard, Théo, Loic, Timothée, Hee Seung, Zhiqi, Bianca, Camila, Emmanuel, Hadrien, Michael, Ieva, Jules, Thomas, Nassim, Jocelyn and Karteek. To you Romain, thanks for the chess games, for the ski sessions and for your unique spirit. To you Anand at Robotlearn, thank you. To Julyan at Statify, thank you as well.

To my friends and somehow older brothers Adil and Moussab, thank you for your advice and guidance from the very beginning. You have been helping countless times. Thank you Anna, as well, it has been such a pleasure to know you and be around with you and Adil at ICML22. To my best friend, Zakarya, to my friends and older brothers Yassin and Hicham. To my Tanjaoui brothers, Hamza and Zakaria. Thank you from the very beginning. To my friends Fatih and Adil. To my oldest friends in Grenoble, Sofiane, Yassine, Nelson, Anas. Thank you for your countless support, your time with me and your energy. Nothing would have been the same without you. Special thanks to you Sofiane (Jamal), for your exceptional help for this manuscript. Thank you Osman, Hicham, Ghassen, Nassera, Mahmoud, Moustapha. To my friends Siham, Merve, Camille, Naima, Lisa, Flore, Nouzhaty, thank you. To my brothers Imad, Aimine, Yahya, Ahmed, thank you for your spirituality, knowing you has been a haven of peace. To my Sira friends, Shakeel, Oussamah, Bachar, Bassel, Iyad, Rayan, Zeyneb, Hakima thank you.

Last but not least, I want to express to my greatest and most heartfelt appreciation to my family for their unwavering support and endless encouragement throughout my PhD journey. Their love, understanding, and belief in me have been the pillars upon which I built my academic pursuits. To my mother and father, your sacrifices and patience, often during my busiest and most challenging times, have made all the difference. This accomplishment is as much yours as it is mine, and I am profoundly grateful for your boundless dedication to my success. To my brothers Oussama, Tarek, my sister Sarah, thank you for being my source of strength and inspiration. Words cannot express how thankful I am to have you. Needless to say, I extend my deepest appreciation to my beloved family in Algeria.

All the blessings and praises to God.

To my grandfather Ali, to your values and principles, to your dedication to your family. To my grandmother Messaouda, to your ever and unconditional love, to your anecdotes and experiences. To both of you, words cannot convey the depth of my longing for your presence.

Ziama, Algeria

To my fierce and inflexible Azirou, to my warm and loving Zenata.

Contents

Nomenclature

Abbreviations

We provide below a table of some of the most useful abbreviations in this manuscript.

Abbreviation	Definition
CRM	Counterfactual Risk Minimization
SCRM	Sequential Counterfactual Risk Minimization
IPS	Inverse Propensity Scoring
SNIPS	Self-Normalized Inverse Propensity Scoring
DM	Direct Method
DR	Doubly Robust
UCB	Upper Confidence Bound Algorithm
FTRL	Follow the Regularized Leader
OMD	Online Mirror Descent
DA	Dual Averaging

Notations

We define here the most crucial notations that are used throughout the manuscript. Other chapter-specific notations are defined along the text and recalled in the appendices when needed in analysis sections.

Below are some notations related to the learning setting:

- L is an expected risk measure
- $\hat{L}$ is an estimator of that quantity
- λ is a regularization parameter
- n is a sample size
- θ is a parameter and the parameter space is Θ
- θ^* is an unknown optimal parameter
- δ is a confidence level
- Ω is a regularization function
- $\mathcal{L}$ is an objective function to be optimized
- d is the dimension of an input space

Below are some notations related to the bandit setting:

- $\mathcal{A}$ is the action set (set of alternatives)
- $k = |\mathcal{A}|$ is the size of the action set when it is finite
- $a_t \in \mathcal{A}$ is an action played at a step t

- $\mathcal{X}$ is the context space
- $x_t \in \mathcal{X}$ is a context sampled at a step t
- $y_t \in \mathcal{Y}$ is a loss (or a target) induced at time t
- r_t is a reward at time t
- Π is a policy set
- $\pi \in \Pi$ is a policy that can be stochastic or deterministic

Below are some generic notations:

- $[n] := \{1, \dots, n\}$
- $\lesssim$ denotes an approximate inequality up to logarithmic multiplicative or additive terms
- For random variables $x \sim \mathcal{P}_{\mathcal{X}}$, $a \sim \pi_\theta(\cdot|x)$ and $y \sim \mathcal{P}_{\mathcal{Y}}(\cdot|x, a)$, we write the expectation $\mathbb{E}_{x,\theta,y}[\cdot] = \mathbb{E}_{x\sim\mathcal{P}_{\mathcal{X}}, a\sim\pi_\theta(\cdot|x), y\sim\mathcal{P}_{\mathcal{Y}}(\cdot|x,a)}[\cdot]$ and do the same for the variance $\mathrm{Var}_{x,\theta,y}$.

Below are generic notations related to RKHS:

- $\mathcal{S}$ is the input space
- $K : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ is a bounded positive definite Kernel
- $\kappa > 0$ is an upper-bound on the kernel $\kappa^2 \geq \sup_{s\in\mathcal{S}} K(s, s)$.
- $\mathcal{H}$ is the reproducing kernel Hilbert space associated to K
- $\phi : \mathcal{S} \to \mathcal{H}$ is the feature map such that $K(s, s') = \langle \phi(s), \phi(s') \rangle_{\mathcal{H}}$ for any $s, s' \in \mathcal{S} \times \mathcal{S}$
- $\langle \varphi, \varphi' \rangle_{\mathcal{H}} := \varphi^\top \varphi'$ denotes the inner product for any $\varphi, \varphi' \in \mathcal{H}$
- $\| \cdot \|_{\mathcal{H}}$ denotes the norm associated to $\mathcal{H}$. It is the one induced by the inner product, i.e., $\|\varphi\|^2 = \langle \varphi, \varphi \rangle$
- $\| \cdot \|_V$ denotes for any symmetric positive semi-definite operator $V : \mathcal{H} \to \mathcal{H}$ the norm such that $\|\varphi\|_V = \|V^{1/2}\varphi\|_{\mathcal{H}}$ for all $\varphi \in \mathcal{H}$
- $L \preccurlyeq L'$ means that $L - L'$ is positive semi-definite for two operators L, L' on $\mathcal{H}$
- $\varphi \otimes \varphi' : \mathcal{H} \to \mathcal{H}$ is the tensor product of φ and $\varphi' \in \mathcal{H}$
- $\mathcal{Z} \subset \{s_1, \dots, s_n\}$ is a dictionary of elements of the observation set $\mathcal{S}_n = \{s_1, \dots, s_n\}$
- $m = |\mathcal{Z}|$ is the size of this dictionary

Below are notations related to the sequential setting. Here, $t \in [T]$ denotes the index of the round:

- T is the horizon or number of rounds
- $s_t := (x_t, a_t) \in \mathcal{X} \times \mathcal{A}$ is a state at round t
- $\mathcal{S}_t := \{s_1, \dots, s_t\}$ denotes the history
- $\varepsilon_1, \dots, \varepsilon_T$ are independent centered sub-Gaussian noise
- $\mathcal{F}_t := \sigma(\varepsilon_1, \dots, \varepsilon_t)$ is the natural filtration with respect to $(\varepsilon_i)_{i\geq 1}$
- $\varphi_t := \phi(x_t, a_t) \in \mathcal{H}$ for $\phi : \mathcal{X} \times \mathcal{A} \to \mathcal{H}$

Below are notations related to the online optimization setting.

- $u \in \Delta(\mathcal{A})$ mixed strategy on the alternative set
- P is a choice function that maps score vectors $y \in \mathbb{R}^{\mathcal{A}}$ to mixed strategies via the relation $u = P(y)$
- γ is the learning rate for the choice map
- h is an entropy function
- H is the depth of the entropy function

1

Introduction

By combining statistical methods with machine learning techniques, researchers have unlocked in the recent decades dramatical insights and innovative solutions in fields such as marketing, healthcare, social sciences and finance. Such methods have been instrumental in driving many recent technological advances in statistical decision making where the goal is to understand dependencies on random variables and make decisions based on observational data. To do so, given past outcomes of experiments, the core complexity of such statistical learning methods lies in inferring an underlying probabilistic structure from finite samples of arbitrary sizes.

Yet, even if the full probabilistic model were known, an additional difficulty raise because the knowledge of observational distributions does not determine an underlying *causal* structure, in the sense that *correlation does not imply causation*. While in the recent decade, tremendous and remarkable empirical successes have been achieved on difficult tasks with complex data, a growing interest has been shifting to understanding causal dependencies and properties for tasks in precision medicine, drug dosage, online advertising and personalized recommendations. As a matter of fact, in decision problems where treatments or actions are taken through a policy, an increased attention has been given to *counterfactual* reasoning. The latter aims at providing a probabilistic answer to a *"what would have happened if"* question that occurs in many problems with partial feedback or missing information and for which an a posteriori analysis of past decisions is desirable. In that case, a decision making system can be improved without being deployed in a real-world experiment.

Subsequently, leveraging large amounts of data, counterfactual learning methods have been developed for learning problems. Notably, log data is an extremely widespread type of data that can be easily collected from a variety of systems (such as search engines, ad placement, and recommendation systems) at a low cost. Typically, the logs of such decision systems contain information on user input (such as user features), system predictions (such as a recommended list of news articles), and feedback (such as the number of articles the user read). However, this feedback only provides partial information, known as "bandit feedback," which is limited to the specific prediction made by the system. The feedback for all the other possible predictions is typically unknown. This fundamental difference in feedback makes learning from log data distinct from supervised learning, where full-information feedback is available through "correct" predictions and a loss function.

The latter methods are actually offline variants for a sequential learning setting, where data sets are not immediately available to learn a model, but rather observed sequentially as a data flow. In that so called bandit setting, a decision maker is required to take actions one after another based on past observations. Once the decision is made, the decision maker suffers a loss (or gains a reward, depending on the problem) with partial feedback. Every decision carries the potential for a different loss, which is unknown to the participant beforehand. To analyse such settings, under specific assumptions on the distribution of contexts and losses, it is possible to derive guarantees using statistical learning theory. However, in such a setting, it is worth noting that the environment may be so complex that it is not feasible to select a comprehensive model and apply classical statistical theory and optimization. Specifically, an adversary may arbitrarily choose the losses at each round which necessitates more elaborate decision making.

This thesis follows these main directions and focuses on exploring theoretical and practical questions related to statistical methods for counterfactual policy learning and sequential learning for problems motivated in Section 1.1. These contributions are further described in Section 1.2. The rest of this introduction aims at providing an overview of essential concepts and settings that arise in the contributions of this thesis. Specifically, Section 1.3 introduces the fundamental concepts in the offline counterfactual risk minimization setting that are covered in Chapter 2 and 3. Next, Section 1.4 provides brief explanations on the intuition of acceleration strategies that are used in the Chapter 3 to obtain faster convergence rates. Section 1.5 presents some of the kernel scalability issues that arise in the algorithms presented in Chapter 4 and that are also used in Chapter 2. Then, Section 1.6 provides the essential background to define the stochastic bandit setting that are instrumental to Chapter 3, 4, 5. Eventually, Section 1.7 introduces the basic notions on online optimization that are used in the analysis of the adversarial bandit setting considered in Chapter 5. We provide a summary of the introduction sections and the associated contributions of this thesis in Figure 1.1.

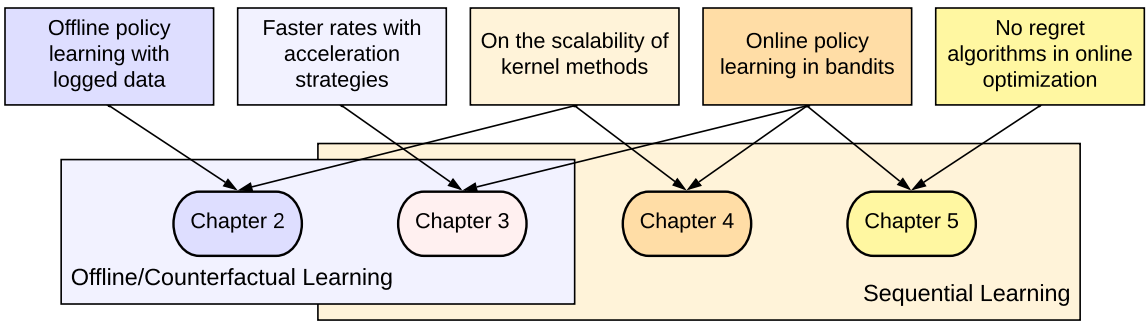

Figure 1.1: Summary of the introduction to the contributions of the thesis.

1.1. Motivations and practical problems

We start by introducing a few practical problems that arise in advertising companies such as Criteo, which motivated this PhD thesis at the Criteo AI Lab in collaboration with INRIA. The introduction of this thesis will then aim at providing foundations to understand existing solutions and algorithms that can be applied for those challenges that may arise in Criteo.

Beyond A/B Testing In traditional A/B testing, the objective of the designer is to decide which option (A or B) is better. For example, company designers could consider whether to place a new feature in a product being deployed. However, such methods have practical limitations. First, if we view the two versions of the site as policies π_A or π_B being deployed, running policies that are "suboptimal" may induce large development and experimentation costs or even be dramatic in sensitive applications. Second, in typical online platforms a large majority of A/B tests yield neutral or negative decisions Kohavi et al. (2009). Thus, offline estimation of policy performance allows experimenters to design plausible option policies for the A/B testing as discussed in Chapter 2. Moreover, traditional A/B testing may require multiple deployments to collect sufficient sample size and enhance variance control. In that case, considering sequential deployments of offline and online A/B options as presented in Chapter 3 could be particularly meaningful especially if the original option policies where under-performing.

Online bidding Today most free-to-use services and content applications are funded by advertising. Different forms of advertising exist yet the most widespread type involves running real-time auctions to sell advertising space in an economic efficient manner. In such industries, billions of auctions come out daily between the same group of buyers and sellers. In such bidding problems, a real valued bid is predicted in response to contextual information from user inputs. Using log data, it is possible to design policies that improve upon a previous system when using continuous action modellings, as discussed in Chapter 2. Leveraging the data accumulated from these interactions, various methods are employed to acquire a thorough understanding of the intricate mechanisms that maximize seller revenue and bidder value. Conversely, when considering online policies, the most straightforward framework is to assume a sequential, stochastic game to design efficient strategies for large scale applications as presented in Chapter 4. More realistic assumptions would model this problem with adversarial settings as we did in Chapter 5, but we note that this problem in itself is a broader concept that we did not aim at solving completely in the latter.

Advert Placement In the context of advertising placement, each user visiting a website can be seen as a round, and the available ads can be considered as the set of actions. A standard multi-armed bandit problem can be used, where a policy chooses an ad at each round, and the reward is 1 if the user clicks on the ad and 0 otherwise. However, for a company like Criteo, targeted advertising is essential, and user context should be taken into account. This can be achieved by using the user context, such as in contextual bandits. The methods used in this PhD thesis tackle complex issues of real-world systems, with the set of available ads changing from round to round, with action set structures of various nature, exploration constraints, and other metrics such as scalability and efficiency being important as well. We highlight

the scalability issues of such contextual bandits methods in online settings in Chapter 4. For sensitive applications where logged data is used instead to learn a policy offline we provide methods in Chapter 3 to learn policies with sequential deployments.

Personalized treatments Another extremely related application is personalized treatment. Example applications entail precision (or personalized) medicine. While sequential learning can be used to continually treat and accurately diagnose patients based on their individual characteristics and medical history as new information about their health becomes available, online deployments can be sensitive and unethical. Therefore, in personalized medicine, offline policy learning is often more suitable than online or sequential learning approaches. To do so, randomized control trials are run on patients to assess the effectiveness of a new treatment or intervention. After the participants are randomly assigned to either a treatment group or a control group, an offline analysis is a posteriori possible to perform counterfactual reasoning and learning as described in Chapter 2 or even Chapter 3.

Resource allocation Maintaining a low infrastructure cost is a key problematic in many tech companies including Criteo. While a significant effort in operations research has involved developing methods for distributing limited resources effectively, the problem can resemble a bandit problem in situations where the fluctuations of demand or supply are not certain. As a matter of fact, with a combinatorial structure that resembles the nested structure we present in Chapter 5, one could design a strategy to allocate resources that have similarities in outcome. Distributing marginally different resources can only provide limited insight into the actual demand, while providing excessive resources can lead to wastage. However, it should be noted that resource allocation is a broad concept and many issues have unique structures that do not fit into the typical framework of bandit problems.

1.2. Contributions of the thesis

This thesis brings various contributions with regard to the study of counterfactual policy learning in the offline logged bandit feedback and in sequential learning problems. We review the contributions hereafter.

- Chapter 2 presents methods in modelling, learning and model selection for counterfactual learning of stochastic policies with continuous actions. Continuous action policies have received little attention in the CRM setting while being ubiquitous in many problems (drug dosage, online bidding), our work introduces an effective modelling in that setting which improves the state of the art. Moreover, closely related to learning, we show how appropriate tools can bring signifcant benefits in the optimization perspectives of non-convex and non-differentiable CRM objective functions that have been overlooked. Eventually, we bring contributions in the problem of reliably evaluating learned policies based on logged data only which is crucial in pratice. We propose an offline model selection protocol and release a new large-scale dataset obtained from a real-world system for evaluation benchmark. All of those are also validated by numerical experiments. This work has led to a workshop paper and a working journal paper that are given below.

H. Zenati, A. Bietti, M. Martin, E. Diemert, and J. Mairal. Optimization approaches for counterfactual risk minimization with continuous actions. *International Conference on Learning Representation (ICLR), Causal Learning for Decision Making Workshop*, 2020b

H. Zenati, A. Bietti, M. Martin, E. Diemert, P. Gaillard, and J. Mairal. Counterfactual learning of stochastic policies with continuous actions: from models to offline evaluation. *arXiv preprint arXiv:2004.11722*, 2020a

- Chapter 3 formalizes an extension of the CRM learning principle that is essential in real-world problems. In the logged bandit feedback, when the logging policy underexplores the action space, importance sampling methods in counterfactual learning are proned to large variance issues which often leads to the failure of CRM. In that case, collecting additional data to increase the sample size is desirable and is more efficient with sequential data collection designs. To that effect, when sequential deployments are possible, we introduce sequential counterfactual risk minimization (SCRM). Our method uses a novel counterfactual estimator with controlled variance, extends the analysis of CRM and provides fast rates under an assumption as in restart strategies in optimization. Moreover, numerical results show the efficiency of our method. This chapter has been published as a conference paper.

H. Zenati, E. Diemert, M. Martin, J. Mairal, and P. Gaillard. Sequential counterfactual risk minimization. *International Conference on Machine Learning (ICML)*, 2023

- Chapter 4 completely shifts to an online setting and introduces an efficient algorithm for Kernel UCB (K-UCB) in stochastic contextual bandits. While the standard K-UCB algorithm requires a $\mathcal{O}(T^3)$ complexity where T is the horizon, we propose an efficient contextual algorithm for large-scale problems using kernel approximations. More specifically, with incremental Nyström approximations of the joint kernel embedding of contexts and actions we achieve a complexity of $\mathcal{O}(CTm^2)$ where m is the number of Nyström points. Typically, m is of order of the effective dimension of the problem, which is at most $\mathcal{O}(\sqrt{T})$ and nearly constant in some cases. We numerically validate this approach and obtain as well empirical improvements upon existing methods in the Bayesian experimental design litterature. This work has led to the following conference paper.

H. Zenati, A. Bietti, E. Diemert, J. Mairal, M. Martin, and P. Gaillard. Efficient kernelized ucb for contextual bandits. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2022

- Chapter 5 presents contributions in online decision-making. When choosing between a large number of similar alternatives that have similar losses, it can be particularly difficult to find an optimal strategy. Standard algorithms may explore irrelevant alternatives, leading to high regret. We introduce a setting that we call nested bandit problems, where there are many distinct alternatives with embedded similarities. To solve this problem, we propose the Nested Exploration Weighting algorithm that explores alternatives layer by layer and the Exponential Weights with Experts and Nesting algorithm when learning with expert advice, resulting in improved regret guarantees. This chapter has been published as a conference paper and is also based on a manuscript in preparation.

M. Martin, P. Mertikopoulos, T. Rahier, and H. Zenati. Nested bandits. *International Conference on Machine Learning (ICML)*, 2022

Manuscript in preparation:

H. Zenati, T. Rahier, M. Martin, and P. Mertikopoulos. Sequential Decision Processes with Outcome Similarities

Moreover, we highlight that this thesis led to open-source softwares related to the contributions above, which are given in Appendices 2.9, 3.8, 4.7, 5.9 and that we restate below:

- Chapter 2: `https://github.com/criteo-research/optimization-continuous-action-crm`
- Chapter 3: `https://github.com/criteo-research/sequential-conterfactual-risk-minimization`
- Chapter 4: `https://github.com/criteo-research/Efficient-Kernel-UCB`
- Chapter 5: `https://github.com/criteo-research/Nested-Exponential-Weights`

Other contributions of this thesis, which are not included in this manuscript are collaborations on Criteo internal technical reports in combinatorial bandits.

1.3. Offline policy learning with logged data

In this section we provide the theoretical foundations of the counterfactual risk minimization (CRM) framework to learn an offline policy in the logged bandit feedback problem. To understand the counterfactual learning methods in (CRM), we introduce an overview of the empirical risk minimization framework in statistical learning.

1.3.1. Empirical Risk Minimization

In order to present the empirical risk minimization framework, we start by introducing the supervised learning setting which is a category of statistical learning. For a more in-depth discussion on the topic, we address the reader to (Bach, 2023).

The supervised learning setting

Given some observations $(x_i, y_i)_{i=1,\cdots,n} \in \mathcal{X} \times \mathcal{Y}$, of pairs of inputs (features or covariates such as images, text, sequences of DNA, times series) and targets (labels that can be binary, categorical or continuous responses), the objective in supervised learning is to predict a new $y \in \mathcal{Y}$ given a new previously unseen $x \in \mathcal{X}$. Note that in supervised learning, a probabilistic formulation is used to see pairs $(x_i, y_i)_{i=1,\cdots,n}$ as realizations of random variables, that are assumed to be independent and identically distributed (i.i.d.). To quantify the prediction objective we define a loss function $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ where $l(y, \hat{y})$ is the loss of predicting $\hat{y}$ while the true target is y.

Then, the criterion is to maximize the expectation of some "performance" measure with respect to the distribution of the data. Given a prediction function $f : \mathcal{X} \to \mathcal{Y}$, we can define the expected risk (also referred to as generalization error) of a function as the expectation of the loss function between the output y and the prediction $f(x)$:

$$R(f) = \mathbb{E}_{x,y}\left[l(y, f(x))\right] = \int_{\mathcal{X}\times\mathcal{Y}} l(y, f(x)) \, \mathrm{d}p(x, y), \tag{1.1}$$

where p is the probability distribution on $\mathcal{X} \times \mathcal{Y}$. As a matter of fact, the risk is taken as the expectation over the randomness of the targets as well since we also consider random predictions. The optimal predictor (also referred to as Bayes optimal predictor) f^* is then the minimizer of R over the measurable elements of $\mathcal{Y}^{\mathcal{X}}$:

$$f^* \in \operatorname*{arg\,min}_{f \in \mathcal{Y}^{\mathcal{X}}} R(f). \tag{1.2}$$

A learning algorithm aims at finding a prediction function $\hat{f}$ from the observational data such that $R(\hat{f})$ is small, ideally close to the optimal (Bayes) risk $R(f^*)$. Therefore, we usually use the following excess risk definition:

$$\Delta_f = R(f) - R(f^*). \tag{1.3}$$

It is now natural to ask when it is possible to obtains guarantees on a learning algorithm with n observations, which is usually obtained in two manners. First, we can consider upper bounding the excess risk by a term that vanishes to zero when n tends to infinity: the algorithm is consistent in expectation. Another way is to guarantee that for any $\varepsilon > 0$,

$$R(\hat{f}) - R(f^*) \leq \varepsilon$$

holds for a given level of confidence, which is called "Probably approximately correct" (PAC) learning. Interestingly, without searching $\hat{f}$ in a particular subset of functions $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$, it is not possible in general to obtain such guarantees, for instance if $\mathcal{X}$ is infinite in a classification task as stated in a form of the no free lunch theorem (Shalev-Shwartz and Ben-David, 2014).

Therefore, to learn a predictor with small risk, ideally close to the Bayes risk $R(f^*)$, we need restrictions on the function class, which creates an inductive bias to learning. Intuitively,

a large function class is more likely to contain f^*, but a small class makes learning easier. This leads to an estimation-approximation tradeoff:

$$R(\hat{f}) - R(f^*) = \underbrace{R(\hat{f}) - \min_{f \in \mathcal{F}} R(f)}_{\text{estimation error}} + \underbrace{\min_{f \in \mathcal{F}} R(f) - R(f^*)}_{\text{approximation error}}. \tag{1.4}$$

The first term, the estimation error, deals with the ability to learn the best function in the class $\mathcal{F}$ from a finite number of samples n and increases as the hypothesis class becomes larger, since this makes learning harder. The second term, approximation error, decreases when $\mathcal{F}$ gets larger and reaches zero once $\mathcal{F}$ is large enough to contain f^*. In the analysis of the methods used in this thesis, we will focus on controlling the estimation error.

Learning from data

In practice, to learn from observational data, since we do not have the full knowledge of the data distribution p, we need to estimate a prediction function from the observational data. To do so we start by defining an empirical risk by averaging the loss on the observational data:

$$\hat{R}(f) = \frac{1}{n} \sum_{i=1}^{n} l(y_i, f(x_i)). \tag{1.5}$$

The empirical risk minimization for a function class $\mathcal{F}$ then consists in solving the following optimization problem:

$$\hat{f} \in \arg\min_{f \in \mathcal{F}} \hat{R}(f). \tag{1.6}$$

Often, we consider a parametrized family of prediction functions $f_\theta : \mathcal{X} \to \mathcal{Y}$ for θ in a parameter (model) space Θ. To not overload the notations on the expect risk, we write:

$$L(\theta) = R(f_\theta) \tag{1.7}$$

the expected risk of the model θ in the model space Θ and $\hat{L}(\theta) = \hat{R}(f_\theta)$.

Example 1.3.1. *The most classic example is linear least-squares regression where we minimize:*

$$\frac{1}{n} \sum_{i=1}^{n} (y_i - \theta^\top \phi(x_i))^2$$

over $\theta \in \Theta \subseteq \mathbb{R}^d$ and a fixed and known feature map $\phi : \mathcal{X} \to \mathbb{R}^d$.

In the decomposition of Eq. (1.4), the estimation error is related with the learning algorithm and the use of a finite sample. One basic approach to control it is through uniform convergence, which control maximal deviations between empirical and expected risk, for

all functions in a function class $\mathcal{F}$. Let $\theta^* \in \arg\min_{\theta\in\Theta} L(\theta)$ and $\hat{\theta} \in \arg\min_{\theta\in\Theta} \hat{L}(\theta)$ the empirical risk minimizer, then:

$$\begin{aligned} L(\hat{\theta}) - L(\theta^*) &= L(\hat{\theta}) - \hat{L}(\hat{\theta}) + \hat{L}(\hat{\theta}) - \hat{L}(\theta^*) + \hat{L}(\theta^*) - L(\theta^*) \\ &\leq \hat{L}(\hat{\theta}) - \hat{L}(\theta^*) + 2\sup_{\theta\in\Theta} |L(\theta) - \hat{L}(\theta)|. \end{aligned}$$

Note that from the definition of empirical risk minimizer, the left term $\hat{L}(\hat{\theta}) - \hat{L}(\theta^*)$ is negative in theory but in practice it may not when using optimization algorithms. The uniform deviation $\sup_{\theta\in\Theta} |L(\theta) - \hat{L}(\theta)|$ grows with the size of Θ and usually decays with n.

Convergence rates To provide convergence rates and assess the performance of the empirical risk minimization, it is then useful to bound the uniform deviation term that we considered. In particular, when the loss is uniformly bounded by a constant C, using concentration and a technique called symmetrization, it is possible (Boucheron et al., 2005; Shalev-Shwartz and Ben-David, 2014) to obtain an upper bound with probability $1 - \delta$:

$$\sup_{f\in\mathcal{F}} |R(f) - \hat{R}(f)| \leq 2\hat{R}_n\left(l \circ \mathcal{F}\right) + C\frac{2\log\frac{2}{\delta}}{n}, \tag{1.8}$$

where $\hat{R}_n\left(l \circ \mathcal{F}\right)$ is the empirical Rademacher complexity of the set of empirical observations $\{l(f(x_1), y_1),\ \ldots\ l(f(x_n), y_n) : f \in \mathcal{F}\}$. This quantity typically grows with the number of parameters and is often unbounded for rich, non-parametric classes (like kernel methods). However, if we consider that $l(\cdots, y)$ is C_l-Lipschitz for any y, then using the contraction lemma (Bartlett and Mendelson, 2002; Boucheron et al., 2005) we obtain that $\hat{R}_n\left(l \circ \mathcal{F}\right) \leq C_l\hat{R}_n(\mathcal{F})$ where $\hat{R}_n(\mathcal{F})$ is the empirical Rademacher complexity of the set $\{f(x_1), \ldots f(x_n) : f \in \mathcal{F}\}$. Eventually, for certain classes such as kernel methods with a bounded norm, we can bound the latter complexity. For example, if we consider an RKHS ball $\mathcal{F}_B = \{f \in \mathcal{H} : \|f\|_{\mathcal{H}} \leq B\}$ of a kernel K, and assume $K(x, x) \leq R^2$ for all $x \in \mathcal{X}$, we can bound (Bartlett and Mendelson, 2002; Boucheron et al., 2005) $\hat{R}_n(\mathcal{F}) \leq \frac{BR}{\sqrt{n}}$. Moreover for parametric decision rules, if we consider for example the linear decision rule $\mathcal{F} = \{f_\theta,$ such that $f_\theta(x) = \theta^\top x : \|\theta\|_2 B$, if we further assume that $\|x\| \leq W$, Kakade et al. (2008) shows that $\hat{R}_n(\mathcal{F}) \leq \frac{BW}{\sqrt{n}}$. It is then possible to bound the empirical risk minimizer excess risk as:

$$L(\hat{\theta}) - L(\theta^*) \leq 2\left((BWC_l + 1)\sqrt{\frac{\log\frac{2}{\delta}}{n}}\right). \tag{1.9}$$

The latter bound is of order $\mathcal{O}(1/\sqrt{n})$. However, one limitation of those upper bounds is that they are dependent on properties that apply consistently across the entire set of possible hypotheses $\mathcal{F}$ (as a result of uniform convergence bounds). As a result, they cannot take advantage of beneficial statistical features that may only be present in functions that perform well on the given data sample. It is possible to obtain better rates (known as fast rates) for instance of order $\mathcal{O}(1/n)$.

Capacity control In order to prevent overfitting, it is necessary to limit the complexity of the model by reducing the number of parameters or constraining the norm of predictors. This is commonly achieved through constrained optimization, which restricts the set of allowed functions and reduces the size of the parameter space Θ. By doing so, it becomes possible to decompose the risk as in Eq. (1.4). Even so, capacity control can be done through regularization, that is by adding a penalty term in the minimization:

$$\hat{\theta} \in \arg\min_{\theta \in \Theta} \hat{L}(\theta) + \Omega(\theta), \tag{1.10}$$

where $\Omega(\theta)$ controls the capacity of the hypothesis f_θ associated to $\theta \in \Theta$. This Ω term can be a classical L2 penalization term in the simple ridge regression setting or a more convoluted empirical variance term as used in sample variance penalization (Maurer and Pontil, 2009) and discussed in the next subsection.

1.3.2. Counterfactual Risk Minimization

Next, we present the logged bandit feedback problem and follow with a presentation of the counterfactual risk minimization (CRM) framework, which is at the core of Chapters 2 and 3.

The logged bandit feedback problem

In the logged bandit problem, we are given n logged observations $(x_i, a_i, y_i)_{i=1,\dots,n}$ where contexts $x_i \in \mathcal{X}$ are sampled from a stochastic environment distribution $x_i \sim \mathcal{P}_\mathcal{X}$, actions $a_i \sim \pi_{\theta_0}(\cdot|x_i)$ are drawn from a logging policy π_{θ_0}. Unlike the supervised learning setting, aside from the contextual information given in the $(x_i)_{i=1,\dots n}$ we also consider actions $(a_i)_{i=1,\dots n}$ from a logging policy π_0. We write $s_0 = (x_i, a_i, y_i)_{i=1,\dots,n}$ the logging dataset for which actions are sampled under the logging policy. We consider in this setting parametric policies and write θ_0 the logging model in the parameter space Θ. The losses are drawn from a conditional distribution $y_i \sim \mathcal{P}_\mathcal{Y}(\cdot|x_i, a_i)$. We define the propensities $\pi_{0,i} = \pi_{\theta_0}(a_i|x_i)$ and assume them to be known. We will assume that the policies in $\pi_\theta, \theta \in \Theta$ admit densities so that the propensities will denote the density function of the logging policy on the actions given the contexts. The expected risk of a model θ is defined as:

$$L(\theta) = \mathbb{E}_{x,\theta,y}\left[y\right]. \tag{1.11}$$

For the logged bandit, the task is to determine a model $\hat{\theta}$ with small risk. A model $\hat{\theta}$ is associated to a policy $\hat{\pi}_\theta$ in a set of *stochastic* policies Π_Θ. Thus, this definition may also include deterministic policies by allowing Dirac measures, unless Π includes a specific constraint *e.g.*, minimum variance, which may be desirable in order to gather data for future offline experiments as we will see in Chapter 3.

To minimize L we typically have access to an empirical estimator $\hat{L}$ and solve the following regularized problem:

$$\hat{\theta} \in \arg\min_{\theta \in \Theta} \hat{L}(\theta) + \Omega(\theta), \tag{1.12}$$

where Ω is a regularizer. When using counterfactual estimators for $\hat{L}$, the solution of (1.12) has been called *counterfactual risk minimization* (Swaminathan and Joachims, 2015a).

Counterfactual Learning

The counterfactual approach tackles the distribution mismatch between the logging policy $\pi_{\theta_0}(\cdot|x)$ and a policy π_θ in Π_Θ via importance sampling. The (IPS) method (Horvitz and Thompson, 1952) relies on correcting the distribution mismatch using the well-known relation

$$L(\theta) = \mathbb{E}_{x,\theta_0,y}\left[y\frac{\pi_\theta(a|x)}{\pi_{\theta_0}(a|x)}\right], \tag{1.13}$$

under the common support assumption (the support of π_θ support is included in the support of π_{θ_0}), which allows to derive an unbiased empirical estimate where we recall $\pi_{0,i} = \pi_{\theta_0}(a_i|x_i)$:

$$\hat{L}^{\text{IPS}}(\theta) = \frac{1}{n}\sum_{i=1}^{n} y_i \frac{\pi_\theta(a_i|x_i)}{\pi_{0,i}}. \tag{IPS}$$

Clipped estimator. Since the empirical estimator $\hat{L}^{\text{IPS}}(\theta)$ may suffer from large variance and is subject to various overfitting phenomena, regularization strategies have been proposed. In particular, this estimator may overfit negative feedback values y_i for samples that are unlikely under π_{θ_0} (see motivation for clipped estimators in Appendix 2.9), resulting in higher variances. Clipping the importance sampling weights in Eq. (cIPS) as Bottou et al. (2013) mitigates this problem, leading to a clipped (cIPS) estimator

$$\hat{L}^{\text{cIPS}}(\theta) = \frac{1}{n}\sum_{i=1}^{n} y_i \min\left\{\frac{\pi_\theta(a_i|x_i)}{\pi_{0,i}}, \alpha\right\}, \tag{cIPS}$$

where α is a clipping parameter. Smaller values of α reduce the variance of $\hat{L}(\theta)$ but induce a larger bias. Swaminathan and Joachims (2015a) also use the sample variance penalization principle (Maurer and Pontil, 2009) and propose adding an empirical variance penalty term controlled by a factor $\lambda > 0$ to the empirical risk $\hat{L}(\theta)$. Specifically, they write $\nu_i(\theta) = y_i \min\left(\frac{\pi_\theta(a_i|x_i)}{\pi_{0,i}}, \alpha\right)$ and consider the empirical variance for regularization:

$$\hat{V}^{\text{cIPS}}(\theta) = \frac{1}{n-1}\sum_{i=1}^{n}(\nu_i(\theta) - \bar{\nu}(\theta))^2, \qquad \text{with} \quad \bar{\nu}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\bar{\nu}_i(\theta), \tag{1.14}$$

which is subsequently used to obtain a regularized objective $\mathcal{L}$ with hyperparameters α for clipping and λ for variance penalization, respectively, so that $\Omega^{\text{cIPS}}(\theta) = \lambda\sqrt{\frac{\hat{V}^{\text{cIPS}}(\theta)}{n}}$ and:

$$\mathcal{L}(\theta) = \hat{L}^{\text{cIPS}}(\theta) + \lambda\sqrt{\frac{\hat{V}^{\text{cIPS}}(\theta)}{n}}. \tag{1.15}$$

The (CRM) learning problem then is formulated as:

$$\hat{\theta}^{\text{CRM}} \in \arg\min_{\theta\in\Theta} \mathcal{L}(\theta). \tag{CRM}$$

A natural question now is to wonder whether we can provide statistical guarantees on $L(\hat{\theta}^{\text{CRM}}) - L(\theta^*)$ as we did in the supervised learning in the case of Eq. (1.9). Luckily, we provide in Chapter 2 guarantees of a modified clipping estimator as well as an extensive discussion on this matter with regards to the related work. In particular, Chapter 2 provides a proposition that we simpplify below.

Proposition 2.5.1. *Let $\hat{\theta}^{CRM}$ be the solution of* (CRM)*. Then, with well chosen parameters λ and M and other regularity assumptions detailed in the full proposition in Chapter 2, denoting the variance $\nu_*^2 = \mathrm{Var}_{\pi_0}\big[\pi_{\theta^*}(a|x)/\pi_{\theta_0}(a|x)\big]$, with probability at least $1-\delta$, the excess risk is upper bounded as:*

$$L(\hat{\theta}^{CRM}) - L(\theta^*) \lesssim \sqrt{\frac{(1+\nu_*^2)\log(n)}{n}},$$

where $\lesssim$ hides universal multiplicative constants.

The latter thus provides us a convergence rate of the (CRM) procedure. We further illustrate in Chapter 3 how to improve those guarantees when sequential redeployments are possible as presented in Section 1.4.

The self-normalized estimator. Swaminathan and Joachims (2015b) also introduce a regularization mechanism for tackling the so-called *propensity overfitting* issue, occuring with rich policy classes, where the method would focus only on maximizing (resp. minimizing) the sum of ratios $\pi_\theta(a_i|x_i)/\pi_{0,i}$ for negative (resp. positive) costs. This effect is corrected through the following *self-normalized importance sampling* (SNIPS) estimator (Owen, 2013, see also):

$$\hat{L}^{\text{SNIPS}}(\theta) = \frac{\sum_{i=1}^n y_i w_i^\theta}{\sum_{i=1}^n w_i^\theta}, \quad \text{with } w_i^\theta = \frac{\pi_\theta(a_i|x_i)}{\pi_{0,i}}. \tag{SNIPS}$$

The (SNIPS) estimator is also associated to $\Omega_{\text{SNIPS}}(\theta) = \lambda\sqrt{\frac{\hat{V}_{\text{SNIPS}}(\theta)}{n}}$ which uses an empirical variance estimator that writes as:

$$\hat{V}^{\text{SNIPS}}(\theta) = \frac{\sum_{i=1}^n \Big(w_i^\theta \Big(y_i - \hat{L}^{\text{SNIPS}}(\theta)\Big)\Big)^2}{\big(\sum_{i=1}^n w_i^\theta\big)^2}. \tag{1.16}$$

Note that another motivation of using (SNIPS) is that the (IPS) is not equivariant, that is to say for a constant c:

$$c + \min_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^n y_i w_i^\theta \neq \min_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^n (y_i + c) w_i^\theta.$$

When the solution of the optimisation problem is affected by a translation $y \leftarrow y + c$ for any real valued c, the estimator is not equivariant. However, the (SNIPS) estimator does verify:

$$c + \min_{\theta\in\Theta} \frac{\sum_{i=1}^n y_i w_i^\theta}{\sum_{i=1}^n w_i^\theta} = \min_{\theta\in\Theta} \frac{\sum_{i=1}^n (y_i + c) w_i^\theta}{\sum_{i=1}^n w_i^\theta}.$$

In that case, the estimator is more robust to propensity overfitting (Swaminathan and Joachims, 2015b) which is a phenomenon where the learned policies either overfit or avoid the training data depending on the sign of the losses y. Eventually with the (SNIPS) estimator, the learning objective becomes:

$$\mathcal{L}(\theta) = \hat{L}^{\text{SNIPS}}(\theta) + \lambda \sqrt{\frac{\hat{V}^{\text{SNIPS}}(\theta)}{n}}, \tag{1.17}$$

and the (2.10) learning principle can be applied as well with the latter.

Direct Methods

It is possible to perform supervised learning in the logged bandit feedback and to infer policies thereof. As a matter of fact, an important quantity is the expected cost given actions and context, denoted by $\eta^*(x, a) = \mathbb{E}[y|x, a]$. If this expected cost was known, an optimal (deterministic) greedy policy π^* would indeed simply select actions that minimize the expected cost

$$\pi^*(x) = \arg\min_{a \in \mathcal{A}} \eta^*(x, a). \tag{DM}$$

Therefore, it is then tempting to use the available data to learn an estimator $\hat{\eta}(x, a)$ of the expected cost, for instance by using ridge regression to fit $y_i \approx \hat{\eta}(x_i, a_i)$ on the training data. Then, we may use the deterministic greedy policy $\hat{\pi}^{\text{DM}}(x) = \arg\min_a \hat{\eta}(x, a)$. This approach, termed *direct method* (DM), has the benefit of avoiding the high-variance problems of IPS-based methods, but may suffer from large bias since it ignores the potential mismatch between $\hat{\pi}^{\text{DM}}$ and π_{θ_0}. Specifically, the bias is problematic when the logging policy provides unbalanced data samples (*e.g.*, only samples actions in a specific part of the action space) leading to overfitting (Bottou et al., 2013; Dudik et al., 2011; Swaminathan and Joachims, 2015b). Conversely, counterfactual methods re-balance these generated data samples with importance weights and mitigate the distribution mismatch to better estimate reward function on less explored actions (see explanations in Appendix 2.9). Nevertheless, such cost estimators can be sometimes effective in practice and may be used to improve IPS estimators in the so-called doubly robust (DR) estimator (Dudik et al., 2011) by applying (IPS) to the residuals $y_i - \hat{\eta}(x_i, a_i)$ as follows:

$$\hat{L}^{\text{DR}}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \hat{\eta}(x_i, a_i)\right) \frac{\pi_\theta(a_i|x_i)}{\pi_{0,i}} + \frac{1}{n} \sum_{i=1}^{n} \sum_{a \in \mathcal{A}} \hat{\eta}(x_i, a) \pi_\theta(a|x_i), \tag{DR}$$

which holds when the summation over $\mathcal{A}$ is possible (discrete action sets for e.g). As a matter of fact, the (DR) estimator uses $\hat{\eta}$ as a control variate to decrease the variance of (IPS). We investigate in Chapter 3 the use of an additional control variate as well to control for the variance of (IPS).

While such greedy deterministic policies may be sufficient for exploitation, stochastic policies may be needed in some situations, for instance when one wants to still encourage some exploration in a future round of data logs. Using a stochastic policy also allows us to obtain more accurate off-policy estimates when performing cross-validation on logged

data. Then, it may be possible to define a stochastic version of the direct method by adding Gaussian noise with variance σ^2:

$$\hat{\pi}^{\text{SDM}}(\cdot|x) = \mathcal{N}(\hat{\pi}^{\text{DM}}(x), \sigma^2), \tag{1.18}$$

In the context of offline evaluation on bandit data, such a smoothing procedure may also be seen as a form of kernel smoothing for better estimation (Kallus and Zhou, 2018).

1.4. Obtaining faster rates with restart acceleration strategies

In this section we provide a very short introduction to restart strategies used in optimization. This will allow to understand key contributions of Chapter 3 that uses similar assumption and analysis techniques for a batched bandit policy learning setting.

1.4.1. Restart strategies

In this section, we present elements of restart strategies that exist in acceleration methods in optimization. Rather than focusing on the sample efficiency of a statistical estimator, the aim of this section will be to provide an intuition on how an objective function satisfying generic Hölderian error bounds (HEB) can be optimized with a faster convergence rate, that is to say fewer optimisation iterations. While the two are not the same, understanding some of the elementary notions in this introduction will dramatically help grasp some of the contributions in Chapter 3. For a thorough presentation of the restart strategies, we point the reader to (d'Aspremont et al., 2021; Iouditski and Nesterov, 2014; Ghadimi and Lan, 2013; Kulunchakov and Mairal, 2019).

An illustrative example: strongly convex objective functions

Typically, first-order methods in optimization (methods that use gradient information of an objective function) have a sublinear convergence rate that depends on the smoothness of the gradient (Beck, 2017). The upper complexity bounds of these methods are usually convex functions of the number of iterations, which means that they converge quickly at first, but their convergence slows down as more iterations are performed. The intuition of restart methods is that it should be possible to speed up convergence by periodically restarting the first-order methods, that is to say running more "early" iterations. Moreover, first order methods implicitly approximate the function around the optimum at each iteration, and restarting should refresh this approximation periodically to discard outdated information as the algorithm approaches the optimal solution.

In restart strategies, the task is similar to what we do in Section 1.3.1 in the sense that we perform a minimization (as in Eq. (1.2)) on a given function L (that can be an expected risk as defined in Eq. (1.7)):

$$\min_{\theta \in \Theta} L(\theta), \tag{1.19}$$

where $\Theta \subset \mathbb{R}^d$ is the compact parameter space that we consider in Section 1.3.1.

The acceleration method in restart strategies is possible through a chaining argument that we will illustrate with a particular case of strongly convex objective functions L. Supposing

Algorithm 1: Restart scheme

Input: Objective function L, initial point θ_0, inner optimization algorithm $A(\theta, k)$, M number of iterations and planning $k_1, \ldots, k_M$

for $m = 1$ *to* M **do**

Obtain θ_m by running k_m iterations of A, starting at θ_{m-1}, i.e:

$$\theta_m = A(\theta_{m-1}, k_m)$$

end

that the gradient of L is Lipschitz continuous with constant C with respect to the Euclidean norm:

$$\|\nabla L(\theta') - \nabla L(\theta)\|_2 \leq C\|\theta - \theta'\|_2 \text{ for all } \theta, \theta' \in \Theta. \tag{1.20}$$

If we use a straightforward fixed gradient method to solve that problem, we have the iterates for $k \in \mathbb{N}$:

$$\theta^{(k+1)} = \theta^{(k)} - \frac{1}{C}\nabla L(\theta^{(k)}) \tag{1.21}$$

The smoothness assumption that stems for the gradient-Lipschitzness in Eq. (1.20) gives off the upper bound:

$$L(\theta^{(k)}) - L(\theta^*) \leq \frac{2C\|\theta_0 - \theta^*\|_2}{k+4} \tag{1.22}$$

after k iterations. If we now assume that L is strongly convex with parameter γ we have:

$$\frac{\gamma}{2}\|\theta - \theta^*\|_2 \leq L(\theta) - L(\theta^*), \tag{1.23}$$

where θ^* is a solution of (1.19). For any m and k_m the number of inner iterations in the optimization algorithm A defined through k_m iterations of gradient descent updates with (1.21), we write $\theta_m = A(\theta_{m-1}, k_m)$ with θ_0 an initial point. This means that we can rewrite Eq. (1.22) as:

$$L(\theta_{m-1}) - L(\theta^*) \leq \frac{2C\|\theta_m - \theta^*\|_2}{k_m + 4} \tag{1.24}$$

Then, combining the latter upper bound and the strong convexity in (1.23), after an iteration of the restart scheme in Algorithm 1, we obtain the chained inequality:

$$L(\theta_{m+1}) - L(\theta^*) \leq \frac{2C\|\theta_m - \theta^*\|_2^2}{k+4} \leq \frac{4C}{\gamma(k+4)}\left(L(\theta_m) - L(\theta^*)\right). \tag{1.25}$$

If we set $k_m = k = \lceil \frac{8C}{\gamma} \rceil$ then:

$$L(\theta_M) - L(\theta) \leq \left(\frac{1}{2}\right)^M \left(L(\theta_0) - L(\theta^*)\right),$$

after M iterations of the restart scheme in Algorithm 1. Therefore, if we run a total of $n = Mk$ gradient steps, we can write the previous upper bound as:

$$L(\theta^{(n)}) - L(\theta) \leq \left(\frac{1}{2^{\frac{\gamma}{8C}}}\right)^n (L(\theta_0) - L(\theta^*)), \tag{1.26}$$

which proves a linear convergence in the strongly convex case.

1.4.2. Hölderian Error Bounds

We now state the following assumption on Hölderian error bounds:

Assumption 1.4.1. *There exist some $\gamma, \beta > 0$ such that*

$$\gamma d(\theta, S^*_\Theta) \leq (L(\theta) - L(\theta^*))^\beta, \tag{1.27}$$

where $d(\theta, S^*_\Theta)$ is some distance to the optimal set ($S^*_\Theta = \arg\min_{\theta\in\Theta} L(\theta)$).

This bound is akin to a local version of strong convexity ($\beta = 1$) or a bounded parameter space ($\beta = 0$) if d is the Euclidean distance. When $\beta \in [0, 1]$, this has also been referred to as the Łojasiewicz assumption introduced in (Łojasiewicz, 1963, 1993).

It is important to note that a large class of functions L verify this bound. Specifically, we address the reader to the details on subanalytic functions in (Bolte et al., 2007) and the Łojasiewicz factorization lemma as stated in (d'Aspremont et al., 2021) to understand that this bound holds for mild conditions (Θ the parameter space is globally subanalytic and L is continuous and subanalytic).

Now, if it is possible to obtain a bound similar to that of Eq. (1.24), it will be possible to use the same chaining argument as done in Eq. 1.25 to demonstrate faster convergence rates. For example, for a C-smooth convex function Nesterov's method (Nesterov, 1983) with an optimal method (d'Aspremont et al., 2021) gives the upper bound:

$$L(\theta_{m+1}) - L(\theta^*) \leq \frac{4C}{{k_m}^2} \|\theta_m - \theta^*\|_2^2 \tag{1.28}$$

after k_m iterations to obtain θ_{m+1} with an initial point θ_m. It is then possible to show improved convergence rates for a smooth convex function satisfying the previous inequality as well as the Höderian error bound (d'Aspremont et al., 2021). In this thesis in Chapter 3, we provide a similar formulation as a Hölderian Error Bound for the (CRM) risk that we try to minimize and that can be previewed below.

Assumption 3.5.1 (Hölderian Error Bound)**.** *We assume that there exist $\gamma > 0$ and $\beta > 0$ such that for any $\theta \in \Theta$, there exists $\theta^* \in \arg\min_{\theta\in\Theta} L(\theta)$ such that*

$$\gamma \mathrm{Var}_{x,\theta}\left(\frac{\pi_{\theta^*}(x|a)}{\pi_\theta(x|a)}\right) \leq (L(\theta) - L(\theta^*))^\beta .$$

This variance term is not a distance but provides a similar intuition of how parameters can be distant to the optimal parameter θ^*. For example, for Gaussian policies with fixed variance, it is the exponential of the euclidean distance. Given that assumption, we show in Chapter 3 how to derive improved variance dependent convergence guarantees (w.r.t Chapter 2) and we use a similar analysis as the restart strategy presented above to derive the following result, which is a fast rate for (CRM).

Proposition 3.5.1 (Excess risk upper-bound). *Let $n \geq 2$ and $\theta^* \in \arg\min_\theta L(\theta)$. Let $M \approx \lfloor \log_2(n) \rfloor$. Then, under Assumption 3.5.1 and other regularity assumptions detailed in the full proposition in Chapter 3, the SCRM procedure (Alg. 6) satisfies the excess risk upper-bound for the round M:*

$$L(\theta_M) - L(\theta^*) \leq O\Big(n^{-\frac{1}{2-\beta}} \log n\Big).$$

1.5. On the scalability of kernel methods

In this part we introduce background notions on kernel methods and reproducing kernel Hilbert spaces (RKHS) that are used in Chapter 2 but more specifically in Chapter 4. We also provide generalizations properties and notions of kernel approximations. For an in-depth presentation of kernel methods, we address the reader to (Schölkopf and Smola, 2002; Shawe-Taylor and Cristianini, 2004; Berlinet and Thomas-Agnan, 2004) and to the lecture notes (Vert and Mairal, 2020).

1.5.1. Kernels and Reproducing Kernel Hilbert Spaces

Kernel methods are a class of algorithms in machine learning that allow learning in rich functional spaces. In particular, they use kernel functions to map the input data into a different high dimensional (or infinite dimensional) space. With this embedding, simple models can be trained on new non linear spaces, and has shown to drastically improve performances of the models.

Definition 1.5.1. *A positive definite kernel is a symmetric function $K : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ such that for any collection of points $s_1, \dots s_n \in \mathcal{S}$ and scalars $\alpha_1, \dots \alpha_n \in \mathbb{R}$, we have:*

$$\sum_{1 \leq i,j \leq n} \alpha_i \alpha_j K(s_i, s_j) \geq 0.$$

Conversely, kernel functions allow to define a gram matrix K_n:

$$K_n = [K(s_i, s_j)]_{1 \leq i,j \leq n}$$

Equivalently, K is a positive definite kernel if for any $n \in \mathbb{N}$ and input data $s_1, \dots, s_n \in \mathcal{S}^n$, the previously defined gram matrix K_n is positive semidefinite. Kernel methods take such matrices as input and have several advantages aside from their embedding properties that we will mention in Section 1.5.2. First, kernel methods always use such $n \times n$ matrices for

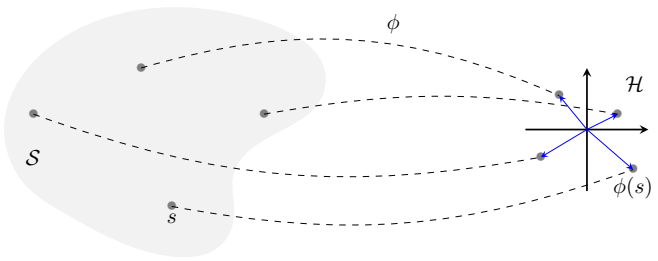

Figure 1.2: Representation of a kernel embedding

any input data (vectors, strings, etc.): the same algorithm can therefore work for multiple problems and applications. Second, the choice of the kernel function K is independent of the algorithm which therefore introduces a great modularity. However, note that they might suffer poor scalability with respect to the dataset size due to the size of K_n as we will see in Section 1.5.3. One contribution of this thesis in Chapter 4 subsequently aims at leveraging methods for this issue.

More importantly, kernel functions introduce a notion of comparison between data points between two objects s, s' in the set $\mathcal{S}$ which may have any arbitrary structure and thus create a similarity measure. Interestingly, it is in fact possible to show that such kernel functions are associated to an inner product on some features that can be non-linear.

Theorem 1.5.1. *(Aronszajn, 1950) A kernel $K : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ is positive definite if and only if there exists a Hilbert space $\mathcal{H}$ and a feature map $\phi : \mathcal{S} \to \mathcal{H}$ such that for any $s, s' \in \mathcal{S}$:*

$$K(s, s') = \langle \phi(s), \phi(s') \rangle_{\mathcal{H}}. \tag{1.29}$$

Such a feature map ϕ may define a space $\mathcal{H}$ in high dimensions on which linear models are effective and can be applied. As a matter of fact, ϕ can be infinite dimensional which makes the embedding of kernel methods very powerful. We now define Reproducing Kernel Hilbert Spaces (RKHS).

Definition 1.5.2. *Let $\mathcal{S}$ be a set and $\mathcal{H} \subset \mathbb{R}^{\mathcal{S}}$ be a class of functions forming a real Hilbert space with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. The function $K : \mathcal{S}^2 \to \mathbb{R}$ is a called a reproducing kernel of $\mathcal{H}$ if:*

- *For any $s \in \mathcal{S}$, let $K_s : t \mapsto K(s, t)$, then $K_s \in \mathcal{H}$*
- *For any $s \in \mathcal{S}$ and $f \in \mathcal{H}$, the reproducing property holds:*

$$f(s) = \langle f, K_s \rangle_{\mathcal{H}}$$

If such a reproducing kernel K exists, then $\mathcal{H}$ is called a reproducing kernel Hilbert space (RKHS).

RKHS are of great interest due to the simplicity they bring in machine learning. As a matter of fact, after mapping a data point $s \in \mathcal{S}$ to the RKHS $\mathcal{H}$ through a kernel mapping $\phi : \mathcal{S} \to \mathcal{H}$ with $\phi(s) = K_s$, simple linear models f are considered in $\mathcal{H}$ with $f(s) = \langle f, \phi(s) \rangle$.

Note also that it is possible to show that the reproducing kernel is unique given a RKHS, and conversely that a positive definite kernel defines a unique RKHS. This space characterizes the functions that are learned in kernel methods and hence allow the modelling of smooth functions.

Smoothness functional The reproducing property and the Cauchy-Schwarz inequality imply that for $s, s' \in \mathcal{S}$, the variations of a function $f \in \mathcal{H}$ can be controlled as:

$$\begin{aligned} |f(s) - f(s')| &= |\langle f, K_s - K_{s'} \rangle_{\mathcal{H}}| \\ &\leq \|f\|_{\mathcal{H}} \times \|K_s - K_{s'}\|_{\mathcal{H}} \end{aligned}$$

The norm of a function in the RKHS controls the variation of a function over $\mathcal{S}$ with respect to the geometry induced by the kernel, as small norm induces small variations. Therefore, the norm in the RKHS is related to its smoothness with regard to the metric defined by the kernel.

1.5.2. The Kernel Trick

Theoretical results on representing positive definite kernels as inner products and the representer theorem allow to use a family of powerful kernel methods algorithms. In this section we will introduce them.

Kernel Trick Recalling that the kernel is exactly the inner product in the feature space, we can state a simple yet extremely powerful statement. Any algorithm to process finite dimensional vectors and that is expressed only with pairwise inner products can be applied to infinite or high dimensional vectors in the feature space of positive definite kernels by replacing inner product evaluation by a kernel evaluation. Thus, vectors in the feature space can be manipulated implicitly through pairwise inner products.

We can provide a more formal statement of this intuition through the following theorem.

Theorem 1.5.2 (Representer theorem)**.** *Let $\mathcal{S}$ be a set endowed with a positive definite kernel K and $\mathcal{H}$ be the corresponding RKHS. Let $s^0 = \{s_1, \cdots, s_n\} \subseteq \mathcal{S}$ a finite set of points. Let $\psi : \mathbb{R}^{n+1} \to \mathbb{R}$ be a function $n+1$ variables, strictly increasing with respect to the last variable. Then any solution of the following optimization problem:*

$$\min_{f \in \mathcal{H}} \psi\big(f(s_1), \cdots, f(s_n)), \|f\|_{\mathcal{H}}\big),$$

admits a representation of the following form, where there exists real numbers $\alpha_1, \cdots, \alpha_n$ such that for any $s \in \mathcal{S}$:

$$f(s) = \sum_{i=1}^{n} \alpha_i K(s_i, s) = \sum_{i=1}^{n} \alpha_i K_{s_i}(s).$$

The solution lives in a finite dimensional subspace:

$$f \in \mathit{Span}(K_{s_1}, \cdots, K_{s_n}) \tag{1.30}$$

Note that the function ψ has the following form, where $c(\cdot)$ measures the "fit" of f to a given problem (regression, classification, $\cdots$) and Ω is a strictly increasing regularization function:

$$\psi\big(f(s_1), \cdots, f(s_n), \|f\|_{\mathcal{H}}\big) = c(f(s_1), \cdots f(s_n)) + \lambda\Omega(\|f\|_{\mathcal{H}})$$

First, from a theoretical perspective, this minimization enforces a small norm $\|f\|_{\mathcal{H}}$, so as to ensure a smoothness for the solution f. Second, we practically search for a solution in a subspace of dimension n which can lead to tractable algorithms even though the RKHS is infinite dimensional.

In the context of supervised learning models, this theorem allows to solve a regularized empirical risk minimization problem in a simpler space than the hypothesis space $\mathcal{H}$.

Example 1.5.1. *Given a set of data $(s_i \in \mathcal{S}, y_i \in \mathbb{R})_{i=1,\dots n}$, to estimate a regression function $f : \mathcal{S} \to \mathbb{R}$ we can solve the classical minimization problem:*

$$\min_{f\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} l(y_i, f(s_i)) + \lambda\|f\|_{\mathcal{H}}$$

for a loss function l. Solving this problem at first sight in the hypothesis space $\mathcal{H}$ that can be infinite-dimensional is possible with the representer theorem, by stating that any solution writes as

$$f(s) = \sum_{i=1}^{n} \alpha_i K(s_i, s)$$

for some $\alpha_1, \cdots \alpha_n \in \mathbb{R}$. Denoting $\alpha = (\alpha_1, \cdots, \alpha_n)$ the problem simplifies into:

$$\min_{\alpha\in\mathbb{R}^n} \frac{1}{n}\sum_{i=1}^{n} l\left((K\alpha)_i, y_i\right) + \lambda\alpha^\top K\alpha$$

which can be solved using standard convex optimization tools when the loss is convex. For the kernel ridge regression, the squared loss $l(\hat{y}, y) = (\hat{y} - y)^2$ induces the solution $\alpha = (K + n\lambda I_n)^{-1}y$, with $y = (y_1, \cdots, y_n)^\top$.

1.5.3. Kernel Approximations to scale and speed up kernel methods

One major problem that arise in kernel methods is the scalability issue. While the previous kernel trick and Representer theorem make kernel algorithms tractable, they can hardly scale up to large sample sizes. Such methods require the computation or inversion of the $n \times n$ Gram matrix which is infeasible when n grows both in terms of memory and computation. In that situation, the use of low-rank approximations of the kernel embedding make such approaches scalable while ensuring controllable properties as the original methods.

Nyström approximations Often the kernel matrix has a low rank, so that approximating the kernel matrix by sampling columns (Smola and Schölkopf, 2000; Williams and Seeger, 2001; Fine and Scheinberg, 2002) allows for efficient computations. The Nyström method consists in replacing any point $K_s = \phi(s)$ of the RKHS $\mathcal{H}$ for $s \in \mathcal{S}$ by its orthogonal projection onto a finite dimensional subspace:

$$\mathcal{F}_{\mathcal{Z}} = \text{Span}(\phi(z_1), \cdots, \phi(z_p))$$

where $\mathcal{Z} = \{z_1, \cdots, z_p\}$ are anchor points with typically $p \ll n$. An illustration of the Nyström approximation is provided in Figure 1.3.

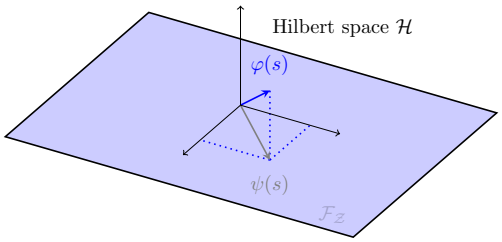

Figure 1.3: Representation of the Nyström approximation

To do so, an orthogonal projection $P_{\mathcal{F}_{\mathcal{Z}}}$ is defined onto the subspace $\mathcal{F}_{\mathcal{Z}}$ so that the points $\phi(s)$ can be approximated by $\psi(s) = P_{\mathcal{F}_{\mathcal{Z}}}\phi(s)$, with an inner product approximation as:

$$\langle \phi(s), \phi(s') \rangle_{\mathcal{H}} \approx \langle P_{\mathcal{F}_{\mathcal{Z}}}\phi(s), P_{\mathcal{F}_{\mathcal{Z}}}\phi(s') \rangle_{\mathcal{H}} \approx \langle \psi(s), \psi(s') \rangle_{\mathbb{R}^p} = K_{\mathcal{Z}}(s)^\top K_{\mathcal{Z}\mathcal{Z}} K_{\mathcal{Z}}(s')$$

where we use the notation $K_{\mathcal{Z}}(s) = [K(z_1, s), \ldots, K(z_p, s)]^\top$ and $K_{\mathcal{Z}\mathcal{Z}}$ is the kernel matrix vector $[K(z, z')]_{z,z' \in \mathcal{Z}}$. In particular, $\psi(s) = K_{\mathcal{Z}\mathcal{Z}}^{-1/2} K_{\mathcal{Z}}(s) \in \mathbb{R}^p$ can be written as the finite dimensional approximated feature map. The corresponding kernel matrix then is defined as:

$$\tilde{K}_n = K_{\mathcal{Z}S}^\top K_{\mathcal{Z}\mathcal{Z}} K_{\mathcal{Z}S}$$

where $K_{\mathcal{Z}S}$ is the kernel matrix vector $[K(z, s)]_{z \in \mathcal{Z}, s \in s^0}$. $\tilde{K}_n$ is thus the low-rank approximation of the original Gram matrix K_n.

As a matter of fact, in Chapter 2, we illustrate how we can provide an embedding over a joint context-action space $\mathcal{S} = \mathcal{X} \times \mathcal{A}$ where using a kernel K and its Nyström approximated feature map ψ, we manage to build an embedding to derive a cost predictor as in Figure 2.1 that we tease below.

To find the anchor points of the Nyström approximation, several strategies have been studied. One can use a naive random sampling, perform a kernel PCA to find largest principal directions in the Gram matrix, a simple K-means algorithm or eventually a greedy approach to find columns with largest residuals. Note that the latter is equivalent to computing an imcomplete Cholesky factorization (Bach and Jordan, 2005; Fine and Scheinberg, 2002). The

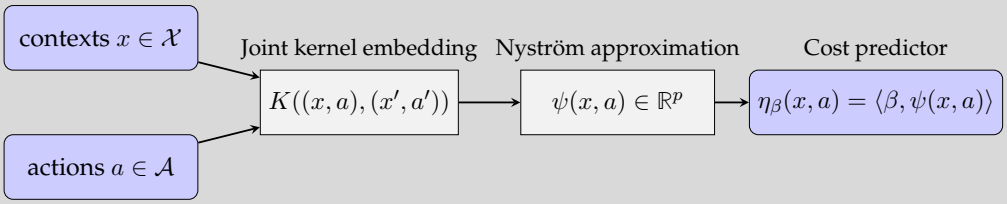

Figure 2.1: Illustration of the joint kernel embedding for the counterfactual loss predictor (CLP).

Nyström sampling has the advantage to admit geometric interpretation and to provide points in the RKHS, so that many operations such as translations, linear combinations on the mapping are valid.

A natural problem is to study the influence of m on the prediction performance of a learning system. In the ridge regression problem Bach (2013); Alaoui and Mahoney (2015) show that it is possible to preserve good convergence rates with an m much smaller than n, which allows large scale learning. In Chapter 4, we also provide an analysis of the number m to preserve the original regret rate of the algorithm we studied in contextual stochastic bandits. We present here one the results that are presented in Chapter 4 as a contribution of this thesis and where we seen the influence of the parameter m on the notion of regret (presented in Section 1.6). Typically this result can be coupled with a capacity condition assumption on the kernel to explicit regimes where it is possible to recover the original regret rate while improving the computational complexity. We present a discussion on this matter in Chapter 4.

Theorem 4.4.1. *Let $T \geq 1$ and $\theta^* \in \mathcal{H}$. Under some boundedness assumptions detailed in the full statement in Chapter 4, the EK-UCB rule in Eq.* (4.4.1) *with a regularization parameter λ and with $m = |\mathcal{Z}_t|$ dictionary updates, satisfies the pseudo-regret bound*

$$R_T \lesssim \sqrt{T}\left(\sqrt{m} + \sqrt{d_{\text{eff}}(\lambda, T)}\right)\left(\sqrt{\lambda} + \sqrt{d_{\text{eff}}(\lambda, T)}\right).$$

where $d_{\text{eff}}(\lambda, T)$, the effective dimension (Hastie et al., 2001) of the kernel matrix K_T replaces the dimension d in the (LinUCB) regret bound and is given formally as:

$$d_{\text{eff}}(\lambda, T) := \text{Tr}(K_T(K_T + \lambda I_T)^{-1}).$$

Random features We also note that another approach exists to perform kernel approximations that is based on sampling techniques. In particular, some kernels K can be written in the form:

$$K(s, s') = \mathbb{E}_{w \sim p}\left[\phi(s, w)\phi(s', w)\right],$$

where $\phi(s, w)$ is termed as a random feature and p is some probability measure. Example of kernels that verify this condition are translation invariant kernels that can be written $K(s, s') = \kappa(s - s')$ where the probability measure p can be obtained with κ using the Bochner

theorem (Vert and Mairal, 2020). Then, the random features $\phi(s, w)$ can be constructed using random Fourier features (Rahimi and Recht, 2007) and samples $w_1, \ldots, w_m$ can be drawn from p to define a finite dimensional mapping $\psi(s) = \frac{1}{\sqrt{m}}(\phi(s, w_1), \ldots, \phi(s, w_m))^\top$ so that when m is large, we have:

$$K(s, s') \approx \langle \psi(s), \psi(s') \rangle_2.$$

It is then possible to study the influence of m to obtain generalization bounds (Rudi and Rosasco, 2017; Bach, 2017) in learning problems.

1.6. Online policy learning in bandits

In this section, we formalize the *bandit* problem and present foundations to understand some of the contributions of this thesis. While the previous section introduced notions on statistical learning and in particular on counterfactual learning methods for the offline logged bandit problem, the present section actually introduces methods on sequential learning for "online" bandits. The interested reader may find more details in (Bubeck and Cesa-Bianchi, 2012; Lattimore and Szepesvári, 2020) as well as the tutorial (Foster and Rakhlin, 2022) and the lecture notes (Gaillard, 2022).

A bandit problem is a sequential game between an *agent* and an *environment*. The agent plays for T rounds where T is called *horizon*. At each round $t \in [T]$, the learner (agent) first chooses an *action* (arm) a_t from a given set $\mathcal{A}$, and the environment then reveals a reward $r_t \in \mathbb{R}$ where $r_t = r(a_t, t)$ where r is a reward function that can be arbitrary or stochastic. In the bandit literature (Lattimore and Szepesvári, 2020) the multi-armed bandit setting (Thompson, 1933; Robbins, 1952; Lai and Robbins, 1985), or k-armed bandit setting refers to the setting where there are at least two arms. The learner only takes action based on the previous history $(a_1, r_1, \cdots, a_{t-1}, r_{t-1})$. A policy π uses all previous information from the history to take actions and the goal for an agent is often to find a policy that chooses actions that lead to the largest possible cumulative reward over all T rounds, which is $\sum_{t=1}^{T} r_t$.

First, the main difficulty in bandits lies in that the environment is unknown to the agent. When an agent learns, it only supposes that the environment lies in an environment class. A large environment class corresponds to less knowledge by the agent. Second, to evaluate an agent, the notion of regret is used, which is the difference between the total expected reward using a policy π for T rounds and the total expected reward collected by the agent over T rounds. The regret relative to a policy class Π is the maximum regret relative to any policy $\pi \in \Pi$. Therefore, if the policy class Π is large enough, it may include the optimal policy for all environments in the environment class. Thus, a large policy class means that the regret is a more demanding criteria. In bandit algorithms (Lattimore and Szepesvári, 2020), the aim is to define algorithms with assumptions that make the regret meaningful and so that there exist policies with small regret.

Stochastic Bandits A simple problem that we focus on in this thesis is that of stochastic bandits. A bandit is stochastic when the sequence of rewards associated to any action is independent and identically distributed according to some distribution. This stochasticity thus corresponds to an assumption on the environment class, when $r(a, t) \sim \nu_a$ where ν_a is a

At each time step $t = 1, \cdots T$
- the agent chooses an action $a_t \in \mathcal{A}$ from the policy π
- given a_t, the environment draws the reward $r(a_t, t) \sim \nu_{a_t}$
- the agent observes the feedback $r_t = r(a_t, t)$ and updates its policy π

Stochastic Bandit setting

stochastic distribution. This assumption will be relaxed in Section 1.7. For some applications indeed, the assumption that the rewards are stochastic may be too restrictive. The objective in stochastic bandits is to minimize the cumulative regret:

$$R_T = \max_{a \in \mathcal{A}} \sum_{t=1}^{T} r(a, t) - \sum_{t=1}^{T} r_t.$$

In stochastic bandits, we generally assume that the sequences to be i.i.d. Each arm a is associated to an unknown probability distribution ν_a over $[0, 1]$ and $r(a, t) \sim \nu_a$. The player aims at finding the arm with the highest reward. We also denote:

$$\mu_a = \mathbb{E}[r(a, t)], \quad \text{and} \quad \mu^* \in \arg\max_{a \in \mathcal{A}} \mu_a$$

As a matter of fact, it is sometimes hard to design algorithms for the true expected regret. In Bernoulli bandits, for example when $\nu_a \sim \mathcal{B}(1/2)$, for $a = 1, \ldots, k$ when $|\mathcal{A}| = k$, we have that for any arm $a \in \mathcal{A}$ $\mathbb{E}[r(a, t)] = 1/2$ and for any chosen action a_t by the learner at round t, $\mathbb{E}[r(a_t, t)] = 1/2$. The maximum cumulated sum of rewards is then a random walk which expected magnitude is of order:

$$\mathbb{E}\left[\max_{a \in \mathcal{A}} \sum_{t=1}^{T} r(a, t)\right] \approx \sqrt{T \log k}$$

Therefore, in that case even though all arms are optimal, the expected regret is of order $\sqrt{T \log k}$ and cannot be less. In the stochastic setting, we thus consider a quantity called the pseudo-regret which corresponds to competing with the best action in expectation, rather than the optimal action on the sequence of realized rewards. The pseudo regret is defined as:

$$\bar{R}_T = T\mu^* - \mathbb{E}\left[\sum_{t=1}^{T} \mu_{a_t}\right] \tag{1.31}$$

Note that the pseudo-regret is upper-bounded by the expected regret $\bar{R}_T \leq \mathbb{E}[R_T]$. This explains why it is harder to design algorithms that minimize the true expected regret. We will then use the pseudo-regret in the following.

In the next parts, we will write $\hat{\mu}_a(t)$ the empirical mean of rewards obtained when pulling arm a after t rounds. Let us also denote for all arms $a = 1, \ldots, k$ the suboptimal gap by:

$$\Delta_a = \mu^* - \mu_a, \tag{1.32}$$

and the following quantity:

$$T_a(t) = \sum_{s=1}^{t} \mathbb{1}\{a_s = a\}, \tag{1.33}$$

be the number of times action a was chosen by the agent after the end of round t. In general, the latter quantity $T_a(t)$ is random, even if the leaner uses a deterministic policy: this stems from the stochasticity of the rewards.

Lemma 1.6.1 (Regret decomposition lemma)**.** *For any policy π and stochastic bandit environment ν with $\mathcal{A}$ finite with $\mathcal{A} = k$ and horizon $T \in \mathbb{N}$, the pseudo-regret $\bar{R}_T$ of policy π in ν satisfies:*

$$\bar{R}_T = \left(\sum_{a=1}^{k} \mathbb{E}[T_a(T)]\right) \mu^* - \mathbb{E}\left(\sum_{a=1}^{k} T_a(T)\mu_a\right) = \sum_{a=1}^{k} \Delta_a \, \mathbb{E}[T_a(T)] \tag{1.34}$$

The latter lemma separates the regret in terms of losses due to each arm is conceptually important. Indeed, an agent should aim at using an arm with a larger suboptimality gap fewer times to minimize the regret.

1.6.1. Optimism in the Face of Uncertainty Learning principle

In this part, we introduce some "optimistic" bandit algorithms, namely the upper confidence bound methods. The Optimism in the Face of Uncertainty Learning (OFUL) principle is at the core of methods developed in Chapter 4.

Upper Confidence Bound Algorithm

The Upper Confidence Bound (UCB) algorithm (Auer et al., 2002) uses the Optimism in the Face of Uncertainty Learning (OFUL) principle (Abbasi-yadkori et al., 2011) which leads to taking actions as if the outcome would be as great as possible, given a level of confidence. This algorithm has the advantage to not rely on an initial exploration phase but rather explores on the fly as observations come. Moreover, the algorithm does not require knowledge of gaps and explores and exploits sequentially throughout the game.

Formally, at any round t, for each arm a, the agent builds a confidence interval $I_a(t)$ on its expected reward based on past observation $(a_1, r_1, \cdots, a_{t-1}, r_{t-1})$:

$$I_a(t) = [\text{LCB}_a(t), \text{UCB}_a(t)],$$

where LCB is a lower confidence bound of the expected reward of the arms and UCB is an upper confidence bound. The agent then acts optimistically in the sense that it chooses the arm with the best "plausible" reward, that is with the highest upper confidence bound:

$$a_t \in \arg\max_{a \in \mathcal{A}} \text{UCB}_a(t).$$

The intuition is that by pulling arms through all rounds up until the horizon, the agent can optimistically explore and exploit through adjusting the confidence intervals to

discard unconvincing arms. The natural question then is to ask how to design the upper-confidence-bounds. First, we define an empirical estimate of the means of all arms a as follows:

$$\hat{\mu}_a(s) = \frac{1}{s}\sum_{s'=1}^{s} \mathbb{1}\{a = a_{s'}\} r(a, s'), \tag{1.35}$$

which has an expectation mean μ_a. Therefore, to design a confidence interval, we can use the Hoeffding's inequality to have that for all arms $a \in \{1, \dots, k\}$, for all $s \geq 1$ and all $\delta \in [0, 1]$:

$$\mathbb{P}\left(\mu_a \geq \hat{\mu}_a(s) + \sqrt{\frac{\log \frac{1}{\delta}}{2s}}\right) \leq \delta. \tag{1.36}$$

At round t, the learner has observed $T_a(t-1)$ samples from arm a and received rewards from that arm with an empirical mean of $\hat{\mu}_a(t-1)$. Then a reasonable candidate for an upper confidence bound of the unknown mean of arm a is:

$$\text{UCB}_a(t-1) = \begin{cases} \infty & \text{if } T_a(t-1) = 0 \\ \hat{\mu}_a(T_a(t-1)) + \sqrt{\dfrac{2\log(1/\delta)}{T_a(t-1)}} & \text{otherwise.} \end{cases} \tag{UCB}$$

Algorithm 2: Upper Confidence Bound Algorithm (UCB)

Input: Action set $\mathcal{A}$ and level of confidence δ
for $t = 1$ *to* n **do**
  Choose action $a_t = \arg\max_{a \in \mathcal{A}} \text{UCB}_a(t-1)$
  Observe reward $r_t = X_{a_t,t}$ and update $\text{UCB}_a(t)$ with updates of Eq. (1.33) and Eq. (1.35) and δ ;
end

It is then possible to bound the pseudo-regret of (UCB) and provide the following theorem as in (Lattimore and Szepesvári, 2020).

Theorem 1.6.1. *If the distributions ν_a have supports included in $[0, 1]$ then for all a such that Δ_a*

$$\mathbb{E}\left[T_a(T)\right] \leq \frac{8\log(T)}{\Delta_a^2} + 2. \tag{1.37}$$

In particular, this implies that the pseudo-regret of (UCB) *is upper bounded as*

$$\bar{R}_T \leq 2k + \sum_{a, \Delta_a > 0} \frac{8\log(T)}{\Delta_a} \tag{1.38}$$

(UCB) has a regret bound of order

$$\bar{R}_T \leq \frac{8k(\log(T)+1)}{\Delta}, \tag{1.39}$$

where $\Delta = \min_{a,\Delta_a>0} \Delta_a$. Then, by reformulating Eq. (1.37) into:

$$\Delta_a \leq 2\sqrt{\frac{2\log T}{\mathbb{E}\left[T_a(T)\right]-2}}, \tag{1.40}$$

and using Eq. (1.34) we can obtain by using a Jensen inequality:

$$\bar{R}_T \lesssim \sqrt{Tk\log(T)}. \tag{1.41}$$

The bound is close to the lower bound that is of order $O(\sqrt{kT})$. As a matter of fact, it is possible to match that lower bound and remove the logarithmic term. The MOSS (Minimax Optimal Strategy in the Stochastic case) algorithm (Audibert and Bubeck, 2009) in particular depends on the smallest gap Δ but achieves the upper bound:

$$\bar{R}_T \lesssim \min\left\{\sqrt{Tk}, \frac{k}{\Delta}\log\frac{T\Delta^2}{k}\right\}. \tag{1.42}$$

Note eventually that there exists other algorithms in the literature for this problem. Two of the most used algorithms in practice are ε-greedy algorithm and Thompson sampling (Thompson, 1933). ε-greedy algorithm samples the arm with the best empirical mean with probability $\varepsilon \in [0,1]$ and explores by playing a random arm with probability $1-\varepsilon$. When Δ is known, such an algorithm can be calibrated to obtain an upper bound of order $\bar{R}_T \lesssim k\log(T)/\Delta^2$. Thompson sampling assumes a prior over the expected rewards μ_a, then at each round $t \geq 1$, for each arm, it computes $\hat{\nu}_{a,t}$ the posterior distribution of the rewards of an arm a given the rewards observed in history. Then, it samples a parameter $\theta_{a,t} \sim \hat{\nu}_{a,t}$ independently and selects an arm subsequently $a_t \in \arg\max_{a\in\mathcal{A}} \theta_{a,t}$. Thompson sampling has a similar bound as UCB of order $\bar{R}_T \lesssim k\log(T)/\Delta$ but has the advantage to easily incorporate prior knowledge on arms.

1.6.2. Stochastic Linear Bandits

Stochastic linear bandits (Li et al., 2010; Abbasi-yadkori et al., 2011) use another model: at round t, the agent is given the time-dependent decision set $\mathcal{A}_t \subset \mathbb{R}^d$ from which it selects an action $a_t \in \mathcal{A}_t$ and receives the reward:

$$r_t = \langle \theta^*, a_t \rangle + \varepsilon_t, \tag{1.43}$$

where ε_t are i.i.d. centered subGaussian noise given $\mathcal{A}_1, a_1, r_1, \ldots, \mathcal{A}_{t-1}, a_{t-1}, r_{t-1}, \mathcal{A}_t, a_t$ and $\theta^* \in \mathcal{H}$ is an unknown parameter. The pseudo regret then writes as:

$$\bar{R}_T = \mathbb{E}\left[\sum_{t=1}^{T} \max_{a\in\mathcal{A}_t} \langle \theta^*, a\rangle - \sum_{t=1} r_t\right] \tag{1.44}$$

Note here that the time-dependency of $\mathcal{A}_t$ is crucial as it allows to consider a contextual linear bandit problem $\mathcal{A}_t = \{\phi(x_t, a) : a \in \mathcal{A}\}$ where x_t is a context in a space $\mathcal{X}$. This allows for the contextual bandit extension that is studied in Chapter 4. Note also that it is clearly possible to recover the multi-arm bandit setting with $\mathcal{A}_t = \{e_1, \dots, e_d\}$ where $(e_i)_{i=1,\dots,d}$ are the unit vectors of $\mathbb{R}^d$.

The generalization of the previous UCB algorithm is based on the intuition to maintain for each possible action an estimate of the mean reward as well as a confidence interval around that mean. Then, at each time the agent chooses the highest upper confidence bound. Formally, if we have a confidence set $\mathcal{C}_t \subset \mathbb{R}^d$ based on samples (x_t, a_t, y_t), for $t \in \{1, \dots, T\}$ that contains the unknown parameter vector θ^* with high probability, we may define:

$$\text{LinUCB}_t(a) = \max_{\theta \in \mathcal{C}_t} \langle \theta, a \rangle \tag{1.45}$$

as an upper bound on the mean pay-off $\langle \theta^*, a \rangle$ of a. To choose the highest upper confidence bound from the confidence set at time t, the algorithm then selects:

$$a_t \in \arg\max_{a \in \mathcal{A}_t} \text{LinUCB}_t(a). \tag{1.46}$$

The next step is to construct a confidence set $\mathcal{C}_t$. To do so, we look for two essential properties: (i) $\mathcal{C}_t$ should contain θ^* with high probability and (ii), $\mathcal{C}_t$ should be as small as possible to control the actions selected. Therefore, following the idea of UCB, instead of empirically estimating the arms' unknown means, we will estimate θ^*. To do so, we build an empirical estimate of θ^* using regression. More precisely, we use the regularized least square estimator:

$$\hat{\theta}_t \in \arg\min_{\theta} \sum_{s=1}^{t-1} \left(\langle \theta, a_s \rangle - r_s\right)^2 + \lambda \|\theta\|_2^2 \tag{1.47}$$

where λ is a regularization parameter. $\lambda > 0$ ensures the minimization problem is well posed when previously sampled actions $a_1, \dots, a_t$ do not span $\mathbb{R}^d$. When defining the following quantities:

$$V_t = \sum_{s=1}^{t} a_s a_s^\top + \lambda I \text{ and } V_0 = \lambda I, \tag{1.48}$$

the solution to Eq. (1.47) is analytically obtained as:

$$\hat{\theta}_t = V_t^{-1} \sum_{s=1}^{t} a_s r_s \tag{1.49}$$

Since $\hat{\theta}_t$ is an estimate of θ^*, we can design an ellipsoidal confidence set $\mathcal{C}_t$ centered around $\hat{\theta}_t$. We then define

$$\mathcal{C}_t = \{\theta \in \Theta : \|\theta - \hat{\theta}_t\|_{V_t} \leq B(\delta)\} \tag{1.50}$$

where $\|\theta\|_V = \theta^T V \theta$, and B is a bound on δ that is to be defined. When the rounds t pass, the matrix V_t has increasing eigenvalues, therefore the volume of the ellipse is also shrinking so long as the latter quantity B does not grow too fast. Eventually, with $\mathcal{C}_t$ in the form of an ellipsoid with center $\hat{\theta}$ and radius $\beta = B(\delta)$, we can write analytically the solution of Eq. (1.45). Indeed, note that by defining $B_2 = \{x \in \mathbb{R}^d : \|x\|_2 \leq 1\}$ the unit ball with the Euclidean norm, it is easy to see that $\mathcal{C}_t = \hat{\theta}_t + \beta^{1/2} V^{-1/2} B_2$. Therefore, for $\theta \in B_2$ maximising the quantity $\langle \theta, a \rangle = a^T \hat{\theta}_t + \beta^{1/2} a^T V^{-1/2} \theta$, immediately gives that:

$$\text{LinUCB}_t(a) = \langle \hat{\theta}_t, a \rangle + \beta^{1/2} \|a\|_{V_t^{-1}} \tag{LinUCB}$$

The (LinUCB) policy is summarized in Algorithm 3.

Algorithm 3: Linear Upper Confidence Bound Algorithm (LinUCB)

Input: Action set $\mathcal{A}$ and tuning parameter β
for $t = 1$ *to* n **do**
  Choose action $a_t = \arg\max_{a \in \mathcal{A}} \langle \hat{\theta}_{t-1}, a \rangle + \beta^{1/2} \|a\|_{V_{t-1}^{-1}}$
  Observe reward r_t and update V_t in (1.48) and estimate $\hat{\theta}_t$ with (1.49);
end

We now introduce some results for discussing the regret guarantees of the (LinUCB) algorithm. We start by the following lemma (Lattimore and Szepesvári, 2020).

Lemma 1.6.2. *Let $\delta \in [0, 1]$. Then, with probability at least $1 - \delta$, if $\max_{a \in \mathcal{A}_t} \|a\|_2 \leq 1$, for all $t \geq 1$:*

$$\|\hat{\theta}_t - \theta^*\| \leq \sqrt{\lambda} \|\theta^*\| + \sqrt{2 \log \frac{1}{\delta} + d \log \left(1 + \frac{T}{\lambda}\right)} \tag{1.51}$$

It is then natural to define $B(\delta)$ as follows:

$$B(\delta) = \sqrt{\lambda} \|\theta^*\| + \sqrt{2 \log \frac{1}{\delta} + d \log \left(1 + \frac{T}{\lambda}\right)} \tag{1.52}$$

Remark 1.6.1. *Note that here, the definition of $B(\delta)$ depends on the quantity $\|\theta^*\|$ which is unknown. When running the algorithm, to define $\beta = B(\delta)$, we need to use an upper bound of the value $\|\theta^*\|$.*

Eventually, we can show the following upper bound on the pseudo-regret.

Theorem 1.6.2. *Let $T \geq 1$ and $\theta^* \in \mathbb{R}^d$. Assume that for all $a \in \mathcal{A}_t$, $|\langle \theta^*, a \rangle| \leq 1$, with $\|\theta^*\| \leq 1$ and $\|a_t\| \leq 1$, then LinUCB satistifies the pseudo regret bound:*

$$\bar{R}_T \leq C_\lambda d \sqrt{T} \log T, \tag{1.53}$$

where C_λ is a universal constant that depends on λ.

Note that under further assumptions, it is possible to improve the latter bound. When the set of available actions at time t is fixed and finite, with cardinal $|\mathcal{A}| = k$, elimination

algorithms (Chu et al., 2011; Lattimore and Szepesvári, 2020) allow to achieve the upper bound:

$$\bar{R}_T \leq C_\lambda \sqrt{Td \log(Tk)}$$

which improves the previous (LinUCB) bound by a factor $\sqrt{d}/\log k$ and thus achieves the optimal rate of order $O(\sqrt{dT})$ (Foster and Rakhlin, 2022). However such methods are not practical (Valko et al., 2013) for real-world applications in the sense that the elimination phases too often take suboptimal actions.

Contextual Stochastic Bandits

Contextual stochastic bandits are a class of problem where at each round t, the agent receives a context $x_t \in \mathcal{X}$ that is drawn from a stochastic distribution. The agent then chooses an action conditionally on that context. This setting generalise the previous multi-armed setting by allowing the learner to make use of side information, which is more realistic for many applications.

As stated before, a contextual bandit extension is possible (Chu et al., 2011) with the previous LinUCB algorithm by considering a time-dependent set of action $\mathcal{A}_t$ as $\mathcal{A}_t = \{\phi(x_t, a) : a \in \mathcal{A}\}$ where x_t is a context in a space $\mathcal{X}$ and ϕ is a feature map associated to a kernel. An extension of the (LinUCB) setting to a contextual bandit setting with kernel methods is presented in Chapter 4 as a (K-UCB) rule. In particular, we extend in this thesis the standard analysis of the OFUL algorithm for linear bandits (Abbasi-yadkori et al., 2011; Chowdhury and Gopalan, 2017) to the kernel setting using martingale argument and non-trivial extensions of concentration bounds to infinite-dimensional objects. We present below a result that is given in Chapter 4.

Theorem 4.3.1. *Let $T \geq 2$ and $\theta^* \in \mathcal{H}$. Under some boundedness assumptions detailed in the full statement in Chapter 4, the K-UCB rule defined in Eq.* (1.46) *satisfies the pseudo-regret bound*

$$R_T \lesssim \sqrt{T}\Big(\|\theta^*\|\sqrt{\lambda d_{\text{eff}}(\lambda, T)} + d_{\text{eff}}(\lambda, T)\Big)\,.$$

We will present in the next subsection the details on kernel methods that are key to understand the other contributions made in Chapter 4. Eventually, note that other methods relying on regression oracles (Agarwal et al., 2014; Foster and Rakhlin, 2020; Simchi-Levi and Xu, 2022) have been proposed for the contextual bandit task but are out of the scope of the (OFUL) principle.

1.6.3. Batch sequential policy learning

For the practical problems presented in Section 1.1, a realistic setting is to assume that the data collected is used to design a policy that is redeployed and used to collect additional data and reiterate the learning process, as in Chapter 3. As a matter of fact, many experiments such as clinical trials are typically conducted in batches, where groups of patients are treated

concurrently, and the data collected from each batch is indeed utilized to inform the design of subsequent batches.

This setting has been clarified by important theoretical works under different settings. In the reinforcement learning (RL) literature (Sutton and Barto, 1998), such methods have been referred to as *off-policy* learning algorithms. However, RL methods assume transitions in states that are thus more complicated than the basic assumptions of the basic bandit settings, where an action does not influence the sampled contexts. Using the multi-armed bandit framework instead as in (Perchet et al., 2015), it is possible to answer important questions related to conducting such experiments, for example what can be accomplished with a limited number of batches, how large these batches should be, and how outcomes in one batch should inform the structure of subsequent batches. This framework presents an exploration-exploitation dilemma that needs to be carefully considered. We will present a basic algorithm in the contextual batch bandit setting to understand methods that are closely related to the one we study in Chapter 3.

Let T be the time horizon of the problem. At the beginning of each time $t \in [T]$, the decision maker observes contexts $x_t \in \mathcal{X}$ where $\mathcal{X} \subset \mathbb{R}^d$. When the decision maker selects an action $a \in \mathcal{A}$, a reward r_t as in Eq. (1.43):

$$r_t = \langle \theta^*, \phi(x_t, a_t) \rangle + \varepsilon_t, \tag{1.54}$$

where ε_t is a sequence of zero-mean independent sub-Gaussian random variables that we can assume to be 1-sub-Gaussian. Unlike the standard online setting where the decision maker immediately observes the reward r_t after taking an action a_t, the reward can only be seen at the end of batch $m \in [1, \dots M]$ where the horizon T is partitionned into M units. More specifically, given a total batch size M, a sequential batch bandit algorithm has:

1. A grid $\mathcal{T} = \{t_1, t_2, \dots, t_M\}$ with $t_0 = 0$ and $t_M = T$. Intuitively, this grid partitions the T units into M batches: the m-batch contains units of samples t_{m-1} to t_m. The decision maker can choose in its strategy a grid $\mathcal{T}$ or that grid can be imposed in the problem.
2. A sequential batch policy $\pi = (\pi_1, \pi_2, \cdots, \pi_M)$ such that each π_m can only use information from all the prior batches (contexts, actions and rewards $(x_t, a_t, r_t)_{t=1,\cdots,t_m}$)

To assess the performance of a sequential batch bandit algorithm, we also use a pseudo regret metric as in Eq. (1.44):

$$\bar{R}_T = \mathbb{E}\left[\sum_{t=1}^{T} \max_{a \in \mathcal{A}_t} \langle \theta^*, \phi(x_t, a) \rangle - \sum_{t=1} r_t\right] \tag{1.55}$$

Even though the pseudo regret defined in this context aligns with that of standard online learning in Eq. (1.44), it encompasses a more ambitious objective due to the presence of delays induced by batches in obtaining reward feedback. This results in a situation where the decision maker cannot promptly incorporate feedback into their subsequent decision-making process. Moreover, this allows to compare the performance to that of an oracle utilized in the standard online learning scenario.

A sequential Batch UCB algorithm

Following the (OFUL) principle, the most straightforward approach in batch contextual bandits is to extend (Han et al., 2020) the (LinUCB) algorithm to a batch setting as follows. It is also possible to define the following quantities, as in Eq. (1.48) and Eq. (1.57):

$$V_m = V_{m-1} + \sum_{s=t_{m-1}}^{t_m} \phi(x_s, a_s)\phi(x_s, a_s)^\top \text{ and } V_0 = \lambda I, \tag{1.56}$$

the solution to Eq. (1.47) is analytically obtained as:

$$\hat{\theta}_m = V_m^{-1} \sum_{s=1}^{t_m} r_s \phi(x_s, a_s) \tag{1.57}$$

Using the latter quantities, it is then possible to define a confidence interval $\mathcal{C}_m$ as in Eq. (1.50) and an update rule as in (LinUCB). The Sequential Batch UCB algorithm is then summarized in Algorithm 4.

Algorithm 4: Sequential Batch UCB (SBUCB) (Han et al., 2020)

Input: Action set $\mathcal{A}$, grid $\mathcal{T} = \{n_1, \cdots n_M\}$, tuning parameter β
for $m = 1$ *to* M **do**
  for $n = 1$ *to* n_m **do**
    Choose action $a_t = \arg\max_{a \in \mathcal{A}} \langle \hat{\theta}_{m-1}, \phi(x_t, a) \rangle + \beta^{1/2} \|\phi(x_t, a)\|_{V_{m-1}^{-1}}$
  end
  Observe rewards of the m=th batch $(r_t)_{t=t_{m-1},\cdots t_m}$ and update V_m in (1.56) and estimate $\hat{\theta}_m$ with (1.57);
end

Han et al. (2020) shows the following theorem in the finite action case and under the assumption that the action set is finite and that is the dimension d is relatively low compared to T.

Theorem 1.6.3. *(Han et al., 2020) Let $T \geq 1$ and $\theta^* \in \mathbb{R}^d$. Assume that for all $a \in \mathcal{A}_t$, $|\langle \theta^*, a \rangle| \leq 1$, with $\|\theta^*\| \leq 1$ and $\|a_t\| \leq 1$, then SBUCB satistifies the pseudo regret bound:*

$$\bar{R}_T \leq C_\lambda \sqrt{\frac{T}{M}} \left(d\sqrt{\frac{T}{M}} + \sqrt{Md} \right) \log T \log Tk, \tag{1.58}$$

where C_λ is a universal constant that depends on λ.

This theorem is important as it shows that taking a number of batches in the order of $\sqrt{dT}$ should allow to recover the same optimal rate of $O(\sqrt{dT})$ as in (Chu et al., 2011). Nevertheless, $O(\sqrt{dT})$ can be a large number and conversely, if only a constant number of batches are available, then the regret is linear.

In Chapter 3, we instead require less assumptions on the action set $\mathcal{A}$, nor on the dimension d of the context space $\mathcal{X}$. Moreover, instead of deriving adaptive strategies from the (OFUL) principle, companies might be interested in sequential designs of policies that are learned with the conservative CRM offline learning principle.

1.7. Going further: no-regret algorithms in online optimization

We now end this introduction with this section which objective is to provide background and an introduction to the *dual averaging* technique (Nesterov, 2009) that is at the core of the analysis of the algorithms discussed in Chapter 5. In particular, we present *no-regret* algorithm in online optimization which is the sequential learning setting that is considered in that chapter.

Notations Throughout what follows, $\mathcal{V}$ will denote a finite-dimensional real space with norm $\|\cdot\|$ and $\mathcal{A} \subset \mathcal{V}$ will be a closed convex subset thereof. We will also write $\mathcal{V}^*$ for the (algebraic) dual of $\mathcal{V}$, $\langle y, a\rangle$ for the canonical pairing between $y \in \mathcal{V}^*$ and $x \in \mathcal{V}$, and $\|y\|_* = \sup\{\langle y, a\rangle : \|a\| \leq 1\}$ for the dual norm of $y \in \mathcal{V}^*$.

Online optimization is concerned with solving a series of decision problems over time and is more general than the bandit problems presented before. The goal is to minimize the overall loss experienced over a sequence of unknown loss functions that are arbitrary unlike the stochastic assumptions considered in Section 1.6. In essence, the standard online optimization scenario can be described as a sequence of steps where at each stage, the agent (learner) chooses an action $a_t \in \mathcal{A}$ that incurs a loss $l_t(a_t)$ based on a loss function $l_t : \mathcal{A} \to \mathbb{R}$ which is received by the learner. The learner then updates their actions and the process repeats.

At each time step $t = 1, \cdots T$
- the agent chooses an action $a_t \in \mathcal{A}$
- given a_t, an arbitrary loss $l_t(a_t)$ is incurred
- the agent observes a feedback and updates its action a_{t+1}

Online optimization setting

Conceptually, it is extremely important to note here that an action a here can refer to a distribution on a set of arms $1, \ldots, k$, that is to say $\mathcal{A} = \Delta(k)$ is the simplex on $\mathbb{R}^k$. This stems from the idea that a learner does not define a strategy by choosing a deterministic arm as in stochastic bandits, but rather defines a distribution on possible arms to sample arms and prevent *adversaries* to manipulate the losses l_t.

Based on some properties of l_t, we can consider the following basic problems: the online convex (respectively strongly convex) optimization where l_t is assumed convex (respectively strongly convex) or online linear optimization where each l_t is assumed linear, i.e of the form $l_t(a) = -\langle v_t, a\rangle$ for some payoff vector $v_t \in \mathcal{V}$. Note that linear and strongly convex problems are both convex problems but otherwise different. For the rest of this introduction, we will

consider linear problems and require that each l_t is differentiable and attains its minimum in $\mathcal{A}$.

To measure the performance of learner, the notion of regret, similarly to Section 1.6, is defined as:

$$R_T = \max_{a \in \mathcal{A}} \sum_{t=1}^{T} \left[l_t(a_t) - l_t(a) \right]. \tag{1.59}$$

which is also a difference of the cumulated loss incurred by the agent after T stages and that of the best action in hindsight. Contrary to the definition given in 1.6, the loss l_t is arbitrary and is not drawn from a fixed distribution as before. This makes a drastic difference and will enable us to consider *adversarial* bandits (Lattimore and Szepesvári, 2020). For both cases, the agent's regret contrasts the performance of the agent's policy a_t to that of an action $a^* \in \arg\min_{a \in \mathcal{A}} \sum_{t=1}^{T} l_t(a)$ which minimizes the cumulated incurred loss over all rollouts. Instead, we also consider a pseudo metric regret that is defined as in Eq. (1.31)

$$\bar{R}_T = \max_{a \in \mathcal{A}} \mathbb{E} \left[\sum_{t=1}^{T} \left[l_t(a_t) - l_t(a) \right] \right]. \tag{1.60}$$

The goal in online optimization is to define algorithms that achieve *no regret*, that is:

$$\frac{R_T}{T} \underset{T \to +\infty}{\longrightarrow} 0. \tag{1.61}$$

As in stochastic bandits in Section 1.6, the effectiveness of a policy is then assessed based on the rate of the regret that is actually achieved, which is determined by examining the specific expression within which $\frac{R_T}{T}$ vanishes to zero.

Feedback assumptions In online optimization, it is possible to assume different level of information available to the learner. The access to the entire loss l_t can be given to the optimizer after an action a_t is chosen, which is termed as the *full information* feedback setting. This opposes to the *bandit feedback* as presented before (in the logged bandit feedback problem in Section 1.3 and in bandit problems in Section 1.6, 1.4) where only $l_t(a_t)$ is revealed to the learner.

Moreover, many online learning algorithm require gradient information. In the analysis that is used in works of the literature (Shalev-Shwartz, 2007; Zinkevich, 2003; Mertikopoulos, 2019) an assumption on imperfect gradient feedback is made so that when a gradient "oracle" is called at a point a_t, the learner has access to a gradient vector of the form:

$$\nabla_t = \nabla l_t(a_t) + Z_t, \tag{1.62}$$

where Z_t is defined as the "observational error" in the oracle gradient. Typically, we decompose Z_t in the form of:

$$Z_t = u_t + b_t \tag{1.63}$$

where u_t is zero mean and b_t captures the mean of Z_t. We then define the following statistics to classify the problems that return imperfect gradient feedback.

$$\begin{cases} B_t = \mathbb{E}\left[\|b_t\|\right] & \text{(bias)} \\ \sigma_t^2 = \mathbb{E}\left[\|u_t\|^2\right] & \text{(variance)} \\ M_t^2 = \mathbb{E}\left[\|\nabla_t\|^2\right] & \text{(second moment).} \end{cases} \tag{1.64}$$

No-regret algorithms

We now present no-regret algorithms and specifically make the link between *Follow-the-Regularized-Leader* and *Mirror Descent* strategies that are used in the literature (Lattimore and Szepesvári, 2020, see Chapter 28) to analyze adversarial bandit algorithms as we do in Chapter 5 of this thesis. Specifically, we introduce such strategies to eventually present the *dual averaging* method.

Leader-following policies Starting from the intuition that the optimizer can play the action that is optimal in hindsight up to stage t, it is possible to derive a no-regret strategy. This strategy is known as *follow-the-leader* (FTL) and can be expressed as:

$$a_{t+1} \in \arg\min_{a \in \mathcal{A}} \sum_{s=1}^{t} l_s(a). \tag{FTL}$$

However, it is known that this strategy induces a positive regret in simple examples where the loss l_t can oscillate from one round to another and be manipulated by an adversary. To circumvent this, it is possible to regularize the update rule with penalty term which that leads to the so-called *follow the regularized leaser* (FTRL) that can be given as:

$$a_{t+1} \in \arg\min_{a \in \mathcal{A}} \sum_{s=1}^{t} l_s(a) + \frac{1}{\gamma} h(a). \tag{FTRL}$$

Here, $h : \mathcal{A} \to \mathbb{R}$ is a regularization function and $\gamma > 0$ is a parameter that can be chosen by the learner to optimize its learning guarantees. It is then standard to require additional assumptions on h to provide such guarantees.

Assumption 1.7.1. *h is continuous and it is strongly convex, that is, there exists $C > 0$ such that:*

$$[\lambda h(a') + (1-\lambda)h(a)] - h(\lambda a' + (1-\lambda)a) \geq \frac{C}{2}\lambda(1-\lambda)\|a - a'\|^2 \tag{1.65}$$

for all $a, a' \in \mathcal{A}$ and $\lambda \in [0, 1]$.

Moreover, the regret analysis of (FTRL) is typically performed under the following assumption on the loss l_t.

Assumption 1.7.2. *Each l_t is convex and it is Lipschitz continuous, i.e:*

$$|l_t(a') - l_t(a)| \leq L_t\|a' - a\| \tag{1.66}$$

for some $L_t > 0$ and all $a, a' \in \mathcal{A}$.

Then, in this setting, the following result applies:

Theorem 1.7.1. *(Shalev-Shwartz, 2007) Assuming Assumptions 1.7.1 and 1.7.2 and supposing that* (FTRL) *is run against a sequence of loss functions l_t, $t = 1, \dots T$, then* (FTRL) *achieves no-regret with the regret bound:*

$$R_T \leq \frac{H}{\gamma} + \frac{\gamma}{C} \sum_{t=1}^{T} L_t^2 \tag{1.67}$$

where $H = \max h - \min h$ is the "depth" of h over $\mathcal{A}$. In particular, if $\sup_t L_t < \infty$ and writing $L = \sup_t L_t$ and if we set $\gamma = \frac{1}{L}\sqrt{\frac{HC}{T}}$, the incurred regret is bounded as:

$$R_T \leq 2L\sqrt{\frac{H}{C}T}. \tag{1.68}$$

This theorem illustrates how it is possible to achieve no-regret under a simple strategy and the dependencies of the regret on the regularization function h. Although the dependency on the horizon T is of order $\sqrt{T}$, the dependencies on the alternative set as discussed in Chapter 5 typically stem for the quantity H presented above.

Online gradient descent Another simple way to minimize the online loss is to use its gradient to take a step against it and repeat the process, as it is done in gradient descent in optimization. When faced with a different loss function at each stage, the policy derived from such a process is known as *online gradient descent* (OGD). Using the projection operator

$$P(a) = \arg\min_{a' \in \mathcal{A}} \|a' - a\|^2, \tag{1.69}$$

we define the update rules:

$$a_{t+1} = P(a_t + \gamma_t V_t) \tag{OGD}$$

where $\gamma_t > 0$ is the algorithm step size and V_t defined as

$$V_t = -[\nabla l_t(a_t) + Z_t] \tag{1.70}$$

with Z_t being defined as the "observational error" in the oracle gradient in Eq. (1.63). We illustrate the online gradient descent procedure in Figure 1.5.

We can now establish the following regret bound:

Theorem 1.7.2. *(Zinkevich, 2003) Assuming Assumption 1.7.2 and supposing that* (OGD) *is run against a sequence of loss functions l_t, $t = 1, \dots T$ with step size $\gamma_t = \gamma$, then* (OGD) *achieves satisfies the pseudo-regret bound:*

$$\bar{R}_T \leq \frac{\operatorname{diam}(\mathcal{A})^2}{2\gamma} + \frac{\gamma}{2} \sum_{t=1}^{T} M_t^2 + \operatorname{diam}(\mathcal{A}) \sum_{t=1}^{T} B_t \tag{1.71}$$

where $\operatorname{diam}(\mathcal{A}) = \max\{\|a - a'\| : a, a' \in \mathcal{A}\}$ denotes the diameter of $\mathcal{A}$. In particular, if $\sup_t M_t < \infty$ and writing $M = \sup_t M_t$ and if we set $\gamma = (1/M)\operatorname{diam}(\mathcal{A})/\sqrt{T}$, with unbiased feedback $B_t = 0$ the incurred regret is bounded as:

$$R_T \leq \operatorname{diam}(\mathcal{A}) M \sqrt{T}. \tag{1.72}$$

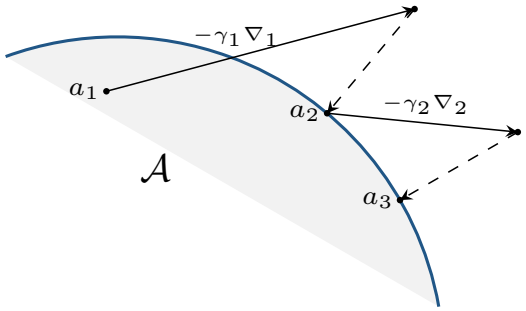

Figure 1.5: Representation of online gradient descent

Up to multiplicative constants, the bound in (1.72) is essentially the same as the corresponding bound in (1.68) for (FTRL) and are in order of $O(\sqrt{T})$; as long as the oracle does not suffer from systematic errors in (OGD). In other words, (OGD) achieves the same regret minimization rate as (FTRL), even though the latter requires a full information oracle.

Online Mirror Descent There are cases where taking into account the problem's geometry may allow for improved regret guarantees. A natural question that arises is whether running (OGD) with a non-Euclidean norm can lead to better regret bounds. For this, Online Mirror Descent (OMD) is a generalization of (OGD) to better exploit the geometry of the decision space $\mathcal{A}$. This algorithm is the online counterpart of the Mirror Descent algorithm from convex optimization.

To define it, let us first rewrite the projection defined in (1.69) as:

$$\begin{aligned} P(a+y) &= \arg\min_{a' \in \mathcal{A}} \{\|a + y - a'\|^2\} \\ &= \arg\min_{a' \in \mathcal{A}} \{\|a - a'\|^2 + \|y\|^2 + 2\langle y, a - a'\rangle\} \\ &= \arg\min_{a' \in \mathcal{A}} \{\langle y, a - a'\rangle + D(a', a)\} \end{aligned}$$

where

$$D(a', a) = \frac{1}{2}\|a' - a\|^2 = \frac{1}{2}\|a'\|^2 - \frac{1}{2}\|a\|^2 - \langle a, a - a'\rangle \tag{1.73}$$

is the squared Euclidean distance between a and a'. The generality of (OMD) comes from the updates being performed into a dual space which is defined by a C-strongly convex "distance generating function" $h : \mathcal{A} \to \mathbb{R}$. More particularly, by replacing the latter Euclidean distance with what we call the *Bregman divergence* induced by h:

$$D_h(a', a) = h(a') - h(a) - \langle \nabla h(a), a - a'\rangle \tag{1.74}$$

by then defining the *prox-mapping* for any $a \in \mathcal{A}$, $\text{Prox}_a(y) : \mathcal{V}^* \to \mathcal{A}$ as:

$$\underset{a}{\text{Prox}}(y) = \arg\min_{a' \in \mathcal{A}} \{\langle y, a - a'\rangle + D_h(a', a)\} \text{ for all } y \in \mathcal{V}^*. \tag{1.75}$$

Online Mirror Descent (OMD) is then defined as follows:

$$a_{t+1} = \underset{a_t}{\mathrm{Prox}}(\gamma_t V_t) \tag{OMD}$$

where γ_t is a variable step-size sequence, and the signals V_t as in (1.70).

Example 1.7.1. *Consider the quadratic distance generating function $h(a) = \frac{1}{2}\|a\|^2$ that gives the Euclidean prox-mapping:*

$$\underset{a}{\mathrm{Prox}}(y) = \underset{a' \in \mathcal{A}}{\arg\min}\{\langle y, a - a'\rangle + \frac{1}{2}\|a' - a\|^2\} = P(a + y). \tag{1.76}$$

We therefore recover the Euclidean gradient descemt with (OMD).

Example 1.7.2. *As an example, consider $\mathcal{A} = \Delta(k)$ the standard unit simplex of $\mathbb{R}^k$, and consider the entropic regularizer:*

$$h(a) = \sum_{j=1}^{k} a_j \log a_j. \tag{1.77}$$

A standard calculation shows that h is strongly convex and that the induced prox-mapping is given as:

$$\underset{a}{\mathrm{Prox}}(y) = \left(\frac{a_j \exp(y_j)}{\sum_{j'=1}^{k} a_{j'} \exp(y_{j'})}\right)_{1 \leq j \leq k} \tag{EGD}$$

which provides the entropic gradient descent update of :

$$a_{j,t+1} = \frac{a_{j,t} \exp(\gamma_t V_{j,t})}{\sum_{j'=1}^{k} a_{j',t} \exp(\gamma_t V_{j',t})}. \tag{1.78}$$

In the bandit literature, this algorithm update is known as exponential weights or Exponentiated Gradient forecaster.

We now provide the basic regret guarantees of (OMD).

Theorem 1.7.3. *(Shalev-Shwartz, 2007) Assuming Assumption 1.7.2 and supposing that* (OMD) *is run against a sequence of loss functions l_t, $t = 1, \dots T$ with step size $\gamma_t = \gamma$, then* (OMD) *achieves satisfies the regret bound:*

$$\bar{R}_T \leq \frac{H}{\gamma} + \frac{\gamma}{2C}\sum_{t=1}^{T} M_t^2 + \mathrm{diam}(\mathcal{A})\sum_{t=1}^{T} B_t, \tag{1.79}$$

where $\mathrm{diam}(\mathcal{A}) = \max\{\|a - a'\| : a, a' \in \mathcal{A}\}$ denotes the diameter of $\mathcal{A}$ and $H = \max h - \min h$ denotes the "depth" of h over $\mathcal{A}$. In particular, if $\sup_t M_t < \infty$ and writing $M = \sup_t M_t$ and if we set $\gamma = (1/M)\sqrt{2CH/T}$, with unbiased feedback $B_t = 0$ the incurred regret is bounded as:

$$R_T \leq M\sqrt{(2H/C)T}. \tag{1.80}$$

The main difference between the bounds of (OGD) and that of (OMD) is the factor $2H/C$. As a matter of fact, instead of the quantity $\mathrm{diam}(\mathcal{A})$, the geometry of the problem is taken into account in the terms H and C, as explained in the following example.

Example 1.7.3. *Going back to Example 1.7.2, the entropic regularizer in* (1.77) *has a strong convexity modulus $C = 1$ and its depth H over $\mathcal{A}$ is:*

$$H = \max h - \min h = 0 - \sum_{j=1}^{k} (1/k)\log(1/k) = \log k. \tag{1.81}$$

Hence if (EGD) *is run against a multi-armed bandit with bounded payoffs $\|l_t\|_\infty \leq 1$, we obtain a regret bound of the form:*

$$R_T \leq \sqrt{2T \log k}. \tag{1.82}$$

By comparison, the corresponding bound for (OGD) *is $R_T \leq 2\sqrt{kT}$ so* (EGD) *improves upon it by a factor $\sqrt{2}k/\log k$.*

Therefore, even both algorithms enjoy the same $O(\sqrt{T})$ regret bound, the difference in multiplicative constants can result in a substantial enhancement compared to the problem's dimension. This can be extremely beneficial for real-world machine learning, and is something that we specifically work on in Chapter 5.

The link between FTRL and OMD, the Dual Averaging We will now establish the relation between (FTRL) and (OMD) and introduce the *dual averaging* method. We first start by providing a simple example where (FTRL) and (OGD) strategies coincide.

Example 1.7.4. *Consider an unconstrained linear problem with action set $\mathcal{A} = \mathbb{R}^d$, with regularization function $h(a) = \frac{1}{2}\|a\|^2$, and linear losses of the form $l_t(a) = -\langle v_t, a\rangle$, for some sequence $v_t \in \mathbb{R}^d$. In that case, the* (FTRL) *update is expressed as:*

$$a_{t+1} = \operatorname*{arg\,min}_{a\in\mathcal{A}} \Big\{ \sum_{s=1}^{t} l_s(a) + \frac{1}{\gamma} h(a) \Big\} = \operatorname*{arg\,min}_{a\in\mathbb{R}^d} \Big\{ \|a\| - 2\gamma \sum_{s=1}^{t} \langle v_s, a\rangle \Big\} \tag{1.83}$$

$$= \operatorname*{arg\,min}_{a\in\mathbb{R}^d} \Big\| a - \gamma \sum_{s=1}^{t} \langle v_s, a\rangle \Big\| = \gamma \sum_{s=1}^{t} v_s = a_t + \gamma v_t \tag{1.84}$$

which is the unprojected gradient update of (OGD).

Actually, it is possible to modify the (FTRL) strategy with a gradient trick to require the same assumptions as (OMD) and establish relations between the two strategies. Specifically, it possible to define a variant of (FTRL) which only requires first-order oracle information that is, the same type of feedback as (OMD). The idea is to replace the loss $l_t(a)$ with the *linear* surrogate:

$$\tilde{l}_t(a) = l_t(a_t) + \langle \nabla l_t(a_t), a_t - a\rangle, \tag{1.85}$$

which yields the update:

$$a_{t+1} = \operatorname*{arg\,min}_{a\in\mathcal{A}} \Big\{ \sum_{s=1}^{t} \tilde{l}_s(a) + \frac{1}{\gamma} h(a) \Big\} = \operatorname*{arg\,max}_{a\in\mathcal{A}} \Big\{ \gamma \sum_{s=1}^{t} \langle \nabla l_s(a_s), a\rangle - h(a) \Big\}. \tag{1.86}$$

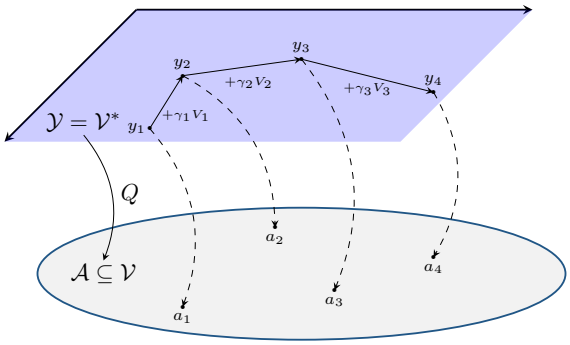

Figure 1.6: Representation of dual averaging

Contrary to (FTRL), this policy only requires first-order information on l_t (and coincides with it in case of linear losses). When the feedback available to the optimizer is a gradient signal V_t of the form (1.70), we can define the *follow-the-linearized-leader* (FTLL) policy:

$$a_{t+1} = \arg\max_{a \in \mathcal{A}} \Big\{ \gamma \sum_{s=1}^{t} V_s - h(a) \Big\}. \tag{FTLL}$$

Thus if we introduce the notion of the "mirror map" of h being defined for all $y \in \mathcal{V}^*$ as:

$$Q(y) = \arg\max_{a \in \mathcal{A}} \big\{ \langle y, a \rangle - h(a) \big\}. \tag{1.87}$$

We can write the (FTLL) in a recursive form with to yield the *dual averaging* method, with $\gamma_t > 0$ a variable step size parameter:

$$\left\{ \begin{array}{l} y_{t+1} = y_t + \gamma_t V_t \\ a_{t+1} = Q(y_{t+1}). \end{array} \right. \tag{DA}$$

Here, $y_t \in \mathcal{V}^*$ is an auxiliary dual variable that aggregates gradient steps. The name "dual averaging" is due to Nesterov (2009) and illustrates how gradients are "averaged" directly where they are in the dual space $\mathcal{V}^*$ before being "mirrorred" back through Q to the problem's original space $\mathcal{A}$. We provide in Figure 1.6 a schematic representation of the dual averaging procedure.

Example 1.7.5. *Going back to the quadratic regularizer $h(a) = \frac{1}{2}\|a\|^2$ that yielded an Euclidean projection for* (OMD)*, we now obtain the mirror map:*

$$Q(y) = \arg\max_{a \in \mathcal{A}} \big\{ \langle y, a \rangle - \frac{1}{2}\|a\|^2 \big\} = P(y), \tag{1.88}$$

where P is the projection defined in (1.69)*. We thus obtain the so-called lazy gradient descent update:*

$$\left\{ \begin{array}{l} y_{t+1} = y_t + \gamma_t V_t \\ a_{t+1} = P(y_{t+1}). \end{array} \right. \tag{LGD}$$

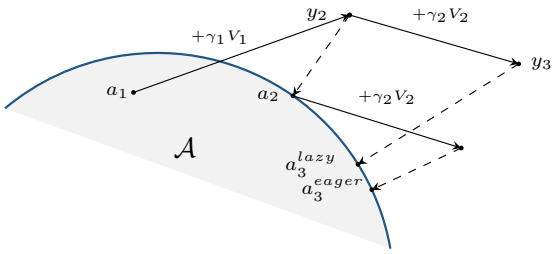

Figure 1.7: Representation of "lazy" and "eager" gradient descent

As a matter of fact, in the online learning literature, (DA) is often referred to as the “lazy” variant of (OGD) and (OMD) (Zinkevich, 2003; Shalev-Shwartz, 2012). This statement refers to the idea that the algorithm performs a "lazy" aggregation of gradient steps, in the sens that it doesn't transport the gradient steps to their original state. Instead, it only projects them to $\mathcal{A}$ in order to create a new gradient signal.

Actually, the selection of the distance-generating function h has a significant impact on establishing the relationship between (DA) and (OMD), which performs "eager" updates unlike (DA). A schematic difference between "lazy" and "eager" updates is provided in Figure 1.7.

As a matter of fact, Mertikopoulos (2019) studies under which conditions on h induces the same updates for the lazy and eager variants. In particular, h needs to intuitively be "steep" at the boundary of $\mathcal{A}$, but the discussions on this matter are out of the scope of this thesis. More importantly, we detail again the entropic regularization example with the (DA) update.

Example 1.7.6. *Going back to example 1.7.2, it is easy to show that the mirror map associated to the entropic regularizer* $h(a) = \sum_{j=1}^{k} a_j \log a_j$ *is the logit choice rule:*

$$\Gamma(y) = \left(\frac{\exp(y_j)}{\sum_{j'=1}^{k} \exp(y_{j'})} \right)_{j=1,\dots k}, \tag{1.89}$$

which gives the "Hedge" policy:

$$\begin{cases} y_{t+1} = y_t + \gamma_t V_t \\ a_{t+1} = \Gamma(y_{t+1}) \end{cases} \tag{Hedge}$$

when unfolding $a_{j,t+1} \propto a_{j,t} \exp(V_{j,t})$ *and given that* $\sum_{j=1}^{k} a_{j,1} = 1$ *in the simplex, the sequence of iterates of* (Hedge) *is the same as* (EGD).

The latter example demonstrates that for entropic regularization, the "lazy" and "eager" versions of (OMD) are the same. Therefore, we will not make much differentiation between the two variants of (OMD) due to the aforementioned reasons and will use the regret guarantees of Theorem 1.7.3 in the methods we will develop.

In Chapter 5, we provide results on the pseudo-regret of algorithms for adversarial multi-armed bandit problems. Specifically, when a similarity structure is known we can define a effective number of arm k_{eff} that is typically much smaller that k. Then, it is possible to improve the regret bound given by EXP4 (Auer et al., 2003) and prove the following with a (DA) analysis:

Theorem 5.5.1 (EWEN Regret)**.** *Suppose that Algorithm 15 is run with a non-increasing learning rate $\gamma_t > 0$ against a sequence of cost vectors $l_t \in [0,1]^{\mathcal{E}}$, $t = 1, 2, \ldots$, as per* (5.4)*. Then, the learner enjoys the regret bound*

$$\mathbb{E}[R_T] \leq \frac{H_{\mathcal{E}}}{\gamma_{T+1}} + \frac{k_{\text{eff}}}{2} \sum_{t=1}^{T} \gamma_t \tag{1.90}$$

with k_{eff} given by (5.13) *and $H_{\mathcal{E}}$ is defined as the depth over $\Delta(\mathcal{E})$ of the entropic regularizer $h_{\mathcal{E}}$ in* (5.12.1)*, i.e.,*

$$H_{\mathcal{E}} = \max h_{\mathcal{E}} - \min h_{\mathcal{E}} = \log M \tag{1.91}$$

In particular, if Algorithm 15 is run with $\gamma_t = \sqrt{\log M/(2t \cdot k_{\text{eff}})}$, we have

$$\mathbb{E}[R_T] \leq 2\sqrt{k_{\text{eff}} \log M \cdot T}. \tag{1.92}$$

Moreover, we propose in Chapter 5 a nested entropy on the similarity structure to propose the following regret bound that in turn improves the regret of EXP3 (Vovk, 1990; Littlestone and Warmuth, 1994; Auer et al., 1995) with a (DA) analysis.

Part I

Effective Counterfactual Learning in the Logged Bandit Feedback Problem

2

Counterfactual Learning of Stochastic Policies with Continuous Actions

Counterfactual reasoning from logged data has become increasingly important for many applications such as web advertising or healthcare. In this chapter, we address the problem of learning stochastic policies with continuous actions from the viewpoint of counterfactual risk minimization (CRM). While the CRM framework is appealing and well studied for discrete actions, the continuous action case raises new challenges about modelization, optimization, and offline model selection with real data which turns out to be particularly challenging. Our work contributes to these three aspects of the CRM estimation pipeline. First, we introduce a modelling strategy based on a joint kernel embedding of contexts and actions, which overcomes the shortcomings of previous discretization approaches. Second, we empirically show that the optimization aspect of counterfactual learning is important, and we demonstrate the benefits of proximal point algorithms and differentiable estimators. Finally, we propose an evaluation protocol for offline policies in real-world logged systems, which is challenging since policies cannot be replayed on test data, and we release a new large-scale dataset along with multiple synthetic, yet realistic, evaluation setups.

This chapter is based on the following material:

H. Zenati, A. Bietti, M. Martin, E. Diemert, and J. Mairal. Optimization approaches for counterfactual risk minimization with continuous actions. *International Conference on Learning Representation (ICLR), Causal Learning for Decision Making Workshop*, 2020b

H. Zenati, A. Bietti, M. Martin, E. Diemert, P. Gaillard, and J. Mairal. Counterfactual learning of stochastic policies with continuous actions: from models to offline evaluation. *arXiv preprint arXiv:2004.11722*, 2020a

2.1. Introduction

Logged interaction data is widely available in many applications such as drug dosage prescription (Kallus and Zhou, 2018), recommender systems (Li et al., 2012), or online auctions (Bottou et al., 2013). An important task is to leverage past data in order to find a good *policy* for selecting actions (*e.g.*, drug doses) from available features (or *contexts*), rather than relying on randomized trials or sequential exploration, which may be costly to obtain or subject to ethical concerns.

More precisely, we consider offline logged bandit feedback data, consisting of contexts and actions selected by a given *logging policy*, associated to observed rewards. This is known as *bandit feedback*, since the reward is only observed for the action chosen by the logging policy. The problem of finding a good policy thus requires a form of *counterfactual* reasoning to estimate what the rewards would have been, had we used a different policy. When the logging policy is stochastic, one may obtain unbiased reward estimates under a new policy through importance sampling with inverse propensity scoring (IPS, Horvitz and Thompson, 1952). One may then use this estimator or its variants for optimizing new policies without the need for costly experiments (Bottou et al., 2013; Dudik et al., 2011; Swaminathan and Joachims, 2015a,b), an approach also known as counterfactual risk minimization (CRM). While this setting is not sequential, we assume that learning a *stochastic* policy is required so that one may gather new exploration data after deployment.

In this chapter, we focus on stochastic policies with continuous actions, which, unlike the discrete setting, have received little attention in the context of counterfactual policy optimization (Demirer et al., 2019; Kallus and Zhou, 2018; Chen et al., 2016). As noted by Kallus and Zhou (2018) and as our experiments confirm, addressing the continuous case with naive discretization strategies performs poorly. Our first contribution is about *data modeling*: we introduce a joint embedding of actions and contexts relying on kernel methods, which takes into account the continuous nature of actions, leading to rich classes of estimators that prove to be effective in practice.

In the context of CRM, the problem of *estimation* is intrinsically related to the problem of *optimization* of a non-convex objective function. In our second contribution, we underline the role of optimization algorithms (Bottou et al., 2013; Swaminathan and Joachims, 2015b). We believe that this aspect was overlooked, as previous work has mostly studied the effectiveness of estimation methods regardless of the optimization procedure. In this chapter, we show that appropriate tools can bring significant benefits. To that effect, we introduce differentiable estimators based on soft-clipping the importance weights, which are more amenable to gradient-based optimization than previous hard clipping procedures (Bottou et al., 2013; Wang et al., 2017). We provide a statistical analysis of our estimator and discuss its theoretical performance with regards to the literature. We also find that proximal point algorithms (Rockafellar, 1976) tend to dominate simpler off-the-shelf optimization approaches, while keeping a reasonable computation cost.

Finally, an open problem in counterfactual reasoning is the difficult question of reliable *evaluation* of new policies based on logged data only. Despite significant progress thanks to various IPS estimators, we believe that this issue is still acute, since we need to be able to estimate the quality of policies and possibly select among different candidate ones *before* being

able to deploy them in practice. Our last contribution is a small step towards solving this challenge, and consists of a new offline evaluation benchmark along with a new large-scale dataset, which we call CoCoA, obtained from a real-world system. The key idea is to introduce importance sampling diagnostics (Owen, 2013) to discard unreliable solutions along with significance tests to assess improvements to a reference policy. We believe that this contribution will be useful for the research community; in particular, we are not aware of similar publicly available large-scale datasets for continuous actions.

2.2. Related Work

A large effort has been devoted to designing CRM estimators that have less variance than the IPS method, through clipping importance weights (Bottou et al., 2013; Wang et al., 2017), variance regularization (Swaminathan and Joachims, 2015a), or by leveraging reward estimators through doubly robust methods (Dudik et al., 2011; Robins and Rotnitzky, 1995). In order to tackle an overfitting phenomenon termed "propensity overfitting", Swaminathan and Joachims (2015b) also consider self-normalized estimators (Owen, 2013). Such estimation techniques also appear in the context of sequential learning in contextual bandits (Agarwal et al., 2014; Langford and Zhang, 2008), as well as for off-policy evaluation in reinforcement learning (Jiang and Li, 2016). In contrast, the setting we consider is not sequential. Moreover, unlike direct approaches (Dudik et al., 2011) which learn a cost predictor to derive a deterministic greedy policy, our approach learns a model indirectly by rather minimizing the policy risk.

While most approaches for counterfactual policy optimization tend to focus on discrete actions, few works have tackled the continuous action case, again with a focus on estimation rather than optimization. In particular, propensity scores for continuous actions were considered by Hirano and Imbens (2004). More recently, evaluation and optimization of continuous action policies were studied in a non-parametric context by Kallus and Zhou (2018), and by Demirer et al. (2019) in a semi-parametric setting.

In contrast to these previous methods, (i) we focus on stochastic policies while they consider deterministic ones, even though the kernel smoothing approach of Kallus and Zhou (2018) may be interpreted as learning a deterministic policy perturbed by Gaussian noise. (ii) The terminology of *kernels* used by Kallus and Zhou (2018) refers to a different mathematical tool than the kernel embedding used in our work. We use positive definite kernels to define a nonlinear representation of actions and contexts in order to model the reward function, whereas Kallus and Zhou (2018) use *kernel density estimation* to obtain good importance sampling estimates and not model the reward. Chen et al. (2016) also use a kernel embedding of contexts in their policy parametrization, while our method jointly models contexts and actions. Moreover, their method requires computing an $n \times n$ Gram matrix, which does not scale with large datasets; in principle, it should be however possible to modify their method to handle kernel approximations such as the Nyström method (Williams and Seeger, 2001). Besides, their learning formulation with a quadratic problem is not compatible with CRM regularizers introduced by (Swaminathan and Joachims, 2015a,b) which would change their optimization procedure. Eventually, we note that Krause and Ong (2011) use similar kernels to ours for jointly modeling contexts and actions, but in the different setting of

sequential decision making with upper confidence bound strategies. (iii) While Kallus and Zhou (2018) and Demirer et al. (2019) focus on policy *estimation*, our work introduces a new continuous-action data *representation* and encompasses *optimization*: in particular, we propose a new contextual policy parameterization, which leads to significant gains compared to baselines parametrized policies on the problems we consider, as well as further improvements related to the optimization strategy. We also note that, apart from Demirer et al. (2019) that uses an internal offline cross-validation for model selection, previous works did not perform offline model selection nor evaluation protocols, which are crucial for deploying methods on real data. We provide a brief summary in Table 2.1 to summarize the key differences with our work.

Method	Stochastic	Policy Parametrization	Kernels	CRM Regularizers	Offline evaluation protocol	Large-scale
Chen et al. (2016)	✗	Linear	Embedding of contexts	✗	✗	✗ \ ✓
Kallus and Zhou (2018)	✗ \ ✓	Linear	Kernel Density Estimation	✓	✗	✓
Demirer et al. (2019)	✗	Any	Not used	✗	✓	✓
Ours	✓	CLP	Joint embedding of contexts/actions	✓	✓	✓

Table 2.1: Comparison to Chen et al. (2016); Kallus and Zhou (2018); Demirer et al. (2019), CLP refers to our continuous action model, see section 2.3.1. For discussions on stochastic interpretation of Kallus and Zhou (2018) and the application of Chen et al. (2016) to large-scale data, see main text.

Optimization methods for learning stochastic policies have been mainly studied in the context of reinforcement learning through the policy gradient theorem (Ahmed et al., 2019; Sutton et al., 2000; Williams, 1992). Such methods typically need to observe samples from the new policy at each optimization step, which is not possible in our setting. Other methods leverage a form of off-policy estimates during optimization (Kakade and Langford, 2002; Schulman et al., 2017), but these approaches still require to deploy learned policies at each step, while we consider objective functions involving only a fixed dataset of collected data. In the context of CRM, Su et al. (2019) introduce an estimator with a continuous clipping objective that achieves an improved bias-variance trade-off over the doubly-robust strategy. Nevertheless, this estimator is non-smooth, unlike our soft-clipping estimator.

2.3. Modeling of Continous Action Policies

We now review the CRM framework, and then present our modelling approach for policies with continuous actions.

2.3.1. The Counterfactual Loss Predictor (CLP) for Continuous Actions Policies

We recall that our estimator $\hat{\pi}$ is designed by optimizing (1.12) over a class of policies Π. In this subsection, we discuss how to choose Π when dealing with continuous actions. We emphasize that when considering continuous action spaces, the choice of policies is more involved than in the discrete case. One may indeed naively discretize the action space into buckets and leverage discrete action strategies, but then local information within each bucket gets lost and it is non-trivial to choose an appropriate bucketization of the action space based on logged data, which contains non-discrete actions.

We focus on stochastic policies belonging to certain classes of continuous distributions, such as Normal or log-Normal. Specifically, we consider a set of context-dependent policies of the form

$$\Pi_\Theta = \Big\{ \pi_\theta \quad \text{s.t. for any } x \in \mathcal{X}, \quad \pi_\theta(\cdot|x) = \mathcal{D}(\mu_\beta(x), \sigma^2) \quad \text{with} \quad \theta = (\beta, \sigma) \in \Theta \Big\} \tag{2.1}$$

where $\mathcal{D}(a, b)$ is a probability distribution with mean a and variance b, such as the Normal distribution, and Θ is a parameter space.

Here, the parameter space Θ can be written as $\Theta = \Theta_\beta \times \Theta_\sigma$. The space Θ_σ is either a singleton (if σ is considered as a fixed parameter specified by the user) or $\mathbb{R}_+^*$ (if σ is a parameter to be optimized). The space Θ_β is the parameter space which models the contextual mean $x \mapsto \mu_\beta(x)$.

Counterfactual baselines for Θ_β only consider contexts Before introducing our flexible model for Θ_β, we consider the following simple baselines that will be compared to our model in the experimental Section 2.7. Given a context x in $\mathcal{X} \subset \mathbb{R}^{d_x}$:

- *constant*: $\mu_\beta(x) = \beta$ (context-independent);
- *linear*: $\mu_\beta(x) = \langle x, \beta_1 \rangle + \beta_0$ with $\beta = (\beta_0, \beta_1) \in \mathbb{R}^{d_x+1}$;
- *poly*: $\mu_\beta(x) = \langle xx^\top, \beta_1 \rangle + \beta_0$ with $\beta = (\beta_0, \beta_1) \in \mathbb{R}^{d_x^2+1}$.

These baselines require learning the parameters β by using the CRM approach (1.12). Intuitively, the goal is to find a stochastic policy that is close to the optimal deterministic one from Eq. (DM). Yet, these approach consider function spaces Θ_μ of the mean functions μ that only use the context. While these approaches, adopted by Chen et al. (2016), Kallus and Zhou (2018) can be effective in simple problems, they may be limited in more difficult scenarios where the expected cost $\eta^*(x, a)$ has a complex behavior as a function of a. This motivates the need for classes of policies which can better capture such variability by considering a joint model $\eta(x, a)$ of the cost.

The counterfactual loss predictor (CLP) model for Θ_β. Assuming that we are given such a parametric model $\eta_\beta(x, a)$, which we call *loss predictor* and will be detailed thereafter, we parametrize the mean of a stochastic policy by using a soft-argmin operator with temperature $\gamma > 0$:

$$\text{CLP:} \qquad \mu_\beta^{\text{CLP}}(x) = \sum_{i=1}^m a_i \frac{\exp(-\gamma \eta_\beta(x, a_i))}{\sum_{j=1}^m \exp(-\gamma \eta_\beta(x, a_j))}, \tag{2.2}$$

where $a_1, \ldots, a_m \in \mathcal{A}$ are anchor points (e.g.,, a regular grid or quantiles of the action space), and μ_β may be viewed here as a smooth approximation of a greedy policy $\mu_{\text{greedy}}(x) = \arg\min_a \eta(x, a)$. This allows CLP policies to capture complex behavior of the expected loss as a function of a. The motivation for introducing a soft-argmin operator is to avoid the optimization over actions and to make the resulting CRM problem differentiable.

Modeling of the loss predictor $\eta_\beta(x, a)$. The above CLP model is parameterized by $\eta_\beta(x, a)$ that may be interpreted as a loss predictor. We choose it of the form

$$\eta_\beta(x, a) = \langle \beta, \psi(x, a) \rangle$$

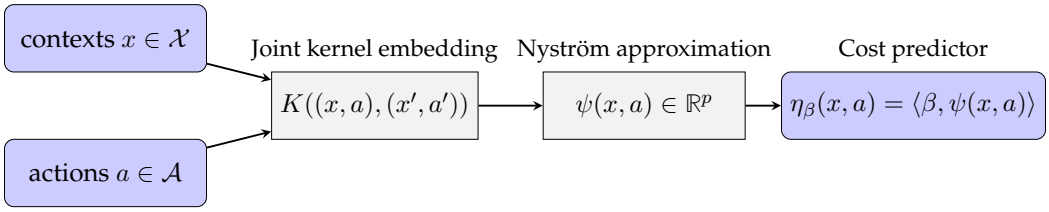

Figure 2.1: Illustration of the joint kernel embedding for the counterfactual loss predictor (CLP).

for some parameter $\beta \in \mathbb{R}^p$, which norm controls the smoothness of η, and a feature map $\psi(x,a) \in \mathbb{R}^p$ that we detail in two parts: a joint kernel embedding between the actions and the contexts and a Nyström approximation. The complete modeling of $\eta_\beta(x,a)$ is summarized in Figure 2.1.

1. Joint kernel embedding. In a continuous action problem, a reasonable assumption is that losses y vary smoothly as a function of actions. Thus, a good choice is to take η in a space of smooth functions such as the reproducing kernel Hilbert space $\mathcal{H}$ (RKHS) defined by a positive definite kernel (Schölkopf and Smola, 2002), so that one may control the smoothness of η through regularization with the RKHS norm. More precisely, we consider kernels of the form

$$K((x,a),(x',a')) = \langle \psi_{\mathcal{X}}(x), \psi_{\mathcal{X}}(x') \rangle e^{-\frac{\alpha}{2}\|a-a'\|^2}, \tag{2.3}$$

where, for simplicity, $\psi_{\mathcal{X}}(x)$ is either a linear embedding $\psi_{\mathcal{X}}(x) = x$ or a quadratic one $\psi_{\mathcal{X}}(x) = (xx^T, x)$, while actions are compared via a Gaussian kernel, allowing to model complex interactions between contexts and actions.

2. Nyström method and explicit embedding. Since traditional kernel methods lack scalability, we rely on the classical Nyström approximation (Williams and Seeger, 2001) of the Gaussian kernel, which provides us a finite-dimensional approximate embedding $\psi_{\mathcal{A}}(a)$ in $\mathbb{R}^m$ such that $e^{-\frac{\alpha}{2}\|a-a'\|^2} \approx \langle \psi_{\mathcal{A}}(a), \psi_{\mathcal{A}}(a') \rangle$ for all actions a, a'. This allows us to build a finite-dimensional embedding

$$\psi(x,a) = \psi_{\mathcal{X}}(x) \otimes \psi_{\mathcal{A}}(a), \tag{2.4}$$

where $\otimes$ denotes the tensorial product, such that

$$K((x,a),(x',a')) \approx \langle \psi_{\mathcal{X}}(x), \psi_{\mathcal{X}}(x') \rangle \langle \psi_{\mathcal{A}}(a), \psi_{\mathcal{A}}(a') \rangle = \langle \psi(x,a), \psi(x',a') \rangle.$$

More precisely, Nyström's approximation consists of projecting each point from the RKHS to a m-dimensional subspace defined as the span of m anchor points, representing here the mapping to the RKHS of m actions $\bar{a}_1, \bar{a}_2, \ldots, \bar{a}_m$ of the Nyström dictionary $\mathcal{Z}$. In practice, we may choose $\bar{a}_i$ to be equal to the a_i in (2.2), since in both cases the goal is to choose a set of "representative" actions. For one-dimensional actions ($\mathcal{A} \subseteq \mathbb{R}$), it is reasonable to consider a uniform grid, or a non-uniform ones based on quantiles of the empirical distribution of actions in the dataset. In higher dimensions, one may simply use a K-means algorithms and assign anchor points to centroids.

From an implementation point of view, Nyström's approximation considers the embedding $\psi_{\mathcal{A}}(a) = K_{\mathcal{Z}\mathcal{Z}}^{-1/2} K_{\mathcal{Z}}(a)$, where $K_{\mathcal{Z}\mathcal{Z}} = [K_{\mathcal{A}}(\bar{a}_i, \bar{a}_j)]_{ij}$ and $K_{\mathcal{Z}}(a) = [K_{\mathcal{A}}(a, \bar{a}_i)]_i$ and $K_{\mathcal{A}}$ is the Gaussian kernel.

The anchor points that we use can be seen as the parameters of an *interpolation* strategy defining a smooth function, similar to knots in spline interpolation. Naive discretization strategies would prevent us from exploiting such a smoothness assumption on the cost with respect to actions and from exploiting the structure of the action space. Note that Section 2.7 provides a comparison with naive discretization strategies, showing important benefits of the kernel approach. Our goal was to design a stochastic, computationally tractable, differentiable approximation of the optimal (but unknown) greedy policy (DM).

Summary of the CLP policy class definition We provide below a shortened description of the CLP parametrization. In particular the policy class construction requires input parameters and yields a parametric policy class:

Input: Temperature $\gamma > 0$, kernel K, Nyström dictionary $\mathcal{Z} \subseteq \mathcal{A}$, parametric distribution $\mathcal{D}$ (such as Normal or log-Normal).

1. Define the d-dimensional feature map ψ as in Eq. (2.4) by using K and $\mathcal{Z}$.
2. For any $\beta \in \mathbb{R}^d$ and $(x, a) \in \mathcal{X} \times \mathcal{A}$, define

$$\eta_\beta(x, a) = \langle \beta, \psi(x, a) \rangle \qquad \text{and} \qquad \mu_\beta^{\text{CLP}}(x) = \sum_{a \in \mathcal{Z}} \frac{\exp(-\gamma \eta_\beta(x, a))}{\sum_{a' \in \mathcal{Z}} \exp(-\gamma \eta_\beta(x, a'))}.$$

3. Define the policy set

$$\Pi_\Theta^{\text{CLP}} = \left\{ \pi \text{ s.t. } \forall x \in \mathcal{X},\ \pi(\cdot|x) = \mathcal{D}(\mu_\beta^{\text{CLP}}(x), \sigma^2),\ \text{with } (\beta, \sigma) \in \Theta \right\}.$$

2.4. On Optimization Perspectives for CRM

Because our models yield non-convex CRM problems, we believe that it is crucial to study optimization aspects. Here, we introduce a differentiable clipping strategy for importance weights and discuss optimization algorithms.

2.4.1. Soft Clipping IPS

The classical hard clipping estimator

$$\hat{L}^{\text{cIPS}}(\theta) = \frac{1}{n} \sum_{i=1}^{n} y_i \min \left\{ \pi_\theta(a_i|x_i)/\pi_{0,i}, M \right\} \tag{2.5}$$

makes the objective function non-differentiable, and yields terms in the objective with clipped weights to have zero gradient. In other words, a trivial stationary point of the objective function is that of a stochastic policy that differs enough from the logging policy such that all importance weights are clipped. To alleviate this issue, we propose a differentiable

logarithmic soft-clipping strategy. Given a threshold parameter $M \geq 0$ and an importance weight $w_i = \pi_\theta(a_i|x_i)/\pi_{0,i}$, we consider the soft-clipped weights:

$$\zeta(w_i, M) = \begin{cases} w_i & \text{if } w_i \leq M \\ \alpha(M)\log\left(w_i + \alpha(M) - M\right) & \text{otherwise,} \end{cases} \tag{2.6}$$

where $\alpha(M)$ is such that $\alpha(M)\log(\alpha(M)) = M$, which yields a differentiable operator. We illustrate the soft clipping expression in Figure 2.2 and give further explanations about the benefits of clipping strategies in Appendix 2.9.

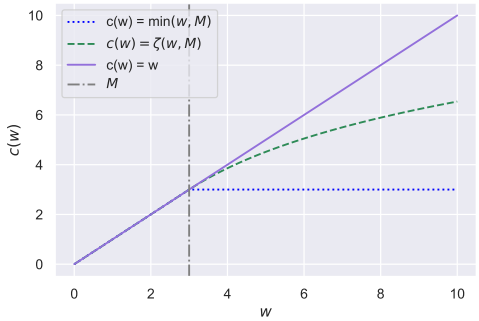

Figure 2.2: Different clipping strategies c on the importance weights w. Weights are clipped for $M = 3$, the hard clipping $c(w) = \min(w, M)$ provides no gradient for $w > M$, while the soft clipping $c(w) = \zeta(w, M)$ and the unclipped estimators $c(w) = w$ do.

Then, the IPS estimator with soft clipping becomes

$$\hat{L}^{\text{scIPS}}(\theta) = \frac{1}{n}\sum_{i=1}^{n} y_i \zeta\left(\frac{\pi_\theta(a_i|x_i)}{\pi_{0,i}}, M\right). \tag{2.7}$$

We now provide a similar generalization bound to that of Swaminathan and Joachims (2015a) (for the hard-clipped version) for the variance-regularized objective of this soft-clipped estimator, justifying its use as a good optimization objective for minimizing the expected risk. Writing $\chi_i(\theta) = y_i\zeta\left(\frac{\pi_\theta(a_i|x_i)}{\pi_{\theta_0}(a_i|x_i)}, M\right)$, we recall the empirical variance with scIPS that is used for regularization:

$$\hat{V}^{\text{scIPS}}(\theta) = \frac{1}{n-1}\sum_{i=1}^{n}(\chi_i(\theta) - \bar{\chi}(\theta))^2, \qquad \text{with} \quad \bar{\chi}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\chi_i(\theta). \tag{2.8}$$

We assume that costs $y_i \in [-1, 0]$ almost surely, as in (Swaminathan and Joachims, 2015a), and make the additional assumption that the importance weights $\pi_\theta(a_i|x_i)/\pi_{\theta_0}(a_i|x_i)$ are upper bounded by a constant W almost surely for all $\pi \in \Pi$. This is satisfied, for instance, if all policies have a given compact support (*e.g.*, actions are constrained to belong to a given interval) and π_{θ_0} puts mass everywhere in this support.

Proposition 2.4.1 (Generalization bound for $\hat{L}^{\text{scIPS}}(\theta)$)**.** *Let Θ be a parameter space for the policy class Π_Θ and π_{θ_0} be a logging policy. Let $s_0 = (x_i, a_i, y_i)_{i=1,\dots,n}$ the logging dataset for which actions are sampled under π_{θ_0}. Assume that the losses $y \in [-1, 0]$ to be bounded a.s. and that the*

importance weights are bounded by W. Then, with probability at least $1-\delta$, the IPS estimator with soft clipping (2.7) *on n samples from s_0 satisfies*

$$\forall \pi \in \Pi, \qquad L(\theta) \leq \hat{L}_{scIPS}(\theta) + O\left(\sqrt{\frac{\hat{V}_{scIPS}(\theta)\big(C_n(\Theta, M) + \log\frac{1}{\delta}\big)}{n}} + \frac{S\big(C_n(\Theta, M) + \log\frac{1}{\delta}\big)}{n}\right),$$

where $S = \zeta(W, M) = O(\log W)$, $\hat{V}^{scIPS}(\theta)$ denotes the empirical variance of the cost estimates (1.16)*, and $C_n(\Theta, M)$ is a complexity measure of the policy class defined in* (2.14)*.*

We prove the Proposition 2.4.1 in Appendix 2.9. This generalization error bound motivates the use of the empirical variance penalization as in Swaminathan and Joachims (2015a) and shows that minimizing both the empirical risk and penalization of the soft clipped estimator minimize the true risk of the policy.

Note that while the bound requires importance weights bounded by a constant W, the bound only scales logarithmically with W when $W \gg M$, compared to a linear dependence for IPS. However we gain significant benefits in terms of optimization by having a smooth objective.

Remark 2.4.1. *If costs are in the range $[-c, 0]$, the constant S can be replaced by cS, making the bound homogeneous in the scale (indeed, the variance term is also scaled by c).*

Remark 2.4.2. *For a fixed parameter M, the scIPS estimator is less biased than the cIPS. Indeed, we can bound the importance weights as $\min\{\frac{\pi_\theta(a|x)}{\pi_{\theta_0}(a|x)}, M\} \leq \zeta\left(\frac{\pi_\theta(a|x)}{\pi_{\theta_0}(a|x)}, M\right) \leq \frac{\pi_\theta(a|x)}{\pi_{\theta_0}(a|x)}$ and subsequently derive the bound on the different biases:*

$$\left|\, \mathbb{E}_{x,a\sim\pi_{\theta_0}(\cdot|x)}\left[y \min\{\frac{\pi_\theta(a|x)}{\pi_{\theta_0}(a|x)}, M\} - y\right] \right| \geq \left|\, \mathbb{E}_{x,a\sim\pi_{\theta_0}(\cdot|x)}\left[y\zeta\left(\frac{\pi_\theta(a|x)}{\pi_{\theta_0}(a|x)}, M\right) - y\right] \right| \geq 0$$

We emphasize however that the M parameter may have different optimal values for both methods, and that the main motivation for such a clipping strategy is to provide a differentiable estimator which is not the case for cIPS in areas where all point are clipped.

2.4.2. Proximal Point Algorithms

Non-convex CRM objectives have been optimized with classical gradient-based methods (Swaminathan and Joachims, 2015a,b) such as L-BFGS (Liu and Nocedal, 1989), or the stochastic gradient descent approach (Joachims et al., 2018). Proximal point methods are classical approaches originally designed for convex optimization (Rockafellar, 1976), which were then found to be useful for nonconvex functions (Fukushima and Mine, 1981; Paquette et al., 2018). In order to minimize a function $\mathcal{L}$, the main idea is to approximately solve a sequence of subproblems that are better conditioned than $\mathcal{L}$, such that the sequence of iterates converges towards a better stationary point of $\mathcal{L}$. More precisely, for our class of parametric policies, the proximal point method consists of computing a sequence

$$\theta^{(k)} \approx \arg\min_\theta \left(\mathcal{L}(\theta) + \frac{\kappa}{2}\|\theta - \theta^{(k-1)}\|_2^2\right), \tag{2.9}$$

where $\mathcal{L}(\theta) = \hat{L}(\theta) + \Omega(\theta)$ and $\kappa > 0$ is a constant parameter. The regularization term Ω often penalizes the variance (Swaminathan and Joachims, 2015b), see Appendix 2.9. The role of the

quadratic function in (2.9) is to make subproblems "less nonconvex" and for many machine learning formulations, it is even possible to obtain convex sub-problems with large enough κ (see Paquette et al., 2018). In this chapter, we consider such a strategy (2.9) with a parameter κ, which we set to zero only for the last iteration.

Note that the effect of the proximal point algorithm differs from the proximal policy optimization (PPO) strategy used in reinforcement learning (Schulman et al., 2017), even though both approaches are related. PPO encourages a new stochastic policy to be close to a previous one in Kullback-Leibler distance. Whereas the term used in PPO modifies the objective function (and changes the set of stationary points), the proximal point algorithm optimizes and finds a stationary point of the original objective $\mathcal{L}$, even with fixed κ.

The proximal point algorithm (PPA) introduces an additional computational cost as it leads to solving multiple sub-problems instead of a single learning problem. In practice for 10 PPA iterations and with the L-BFGS solver, the computational overhead was about $3\times$ in comparison to L-BFGS without PPA. This overhead seems to be the price to pay to improve the test reward and obtain better local optima, as we show in the experimental section 2.7.2. Nevertheless, we would like to emphasize that computational time is often not critical for the applications we consider, since optimization is performed offline.

2.5. Analysis of the Excess Risk

In the previous section, we have introduced a new counterfactual estimator $\hat{L}^{\text{scIPS}}$ (2.7) of the risk, which satisfies good optimization properties. Motivated by the generalization bound in Proposition 2.4.1, for any policy class Π_Θ, we associate $\hat{L}^{\text{scIPS}}$ with the data-dependent regularizer and define the following CRM estimator

$$\hat{\theta}^{CRM} = \operatorname*{arg\,min}_{\theta\in\Theta} \left\{ \hat{L}^{\text{scIPS}}(\theta) + \lambda \sqrt{\frac{\hat{V}^{\text{scIPS}}(\theta)}{n}} \right\}, \tag{2.10}$$

where $\hat{V}^{\text{scIPS}}(\pi)$ is the empirical variance defined in (2.8). In this section, we provide theoretical guarantees on the excess risk of $\hat{\theta}^{CRM}$, first for any general policy class Π_Θ, then for our newly introduced policy class Π_Θ^{CLP} (Section 2.3.1). We now define what is the expected risk of a model $\theta \in \Theta$.

Definition 2.5.1 (Excess Risk)**.** *Given an optimal model $\theta^* \in \arg\min_{\theta\in\Theta} L(\theta)$, we write the excess risk:*

$$\Delta(\theta) = L(\theta) - L(\theta^*), \tag{2.11}$$

We now provide the following high-probability upper-bound on the excess-risk.

Proposition 2.5.1 (Excess risk upper bound)**.** *Consider the notations and assumptions of Proposition 2.4.1. Let $\hat{\theta}^{CRM}$ be the solution of the CRM problem in Eq.* (2.10)*. Then, with well chosen parameters λ and M, denoting the variance $\nu_*^2 = \mathrm{Var}_{\pi_{\theta_0}}\big[\pi_{\theta^*}(a|x)/\pi_{\theta_0}(a|x)\big]$, with probability at least $1-\delta$, the excess risk is upper bounded as:*

$$\Delta(\hat{\theta}^{CRM}) \lesssim \sqrt{\frac{(1+\nu_*^2)\log(W+e)\big(C_n(\Theta,M)+\log\frac{1}{\delta}\big)}{n}} + \frac{\log(W+e)(C_n(\Theta,M)+\log\frac{1}{\delta})}{n},$$

where $\lesssim$ hides universal multiplicative constants. In particular, assuming also that $\pi_{\theta_0}(x|a)^{-1}$ are uniformly bounded, the complexity of the class Π^{CLP} described in Section 2.3.1, applied with a bounded kernel and $\Theta = \{\beta \in \mathbb{R}^m,\ s.t\ \|\beta\| \leq C\} \times \{\sigma\}$, is of order

$$C_n(\Theta^{CLP}, M) \leq O(m \log n)\,,$$

where m is the size of the Nyström dictionary and $O(\cdot)$ hides multiplicative constants independent of n and m (see (2.28)*).*

The proof and the exact definition of $C_n(\Theta, M)$ are provided in Appendix 2.9. Our analysis relies on Theorem 15 of Maurer and Pontil (2009).

Comparison with related work The closest works are the ones of Chen et al. (2016) and Kallus and Zhou (2018). Chen et al. (2016) analyze their method for Besov policy classes $B^{\alpha}_{1,\infty}(\mathbb{R}^d)$. When $\alpha \to \infty$, they obtain a rate of order $\mathcal{O}(n^{-1/4})$. In this case, their setting is parametric and their rate can be compared to our $O(n^{-1/2})$ when m is finite. Kallus and Zhou (2018) provide bounds with respect to general deterministic classes of functions, whose complexity is measured by their Rademacher complexity. For parametric classes, their excess risk for an estimated $\hat{\theta}$ is bounded (up to logs) by $\Delta(\hat{\theta}) \lesssim h^{-2}n^{-1/2} + h^{-1}n^{-1/2} + h^2$, where h is a smoothing parameter. By optimizing the bandwidth $h = \mathcal{O}(n^{-1/8})$, their method also yields a rate of order $\mathcal{O}(n^{-1/4})$.

Yet, a key difference between their setting and ours explains the gap between their rate $\mathcal{O}(n^{-1/4})$ and $\mathcal{O}(n^{-1/2})$ of Proposition 2.5.1. Both consider deterministic policy classes, while we only consider stochastic policies. Indeed, W and ν^* would be unbounded for deterministic policies in Proposition 2.5.1. Therefore, to leverage deterministic policies, they both need to smooth their predictions and suffer an additional bias that we do not incur. This is why there is a difference between their rate and ours. For instance, for stochastic classes with variance σ^2, Kallus and Zhou (2018) would satisfy $R_{\hat{\pi}} \lesssim \sigma^{-2}n^{-1/2} + \sigma^{-1}n^{-1/2}$ for $h \approx \sigma$, which would also entail a rate of order $\mathcal{O}(n^{-1/2})$. Interestingly, on the other hand, our approach would satisfy a rate $O(n^{-1/3})$ for deterministic policies, i.e., $\sigma^2 \to 0$ (see Appendix 2.9). This may be explained by the fact that, contrary to Kallus and Zhou (2018); Chen et al. (2016) who only use it in practice, we consider variance regularization and clipping in our analysis.

Another related work is (Demirer et al., 2019). They obtain an excess risk rate of $\mathcal{O}(n^{-1/2})$ when learning deterministic continuous action policies with a policy space of finite and small VC-dimension. Under a margin condition, as in bandit problems, their rate may be improved to $O(\log(n)/n)$. However, their method significantly differs from ours and Chen et al. (2016), Kallus and Zhou (2018) because it relies on a two steps plug-in procedure: first estimate a nuisance function, then learn a policy using with a value function using this estimate. Eventually, we note that Majzoubi et al. (2020) also enjoys a regret of $\mathcal{O}(n^{-1/2})$ (up to logarithmic factors) but learns tree policies that are hardly comparable to ours. Both approaches turn out to perform worse in all our benchmarks, as seen in Section. 2.7.2.

2.6. On Evaluation and Model Selection for Real World Data

The CRM framework helps finding solutions when online experiments are costly, dangerous or raising ethical concerns. As such it needs a reliable validation and evaluation

procedure before rolling-out any solution in the real world. In the continuous action domain, previous work have mainly considered semi-simulated scenarios (Bertsimas and McCord, 2018; Kallus and Zhou, 2018), where contexts are taken from supervised datasets but rewards are synthetically generated. To foster research on practical continuous policy optimization, we release a new large-scale dataset called CoCoA, which to our knowledge is the first to provide logged exploration data from a real-world system with continuous actions. Additionally, we introduce a benchmark protocol for reliably evaluating policies using off-policy evaluation.

2.6.1. The CoCoADataset

The CoCoAdataset comes from the Criteo online advertising platform which ran an experiment involving a randomized, continuous policy for real-time bidding. Data has been properly anonymized so as to not disclose any private information. Each sample represents a bidding opportunity for which a multi-dimensional context x in $\mathbb{R}^d$ is observed and a continuous action a in $\mathbb{R}^+$ has been chosen according to a stochastic policy π_{θ_0} that is logged along with the reward $-y$ (meaning cost y) in $\mathbb{R}$. The reward represents an advertising objective such as sales or visits and is jointly caused by the action and context (a, x). Particular care has been taken to guarantee that each sample $(x_i, a_i, \pi_{\theta_0}(a_i|x_i), y_i)$ is independent. The goal is to learn a contextual, continuous, stochastic policy $\pi_\theta(a|x)$ that generates more reward in expectation than π_{θ_0}, evaluated offline, while keeping some exploration (stochastic part). As seen in Table 2.2, a typical feature of this dataset is the high variance of the cost ($\mathbb{V}[Y]$), motivating the scale of the dataset N to obtain precise counterfactual estimates. The link to download the dataset is available in the code repository: `https://github.com/criteo-research/optimization-continuous-action-crm`.

Table 2.2: CoCoAdataset summary statistics.

N	d	$\mathbb{E}[-Y]$	$\mathbb{V}[Y]$	$\mathbb{V}[A]$	$\mathbb{P}(Y \neq 0)$
120.10^6	3	11.37	9455	.01	.07

2.6.2. Evaluation Protocol for Logged Data

In order to estimate the test performance of a policy on real-world systems, off-policy evaluation is needed, as we only have access to logged exploration data. Yet, this involves in practice a number of choices and difficulties, the most documented being i) potentially infinite variance of IPS estimators (Bottou et al., 2013) and ii) propensity over-fitting (Swaminathan and Joachims, 2015a,b). The former implies that it can be difficult to accurately assess the performance of new policies due to large confidence intervals, while the latter may lead to estimates that reflect large importance weights rather than rewards.

A proper evaluation protocol should therefore guard against such outcomes.

A first, structuring choice is the IPS estimator. While variants of IPS exist to reduce variance, such as clipped IPS, we found Self-Normalized IPS (SNIPS, Swaminathan and Joachims, 2015b; Lefortier et al., 2016; Owen, 2013; Nedelec et al., 2017) to be more effective in practice. Indeed, it avoids the choice of a clipping threshold, generally reduces variance and is equivariant with respect to translation of the reward.

Algorithm 5: Evaluation Protocol

Input: $1 - \delta$: confidence of statistical test (def: 0.95); ν: a max deviance ratio for effective sample size (def: 0.01);

Output: counterfactual estimation of $L(\theta)$ and decision to reject the null hypothesis $\{H_0\text{: } L(\theta) \geq L(\theta_0)\}$.

1. Split observation dataset $s_0 \mapsto s^{\text{train}}, s^{\text{valid}}, s^{\text{test}}$
2. Train θ on s^{train} and tune policy class and optimization hyper-parameters on s^{valid} (for e.g by internal cross-validation)
3. Estimate effective sample size n_{eff} on s^{valid}

if $\frac{n_{\text{eff}}}{n} > \nu$ **then**

  Estimate $\hat{L}^{\text{SNIPS}}(\theta)$ on s^{test} and test $\hat{L}^{\text{SNIPS}}(\theta) < \hat{L}(\theta_0)$ on s^{test} with confidence $1 - \delta$. If the test is valid, reject H_0, otherwise keep it.

else

  Keep H_0, consider the estimate to be invalid.

end

A second component is the use of importance sampling diagnostics to prevent propensity over-fitting. Lefortier et al. (2016) propose to check if the empirical average of importance weights deviates from 1. However, there is no precise guideline based on this quantity to reject estimates. Instead, we recommend to use a diagnostic on the *effective sample size* $n_{\text{eff}} = (\sum_{i=1}^{n} w_i)^2 / \sum_{i=1}^{n} w_i^2$, which measures how many samples are actually usable to perform estimation of the counterfactual estimate; we follow Owen (2013), who recommends to reject the estimate when the relative effective sample size n_{eff}/n is less than 1%.

A third choice is a statistical decision procedure to check if $L(\theta) < L(\theta_0)$. In theory, any statistical test against a null hypothesis H_0: $L(\theta) \geq L(\theta_0)$ with confidence level $1 - \delta$ can be used.

Finally, we present our protocol in Algorithm 5. Since we cannot evaluate such a protocol on purely offline data, we performed an empirical evaluation on synthetic setups where we could analytically design true positive ($L(\theta) < L(\theta_0)$) and true negative policies. We discuss in Section 2.7 the concrete parameters of Algorithm 5 and their influence on false (non-)discovery rates in practice.

Model selection with the offline protocol In order to make realistic evaluations, hyper-parameter selection is always conducted by estimating the loss of a new policy π_θ in a counterfactual manner. This requires using a validation set (or cross-validation) with propensities obtained from the logging policy π_{θ_0} of the training set. Such estimates are less accurate than online ones, which would require to gather new data obtained from π, which we assume is not feasible in real-world scenarios.

To solve this issue, we have chosen to discard unreliable estimates that do not pass the effective sample size test from Algorithm 5. When doing cross-validation, it implies discarding folds that do not pass the test, and averaging estimates computed on the remaining folds. Although this induces a bias in the cross-validation procedure, we have found it to significantly reduce the variance and dramatically improve the quality of model selection

when the number of samples is small, especially for the Warfarin dataset in Section 2.7.

2.7. Experimental Setup and Evaluation

We now provide an empirical evaluation of the various aspects of CRM addressed in this chapter such as policy class modelling (CLP), estimation with soft-clipping, optimization with PPA, offline model selection and evaluation. We conduct such a study on synthetic and semi-synthetic datasets and on the real-world CoCoAdataset.

2.7.1. Experimental Validation of the Protocol

In this section, we study the ability of Algorithm 5 to accurately decide if a candidate policy π is better than a reference logging policy π_{θ_0} (condition $L(\theta) \leq L(\theta_0)$) on synthetic data. Here we simulate logging policy π_{θ_0} being a lognormal distribution of known mean and variance, and an optimal policy π_{θ^*} being a Gaussian distribution. We generate a logged dataset by sampling actions $a \sim \pi_{\theta_0}$ and trying to evaluate policies $\hat{\pi}_\theta$ with costs observed under the logging policy. We compare the costs predicted using IPS and SNIPS offline metrics to the online metric as the setup is synthetic, it is then easy to check that indeed they are better or worse than π_{θ_0}. We compare the IPS and SNIPS estimates along with their level of confidences and the influence of the effective sample size diagnostic. Offline evaluations of policies $\hat{\pi}_\theta$ illustrated in Figure 2.3 are estimated from logged data $(x_i, a_i, y_i, \pi_{\theta_0})_{i=1\ldots n}$ where $a_i \sim \pi_{\theta_0}(\cdot|x_i)$ and where the policy risk would be optimal under the oracle policy π_{θ^*}.

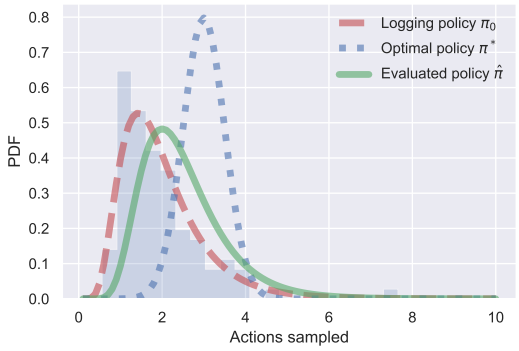

Figure 2.3: Illustration of policies: logging policy π_{θ_0}, optimal π_{θ^*} and example policy $\hat{\pi}_\theta$.

While the goal of counterfactual learning is to find a policy $\hat{\pi}_\theta$ which is as close a possible to the optimal policy π_{θ^*}, based on samples drawn from a logging policy π_{θ_0}, it is in practice hard to assess the statistical significance of a policy that is too "far" from the logging policy. Offline importance sampling estimates are indeed limited when the distribution mismatch between the evaluated policy and the logging policy (in terms of KL divergence $D_{KL}(\pi_{\theta_0}||\hat{\pi}_\theta)$) is large. Therefore we create a setup where we evaluate the quality of offline estimates for policies (i) "close" to the logging policy (meaning the KL divergence $D_{KL}(\pi_{\theta_0}||\hat{\pi}_\theta)$ is low) and (ii) "close" to the oracle optimal policy (meaning the KL divergence $D_{KL}(\pi_{\theta^*}||\hat{\pi}_\theta)$ is low). In this experiment, we focus on evaluating the ability of the offline protocol to correctly assess whether $L(\theta) \leq L(\theta_0)$ or not by comparing to online truth estimates. Specifically, for

both setups (i) and (ii), we compare the number of False Positives (FP) and False Negatives (FN) of the two offline protocols for $N = 2000$ initializations, by adding Gaussian noise to the parameters of the closed form policies. False negatives are generated when the offline protocol keeps $H_0 : L(\theta) \geq L(\theta_0)$ while the online evaluation reveals that $L(\theta) \leq L(\theta_0)$, while false positives are generated in the opposite case when the protocol rejects H_0 while it is true. We also show histograms of the differences between online and offline boundary decisions for $(L(\theta) < L(\theta_0))$, using bootstrapped distribution of SNIPS estimates to build confidence intervals.

Validation of the use of SNIPS estimates for the offline protocol. To assess the performance of our evaluation protocol, we first compare the use of IPS and SNIPS estimates for the offline evaluation protocol and discard solutions with low importance sampling diagnostics $\frac{n_{\text{eff}}}{n} < \nu$ with the recommended value $\nu = 0.01$ from Owen (2013). In Table 2.3, we provide an analysis of false positives and false negatives in both setups. We first observe that for setup (i) the SNIPS estimates has both fewer false positives and false negatives. Note that is setup is probably more realistic for real-world applications where we want to ensure incremental gains over the logging policy. In setup (ii) where importance sampling is more likely to fail when the evaluated policy is too "far" from the logging policy, we observe that the SNIPS estimate has a drastically lower number of false negatives than the IPS estimate, though it slightly has more false positives, thus illustrating how conservative this estimator is.

Table 2.3: Comparison of false positives and false negatives: Perturbation to the logging policy π_{θ_0} (setup (i)) and perturbation to the optimal policy (setup (ii)). The SNIPS estimator yields less FN and FP on setup (i), while being more effective on setup (ii) as well by inducing a drastically lower FP rate than IPS and a low FN rate. The effective sample size threshold is fixed at $\nu = 0.01$

Offline Protocol		Setup (i)				Setup (ii)			
		IPS		SNIPS		IPS		SNIPS	
		$\hat{\pi}_\theta \succeq \pi_{\theta_0}$	Keep H_0	$\hat{\pi}_\theta \succeq \pi_{\theta_0}$	Keep H_0	$\hat{\pi}_\theta \succeq \pi_{\theta_0}$	Keep H_0	$\hat{\pi} \succeq \pi_{\theta_0}$	Keep H_0
"Truth"	$\hat{\pi}_\theta \succeq \pi_{\theta_0}$	1282	24	1296	**10**	1565	67	1631	**1**
	Keep H_0	19	675	**0**	694	**0**	368	6	362

We then provide in Fig. 2.4 histograms of the differences of the upper boundary decisions between online estimates and bootstrapped offline estimates over all samples for both setups (i, left) and (ii, right). Both histograms illustrate how the IPS estimate underestimates the value of the reward with regard to the online estimate, unlike the SNIPS estimates. In the setup (ii) in particular, the IPS estimate underestimates severely the reward, which may explain why IPS has lower number of false positives when the evaluated policy is far from the logging policy. However in both setups, IPS has a higher number of false negatives. We also observed that our SNIPS estimates were highly correlated to the true (online) reward (average correlation $\xi = .968$, 30% higher than IPS, see plots in Appendix 2.9) for the synthetic setups presented in section 2.7.2, which therefore confirms our findings.

Influence of the effective sample size criteria in the evaluation protocol In this setup we vary the effective sample size (ESS) threshold and show in Fig. 2.5 how it influences the performance of the offline evaluation protocol for the two previously discussed setups where we consider evaluations of (i) perturbations of the logging policy (left) and (ii) perturbations of the optimal policy (right) in our synthetic setup. We compute precision, recall and F1 scores

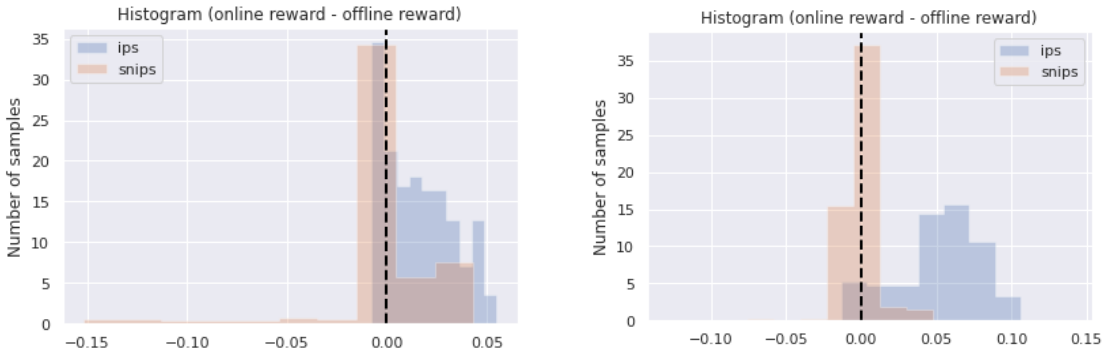

Figure 2.4: Histogram of differences between online reward and offline lower confidence bound. Perturbation to the logging policy π_{θ_0} (left), perturbation to the optimal policy π^* (right). Effective sample size threshold $\nu = 0.01$

for each threshold values between 0 and 1. One can see that for low threshold values where no policies are filtered, precision, recall and F1 scores remain unchanged. Once the ESS raises above a certain threshold, undesirable policies start being filtered but more false negatives are created when the ESS is too high. Overall, ESS criterion is relevant for both setups. However, we observe that on simple synthetic setups the effective sample size criterion $\nu = n_{\text{eff}}/n$ is seldom necessary for policies close to the logging policy ($\pi_\theta \approx \pi_{\theta_0}$). Conversely, for policies which are not close to the logging policy the standard statistical significance testing at $1 - \delta$ level was by itself not enough to guarantee a low false discovery rate (FDR) which justified the use of ν. Adjusting the effective sample size can therefore influence the performance of the protocol (see Appendix 2.9 for further illustrations of importance sampling diagnostics in what-if simulations).

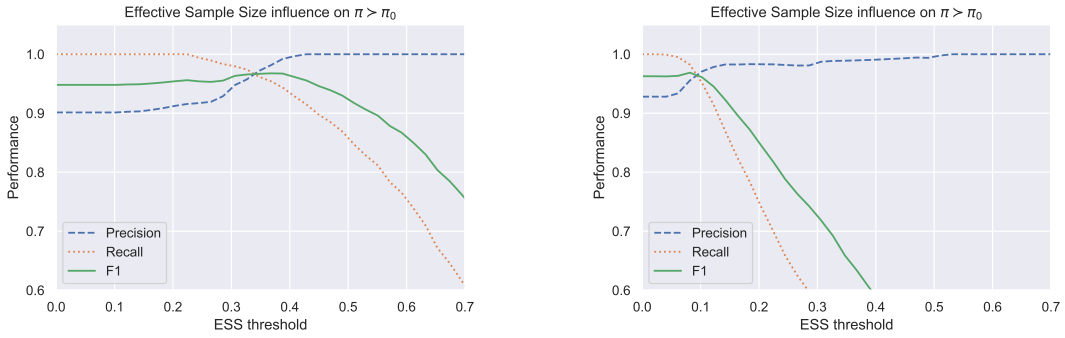

Figure 2.5: Precision, recall and F1 score varying with the ESS threshold on synthetic setups (i) and (ii). Setup (i) perturbation of the logging policy (left) and setup (ii) perturbation to the optimal policy (right). The ESS threshold can maximize the F1 score.

2.7.2. Experimental Evaluation of the Continuous Modelling and the Optimization Perspectives

In this section we introduce our empirical settings for evaluation and present our proposed CLP policy parametrization, and the influence of optimization in counterfactural

risk minimization problems.

Experimental Setup

We present the synthetic potential prediction setup, a semi-synthetic setup as well as our real-world setup.

Synthetic potential prediction. We introduce simple synthetic environments with the following generative process: an unobserved random group index g in $\mathcal{G}$ is drawn, which influences the drawing of a context x and of an unobserved "potential" p in $\mathbb{R}$, according to a joint conditional distribution $\mathcal{P}_{\mathcal{X},P|\mathcal{G}}$. Intuitively, the potential p may be compared to users a priori responsiveness to a treatment. The observed reward $-y$ is then a function of the context x, action a, and potential p. The causal graph corresponding to this process is given in Figure 2.6.

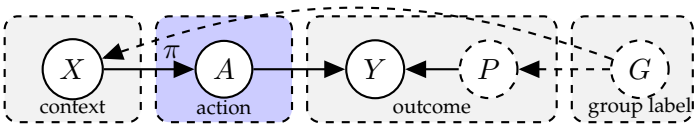

Figure 2.6: Causal Graph of the synthetic setting. A denotes action, X context, G unobserved group label, Y outcome and P unobserved potentials. Unobserved elements are dotted.

Then, we generate three datasets ("noisymoons, noisycircles, and anisotropic", abbreviated respect. "moons, circles, and GMM" in Table 2.4 and illustrated in Figure 2.7, with two-dimensional contexts on 2 or 3 groups and different choices of $\mathcal{P}_{\mathcal{X},P|\mathcal{G}}$.

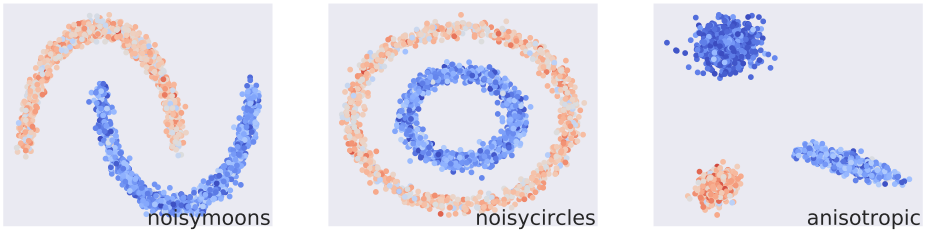

Figure 2.7: Contexts (points in $\mathbb{R}^2$), and potentials represented by a color map for the synthetic datasets. Learned policies should vary with the context to adapt to the underlying potentials.

The goal is then to find a model θ associated to a policy $\pi_\theta(a|x)$ that maximizes reward by adapting to an unobserved potential. For our experiments, potentials are normally distributed conditionally on the group index, $p|g \sim \mathcal{N}(\mu_g, \sigma^2)$. As many real-world applications feature a reward function that increases first with the action up to a peak and finally drops, we have chosen a piecewise linear function peaked at $a = p$ (see Appendix 2.9, Figure 2.18), that mimics reward over the CoCoAdataset presented in Section 2.6. In bidding applications, a potential may represent an unknown true value for an advertisement, and the reward is then maximized when the bid (action) matches this value. In medicine, increasing drug dosage may increase treatment effectiveness but if dosage exceeds a threshold, secondary effects may appear and eclipse benefits (Barnes and Eltherington, 1966).

vs.

Semi-synthetic setting with medical data. We follow the setup of Kallus and Zhou (2018) using a dataset on dosage of the Warfarin blood thinner drug (War, 2009). The dataset consists of covariates about patients along with a dosage treatment prescription by a medical expert, which is a scalar value and thus makes the setting useful for continuous action modelling. While the dataset is supervised, we simulate a contextual bandit environment by using a hand-crafted reward function that is maximal for actions a that are within 10% of the expert's therapeutic drug dosage, following Kallus and Zhou (2018).

Specifically, the semi-synthetic cost inputs prescriptions from medical experts to obtain $y(a, x) = \max(|a - t^*| - 0.1t^*, 0)$, so as to mimic the expert prediction. The logging policy π_{θ_0} samples actions $a \sim \pi_{\theta_0}$ contextually to a patient's body mass index (BMI) score $Z_{BMI} = \frac{x_{\text{BMI}} - \mu_{\text{BMI}}}{\sigma_{\text{BMI}}}$ and can be analytically written with i.i.d noise $e \sim \mathcal{N}(0, 1)$, moments of the therapeutic dose distribution μ_T^*, σ_T^* such that $a = \mu_T^* + \sigma_T^* \sqrt{\theta} Z_{BMI} + \sigma_T^* \sqrt{1 - \theta}\varepsilon$ ($\theta = 0.5$ in the setup of Kallus and Zhou (2018)). The logging probability density function thus is a continuous density of a standard normal distribution over the quantity $\frac{a - \mu_T^* + \sigma_T^* \sqrt{\theta} Z_{BMI}}{\sigma_T^* \sqrt{1-\theta}}$.

Evaluation methodology For synthetic datasets, we generate training, validation, and test sets of size 10 000 each. For the CoCoA dataset, we consider a 50%-25%-25% training-validation-test sets. We then run each method with 5 different random intializations such that the initial policy is close to the logging policy. Hyperparameters are selected on a validation set with logged bandit feedback as explained in Algorithm 5. We use an offline SNIPS estimate of the obtained policies, while discarding solutions deemed unsafe with the importance sampling diagnostic. On the semi-synthetic Warfarin dataset we used a cross-validation procedure to improve model selection due to the low dataset size. For estimating the final test performance and confidence intervals on synthetic and on semi-synthetic datasets, we use an online estimate by leveraging the known reward function and taking a Monte Carlo average with 100 action samples per context: this accounts for the randomness of the policy itself over given fixed samples. For offline estimates we leverage the randomness across samples to build confidence intervals: we use a 100-fold bootstrap and take percentiles of the distribution of rewards. For the CoCoA dataset, we report SNIPS estimates for the test metrics.

Empirical Evaluation

We now evaluate our proposed CLP policy parametrization and the influence of optimization in counterfactural risk minimization problems.

Continuous action space requires more than naive discretization. In Figure 2.8, we compare our continuous parametrization to discretization strategies that bucketize the action space and consider stochastic discrete-action policies on the resulting buckets, using IPS and SDM. We add a minimal amount of noise to the deterministic DM in order to pass the $n_{\text{eff}}/n > \nu$ validation criterion, and experimented different hyperparameters and models which were selected with the offline evaluation procedure. On all synthetic datasets, the CLP continuous modeling associated to the IPS perform significantly better than discrete approaches (see also Appendix 2.9), across all choices considered for the number of anchor points/buckets. To achieve a reasonable performance, naive discretization strategies require a much finer grid, and are thus also more computationally costly. The plots also show that our (stochastic) direct

method strategy, where we use the same parametrization is overall outperformed by the CLP parametrization combined to IPS, highlighting a benefit of using counterfactual methods compared to a direct fit to observed rewards.

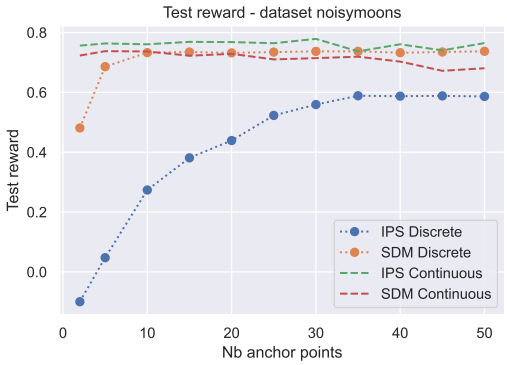

Figure 2.8: Continuous vs discretization strategies. Test rewards on NoisyMoons dataset with varying numbers of anchor points for our continuous parametrization for IPS and SDM, versus naive discretization with softmax policies. Note that few anchor points are sufficient to achieve good results on this dataset; this is not the case for more complicated ones (e.g.,, Warfarin requires at least 15 anchor points).

Counterfactual cost predictor (CLP) provides a competitive parameterization for continuous-action policy learning. We compare our CLP modelling approach to other parameterized modelings (constant, linear and non-linear described in Section 2.3.1) on our synthetic and semi-synthetic setups described in Section 2.7.2 as well as the CoCoAdataset presented in Section 2.6.1.

In Table 2.4, we show a comparison of test rewards for different contextual modellings (associated to different parametric policy classes). We show the performance and the associated variance of the best policy obtained with the offline model selection procedure (Section 2.6.2). Specifically, we consider a grid of hyperparameters and optimized the associated CRM problem with the PPA algorithm (Section 2.4.2). We report here the performances of scIPS and SNIPS estimators. For the Warfarin dataset, following Kallus and Zhou (2018), we only consider the linear context parametrization baseline, since the dataset has categorical features and higher-dimensional contexts. Overall, we find our CLP parameterization to improve over all other contextual modellings, which highlights the effectiveness of the cost predictor at exploiting the continuous action structure. As all the methods here have the same sample efficiency, the superior performance of our method can be imputed to the richer policy class we use and which better models the dependency of contexts and actions that may reduce the approximation error. We can also draw another conclusion: unlike synthetic setups, it is harder to obtain policies that beat the logging policy with large statistical significance on the CoCoAdataset where the logging policy already makes a satisfactory baseline for real-world deployment. Only CLP passes the significance test on this dataset. This corroborates the need for offline evaluation procedures, which were absent from previous works.

Table 2.4: Test rewards (higher the better) for several contextual modellings (see main text for details).

		Noisycircles	NoisyMoons	Anisotropic	Warfarin	CoCoA
Logging policy π_{θ_0}		0.5301	0.5301	0.4533	-13.377	11.34
scIPS	Constant	0.6115 ± 0.0000	0.6116 ± 0.0000	0.6026 ± 0.0000	-8.964 ± 0.001	11.36 ± 0.13
	Linear	0.6113 ± 0.0001	0.7326 ± 0.0001	0.7638 ± 0.0005	-12.857 ± 0.002	11.35 ± 0.02
	Poly	0.6959 ± 0.0001	0.7281 ± 0.0001	0.7448 ± 0.0008	-	10.36 ± 0.11
	CLP	**0.7674** ± 0.0008	**0.7805** ± 0.0004	**0.7703** ± 0.0002	**-8.720** ± 0.001	**11.44*** ± 0.10
SNIPS	Constant	0.6115 ± 0.0001	0.6115 ± 0.0001	0.5930 ± 0.0001	-9.511 ± 0.001	11.32 ± 0.13
	Linear	0.6115 ± 0.0001	0.7360 ± 0.0001	0.7103 ± 0.0003	-10.583 ± 0.005	10.34 ± 0.12
	Poly	0.6969 ± 0.0001	0.7370 ± 0.0001	0.5801 ± 0.0002	-	11.13 ± 0.08
	CLP	**0.6972** ± 0.0001	**0.74091** ± 0.0004	**0.7899** ± 0.0002	**-9.161** ± 0.001	**11.48*** ± 0.14

Soft-clipping improves performance of the counterfactual policy learning. Figure 2.9 shows the improvements in test reward of our optimization-driven strategies for the soft-clipping estimator for the synthetic datasets (see also Appendix 2.9). The points correspond to different choices of the clipping parameter M, models and initialization, with the rest of the hyper-parameters optimized on the validation set using the offline evaluation protocol. This plot also shows that soft clipping provides benefits over hard clipping, perhaps thanks to a more favorable optimization landscape. Overall, these figures confirm that the optimization perspective is important when considering CRM problems.

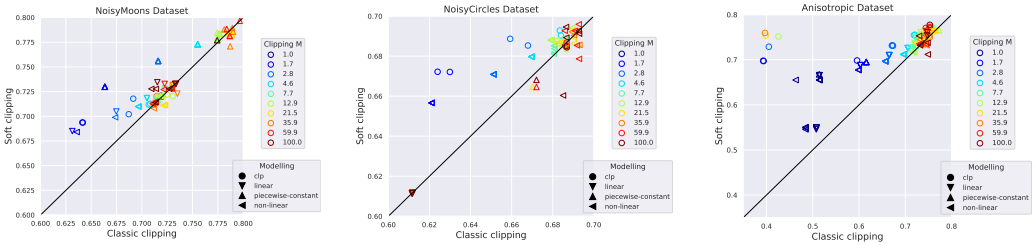

Figure 2.9: Influence of soft-clipping. Relative improvements in the test performance for soft- vs hard-clipping on synthetic datasets. The points correspond to different choices of the clipping parameter, models and initialization.

Soft-clipping improves or competes with other importance weighting transformation strategies. We also experiment on the synthetic datasets comparing our soft clipping approach with other methods which focus is to improve upon the classic clipping strategy for the same optimization purposes. Notably, we consider the (Metelli et al., 2021) method which we adapt to our continuous modeling strategy to enable fair comparison. Moreover, we also added the SWITCH (Wang et al., 2017) as well as the CAB (Su et al., 2019) methods. However, both methods use a direct method term in their estimation, which is difficult to adapt for stochastic policies with continuous actions, as explained in our discussions on doubly robust estimators (see Appendix 2.9). Therefore, we considered discretized strategies and compared them with soft clipped estimator applied to discretized policies. For the discretized strategies, we have used the same anchoring strategies as described before, namely using empirical quantiles of the logged actions (for 1D actions), and have optimized the number of anchor points along with the other hyperparameters using the offline evaluation protocol. We see overall in Table 2.5 that our soft-clipping strategy provides satisfactory performance or

improves upon all weight transforming strategies on the synthetic datasets.

	Noisymoons	Noisycircles	Anisotropic
Logging policy π_{θ_0}	0.5301	0.5301	0.4533
(Wang et al., 2017) (discrete)	0.5786 ± 0.0025	0.5520 ± 0.0026	0.5741 ± 0.0024
(Su et al., 2019) (discrete)	0.5761 ± 0.0024	0.5534 ± 0.0025	0.5705 ± 0.0021
scIPS (discrete)	$\mathbf{0.5888} \pm 0.0022$	$\mathbf{0.5637} \pm 0.0024$	$\mathbf{0.5941} \pm 0.0019$
Metelli et al. (2021) (CLP)	0.7244 ± 0.0005	0.7189 ± 0.0004	$\mathbf{0.7739} \pm 0.0008$
scIPS (CLP)	$\mathbf{0.7674} \pm 0.0008$	$\mathbf{0.7805} \pm 0.0004$	0.7703 ± 0.0002

Table 2.5: Comparison of importance weight transformations on the synthetic datasets, for discretized strategies and for continuous action policies.

Proximal point algorithm (PPA) influences optimization of non-convex CRM objective functions and policy learning performance. We illustrate in Figure 2.10 the improvements in test reward and in training objective of our optimization-driven strategies with the use of the proximal point algorithm (see also Appendix 2.9). Here, each point compares the test metric for fixed models as well as initialization seeds, while optimizing the remaining hyperparameters on the validation set with the offline evaluation protocol. Figure 2.10 (left) illustrates the benefits of the proximal point method when optimizing the (non-convex) CRM objective in a wide range of hyperparameter configurations, while Figure 2.10 (center) shows that in many cases this improves the test reward as well. In our experiments, we have chosen L-BFGS because it was performing best among the solvers we tried (nonlinear conjugate gradient (CG) and Newton) and used 10 PPA iterations. For further information, Figure 2.10 (right) presents a comparison between CG and L-BFGS for different parameters κ and number of iterations. As for computational time, for 10 PPA iterations, the computational overhead was about $3\times$ in comparison to L-BFGS without PPA. This overhead seems to be the price to pay to improve the test reward and obtain better local optima. Overall, these figures confirm that the proximal point algorithm improves performance in CRM optimization problems.

The scIPS estimator along with CLP parametrization and PPA optimization improves upon previous state of the art methods. We also provide a baseline comparison to stochastic direct methods, to Chen et al. (2016) using their surrogate loss formulation for continous actions, to Kallus and Zhou (2018) who propose a counterfactual method using kernel density estimation. Their approach is based on an automatic kernel bandwidth selection procedure which did not perform well on our datasets except Warfarin; instead, we select the best bandwidth on a grid through cross-validation and selecting it through our offline protocol. We also investigate their self-normalized (SN) variant, which is presented in their paper but not used in their experiments; it turned out to have lower performances in practice. Moreover, we experimented using the generic doubly robust method from Demirer et al. (2019) but could not reach satisfactory results using the parameters and feature maps that were used in their empirical section and with the specific closed form estimators for their applications. Nevertheless, by adapting their method with more elaborated models and feature maps, we managed to obtain performances beating the logging policy; these modifications would make

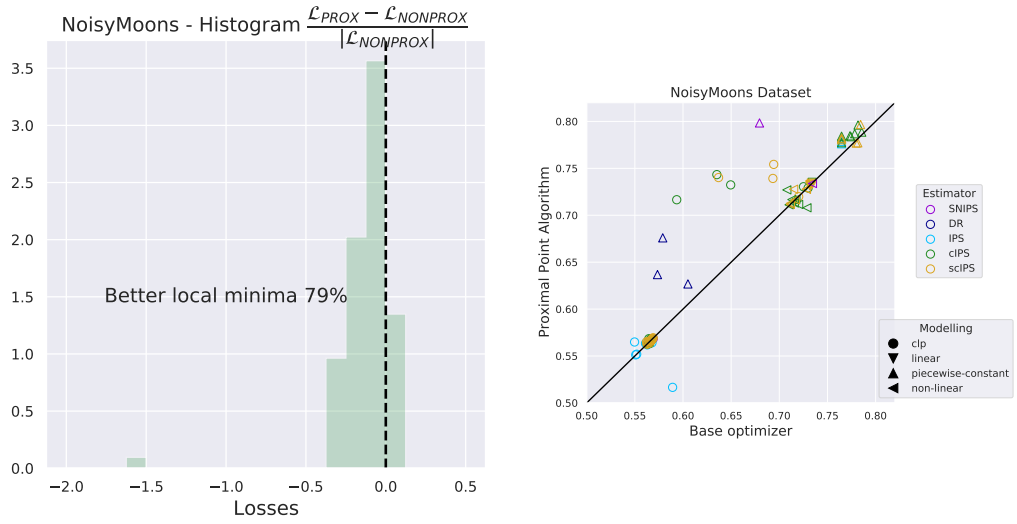

Figure 2.10: Influence of proximal point optimization. Relative improvements in the training objective w and w/o using the proximal point method (left), relative improvements in the test performance w and w/o using the proximal point method (right).

promising directions for future research venues. We eventually also compare to Majzoubi et al. (2020) who propose an offline variant of their contextual bandits algorithm for continuous actions. We used the code of the authors and obtained poor performances with their offline variant but achieved satisfactory performances with their online algorithm, that we provide as a comparison but which does not compare to all previous offline methods. We do not provide results on their method for the CoCoAdataset as we could not access the logged propensities to use the offline evaluation protocol. For the SDM on CoCoA, we did not manage to simultaneously pass the ESS diagnostic and achieve statistical significance, probably due to the noise and variance of the dataset.

	Noisycircles	NoisyMoons	Anisotropic	Warfarin	CoCoA
Stochastic Direct Method	0.6205 ± 0.0004	0.7225 ± 0.0006	0.6383 ± 0.0003	-9.714 ± 0.013	-
Chen et al. (2016)	0.608 ± 0.0002	0.645 ± 0.0003	0.754 ± 0.0002	-9.407 ± 0.004	11.03 ± 0.15
Kallus and Zhou (2018)	0.612 ± 0.0001	0.734 ± 0.0001	0.785 ± 0.0002	-10.19^*	11.38 ± 0.07
SN-Kallus and Zhou (2018)	0.609 ± 0.0001	0.595 ± 0.0001	0.652 ± 0.0001	-12.569 ± 0.001	9.14 ± 0.94
Majzoubi et al. (2020) offline	0.589 ± 0.0011	0.592 ± 0.0011	0.569 ± 0.0012	-12.236 ± 0.2548	-
Ours	$\mathbf{0.767} \pm 0.0008$	$\mathbf{0.781} \pm 0.0004$	$\mathbf{0.770} \pm 0.0002$	$\mathbf{-8.720} \pm 0.001$	$\mathbf{11.44}^* \pm 0.10$
Majzoubi et al. (2020) online	0.713 ± 0.0041	0.710 ± 0.0026	0.771 ± 0.011	-11.672 ± 0.221	-

Table 2.6: Test rewards (higher the better) for previous methods for the logged bandit problem with continuous actions

2.8. Discussions

In this chapter, we addressed the problem of counterfactual learning of stochastic policies on real data with continuous actions. This raises several challenges about different steps of the CRM pipeline such as (i) modelization, (ii) optimization, and (iii) evaluation. First, we propose a new parametrization based on a joint kernel embedding of contexts and actions,

showing competitive performance. Second, we underline the importance of optimization in CRM formulations with soft-clipping and proximal point methods. We provide statistical guarantees of our estimator and the policy class we introduced. Third, we propose an offline evaluation protocol and a new large-scale dataset, which, to the best of our knowledge, is the first with real-world logged propensities and continuous actions. For future research directions, we would like to discuss the doubly robust estimator (which achieves the best results in the discrete action case) with the CLP parametrization of stochastic policies with continuous actions, as well as further optimization perspectives and the offline model selection.

Doubly-robust estimators for continuous action policies While Demirer et al. (2019) provide a doubly robust (DR) estimator on continuous action using a semi-parametric model of the policy value function, we did not propose a doubly-robust estimator along with our CLP modelling. Indeed, their policy learning is performed in two stages (i) estimate a doubly robust parameter $\theta^{DR}(x, a, r)$ in the semi-parametric model of the value function $\mathbb{E}[y|a, x] = V(a, x) = \langle \theta_*(x), \phi(a, x) \rangle$ and (ii) learn a policy in the empirical Monte Carlo estimate of the policy value by solving

$$\min_{\pi \in \Pi} \left\{ \hat{V}^{DR}(\pi) := \frac{1}{n} \sum_{i=1}^{n} \langle \hat{\theta}^{DR}(x_i, a_i, r_i), \phi(\pi(x_i), x_i) \rangle \right\}.$$

The doubly robust estimation is performed with respect to the first parameter learned in (i) for the value function, while we follow the CRM setting Swaminathan and Joachims (2015a) and directly derive estimators of the policy value (risk) itself, which would correspond to the phase (ii). To derive an estimate a DR estimator of such policy values, we tried extending the standard DR approach for discrete actions from Dudik et al. (2011) to continuous actions by using our anchors points, but these worked poorly in practice, as detailed in Appendix 2.9. Actually, a proper DR method for estimating the expectation of a policy risk likely requires new techniques for dealing with integration over the training policy in the direct method term, which is non-trivial and goes beyond the scope of this work. We hope to be able to do this in the future.

On further optimization perspectives and offline model selection As mentioned in Section 2.4, our use of the proximal point algorithm differs from approaches that enhance policies to stay close to the logging policies which modify the objective function as in (Schulman et al., 2017) in reinforcement learning. Another avenue for future work would be to investigate distributionnally robust methods that do such modifications of the objective function or add constraints on the distribution being optimized. The policy thereof obtained would thus be closer to the logging policy in the CRM context. Moreover, as we showed in Section 2.6.2 with importance sampling estimates and diagnostics, the offline decision becomes less statistically significant as the evaluated policy is far from the logging policy. Investigating how the distributionally robust optimization would yield better CRM solutions with regards to the offline evaluation protocol would make an interesting future direction of research.

2.9. Appendices

This appendix is organized as follows. Appendix 2.9 provides motivation for counterfactual methods as opposed to direct approaches. Appendix 2.9 motivates the need for clipping strategies on real datasets. Appendix 2.9 motivates the offline evaluation protocol with experiments justifying the need for appropriate diagnostics and statistical testing for importance sampling. Appendix 2.9 provides the omitted proofs and details of Section2.4 and 2.5. Then, Appendix 2.9 is devoted to experimental details that were omitted from the main chapter for space limitation reasons, and which are important for reproducing our results (see also the code provided with the submission). In Appendix 2.9, we present additional experimental results to those in the main chapter.

2.10. Motivation for Counterfactual Methods

Direct methods (DM) learns a reward/cost predictor over the joint context-action space $\mathcal{X} \times \mathcal{A}$ but ignore the potential mismatch between the evaluated policy and the logging policy and π_{θ_0}. When the logged data does not cover the joint context-action space $\mathcal{X} \times \mathcal{A}$ sufficiently, direct methods rather fit the region where the data has been sampled and may therefore lead to overfitting (Bottou et al., 2013; Dudik et al., 2011; Swaminathan and Joachims, 2015b). Counterfactual methods instead learn probability distributions directly with a re-weighting procedure which allow them to fit the context-action space even with fewer samples.

In this toy setting we aim to illustrate this phenomenon for the DM and the counterfactual method. We create a synthetic 'Chess' environment of uni-dimensional contexts and actions where the logging policy purposely covers only a small area of the action space, as illustrated in Fig. 2.11. The reward function is either 0, 0.5 or 1 in some areas which follow a chess pattern. We use a lognormal logging policy which is peaked in low action values but still has a common support with the policies we optimize using the CRM or the DM.

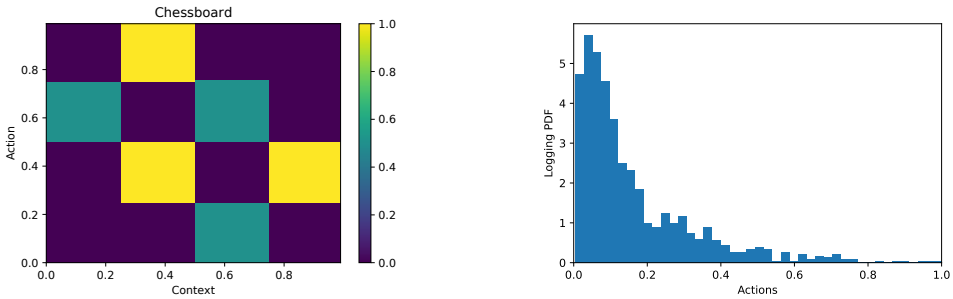

Figure 2.11: 'Chess' toy synthetic setting (left) and lognormal logging policy (right).

Having set that environment and the logging policy, we illustrate in Fig. 2.12 the logging dataset, the actions sampled by the policy learned by a Direct Method and eventually the actions sampled by a counterfactual IPS estimator. To assess a fair comparison between the two methods, we use the same continuous action modelling with the same parameters (CLP parametrization with $m = 5$ anchor points).

This toy example illustrate the mentioned phenomenon in how the counterfactual

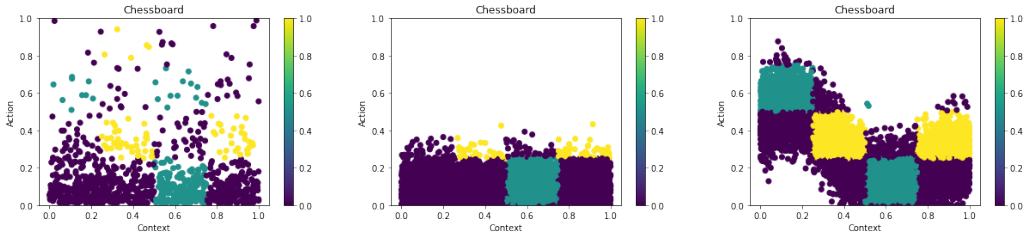

Figure 2.12: Logged data (left), action sampled by direct method (middle) and action sampled by counterfactual policy.

estimator learns a re-balanced distribution that maps the contexts to the actions generating higher rewards than the DM. The latter only learns a mapping that is close to the actions sampled by the logging and only covers a smaller set of actions.

2.11. Motivation for Clipped Estimators

In this section we provide a motivation example for clipping strategies in counterfactual systems in a toy example.

In Figure 2.13 we provide an example of large variance and loss overfitting problem.

We recall the data generation: a hidden group label g in $\mathcal{G}$ is drawn, and influences the associated context distribution x and of an unobserved potential p in $\mathbb{R}$, according to a joint conditional distribution $P_{X,P|G}$ The observed reward r is then a function of the context x, action a, and potential p. Here, we design one outlier (big red dark dot on Figure 2.13 left). This point has a noisy reward r, higher than neighbors, and a potential p high as its neighbors have a low potential. We artificially added a noise in the reward function f that can be written as:

$$r = f(a, x, p) + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, 1)$$

As explained in Section 2.7.2, the reward function is a linear function, with its maximum localized at the point $x = p(x)$, i.e. at the potential sampled. The observability of the potential p is only through this reward function f. Hereafter, we compare the optimal policy computed, using different types of estimators.

The task is to predict the high potentials (red circles) and low potentials (blue circles) in the ground truth data (left). Unfortunately, a rare event sample with high potential is put in the low potential cluster (big dark red dot). The action taken by the logging policy is low while the reward is high: this sample is an outlier because it has a high reward while being a high potential that has been predicted with a low action. The resulting unclipped estimator is biased and overfits this high reward/low propensity sample. The rewards of the points around this outlier are low as the diameter of the points in the middle figure show. Inversely, clipped estimator with soft-clipping succeeds to learn the potential distributions, does not overfit the outlier, and has larger rewards than the clipping policies as the diameter of the

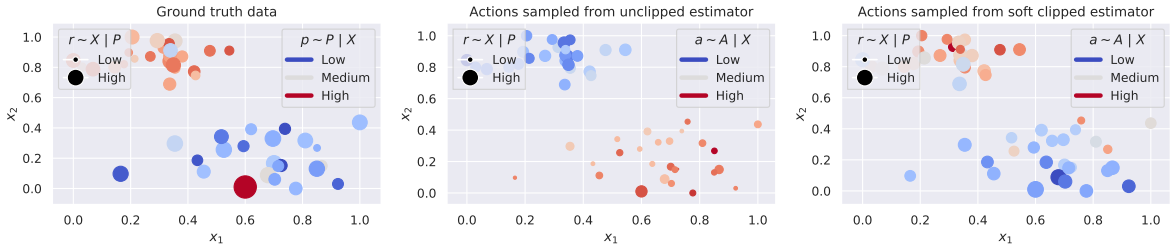

Figure 2.13: High variance and loss overfitting. Unlikely ($\pi_{0,i} \approx 0$) sample $(x_1, x_2) = (0.6, 0.)$ with high reward r (left) results in larger variance and loss overfitting for the unclipped estimator (middle) unlike clipped estimator (right).

points show.

2.12. Motivation for Offline Evaluation Protocol

In this part we demonstrate the offline/online correlation of the estimator we use for real-world systems and for validation of our methods even in synthetic and semi-synthetic setups. We provide further explanations of the necessity of importance sampling diagnostics and we perform experiments to empirically assess the rate of false discoveries of our protocol.

2.12.1. Correlation of Self-Normalized Importance Sampling with Online Rewards

We show in Figures 2.14,2.16,2.15 comparisons of IPS and SNIPS against an on-policy estimate of the reward for policies obtained from our experiments for linear and non-linear contextual modellings on the synthetic datasets, where policies can be directly evaluated online. Each point represents an experiment for a model and a hyperparameter combination. We measure the R^2 score to assess the quality of the estimation, and find that the SNIPS estimator is indeed more robust and gives a better fit to the on-policy estimate. Note also that overall the IPS estimates illustrate severe variance compared to the SNIPS estimate. While SNIPS indeed reduces the variance of the estimate, the bias it introduces does not deteriorate too much its (positive) correlation with the online evaluation.

These figures further justify the choice of the self-normalized estimator SNIPS (Swaminathan and Joachims, 2015b) for offline evaluation and validation to estimate the reward on held-out logged bandit data. While the figures show here that the SNIPS estimator achieves a better bias-variance tradeoff, we note also that the SNIPS estimator has low variance for both low and high reward policies. It is indeed more robust to the reward distribution thanks to its equivariance property (Swaminathan and Joachims, 2015b) to additive shifts and does not require hyperparameter tuning.

2.12.2. Importance Sampling Diagnostics in What-If simulations

Importance sampling estimators rely on weighted observations to address the distribution mismatch for offline evaluation, which may cause large variance of the estimator. Notably, when the evaluated policy differs too much from the logging policy, many importance weights are large and the estimator is inaccurate. We provide here a motivating example to illustrate the effect of importance sampling diagnostics in a simple scenario.

Figure 2.14: Correlation between offline and online estimates on Anisotropic synthetic data. Linear (left) and non-linear (right) contextual modellings. Ideal fit would be $y = x$.

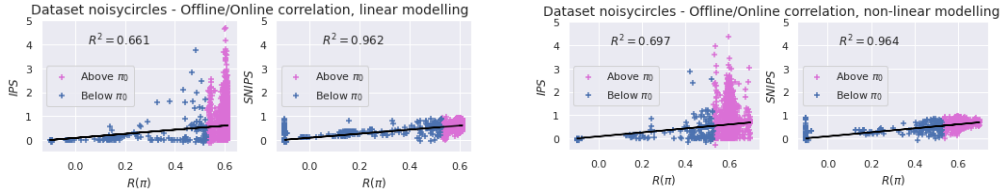

Figure 2.15: Correlation between offline and online estimates on NoisyCircles synthetic data. Linear (left) and non-linear (right) contextual modellings. Ideal fit would be $y = x$.

When evaluating with SNIPS, we consider an "effective sample size" quantity given in terms of the importance weights $w_i = \pi_\theta(a_i|x_i)/\pi_{\theta_0}(a_i|x_i)$ by $n_e = (\sum_{i=1}^n w_i)^2 / \sum_{i=1}^n w_i^2$. When this quantity is much smaller than the sample size n, this indicates that only few of the examples contribute to the estimate, so that the obtained value is likely a poor estimate. Apart from that, we note also that IPS weights have an expectation of 1 when summed over the logging policy distribution (that is $\mathbb{E}_{(x,a)\sim\pi_{\theta_0}}[\pi_\theta(a|x)/\pi_{\theta_0}(a|x)] = 1.$). Therefore, another sanity check, which is valid for any estimator, is to look for the empirical mean $1/n \sum_{i=1}^n \pi_\theta(a_i|x_i)/\pi_{0,i}$ and compare its deviation to 1. In the example below, we illustrate three diagnostics: (i) the one based on effective sample size described in Section 2.6; (ii) confidence intervals, and (iii) empirical mean of IPS weights. The three of them coincide and allow us to remove test estimates when the diagnostics fail.

Example 2.12.1. *What-if simulation: For x in $\mathbb{R}^d$, let $\max(x) = \max_{1\leq j\leq d} x_j$; we wish to estimate $\mathbb{E}(\max(X))$ for X i.i.d $\sim \pi_\mu = \mathcal{N}(\mu, \sigma)$ where samples are drawn from a logging policy $\pi_{\theta_0} = \log\mathcal{N}(\lambda_0, \sigma_0)$ ($d = 3, (\lambda_0, \sigma_0) = (1, 1/2)$) and analyze parameters μ around the mode of the logging policy μ_0 with fixed variance $\sigma = 1/2$. In this parametrized policy example, we see in Fig. 2.17 that $n_e/n \ll 1$, confidence interval range increases and $\sum_{i=1}^n \frac{\pi_\mu(a_i|x_i)}{\pi_{0,i}} \neq 1$ when the parameter μ of the policy being evaluated is far away from the logging policy mode μ_0.*

Note that in this example, the parameterized distribution that is learned (multivariate Gaussian) is not the same as the parameterized distribution of the logging policy (multivariate Lognormal). The skewness of the logging policy may explain the asymmetry of the plots. This points out another practical problem: even though different parametrization of policies is theoretically possible, the probability density masses overlap is in practice what is most important to ensure successful importance sampling. This observation is of utmost interest for real-life applications where the initialization of a policy to be learned needs to be "close" to the logging policy; otherwise importance sampling may fail from the very first iteration of an optimization in learning problems.

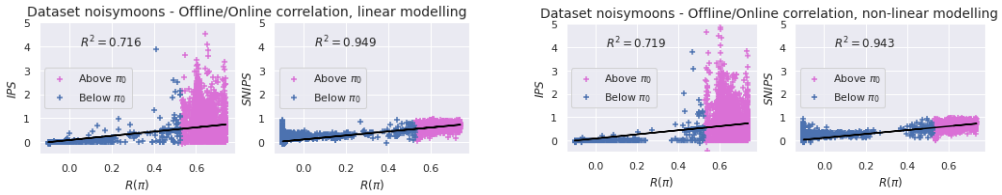

Figure 2.16: Correlation between offline and online estimates on NoisyMoons synthetic data. Linear (left) and non-linear (right) contextual modellings. Ideal fit would be $y = x$.

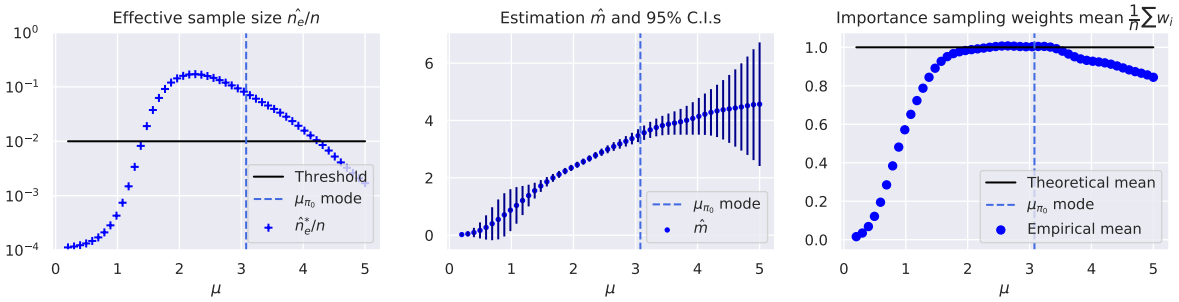

Figure 2.17: Importance sampling diagnostics. Ideal importance sampling: i) effective sample n_e/n close to 1, ii) low confidence intervals (C.I.s) for $\hat{m}$, iii) empirical mean $\frac{1}{n}\sum_i w_i$ close to 1. Note that when μ differs too much from μ_0, importance sampling fails.

2.13. Analysis of the Excess Risk

In this appendix, we provide details and proofs on the excess risk guarantees that are given in Section 2.4 and 2.5.

We start by recalling the definitions of an ε covering and the one of our soft-clipping operator ζ provided in Eq. (2.6).

Definition 2.13.1 (Epsilon Covering and Metric Entropy). *An ε-covering is the smallest cardinality $|A_0|$ of a subset $A_0 \subseteq A$ such that A is contained in the union of balls of radius ε centered in points in A_0, in the metric induced by a norm $\|\cdot\|$. The cardinality of the smallest ε-covering is denoted by $\mathcal{H}(\varepsilon, A, \|\cdot\|)$ and its logarithm is called the metric entropy.*

For any threshold parameter $M \geq 0$ and importance weight $w \geq 0$, the soft-clip operator ζ is defined by

$$\zeta(w, M) = \begin{cases} w & \text{if } w \leq M \\ \alpha(M)\log\left(w + \alpha(M) - M\right) & \text{otherwise} \end{cases},$$

where $\alpha(M)$ is such that $\alpha(M)\log(\alpha(M)) = M$.

2.13.1. Omitted Proofs

We start by defining our complexity measure $C_n(\Theta, M)$, which will be upper-bounded by the metric entropy in sup-norm at level $\varepsilon = 1/n$ of the following function set,

$$\mathcal{F}_{\Theta,M} := \left\{ f_\theta : (x, a, y) \mapsto 1 + \frac{y}{S}\zeta\left(\frac{\pi_\theta(a|x)}{\pi_{\theta_0}(a|x)}, M\right) \quad \text{for some } \theta \in \Theta \right\}, \tag{2.12}$$

where $S=\zeta(W,M)$. The function set corresponds to clipped prediction errors of policies π normalized into $[0,1]$. More precisely, to define rigorously $C_n(\Theta,M)$, we denote for any $n\geq 1$ and $\varepsilon>0$, the complexity of a class $\mathcal{F}$ by

$$\mathcal{H}_\infty(\varepsilon,\mathcal{F},n)=\sup_{(x_i,a_i,y_i)\in(\mathcal{X}\times\mathcal{A}\times\mathcal{Y})^n}\mathcal{H}(\varepsilon,\mathcal{F}\big(\{x_i,a_i,y_i\}\big),\|\cdot\|_\infty)\,, \tag{2.13}$$

where $\mathcal{F}\big(\{x_i,a_i,y_i\}\big)=\big\{\big(f(x_1,a_1,y_1),\dots,f(x_n,a_n,y_n)\big),f\in\mathcal{F}\big\}\subseteq\mathbb{R}^n$. Then, $C_n(\Theta,M)$ is defined by

$$C_n(\Theta,M)=\log\mathcal{H}_\infty(1/n,\mathcal{F}_{\Theta,M},2n)\,. \tag{2.14}$$

We are now ready to prove Proposition 2.4.1 that we restate below.

Proposition 2.4.1 (Generalization bound for $\hat{L}^{\text{scIPS}}(\theta)$)**.** *Let Θ be a parameter space for the policy class Π_Θ and π_{θ_0} be a logging policy. Let $s_0=(x_i,a_i,y_i)_{i=1,\dots,n}$ the logging dataset for which actions are sampled under π_{θ_0}. Assume that the losses $y\in[-1,0]$ to be bounded a.s. and that the importance weights are bounded by W. Then, with probability at least $1-\delta$, the IPS estimator with soft clipping* (2.7) *on n samples from s_0 satisfies*

$$\forall\pi\in\Pi,\qquad L(\theta)\leq\hat{L}_{scIPS}(\theta)+O\left(\sqrt{\frac{\hat{V}_{scIPS}(\theta)\big(C_n(\Theta,M)+\log\frac{1}{\delta}\big)}{n}}+\frac{S\big(C_n(\Theta,M)+\log\frac{1}{\delta}\big)}{n}\right),$$

where $S=\zeta(W,M)=O(\log W)$, $\hat{V}^{scIPS}(\theta)$ denotes the empirical variance of the cost estimates (1.16)*, and $C_n(\Theta,M)$ is a complexity measure* (2.14) *of the policy class.*

Proof. Let Θ be a parameter space and Π_Θ be a policy class, π_{θ_0} be a logging policy, and $\delta>0$. Let $M\geq 0$ be a threshold parameter, $W\geq\sup_{a,x}\{\pi_\theta(a|x)/\pi_{\theta_0}(a|x)\}\geq 0$ a bound on the importance weights, and $S=\zeta(W,M)$.

Let first consider the finite setting, in which case $C_n(\Theta,M)\leq\log|\Theta|$. Since all functions in $\mathcal{F}_{\Theta,M}$ defined in Eq. (2.12) take values in $[0,1]$, we can apply the concentration bound of Maurer and Pontil (2009, Corollary 5) to $\mathcal{F}_{\Theta,M}$, which yields that with probability at least $1-\delta$, for any $\theta\in\Theta$

$$\mathbb{E}_{x,\theta,y}[f_\theta(x,a,y)]-\frac{1}{n}\sum_{i=1}^n f_\theta(x_i,a_i,y_i)\leq\sqrt{\frac{2\hat{V}^{\text{scIPS}}(\theta)\log(2|\Theta|/\delta)}{n}}+\frac{7\log(2|\Theta|/\delta)}{3(n-1)}\,, \tag{2.15}$$

where $\hat{V}_{\text{scIPS}}(\theta)$ is the sample variance defined in (2.8). Furthermore, note that by construction of the f_θ, for any $\theta\in\Theta$,

$$\mathbb{E}_{x,\theta,y}[f_\theta(x,a,y)]=1+\frac{L^M(\theta)}{S}\quad\text{ and }\quad\frac{1}{n}\sum_{i=1}^n f_\theta(x_i,a_i,y_i)=1+\frac{\hat{L}^{\text{scIPS}}(\theta)}{S}\,,$$

where $L^M(\theta)=\mathbb{E}_{x,\theta,y}\big[y\zeta\big(\pi_\theta(a|x)/\pi_{\theta_0}(a|x),M\big)\big]$ denotes the clipped expected risk of the policy θ and $\hat{L}^{\text{scIPS}}(\pi)$ is defined in (2.7). Thus, multiplying (2.15) by S and using that $L(\theta)\leq L^M(\theta)$ (since $y\leq 0$ and $\zeta(w,M)\leq w$ for all w), we get that with probability $1-\delta$,

$$L(\theta)\leq\hat{L}^{\text{scIPS}}(\theta)+\sqrt{\frac{2\hat{V}^{\text{scIPS}}(\theta)\log(2|\Theta|/\delta)}{n}}+S\frac{7\log(2|\Theta|/\delta)}{3(n-1)},\qquad\forall\theta\in\Theta\,.$$

The finite setting may finally be extended to infinite policy classes by leveraging Maurer and Pontil (2009, Theorem 6) as in (Swaminathan and Joachims, 2015a). This essentially consists in replacing $|\Theta|$ above with an empirical ℓ_∞ covering number of $\mathcal{F}_{\Theta,M}$ of size $\mathcal{H}_\infty(1/n, \mathcal{F}_{\Theta,M}, 2n)$. Note that the number of empirical samples $2n$ is due to the double-sample method used by Maurer and Pontil (2009).

□

We now state the excess risk upper-bound Proposition 2.13.1 and provide the proof. The following proposition is an intermediate result that will allow us to derive the Proposition 2.5.1.

Proposition 2.13.1. *Consider the notations and assumptions of Proposition 2.4.1. Let $\hat{\theta}^{CRM}$ be the solution of the CRM problem in Eq.* (2.10)*. Let $\theta^* \in \arg\min_{\theta\in\Theta} L(\theta)$. Then, the choice $\lambda = 3\sqrt{3}\big(C_n(\Theta, M) + \log(30/\delta)\big)^{1/2}$ implies with probability at least $1-\delta$ the following upper-bound on the excess risk*

$$\Delta(\hat{\theta}^{CRM}) \leq \sqrt{\frac{32V_M(\theta^*)\big(C_n(\Theta, M) + \log\frac{30}{\delta}\big)}{n}} + \frac{22S\big(C_n(\Theta, M) + \log\frac{30}{\delta}\big)}{n-1} + h_M(\theta^*)\,,$$

where $V_M^2(\theta^*)$ and $h_M(\theta^*)$ are the variance and bias of the is the clipped estimator of θ^* and respectively defined in (2.16) *and* (2.19)*.*

Proof. We consider the notations of the proof of Proposition 2.4.1. Fix $\theta^* \in \arg\min_{\theta\in\Theta} L(\theta)$. Applying, Theorem 15 of Maurer and Pontil (2009)[1] to the function set $\mathcal{F}_{\Theta,M}$ defined in (2.12), we get w.p. $1-\delta$

$$\begin{aligned}\mathbb{E}_{x,\theta_0,y}\left[f_{\hat{\theta}^{CRM}}(x,a,y)\right] - \mathbb{E}_{x,\theta_0,y}\left[f_{\theta^*}(x,a,y)\right] \\ \leq \sqrt{\frac{32\mathrm{Var}_{x,\theta_0,y}\big[f_{\theta^*}(x,a,y)\big]\big(C_n(\Theta,M) + \log\frac{30}{\delta}\big)}{n}} + \frac{22\big(C_n(\Theta,M) + \log\frac{30}{\delta}\big)}{n-1}\,.\end{aligned}$$

Using the definition of $f_\theta(x,a,y)$ (2.12), we have

$$\mathbb{E}_{x,\theta_0,y}\left[f_\theta(x,a,y)\right] = 1 + \frac{L^M(\theta)}{S} \text{ and } \mathrm{Var}_{x,\theta_0,y}\big[f_\theta(x,a,y)\big] = \frac{V_M^2(\theta)}{S^2}\,,$$

where

$$V_M^2(\theta) = \mathrm{Var}_{x,\theta_0,y}\left(y\zeta\left(\frac{\pi_\theta(a|x)}{\pi_{\theta_0}(a|x)}, M\right)\right)\,. \tag{2.16}$$

Substituting into the previous bound, this entails

$$L^M(\hat{\theta}^{CRM}) - L^M(\theta^*) \leq \sqrt{\frac{32V_M(\theta^*)\big(C_n(\Theta,M) + \log\frac{30}{\delta}\big)}{n}} + \frac{22S\big(C_n(\Theta,M) + \log\frac{30}{\delta}\big)}{n-1}\,. \tag{2.17}$$

[1]Note that in their notation, $\log \mathcal{M}_n(\Pi_\theta)$ equals $C_n(\Theta, M) + \log(10)$, $\mathbf{X}$ is the dataset $\{(x_i, a_i, y_i)\}_{1\leq i\leq n}$ where (x_i, a_i, y_i) is the observational dataset s_0, and $P(\cdot, \mu)$ is the expectation with respect to one test sample $\mathbb{E}_{x,\mu,y}[\,\cdot\,]$.

To conclude the proof, it only remains to replace the clipped risk L^M with the true risk L. On the one hand, since the costs y take values into $[-1, 0]$, we have $y\zeta\left(\pi_{\theta^*}(a|x)/\pi_{\theta_0}(a|x), M\right) \geq y\pi_\theta(a|x)/\pi_{\theta_0}(a|x)$, which yields

$$L(\hat{\theta}^{CRM}) \leq L^M(\hat{\theta}^{CRM})\,. \tag{2.18}$$

On the other-hand, by defining the bias

$$h_M(\theta^*) = \mathbb{E}_{x,\theta_0,y}\left[y\zeta\left(M, \frac{\pi_{\theta^*}(a|x)}{\pi_{\theta_0}(a|x)}\right) - y\frac{\pi_{\theta^*}(a|x)}{\pi_{\theta_0}(a|x)}\right] \tag{2.19}$$

we also have $-L(\theta^*) - h_M \leq -L^M(\theta^*)$, which together with (2.17) and (2.18) finally concludes the proof

$$L(\hat{\theta}^{CRM}) - L(\theta^*) \leq \sqrt{\frac{32V_M(\theta^*)\big(C_n(\Theta, M) + \log\frac{30}{\delta}\big)}{n}} + \frac{22S\big(C_n(\Theta, M) + \log\frac{30}{\delta}\big)}{n-1} + h_M(\theta^*)\,.$$

□

We can now use the latter to prove Proposition 2.5.1 that is restated below.

Proposition 2.5.1. *Consider the notations and assumptions of Proposition 2.4.1. Let $\hat{\theta}^{CRM}$ be the solution of the CRM problem in Eq.* (2.10)*. Then, with well chosen parameters λ and M, denoting the variance $\nu_*^2 = \mathrm{Var}_{\pi_{\theta_0}}\left[\pi_{\theta^*}(a|x)/\pi_{\theta_0}(a|x)\right]$, with probability at least $1-\delta$, the excess risk is upper bounded as:*

$$\Delta(\hat{\theta}^{CRM}) \lesssim \sqrt{\frac{(1+\nu_*^2)\log(W+e)\big(C_n(\Theta, M) + \log\frac{1}{\delta}\big)}{n}} + \frac{\log(W+e)(C_n(\Theta, M) + \log\frac{1}{\delta})}{n}\,,$$

where $\lesssim$ hides universal multiplicative constants. In particular, assuming also that $\pi_{\theta_0}(x|a)^{-1}$ are uniformly bounded, the complexity of the class Π^{CLP} described in Section 2.3.1, applied with a bounded kernel and $\Theta = \big\{\beta \in \mathbb{R}^m,\ s.t\ \|\beta\| \leq C\big\} \times \big\{\sigma\big\}$, is of order

$$C_n(\Theta^{CLP}, M) \leq O(m \log n)\,,$$

where m is the size of the Nyström dictionary and $O(\cdot)$ hides multiplicative constants independent of n and m (see (2.28)*).*

Proof. We first consider a general policy class Π and some $\pi^* \in \Pi$. In this proof, to ease the notation, we write $\mathbb{E}_{\pi_{\theta_0}}[\,\cdot\,]$, $\mathrm{Var}_{\pi_{\theta_0}}[\,\cdot\,]$, and $\mathcal{P}_{\pi_{\theta_0}}(\,\cdot\,)$ to respectively refer to $\mathbb{E}_{(x,a,y)\sim\mathcal{P}_{\pi_{\theta_0}}}[\,\cdot\,]$, $\mathrm{Var}_{(x,a,y)\sim\mathcal{P}_{\pi_{\theta_0}}}[\,\cdot\,]$, and $\mathcal{P}_{(x,a,y)\sim\mathcal{P}_{\pi_{\theta_0}}}(\,\cdot\,)$.

We consider the notation of the proof of Proposition 2.13.1 and start from its risk upper-bound

$$L(\hat{\theta}^{CRM}) - L(\theta^*) \leq \sqrt{\frac{32V_M(\theta^*)\big(C_n(\Theta, M) + \log\frac{30}{\delta}\big)}{n}} + \frac{22S\big(C_n(\Theta, M) + \log\frac{30}{\delta}\big)}{n-1} + h_M(\theta^*)\,, \tag{2.20}$$

where we recall, for any threshold M, the definitions of the bias and the variance of the clipped estimator of θ^*,

$$h_M(\theta^*) = \mathbb{E}_{x,\theta_0,y}\left[y\Big(\zeta\Big(\frac{\pi_{\theta^*}(a|x)}{\pi_{\theta_0}(a|x)}, M\Big) - \frac{\pi_{\theta^*}(a|x)}{\pi_{\theta_0}(a|x)}\Big)\right] \text{ and } V_M^2(\theta^*) = \mathrm{Var}_{x,\theta_0,y}\left[y\zeta\left(\frac{\pi_{\theta^*}(a|x)}{\pi_{\theta_0}(a|x)}, M\right)\right].$$

Step 1: *For any threshold M, upper-bound of the variance $V_M(\theta^*)$ and the bias $h_M(\theta^*)$.*
By assumption, the (unclipped) variance of $\pi_{\theta^*}/\pi_{\theta_0}$ is bounded and we write

$$\nu_*^2 = \mathrm{Var}_{x,\theta_0,y}\left[\frac{\pi_{\theta^*}(a|x)}{\pi_{\theta_0}(a|x)}\right] = \mathbb{E}_{(x,a,y)\sim\mathcal{P}_{\pi_{\theta_0}}}\left[\left(\frac{\pi_{\theta^*}(a|x)}{\pi_{\theta_0}(a|x)} - 1\right)^2\right].$$

First, we bound the clipped variance as

$$\begin{aligned}
V_M^2(\theta^*) &= \mathrm{Var}_{x,\theta_0,y}\left[y\zeta\left(\frac{\pi_{\theta^*}(a|x)}{\pi_{\theta_0}(a|x)}, M\right)\right] \\
&= \mathbb{E}_{x,\theta_0,y}\left[y^2\zeta\left(\frac{\pi_{\theta^*}(a|x)}{\pi_{\theta_0}(a|x)}, M\right)^2\right] - \mathbb{E}_{x,\theta_0,y}\left[y\zeta\left(\frac{\pi_{\theta^*}(a|x)}{\pi_{\theta_0}(a|x)}, M\right)\right]^2 \\
&\leq \mathbb{E}_{x,\theta_0,y}\left[\Big(\frac{\pi_{\theta^*}(a|x)}{\pi_{\theta_0}(a|x)}\Big)^2\right] - L^M(\pi^*)^2 = \nu_*^2 + 1 - L^M(\theta^*)^2 \leq \nu_*^2 + 1\,. \quad (2.21)
\end{aligned}$$

Then, by writing $X = \pi_{\theta^*}(a|x)/\pi_{\theta_0}(a|x)$, the bias may be upper-bounded as

$$\begin{aligned}
h_M(\theta^*) &\leq \mathbb{E}_{x,\theta_0,y}\left[X - \zeta(X, M)\right] \\
&\leq \mathbb{E}_{x,\theta_0,y}\left[(X - M)\mathbb{1}\{X > M\}\right] \\
&\leq \int_0^\infty \mathbb{P}_{x,\theta_0,y}\Big((X - M)\mathbb{1}\{X > M\} > t\Big)dt \\
&\leq \int_0^\infty \mathbb{P}_{x,\theta_0,y}\big(X - M > t\big)dt = \int_0^\infty \mathbb{P}_{x,\theta_0,y}\Big((X-1)^2 > (t + M - 1)^2\Big)dt \\
&\leq \int_0^\infty \frac{\mathbb{E}_{x,\theta_0,y}\left[(X-1)^2\right]}{(t + M - 1)^2}dt = \frac{\mathbb{E}_{x,\theta_0,y}\left[(X-1)^2\right]}{M - 1} = \frac{\nu_*^2}{M-1}\,. \quad (2.22)
\end{aligned}$$

Furthermore, if $W \leq M$ then $S = \zeta(W, M) = W \leq M$, else, using $\alpha(M) = M/\log(\alpha(M)) \leq \max\{M, e\} \leq M + e$,

$$S = \zeta(W, M) = \alpha(M)\log\big(W + \alpha(M) - M\big) \leq (M + e)\log(W + e)\,. \quad (2.23)$$

Therefore, substituting (2.21), (2.22), and (2.23) into (2.20), yields the following upper-bound on the excess risk

$$\begin{aligned}
&L(\hat{\theta}^{CRM}) - L(\theta^*) \\
&\leq \sqrt{\frac{32(1+\nu_*^2)\big(C_n(\Theta, M) + \log\frac{30}{\delta}\big)}{n}} + \frac{22(M+e)\log(W+e)\big(C_n(\Theta, M) + \log\frac{30}{\delta}\big)}{n-1} + \frac{\nu_*^2}{M-1}\,.
\end{aligned}$$

We now choose M such that

$$\frac{22(M-1)\log(W+e)\big(C_n(\Theta,M)+\log\frac{30}{\delta}\big)}{n-1}=\frac{\nu_*^2}{M-1} \tag{2.24}$$

which is possible since the left term grows from 0 to infinity and the right term decreases from infinity to 0 for $M>1$. Therefore, from the last two terms we eventually have

$$L(\hat{\theta}^{CRM})-L(\theta^*)\lesssim\sqrt{\frac{(1+\nu_*^2)\log(W+e)\big(C_n(\Theta,M)+\log\frac{1}{\delta}\big)}{n}}+\frac{\log(W+e)(C_n(\Theta,M)+\log\frac{1}{\delta})}{n}\,, \tag{2.25}$$

where $\lesssim$ hides universal multiplicative constants. This concludes the first part of the proof.

Step 2: *Evaluating the policy class complexity* $C_n\big(\Theta^{CLP},M\big)$.
In this part, we provide a bound on the metric entropy $C_n(\Theta^{\text{CLP}},M)=\log\mathcal{H}_\infty(1/n,\mathcal{F}_{\Pi_{\Theta^{\text{CLP}}}},2n)$. We recall that $\mathcal{F}_{\Pi_{\Theta^{\text{CLP}}}}$ is defined in (2.12) and $\Pi_{\Theta^{\text{CLP}}}$ is described in Section 2.3.1. More precisely, let $\mathcal{Z}\subseteq\mathcal{A}$ be a Nyström dictionary of size $m\geq 1$ and $\gamma>0$. Since we use Gaussian distributions, we have

$$\Pi_{\Theta^{\text{CLP}}}=\big\{\pi_\beta \text{ s.t. for any } x\in\mathcal{X},\ \pi_\beta(\cdot|x)=\mathcal{N}(\mu_\beta^{\text{CLP}}(x),\sigma^2),\ \text{with } \beta\in\Theta_\beta\big\},$$

where

$$\Theta_\beta=\big\{\beta\in\mathbb{R}^m,\ \text{s.t}\ \|\beta\|\leq C\big\}$$

where

$$\mu_\beta^{\text{CLP}}(x)=\sum_{a\in\mathcal{Z}}\frac{\exp(-\gamma\eta_\beta(x,a))}{\sum_{a'\in\mathcal{Z}}\exp(-\gamma\eta_\beta(x,a'))}\quad\text{and}\quad\eta_\beta(x,a)=\langle\beta,\psi(x,a)\rangle\,,$$

for some embedding ψ described in Section 2.3.1 which satisfies $\|\psi(x,a)\|\leq\upsilon$ for any (x,a). Fix $x\in\mathcal{X}$. Let us show that $\beta\mapsto\mu_\beta^{\text{CLP}}(x)$ is Lipschitz. Denote by $Z_\beta(x)=\sum_{a\in\mathcal{Z}}\exp(-\gamma\eta_\beta(x,a))$ the normalization factor. We consider the gradient of $\mu_\beta^{\text{CLP}}(x)$ with regards to β

$$\frac{\partial\mu_\beta^{\text{CLP}}}{\partial\beta}(x)=\sum_{a\in\mathcal{Z}}a\bigg(\frac{\psi(x,a)\exp\left(\langle\beta,\psi(x,a)\rangle\right)}{Z_\beta(x)}\\ -\frac{\exp\left(\langle\beta,\psi(x,a)\rangle\right)\sum_{a\in\mathcal{Z}}\psi(x,a)\exp\left(\langle\beta,\psi(x,a)\rangle\right)}{Z_\beta(x)^2}\bigg)\,.$$

Taking the norm, and upper-bounding $\|\psi(x,a)\|\leq\upsilon$ and $\|a\|\leq\alpha_{\mathcal{Z}}$, this yields

$$\Big\|\frac{\partial\mu_\beta^{\text{CLP}}}{\partial\beta}(x)\Big\|\leq 2\upsilon\alpha_{\mathcal{Z}}\,.$$

Therefore, $\beta\mapsto\mu_\beta^{\text{CLP}}(x)$ is $2\upsilon\alpha_{\mathcal{Z}}$-Lipschitz, which implies that

$$\beta\mapsto\pi_\beta(a|x)=\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left(\frac{a-\mu_\beta^{\text{CLP}}(x)}{\sigma}\right)^2\right)$$

are also Lipschitz with parameter

$$\sqrt{\frac{2}{e\pi}}\frac{\upsilon\alpha_{\mathcal{Z}}}{\sigma^2}\,. \tag{2.26}$$

We recall that the metric entropy $C_n(\Theta^{\text{CLP}}) = \log \mathcal{H}_\infty(1/n, \mathcal{F}_{\Theta^{\text{CLP}}}, 2n)$ is applied to the function class

$$\mathcal{F}_{\Pi_{\Theta^{\text{CLP}}}} = \left\{ f_\beta : (x,a,y) \mapsto 1 + \frac{y}{S}\zeta\left(\frac{\pi_\beta(a|x)}{\pi_{\theta_0}(a|x)}, M\right) \quad \text{for some } \beta \in \Theta^{\text{CLP}} \right\}.$$

By assumption, the inverse of the logging policy weights are bounded $\pi_{\theta_0}(a|x)^{-1} \leq M_0$ for any $(x,a) \in \mathcal{X} \times \mathcal{A}$ (as in Kallus and Zhou (2018)). Therefore, together with (2.26), for any $(x,a,y) \in \mathcal{X} \times \mathcal{A} \times \mathcal{Y}$, the function $\beta \mapsto f_\beta(x,a,y)$ is Lipschitz with parameter

$$\sqrt{\frac{2}{e\pi}}\frac{\upsilon\alpha_{\mathcal{Z}} M_0}{S\sigma^2}\,. \tag{2.27}$$

Let $\varepsilon > 0$. Because there exists an ε-covering of the ball $\{\beta \in \mathbb{R}^d : \|\beta\| \leq C\}$ of size $(C/\varepsilon)^d$, together with (2.27), the latter provides a covering of $\mathcal{F}_{\Theta^{\text{CLP}}}$ in sup-norm with parameter

$$\varepsilon\sqrt{\frac{2}{e\pi}}\frac{\upsilon\alpha_{\mathcal{Z}} M_0}{S\sigma^2}\,.$$

Equalizing this with n^{-1} and taking the log of the size of the covering entails

$$C_n(\Theta^{\text{CLP}}, M) \leq d \log\left(\sqrt{\frac{2}{e\pi}}\frac{CM_0\upsilon\alpha_{\mathcal{Z}} n}{S\sigma^2}\right).$$

Now, we recall that d is the dimension of the embedding ψ, which we model as

$$\psi(x,a) = \psi_{\mathcal{X}}(x) \otimes \psi_{\mathcal{A}}(a)$$

where $\psi_{\mathcal{A}}(a) \in \mathbb{R}^m$ is the embedding obtained by using the Nyström dictionary of size m on the action space and $\psi_{\mathcal{X}}(x) \in \mathbb{R}^{d_{\mathcal{X}}}$ is the embedding of the context space $\mathcal{X} \subseteq \mathbb{R}^{d_x}$. Typically $d_{\mathcal{X}} = d_x^2 + d_x + 1$ or $d_{\mathcal{X}} = d_x$ respectively with the polynomial and linear maps considered in practice. Thus, $d = md_{\mathcal{X}}$. Substituting the latter into the complexity upper-bound and using $1 \leq S$ and M, we finally get

$$C_n(\Theta^{\text{CLP}}, M) \leq md_{\mathcal{X}} \log\left(\sqrt{\frac{2}{e\pi}}\frac{CM_0\upsilon\alpha_{\mathcal{Z}} n}{\sigma^2}\right), \tag{2.28}$$

where we recall that $d_{\mathcal{X}}$ is the dimension of the contextual feature map, C a bound on the parameter norm β, M_0 a bound on $\pi_{\theta_0}(a|x)^{-1}$, υ^2 a bound on the kernel, σ^2 the variance of the policies, and $\alpha_{\mathcal{Z}}$ a bound on the action norms $\|a\|$. □

2.13.2. Discussion: on the Rate Obtained for Deterministic Classes

Consider the deterministic CLP class that assigns any input x to the action $\mu_\beta^{\text{CLP}}(x)$ defined in (2.2). Although the chapter focuses on stochastic policies, this appendix provides an excess-risk upper-bound with respect to this deterministic class.

The latter corresponds to the choice $\sigma = 0$ in the CLP policy set defined in Section 2.3.1 and therefore, Proposition 2.5.1 cannot be applied directly. Fix some $\sigma^2 > 0$ to be optimized later. For any $\beta \in \mathbb{R}^m$, we denote by $L(\mu_\beta^{\text{CLP}})$ the risk associated with the deterministic policy $a = \mu_\beta^{\text{CLP}}(x)$. We also define $\pi_\beta^{\text{CLP}}(\cdot|x) \sim \mathcal{N}(\mu_\beta^{\text{CLP}}(x), \sigma^2)$ and note $R(\pi_\beta^{\text{CLP}})$ the expected risk of the policy π_β^{CLP} as in Introduction 1.4. Then, let $\hat{\theta}^{\text{CRM}}$ be the counterfactual estimator obtained by Proposition 2.5.1 on the class $\Theta_{CLP} = \{\pi_\beta^{\text{CLP}}(\cdot|x)\}$, with probability $1 - \delta$

$$\begin{aligned}\hat{L}(\hat{\theta}^{CRM}) - L(\mu_\beta^{\text{CLP}}) &\leq \hat{L}(\hat{\theta}^{CRM}) - R(\pi_\beta^{\text{CLP}}) + R(\pi_\beta^{\text{CLP}}) - L(\mu_\beta^{\text{CLP}}) \\ &\leq \hat{L}(\hat{\theta}^{CRM}) - R(\pi_\beta^{\text{CLP}}) + L_0\sigma\sqrt{2\log\frac{2}{\delta}}\,,\end{aligned}$$

where we assumed that the risk is L_0-Lipschtitz and used that $P\big(|X| < \sigma\sqrt{2\log(2/\delta)}\big) \leq \delta$ for $X \sim \mathcal{N}(0, \sigma^2)$. From Proposition 2.5.1, this yields, with probability $1 - 2\delta$

$$\hat{}L(\hat{\theta}^{CRM}) - L(\mu_\beta^{\text{CLP}}) \lesssim \sqrt{\frac{(1+\sigma_*^2)\log(W+e)\big(C_n(\Theta^{\text{CLP}}, M) + \log\frac{1}{\delta}\big)}{n}} + L_0\sigma\sqrt{2\log\frac{2}{\delta}}\,.$$

Now, note that $C_n(\Pi, M)$ and $\log(W)$ only yield logarithmic dependence on σ^2 and n and will thus not impact the rate of convergence. The variance σ^* has a stronger dependence on σ^2 but can be upper-bounded as follows

$$\begin{aligned}\sigma_*^2 = \text{Var}_{\pi_{\theta_0}}\left[\frac{\pi_\beta^{\text{CLP}}(a|x)}{\pi_{\theta_0}(a|x)}\right] &= \int\left(\frac{\pi_\beta^{\text{CLP}}(a|x) - \pi_{\theta_0}(a|x)}{\pi_{\theta_0}(a|x)}\right)^2 \pi_{\theta_0}(a|x)da \\ &\leq \int\frac{\pi_\beta^{\text{CLP}}(a|x)^2}{\pi_{\theta_0}(a|x)}da \leq \frac{1}{M_0}\int\pi_\beta^{\text{CLP}}(a|x)^2da = \frac{1}{2\sigma M_0\sqrt{\pi}}\,,\end{aligned}$$

where the last equality is because $\pi_\beta^{\text{CLP}}(\cdot|x)$ is a Gaussian distribution with variance σ^2. Therefore, keeping only the dependence on σ and n and neglecting log-factors, we get the high-probability upper-bound

$$\hat{L}(\hat{\theta}^{CRM}) - L(\mu_\beta^{\text{CLP}}) \leq \tilde{O}\Big(\frac{1}{\sqrt{\sigma n}} + \sigma\Big)\,.$$

The choice $\sigma = n^{-1/3}$ entails a rate of order $\mathcal{O}(n^{-1/3})$.

2.14. Details on the Experiment Setup and Reproducibility

In this section we give additional details on synthetic and semi-synthetic datasets, we provide details on the evaluation methodology and information for experiment reproducibility.

2.14.1. Synthetic and Semi-Synthetic setups

Synthetic setups As many real-world applications feature a reward function that increases first with the action, then plateaus and finally drops, we have chosen a piecewise linear function as shown in Fig. 2.18 that mimics reward buckets over the CoCoA dataset presented in Section 2.6.

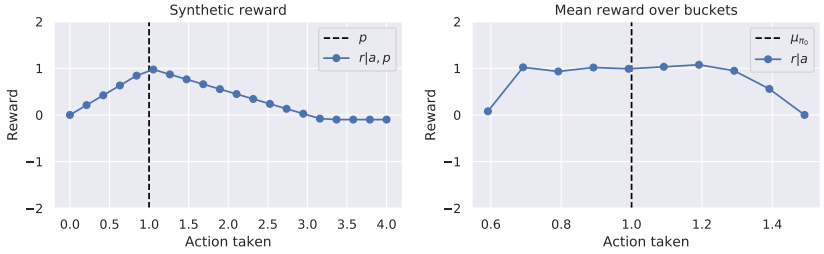

Figure 2.18: Synthetic reward engineering. The synthetic reward (left) is inspired from real-dataset reward buckets (right).

2.14.2. Reproducibility

We provide code for reproducibility and all experiments were run on a CPU cluster, each node consisting on 24 CPU cores (2 x Intel(R) Xeon(R) Gold 6146 CPU@ 3.20GHz), with 500GB of RAM.

Policy parametrization. In our experiments, we consider two forms of parametrizations: (i) a lognormal distribution with $\theta = (\theta_\mu, \sigma), \pi_{(\mu,\sigma)} = \log \mathcal{N}(m, s)$ with $s = \sqrt{\log\left(\sigma^2/\mu^2 + 1\right)}; m = \log(\mu) - s^2/2$, so that $\mathbb{E}_{a\sim\pi_{(\mu,\sigma)}}[a] = \mu$ and $\text{Var}_{a\sim\pi_{(\mu,\sigma)}}[a] = \sigma^2$; (ii) a normal distribution $\pi_{(\mu,\sigma)} = \mathcal{N}(\mu, \sigma)$. In both cases, the mean μ may depend on the context (see Section 2.4), while the standard deviation σ is a learned constant. We add a positivity constraint for σ and add an entropy regularization term to the objective in order to encourage exploratory policies and avoid degenerate solutions.

Models. For parametrized distributions, we experimented both with normal and lognormal distributions on all datasets, and different baseline parameterizations including constant, linear and quadratic feature maps. We also performed some of our experiments on low-dimensional datasets with a stratified piece-wise contextual parameterization, which partitions the space by bucketizing each feature by taking K (for e.g $K = 4$) quantiles, and taking the cross product of these partitions for each feature. However this baseline is not scalable for higher dimensional datasets such as the Warfarin dataset.

Hyperparameters. In Table 2.7 we show the hyperparameters considered to run the experiments to reproduce all the results. Note that the grid of hyperparameters is larger for synthetic data. For our experiments involving anchor points, we validated the number of anchor points and kernel bandwidths similarly to other hyperparameters.

2.15. Additional Results and Additional Evaluation Metrics

In this section we provided additional results on both contextual modeling and optimiza-

Table 2.7: Table of hyperparameters for the Synthetic and CoCoA datasets

	Synthetic	Warfarin	CoCoA
Variance reg. λ	$\{0., 0.001, 0.01, 0.1, 1, 10, 100\}$	$\{0.00010.0010.010.1\}$	$\{0., 0.001, 0.1\}$
Clipping M	$\{1, 1.7, 2.8, 4.6, 7.7, 12.9, 21.5, 35.9, 59.9, 100.0\}$	$\{1, 2.1, 4.5, 9.5, 20\}$	$\{1, 2.1, 4.5, 9.5, 10, 20, 100\}$
Prox. κ	$\{0.001, 0.01, 0.1, 1\}$	$\{0.001, 0.01, 0.1\}$	$\{0.001, 0.01, 0.1\}$
Reg. param. C	$\{0.00001, 0.0001, 0.001, 0.01, 0.1\}$	$\{0.00001, 0.0001, 0.001, 0.01, 0.1\}$	$\{0.00001, 0.0001, 0.001, 0.01, 0.1\}$
Number of anchor points	$\{2, 3, 5, 7, 10\}$	$\{5, 7, 10, 12, 15, 20\}$	$\{2, 3, 5\}$
Softmax γ	$\{1, 10, 100\}$	$\{1, 5, 10\}$	$\{0.1, 0.5, 1, 5\}$

tion driven approaches of CRM.

2.15.1. Continuous vs Discrete strategies in Continuous-Action Space

We provide in Figure 2.19 additional plots for the continuous vs discrete strategies for the synthetic setups described in Section 2.7.2.

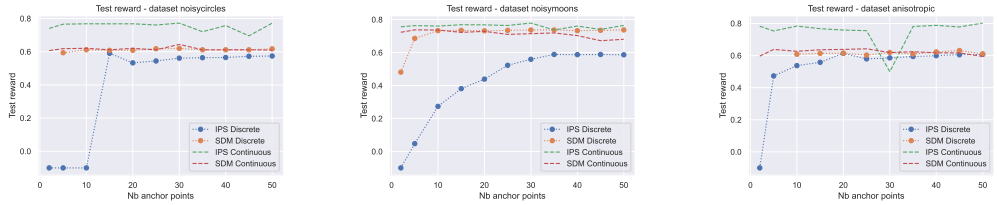

Figure 2.19: Continuous vs discrete. Test rewards for CLP and (stochastic) direct method (DM) with Nyström parameterization, versus a discrete approach, with varying numbers of anchor points. We add a minimal amount of noise to the deterministic DM in order to pass the n_{eff} validation criterion.

2.15.2. Optimization Driven Approaches of CRM

In this part we provide additional results on optimization driven approaches of CRM for the Noisycircles, Anisotropic, Warfarin and CoCoAdatasets.

Both Noisycircles and Anisotropic datasets in Figure 2.20 show the improvements in test reward and in training objective of our optimization-driven strategies, namely the soft-clipping estimator and the use of the proximal point algorithm. Overall we see that for most configurations, the proximal point method better optimizes the objective function and provides better test performances, while the soft-clipping estimator performs better than its hard-clipping variant, which may be attributed to the better optimization properties. For semi-synthetic Warfarin and real-world CoCoA datasets in Figure 2.20 we also show the improvements in test reward and in training objective of our optimization-driven strategies. More particularly we demonstrate the effectiveness of proximal point methods on the Warfarin dataset where most proximal configurations perform better than the base algorithm. Moreover, soft-clipping strategies perform better than its hard-clipping variant on real-world dataset with outliers and noises, which demonstrate the effectiveness of this smooth estimator for real-world setups.

2.15.3. Doubly Robust Estimators

In this section we detail the discussion on doubly robust estimators and the difficulties that exist to obtain a suitable estimator. In policy based methods for discrete actions, the DR

estimator takes the form

$$\hat{L}^{\mathrm{DR}}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{\eta}(x_i, a_i)\right)\frac{\pi_\theta(a_i|x_i)}{\pi_{0,i}} + \frac{1}{n}\sum_{i=1}^{n}\sum_{a\in\mathcal{A}}\hat{\eta}(x_i, a)\pi_\theta(a|x_i),$$

Usually, the DR estimator should only improve on the vanilla IPS estimator thanks to the lower variance induced by the outcome model $\hat{\eta}$. However, in a continuous-action setting with stochastic policies, the second term becomes $\mathbb{E}_{x\sim\mathcal{P}_\mathcal{X}, a\sim\pi(\cdot|x)}\left[\hat{\eta}(x, a)\right]$, which is intractable to optimize in closed form since it involves integrating over actions according to $\pi(\cdot|x)$. Thus, handling this term requires approximations (as described hereafter), which may overall lead to poorer performance compared to an IPW estimator that sidesteps the need for such a term.

The difficulty for stochastic policies with continuous actions is to derive an estimator of the term $\mathbb{E}_{x\sim\mathcal{P}_\mathcal{X}, a\sim\pi(\cdot|x)}\left[\hat{\eta}(x, a)\right]$. Unlike stochastic policies with discrete actions which allow to use a discrete summation over the action set, we would need here to compute here an estimator of the form $\frac{1}{n}\sum_{i=1}^{n}\int_{a\in\mathcal{A}}\pi(a|x_i)\hat{\eta}(x_i, a)$. We note that in the case of deterministic policy π learning this direct method term would easily boil down to $\frac{1}{n}\sum_{i=1}^{n}\hat{\eta}(x_i, \pi(x_i))$, and the DR estimator would be built with smoothing strategies for the IPW term as in (Kallus and Zhou, 2018).

In our experiments for stochastic policies with one dimensional actions $\mathcal{A}\subset\mathbb{R}$, we tried to approximate the direct method term $\frac{1}{n}\sum_{i=1}^{n}\int_{a\in\mathcal{A}}\pi(a|x_i)\hat{\eta}(x_i, a)$ with a finite sum of CDFs differences over the m anchor points $a_1, \dots a_m$ by computing :

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}\int_{a=a_j}^{a_{j+1}}\pi(a|x_i)\hat{\eta}(x_i, a_j)$$

We present a table below of some of the experiments we ran on the synthetic datasets we proposed, along with an evaluation of the baselines that exist in the litterature for discrete actions. We see overall that our model improves indeed upon the logging policy, but does not compare to the performances of the scIPS and SNIPS estimators.

	Noisymoons	Noisycircles	Anisotropic
Logging policy π_{θ_0}	0.5301	0.5301	0.4533
Doubly Robust (discrete)	0.5756 ± 0.0022	0.5500 ± 0.0024	0.5593 ± 0.0026
SWITCH (Wang et al., 2017) (discrete)	0.5786 ± 0.0025	0.5520 ± 0.0026	0.5741 ± 0.0024
CAB-DR (Su et al., 2019) (discrete)	0.5683 ± 0.0023	0.5326 ± 0.0025	0.5361 ± 0.0028
Doubly Robust (ours)	**0.6115** ± 0.0001	**0.6113** ± 0.0002	**0.5977** ± 0.0001

Table 2.8: Comparison of doubly robust estimators, discretized strategies and our model which approximates the direct method term.

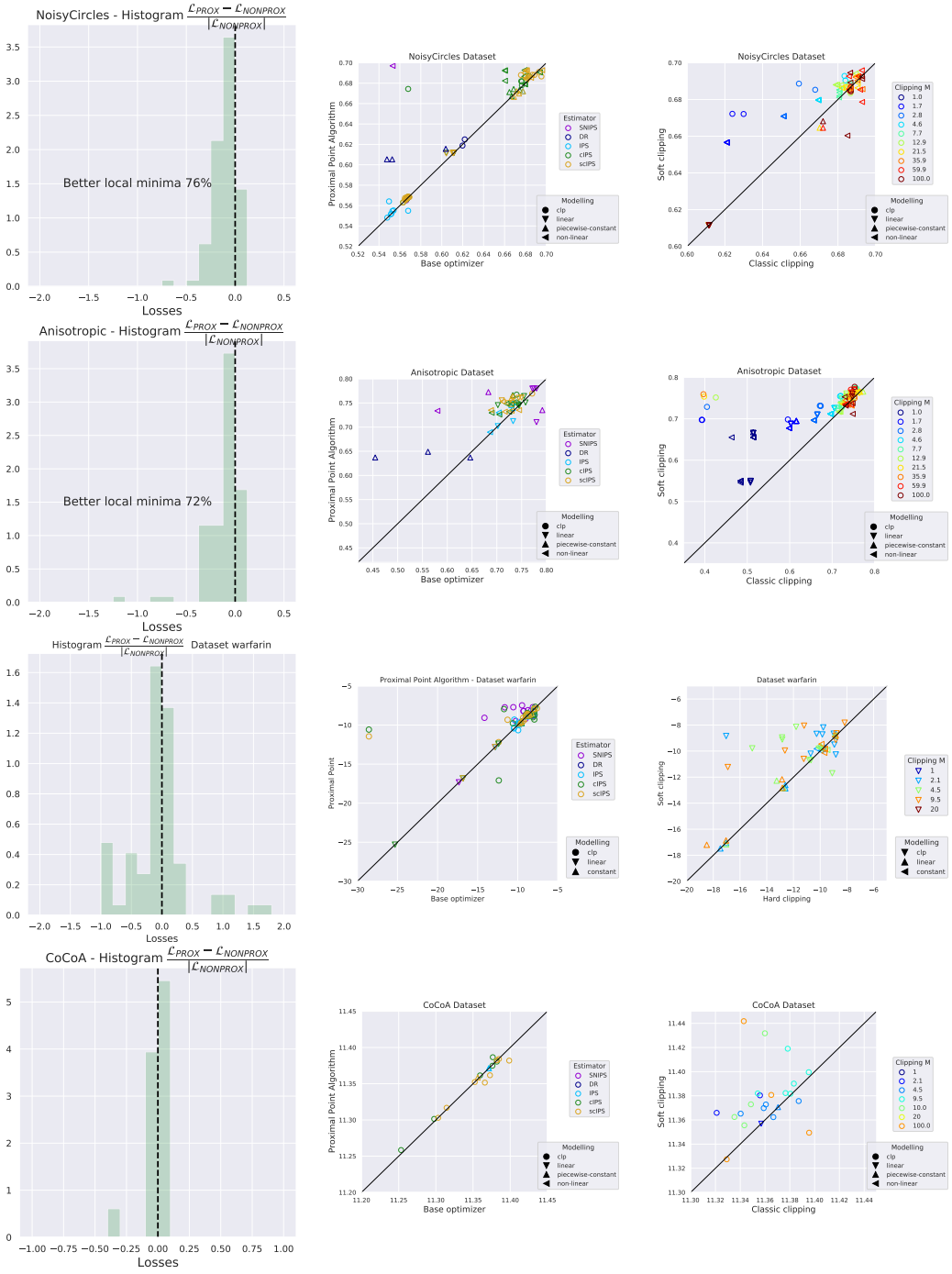

Figure 2.20: Optimization-driven approaches (NoisyCircles, Anisotropic, Warfarin and CoCoA datasets). Relative improvements in the training objective from using the proximal point method (left), comparison of test rewards for proximal point vs the simpler gradient-based method (center), and for soft- vs hard-clipping (right).

3

Sequential Counterfactual Risk Minimization

Counterfactual Risk Minimization (CRM) is a framework for dealing with the logged bandit feedback problem, where the goal is to improve a logging policy using offline data. In this chapter, we explore the case where it is possible to deploy learned policies multiple times and acquire new data. We extend the CRM principle and its theory to this scenario, which we call "Sequential Counterfactual Risk Minimization (SCRM)." We introduce a novel counterfactual estimator and identify conditions that can improve the performance of CRM in terms of excess risk and regret rates, by using an analysis similar to restart strategies in accelerated optimization methods. We also provide an empirical evaluation of our method in both discrete and continuous action settings, and demonstrate the benefits of multiple deployments of CRM.

This chapter is based on the following material:

H. Zenati, E. Diemert, M. Martin, J. Mairal, and P. Gaillard. Sequential counterfactual risk minimization. *International Conference on Machine Learning (ICML)*, 2023

3.1. Introduction

Counterfactual reasoning in the logged bandit problem has become a common task for practitioners in a wide range of applications such as recommender systems (Swaminathan and Joachims, 2015a), ad placements (Bottou et al., 2013) or precision medicine (Kallus and Zhou, 2018). Such a task typically consists in learning an optimal decision policy from logged contextual features and partial feedbacks induced by predictions from a logging policy. To do so, the logged data is originally obtained from a randomized data collection experiment. However, the success of counterfactual risk minimization is highly dependent on the quality of the logging policy and its ability to sample meaningful actions.

Counterfactual reasoning can be challenging due to large variance issues associated with counterfactual estimators (Swaminathan and Joachims, 2015b). Additionally, as pointed out by Bottou et al. (2013), confidence intervals obtained from counterfactual estimates may not be sufficiently accurate to select a final policy from offline data (Dai et al., 2020). This can occur when the logging policy does not sufficiently explore the action space. To address this, one option is to simply collect additional data from the same logging system to increase the sample size. However, it may be more efficient to use already collected data to design a better data collection experiment through a sequential design approach (Bottou et al., 2013, see Section 6.4). It is thus appealing to consider successive policy deployments when possible.

We tackle this sequential design problem and are interested in multiple deployments of the CRM setup of Swaminathan and Joachims (2015a), which we call sequential counterfactual risk minimization (SCRM). SCRM performs a sequence of data collection experiments by determining at each round a policy using data samples collected during previous experiments. The obtained policy is then deployed for the next round to collect additional samples. Such a sequential decision making system thus entails designing an adaptive learning strategy that minimizes the excess risk and expected regret of the learner. In contrast to the conservative learning strategy in CRM, the exploration induced by sequential deployments of enhanced logging policies should allow for improved excess risk and regret guarantees. Yet, obtaining such guarantees is nontrivial and we address it in this work.

In order to accomplish this, we first propose a new counterfactual estimator that controls the variance and analyze its convergence guarantees. Specifically, we obtain an improved dependence on the variance of importance weights between the optimal and logging policy. Second, leveraging this estimator and a weak assumption on the concentration of this variance term, we show how the error bound sequentially concentrates through CRM rollouts. This allows us to improve the excess risk bounds convergence rate as well as the regret rate. Our analysis employs methods similar to restart strategies in acceleration methods (Nesterov, 2012) and optimization for strongly convex functions (Boyd and Vandenberghe, 2004). We also conduct numerical experiments to demonstrate the effectiveness of our method in both discrete and continuous action settings, and how it improves upon CRM and other existing methods in the literature.

3.2. Related Work

Counterfactual learning from logged feedback (Bottou et al., 2013) uses only past interactions to learn a policy without interacting with the environment. Counterfactual risk minimization methods (Swaminathan and Joachims, 2015a,b) propose learning formulations using a variance penalization as in (Maurer and Pontil, 2009) to find policies with minimal variance. Even so, counterfactual methods remain prone to large variance issues (Dudík et al., 2014). These problems may arise when the logging policy under-explores the action space, making it difficult to use importance sampling techniques (Owen, 2013) that are key to counterfactual reasoning. While one could collect additional data to counter this problem, our method focuses on sequential deployments (Bottou et al., 2013, see Section 6.4) to collect data obtained from adaptive policies to explore the action space. Note also that the original motivation is related but different from the support deficiency problem (Sachdeva et al., 2020) where the support of the logging policy does not cover the support of the optimal policy.

Another related literature to our framework is batch bandit methods. Originally introduced by Perchet et al. (2015) and then extended by Gao et al. (2019) in the multi-arm setting, batch bandit agent take decisions and only observe feedback in batches. This therefore differs from the classic bandit setting (Auer et al., 2002; Audibert et al., 2007) where rewards are observed after each action taken by an agent. Extensions to the contextual case have been proposed by Han et al. (2020) and could easily be kernelized (Valko et al., 2013). The sequential counterfactual risk minimization problem is thus closely related to this setting. However, major differences can be noted. First, SCRM does not leverage any problem structure as in stochastic contextual bandits (Li et al., 2010) by assuming a linear reward function (Chu et al., 2011; Goldenshluger and Zeevi, 2013; Han et al., 2020) nor uses regression oracles as (Foster and Rakhlin, 2020; Simchi-Levi and Xu, 2022). Second, deterministic decision rules taken by bandit agents (Lattimore and Szepesvári, 2020) do not allow for counterfactual reasoning or causal inference (Peters et al., 2017), unlike our framework which performs sequential randomized data collection. Third, unlike gradient based methods used in counterfactual methods with parametric policies, batch bandit methods use zero-order methods to learn from data and necessitate approximations to be scalable (Calandriello et al., 2020; Zenati et al., 2022).

The sequential designs that we use are adaptive data collection experiments, which have been studied by Bakshy et al. (2018); Kasy and Sautmann (2021). Closely related to our method is policy learning from adaptive data that has been studied by Zhan et al. (2021) and Bibaut et al. (2021b) in the online setting. In contrast, we consider a batch setting and our analysis achieve fast rates in more general conditions. Zhan et al. (2021) use a doubly robust estimator and provide regret guarantees but assume a deterministic lower bound on the propensity score to control the variance. Instead, our novel counterfactual estimator does not require such an assumption. Bibaut et al. (2021b) propose a novel maximal inequality and derive thereof fast rate regret guarantees under an additional margin condition that can only hold for finite action sets. Our work instead uses a different assumption on the expected risk, which is similar to Hölderian error bounds in acceleration methods (d'Aspremont et al., 2021) that are known to be satisfied for a broad class of subanalytic functions (Bolte et al., 2007).

In the reinforcement learning literature (Sutton and Barto, 1998), off-policy methods (Harutyunyan et al., 2016; Munos et al., 2016) evaluate and learn a policy using actions sampled from a behavior (logging) policy, which is therefore closely related to our setting. Among methods that have shown to be empirically successful are the PPO (Schulman et al., 2017) and TRPO (Schulman et al., 2015) algorithms which learn policies using a Kullback-Leibler distributional constraint to ensure robust learning, which can be compared to our learning strategy that improves the logging policy at each round. However reinforcement learning models transitions in the states (contexts) induced by the agent's actions while bandit problems like ours assume that actions do not influence the context distribution. This enables to design algorithms that exploit the problem structure, have theoretical guarantees and can achieve better performance in practice.

Finally, our method is related to acceleration methods (d'Aspremont et al., 2021) where current iterates are used as new initial points in the optimization of strongly convex functions (Boyd and Vandenberghe, 2004). While different schemes use fixed (Powell, 1977) or adaptive (Nocedal and Wright, 2006; Becker et al., 2011; Nesterov, 2012; Bolte et al., 2007; Gaillard and Wintenberger, 2018) strategies, our method differs in that it does not consider the same original setting, does not require the same assumptions nor provides the same guarantees. Eventually, while current models are also used as new starting points, additional data is effectively collected in our setting unlike those previous works that do not assume partial feedbacks as in our case.

3.3. Sequential Designs

In this section, we introduce the (CRM) framework and motivate the use of sequential designs for (SCRM).

In this section we present a design of data collections that sequentially learn a policy from logged data in order to deploy it and learn from the newly collected data. Specifically, we assume that at a round $m \in \{1, \dots M\}$, a model $\theta_m \in \Theta$ is deployed and a set s_m of n_m observations $s_m = (x_{m,i}, a_{m,i}, y_{m,i}, \pi_{m,i})_{i=1,\dots,n_m}$ is collected thereof, with propensities $\pi_{m,i} = \pi_{\theta_m}(a_{m,i}|x_{m,i})$ to learn a new model θ_{m+1} and reiterate. In this work, we assume that the loss y is bounded in $[-1, 0]$ as in (Swaminathan and Joachims, 2015a) (note however that this assumption could be relaxed to bounded losses) and follows a fixed distribution $\mathcal{P}_{\mathcal{Y}}$. Next, we will introduce useful definitions.

Definition 3.3.1 (Excess Risk and Expected Regret)**.** *Given an optimal model $\theta^* \in \arg\min_{\theta\in\Theta} L(\theta)$, we write for each rollout m the excess risk:*

$$\Delta_m = L(\theta_m) - L(\theta^*), \tag{3.1}$$

and define the expected regret as:

$$R_n = \sum_{m=0}^{M} \Delta_m n_{m+1}. \tag{3.2}$$

The objective is now to find a sequence of models $\{\theta_m\}_{m=1\dots M}$ that have an excess risk and an expected regret R_n that improve upon CRM guarantees. To do so, we define a sequence of

minimization problems for $m \in \{1, \dots M\}$:

$$\hat{\theta}_{m+1} \in \arg\min_{\theta \in \Theta} \mathcal{L}_m(\theta), \tag{SCRM}$$

where $\mathcal{L}_m$ is an objective function that we define in Section 3.4.2. Note that in the setting we consider, samples are i.i.d inside a rollout m but dependencies exist between different sets of observations. From a causal inference perspective (Peters et al., 2017), this does not incur an additional bias because of the successive conditioning on past observations. We provide detailed explanations in Appendix 3.8 on this matter. Note also that the main intuition and motivation of our work is to shed light on how learning intermediate models θ_m to adaptively collect data can improve upon sampling from the same logging system by using the same total sample size $n = \sum_{i=0}^{m} n_m$. To illustrate the learning benefits of SCRM we now provide a simple example.

Example 3.3.1 (Gaussian policies with quadratic loss)**.** *Let us consider Gaussian parametrized policies $\pi_\theta = \mathcal{N}(\theta, \sigma^2)$ and a loss $l_t(a) = (a - y_t)^2 - 1$ where $y_t \sim \mathcal{N}(\theta^*, \sigma^2)$. We illustrate in Figure 3.1 the evolution of the losses of learned models θ_m through 15 rollouts with either i) Batch CRM learning on aggregation of data, being generated by the unique initial logging policy θ_0 or ii) Sequential CRM learning with models $\theta_0, \dots, \theta_{m-1}$ deployed adaptively, with data being generated by the last learned model θ_{m-1} for the batch m. We see that the models learned with SCRM take larger optimization steps than the ones with CRM.*

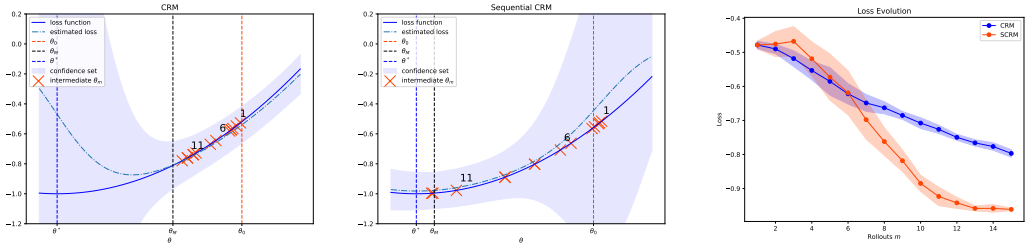

Figure 3.1: Comparison of CRM and SCRM on a simple setting described in Example 3.3.1. The models learned through CRM using re-deployments of θ_0 (left) reach θ^* slower than SCRM (center) that uses intermediate deployments $\theta_1, \dots, \theta_M$ indicated with 'x' markers and rollout numbers. The comparison of the evolution of averaged losses (right) over 10 random runs also shows SCRM converges faster. Here $\theta^* = 1$, $\sigma = 0.3$ and we take $M = 15$ total rollouts with batches m of size $n_m = 100 \times 2^m$. The parameter λ is set to its theoretical value.

We summarize our (SCRM) framework in Algorithm 6 with the different blocks exposed previously. We provide an additional graphical illustration of SCRM compared to CRM in Appendix 3.8. In the next section we will define counterfactual estimators from the observations s_m at each round and define a learning strategy $\mathcal{L}_m$.

3.4. Variance-Dependent Convergence Guarantees

In this part we aim at providing convergence guarantees of counterfactual learning. We show how we can obtain a dependency of the excess risk on the variance of importance weights between the logging model and the optimal model.

Algorithm 6: Sequential Counterfactual Risk Minimization

Input: Logged observations $(x_{0,i}, a_{0,i}, y_{0,i}, \pi_{0,i})_{i=1,\dots,n_0}$, parameter $\lambda > 0$
for $m = 1$ *to* M **do**
  Build $\mathcal{L}_m$ from observations s_m using Eq. (3.5)
  Learn θ_{m+1} using Eq. (SCRM)
  Deploy the model θ_{m+1} and collect observations
  $s_{m+1} = (x_{m+1,i}, a_{m+1,i}, l_{m+1,i}, \pi_{m+1,i})_{i=1,\dots,n_{m+1}}$
end

3.4.1. Implicit exploration and controlled variance

We first introduce a new counterfactual estimator. For this, we will require a common support assumption as in importance sampling methods (Owen, 2013). We will assume that the policies π_θ for $\theta \in \Theta$ have all the same support. We then consider the following estimator of the risk of a model θ:

$$\hat{L}_m^{\text{IPS-IX}}(\theta) = \frac{1}{n_m} \sum_{i=1}^{n_m} \frac{\pi_{\theta,i}}{\pi_{m,i} + \alpha\pi_{\theta,i}} y_{m,i}, \tag{3.3}$$

where $\pi_{\theta,i} = \pi_\theta(a_{m,i}|x_{m,i})$ and α is like a clipping parameter which ensures that the modified propensities $\pi_{m,i} + \alpha\pi_\theta(a_{m,i}|x_{m,i})$ are lower bounded. Noting $\zeta_i(\theta) = \big(\frac{\pi_{\theta,i}}{\pi_{m,i}+\alpha\pi_{\theta,i}} - 1\big) y_{m,i}$, $\bar{\zeta}(\theta) = \frac{1}{n_m}\sum_{i=1}^{n_m} \zeta_i(\theta)$ we can write the empirical variance estimator as:

$$\hat{V}_m^{\text{IPS-IX}}(\theta) = \frac{1}{n_m - 1} \sum_{i=1}^{n_m} (\zeta_i(\theta) - \bar{\zeta}(\theta))^2. \tag{3.4}$$

Here, the empirical variance uses a control variate since it uses the expression of $\zeta_i(\theta)$ above instead of $y_{m,i}\frac{\pi_{\theta,i}}{\pi_{m,i}+\alpha\pi_{\theta,i}}$. This allows to improve the depency on the variance in the excess risk provided in Proposition 3.4.2. Note also that our estimator resembles the implicit exploration estimator in the EXP3-IX algorithm (Lattimore and Szepesvári, 2020), as our motivation is to improve the control of the variance.

3.4.2. Learning strategy

Next, we aim in this part to provide a learning objective strategy $\mathcal{L}_m$, as refered to in Eq. (SCRM). Our approach, like the (CRM) framework, uses the sample variance penalization principle (Maurer and Pontil, 2009) to learn models that have low expected risk with high probability. To do so, we first provide an assumption to be used in our generalization error bound.

Assumption 3.4.1 (Bounded importance weights)**.** *For any models $\theta, \theta' \in \Theta$ and any $(x, a) \in \mathcal{X} \times \mathcal{A}$, we assume $\pi_\theta(a|x)/\pi_{\theta'}(a|x) \leq W$, for some $W > 0$.*

This assumption has been made in previous works (Kallus and Zhou, 2018; Zenati et al., 2020a) and is reasonable when we consider a bounded parameter space Θ. Next, we state an error bound for our estimator.

Proposition 3.4.1 (Generalization Error Bound). *Let $\hat{L}_m^{\text{IPS-IX}}$ and $\hat{V}_m^{\text{IPS-IX}}$ be the empirical estimators defined respectively in Eq.* (3.3) *and Eq.* (3.4). *Let $\theta \in \Theta$, $\delta \in (0,1)$, and $n_m \geq 2$. Then, under Ass. 3.4.1, for $\lambda_m = \sqrt{18(C_m(\Theta) + \log(2/\delta))}$, with probability at least $1-\delta$:*

$$L(\theta) \leq \hat{L}_m^{\text{IPS-IX}}(\theta) + \lambda_m \sqrt{\frac{\hat{V}_m^{\text{IPS-IX}}(\theta)}{n_m}} + \frac{2\lambda_m^2 W}{n_m} + \delta_m,$$

where $C_m(\Theta)$ is a metric entropy complexity measure defined in App. 3.8 and $\delta_m = \sqrt{\log(2/\delta)/(2n_m)}$.

This Proposition is proved in Appendix 3.8 and essentially uses empirical bounds (Maurer and Pontil, 2009). By minimizing the latter high-probability upper bound, we can find models θ with guarantees of minimizing the expected risk. Therefore, at each round, we minimize the following loss:

$$\mathcal{L}_m(\theta) = \hat{L}_m^{\text{IPS-IX}}(\theta) + \lambda_m \sqrt{\frac{\hat{V}_m^{\text{IPS-IX}}(\theta)}{n_m}}, \tag{3.5}$$

where $\lambda_m > 0$ is a positive parameter. Unlike deterministic decision rules used for example in UCB-based algorithms (Lattimore and Szepesvári, 2020), the exploration is naturally guaranteed by the stochasticity of the policies we use.

3.4.3. Excess risk upper bound

Eventually, we establish an upper bound on the excess risk of the IPS-IX estimator for counterfactual risk minimization using the learning strategy that we just defined. For this, we require an assumption on the complexity measure.

Assumption 3.4.2. *We assume that the set Θ is compact and that there exists $d > 0$ such that $C_m(\Theta) \leq d \log(n_m)$.*

This assumption states that the complexity grows logarithmically with the sample size. It holds for parametric policies so long as the propensities are lower bounded, which is verified using our estimator. We now state our variance-dependent excess risk bound.

Proposition 3.4.2 (Conservative Excess Risk). *Let $n_m \geq 1$ and $\theta_m \in \Theta$. Let s_m be a set of n_m samples collected with policy π_{θ_m}. Then, under Assumptions 3.4.1 and 3.4.2, a minimizer θ_{m+1} of Eq.* (3.5) *on the samples s_m satisfies the excess risk upper-bound: w.p. $1-\delta$*

$$\begin{aligned}\Delta_{m+1} = \; & L(\theta_{m+1}) - L(\theta^*) \\ \lesssim \; & \sqrt{\nu_m^2 \frac{d\log n_m - \log\delta}{n_m}} + \frac{W^2 + W(d\log n_m - \log\delta)}{n_m},\end{aligned}$$

where $\nu_m^2 = \operatorname{Var}_{x,\theta_m}\left(\dfrac{\pi_{\theta^*}(a|x)}{\pi_{\theta_m}(a|x)}\right)$.

The proof is postponed to Appendix 3.8. The modified propensities in IPS-IX as well as the control variate used in the variance estimator allow us to improve the dependency in ν_m^2, compared to $\nu_m^2 + 1$ obtained in previous work (Zenati et al., 2020a). This turns out to be a crucial point to use these error bounds sequentially as in acceleration methods since $\nu_m \to 0$ if $\theta_m \to \theta^*$, as explained in the next section.

3.5. SCRM Analysis

In this section we provide the main theoretical result of this work on the excess risk and regret analysis of SCRM. We start by stating an assumption that is common in acceleration methods (d'Aspremont et al., 2021) with restart strategies (Becker et al., 2011; Nesterov, 2012) that we will require to achieve the benefits of sequential designs.

Assumption 3.5.1 (Hölderian Error Bound)**.** *We assume that there exist $\gamma > 0$ and $\beta > 0$ such that for any $\theta \in \Theta$, there exists $\theta^* \in \arg\min_{\theta\in\Theta} L(\theta)$ such that*

$$\gamma \mathrm{Var}_{x,\theta}\left(\frac{\pi_{\theta^*}(x|a)}{\pi_\theta(x|a)}\right) \leq (L(\theta) - L(\theta^*))^\beta \ .$$

Typically, in acceleration methods, Hölderian error bounds (Bolte et al., 2007) are of the form:

$$\gamma d(\theta, S^*_\Theta) \leq (L(\theta) - L(\theta^*))^\beta$$

for some $\gamma, \beta > 0$ and where $d(\theta, S^*_\Theta)$ is some distance to the optimal set ($S^*_\Theta = \arg\min_{\theta\in\Theta} L(\theta)$). This bound is akin to a local version of strong convexity ($\beta = 1$) or a bounded parameter space ($\beta = 0$) if d is the Euclidean distance. When $\beta \in [0, 1]$, this has also been referred to as the Łojasiewicz assumption introduced in (Łojasiewicz, 1963, 1993). Notably, it has been used in online learning (Gaillard and Wintenberger, 2018) to obtain fast rates with restart strategies. This assumption holds for instance for Example 3.3.1 with $\beta = 1$ (see App 3.8). We also discuss this assumption for distributions in the exponential family in Appendix 3.8 notably for distributions that have been used practice (Swaminathan and Joachims, 2015b; Kallus and Zhou, 2018; Zenati et al., 2020a). Next we state our main result that is the acceleration of the excess risk convergence rate and the regret upper bound of SCRM.

Proposition 3.5.1. *Let $n_0, n \geq 2$ and $\theta^* \in \arg\min_\theta L(\theta)$. Let $n_m = n_0 2^m$ for $m = 0, \ldots, M = \left\lfloor \log_2(1 + \frac{n}{n_0}) \right\rfloor$. Then, under Assumptions 3.4.1, 3.4.2 and 3.5.1 with $\beta > 0$, the SCRM procedure (Alg. 6) satisfies the excess risk upper-bound*

$$\Delta_M = L(\theta_M) - L(\theta^*) \leq O\Big(n^{-\frac{1}{2-\beta}} \log n\Big).$$

Moreover, the expected regret is bounded as follows:

$$R_n = \sum_{m=0}^{M} \Delta_m n_{m+1} \leq O\Big(n^{\frac{1-\beta}{2-\beta}} \log(n)^2\Big).$$

The proof of our result is detailed in Appendix 3.8.

Discussion This result illustrates that an excess risk of order $O(\frac{\log(n)}{n})$ may be obtained when $\beta = 1$ (which is implied by a local version of strong convexity assumption in acceleration methods). When $\beta = 0$, which merely accounts that the variance of importance weights are bounded, we simply recover the original rate of CRM of order $O(\log(n)/\sqrt{n})$. The SCRM procedures thus improves the excess risk rate whenever $\beta > 0$. It is worth to emphasize that the knowledge of β is not needed by Alg. 6.

3.6. Empirical Evaluation

In this section we perform numerical experiments to validate our method in practical settings. We present the experimental setup as well as experiments comparing SCRM to related approaches and internal details of the method.

3.6.1. Experimental setup

As our method is able to handle both discrete and continuous actions we experiment in both settings. We now provide a brief description of the setups, with extensive details available in Appendix 3.8. [1]

Continuous actions We perform evaluation on synthetic problems pertaining to personalized pricing problems from (Demirer et al., 2019) (*Pricing*) and advertising from (Zenati et al., 2020a) (*Advertising*). We consider Gaussian policies $\pi_\theta(\cdot|x) = \mathcal{N}(\mu_\theta(x), \sigma^2)$ with linear contextual parametrization $\mu_\theta(x) = \theta^\top x$ and fixed variance σ^2 that corresponds to the exploration budget allowed in the original randomized experiment. The features are up to 10 dimensions and the actions are one-dimensional. We keep the original logging baselines from the settings and compare results to a skyline supervised model trained on the whole training data with full information.

Discrete actions We adapt the setup of (Swaminathan and Joachims, 2015a) that transforms a multilabel classification task into a contextual bandit problem with discrete, combinatorial action space. We keep the original modeling (akin to CRF) with categorical policies $\pi_\theta(a|x) \propto \exp(\theta^\top(x \bigotimes a))$. The baseline (resp. skyline) is a supervised, full information model with identical parameter space than CRM methods trained on 5% (resp. 100%) of the training data. We consider the class of probabilistic policies that satisfy Assumption 3.5.1 by predicting actions in an Epsilon Greedy fashion (Sutton and Barto, 1998)): $\pi_\theta^\varepsilon(a, x) = (1-\varepsilon)\pi_\theta(a, x)+\varepsilon/|\mathcal{A}|$ where $\varepsilon = .1$. Real-world datasets include *Scene*, *Yeast* and *TMC2007* with feature space up to 30,438 dimensions and action space up to 2^{22}. To account for this combinatorial action space we allow a model θ_m to be learned using data from all past rollouts $\{s_l\}_{l<m}$ for better sample efficiency and therefore adjust variance estimation in Appendix 3.8 to take into account sequential dependencies.

3.6.2. SCRM compared to CRM and related methods

We first compare SCRM to CRM and existing methods in the literature.

Comparison between SCRM and CRM First, we provide insights on the performance that SCRM can achieve compared to classical CRM with increasing sample sizes. The key difference between CRM/SCRM is that for each sample size n_m CRM learns from samples generated by the logging model $s_m^{CRM} \leftarrow \theta_0$ (see Alg. 7) whilst SCRM learns from samples generated by a series of optimized models $s_m^{SCRM} \leftarrow \theta_m$ (see Alg. 6). For each sample size we select a posteriori the best λ for both methods based on test set loss value. We report in

[1] All the code to reproduce the empirical results is available at: `https://github.com/criteo-research/sequential-conterfactual-risk-minimization`

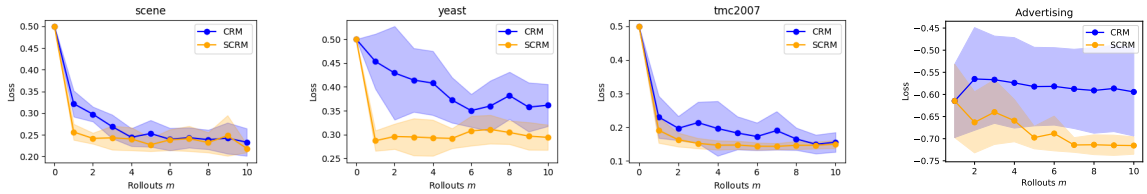

Figure 3.2: Test loss as a function of sample size on *Scene*, *Yeast*, *TMC2007*, *Advertising*, (from left to right). SCRM (in orange) converges faster and with less variance than CRM (in blue).

Percentage p	0.7	0.8	0.9
CRM	100×2^{10}	100×2^{16}	$> 100 \times 2^{22}$
SCRM (ours)	100×2^{8}	100×2^{9}	100×2^{11}

Table 3.1: Needed sample size to achieve test loss $L(\theta) \leq p * L(\theta^*)$ on the setting in Example 3.3.1 over the average of 10 random runs. SCRM needs way less data to converge to near optimal solution. λ is set to its theoretical value.

Figure 3.2 over $M = 10$ rollouts the mean test loss depending on sample size up to 2^{10}, with standard deviation estimated over 10 random runs. We observe that SCRM converges very fast, often within the first rollouts. Conversely, CRM needs more samples and the variance is higher. We conclude that there is a striking benefit to use a sequential design in order to achieve near optimal loss with much fewer samples and better confidence compared to CRM. Complementary results on other datasets are available in Appendix 3.8.

Moreover, to further illustrate this benefit of efficient learning we also report in Table 3.1 the sample size needed to attain near optimal performance when θ^* is known as in Example 3.3.1, where we also observe that SCRM reaches optimal performances faster than CRM. This corroborates the benefits of improved excess risk rates for SCRM.

Hyper-parameter selection for SCRM In our experiments, hyperparameter selection consists in choosing a value for λ. We describe a simple heuristic and evaluate its performance on different datasets. We propose to select $\hat{\lambda}_m$ by estimating the non-penalized CRM loss (Eq. (1.12)) using offline cross-validation on past data $s_{t<m}$. We report in Table 3.2 the test loss obtained when choosing a fixed λ a posteriori (λ') or with this heuristic ($\hat{\lambda}$). We observe that loss confidence intervals for both methods intersect for all discrete datasets, except on *TMC2007* where the degradation shows only at the 3rd digit. On continuous datasets, the heuristic actually improves upon the fixed a posteriori selection. We conclude that this heuristic is usable in practice.

Comparison with other methods In this paragraph we compare our SCRM to related methods to explore practical implications of existing methods in our setting. We first consider batch bandits methods and implement the stochastic sequential batch pure exploitation (SBPE) algorithm in (Han et al., 2020) and a batch version of kernel UCB (Valko et al., 2013) algorithm (BKUCB) with an optimized library (see implementations details in Appendix 3.8). We also experiment with off-policy RL methods PPO (Schulman et al., 2017) and TRPO (Schulman et al., 2015) from the StableBaselines library (Raffin et al., 2021) (see Appendix 3.8).

	Pricing	*Advertising*	*Yeast*	*TMC2007*
λ'	$-5.353 \pm .178$	$-.716 \pm .020$	$.294 \pm .026$	$.146 \pm .012$
$\hat{\lambda}$	$-5.575 \pm .036$	$-.726 \pm .001$	$.299 \pm .039$	$.164 \pm .021$

Table 3.2: Test loss after 10 rollouts when choosing λ by a posteriori selection (λ') or with proposed heuristic ($\hat{\lambda}$). Our heuristic is competitive with the a posteriori selection of a fixed λ'.

$n/\|\mathcal{A}\|/dim(\mathcal{X})\|$	*Pricing* $10^5/\infty/10$	*Advertising* $10^5/\infty/2$	*Scene* $2.10^3/2^6/295$	*Yeast* $2.10^3/2^{14}/104$	*TMC2007* $3.10^4/2^{22}/3.10^4$
Baseline	$-3.414 \pm .162$	$-.431 \pm .120$	$.353 \pm .009$	$.478 \pm .014$	$.511 \pm .003$
SBPE	DNF	DNF	**.179** $\pm .001$	$.302 \pm .003$	DNF
BKUCB	DNF	DNF	$.236 \pm .014$	$.303 \pm .004$	DNF
TRPO	**-5.750** $\pm .020$	$-.670 \pm .030$	$.376 \pm .001$	$.434 \pm .001$	$.396 \pm .001$
PPO	$-5.274 \pm .200$	$-.637 \pm .015$	$.206 \pm .001$	$.463 \pm .001$	$.263 \pm .001$
CRM	$-5.325 \pm .068$	$-.594 \pm .100$	$.233 \pm .031$	$.362 \pm .044$	$.158 \pm .034$
SCRM (ours)	$-5.575 \pm .036$	**-.726** $\pm .020$	$.219 \pm .009$	**.294** $\pm .026$	**.146** $\pm .012$
Skyline	$-5.830 \pm .020$	$-.739 \pm .002$	$.179 \pm .002$	$.312 \pm .003$	$.142 \pm .001$

Table 3.3: Test loss $\pm$ stddev of different methods after 10 rollouts. SCRM achieves optimal or near optimal performance in all datasets. Batch bandit methods did not finish (DNF) on large scale settings, and RL methods perform overall poorly on discrete settings with large action space.

Indeed, such methods model more general state transitions based on past actions, but they could be used in our setting. To fairly compare all methods (in particular those for which no heuristic existing for hyper-parameter selection) we report the mean and standard deviation over 10 random runs of the best test loss a posteriori over hyperparameter grids of the same size. First, we observe that SCRM beats CRM on all datasets, illustrating the benefit of the sequential design. Second, on discrete tasks (where we the combinatorial action space is large) we observe that SCRM achieves nearly the best test loss in all tasks, while RL methods have difficulties maintaining good performances. Third, batch bandits algorithms can achieve good performances in practice because of their deterministic decision rules. However, they involve an $O(n^3)$ matrix inversion and therefore did not finish (DNF) in 24h (per single run) on a 46 CPU / 500G RAM machine in most of our settings with large sample size n, which make them unpractical for large scale experiments. We conclude that SCRM is an effective learning paradigm and that it scales successfully on a variety of settings.

3.6.3. Details on SCRM

Next, we provide additional empirical evaluations of details of our method.

Evaluation of IPS-IX To understand the bias-variance trade-off that IPS-IX can achieve in practice compared to other counterfactual estimators we consider a policy evaluation experiment. The task we consider uses sinusoidal losses $y(a) = \cos(a)$ and evaluated policies are shifted Gaussians $\{\pi_i = \mathcal{N}(i * \pi/4, 1)\}_{i=0,4}$, with π_0 being the logging policy. Evaluated policies with large shifts with π_0 therefore simulate the setting where the logging policy underexplores the action space. The estimators we consider include IPS, SNIPS Swaminathan

and Joachims (2015b), clipped IPS (eq. IPS) with heuristic from Bottou et al. (2013) and IPS-IX (eq. 3.3) with $\alpha = 1/n$. All methods therefore use their respective heuristics to set hyperparameters. We report in Figure 3.3 the bias and variance of estimators for each shift $\mu_0 - \mu = i * \pi/4$ for $i = 0, \ldots, 4$. We observe that IPS-IX shows an empirical bias comparable to IPS, lower than SNIPS and clipped IPS while mainting a lower variance. Moreover its variance is only slightly higher than clipped IPS which introduced a large bias. We conclude that besides being a key component of our analysis IPS-IX also controls the variance with a better tradeoff in practice. More details are available in Appendix 3.8.

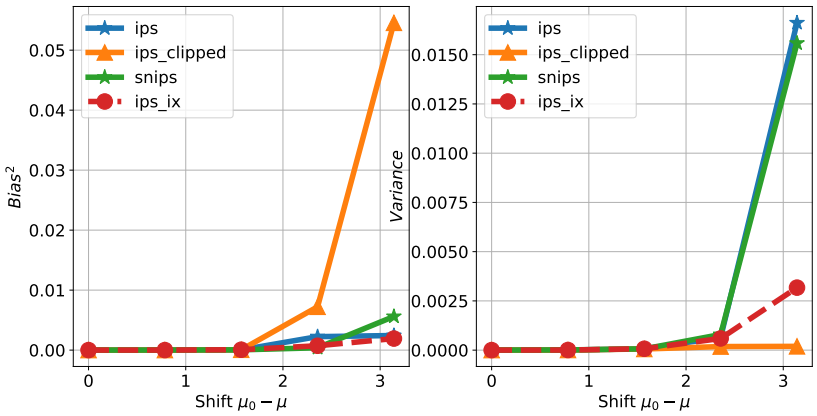

Figure 3.3: Comparison of counterfactual estimators on policy evaluation. Bias (left), Variance (right). IPS-IX shows a low bias and compares favorably to IPS and SNIPS in terms of variance.

When is SCRM useful is a natural question of interest when choosing the method to be used on a given logged bandit feedback problem. Intuitively one can imagine that SCRM will be most useful when the logging policy underexplores the action space, for example when the distance (in parameter space) between the logging and optimal parameters is large. To study this question we proceed to the following experiment on the setup of Example 3.3.1 with Gaussian distributions $\mathcal{N}(\theta, \sigma)$ and fixed loss variance $\sigma^* = \mathrm{Var}_y(y)$. We vary the distance $\delta_0 = \|\theta^* - \theta_0\|$ between the optimal model θ^* and the logging model θ_0. Since the ideal exploration level may be task dependent we choose a posteriori the best σ on a grid, for both CRM and SCRM. We report in Figure 3.4 the best final loss for both CRM and SCRM for a range of values of δ_0. We observe in particular that SCRM achieves better final losses for larger distances δ_0 than CRM. With the same number of rollouts M, SCRM can extend the exploration to further areas while CRM fails for any exploration level in those cases, which advocates for using sequential deployments.

3.7. Discussions

In this work, we have proposed a method to extend the CRM perspective for designing sequential data collection experiments. We have introduced a novel counterfactual estimator to improve variance control in excess risk bounds. Under a weak error bound assumption, we have sequentially applied these excess risk guarantees to achieve faster rates similarly to acceleration methods. Our method also improves upon CRM in practice and is particularly

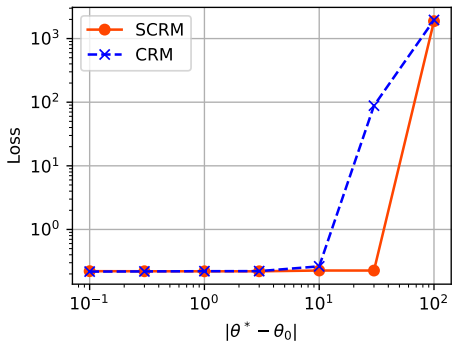

Figure 3.4: Best final loss when varying $\delta_0 = \|\theta^* - \theta_0\|$. SCRM achieves better losses especially for larger δ_0.

well-suited for this setting compared to existing methods in the literature. It is worth noting that, in order to avoid introducing dependencies in the excess risk bounds we analyzed, the theoretical algorithm we have studied uses geometric sample sizes to discard previous samples. However, using all past samples has been found to be also effective in practice and developing guarantees for this case would be an interesting area for future research. Additionally, similar to online settings that involve an exploration-exploitation tradeoff, investigating the use of optimism in the face of uncertainty (OFUL) principle in SCRM would also be a promising avenue for future work.

3.8. Appendices

This appendix is organized as follows: in Appendix 3.8, we provide additional explanations on counterfactual methods related to our approach. In Appendix 3.8, we detail our analysis of our counterfactual estimator as well as the general SCRM procedure, as given in Alg. 6. Next, in Appendix 3.8 we present all the details of the empirical evaluation and eventually in Appendix 3.8 we provide all additional empirical results that were omitted from the main paper due to space limitation.

3.9. Additional details on counterfactual estimators

3.9.1. Unconfoundedness in sequential designs

In these explanations, we recall that the distributions of contexts as well as the distribution of losses are fixed. In other words, the latter do not vary from one batch to another. In the counterfactual risk minimization framework (CRM) (Swaminathan and Joachims, 2015a), the causal graph (using the conventions in (Peters et al., 2017)) can be represented as shown in Figure 3.5.

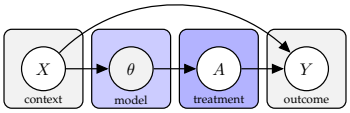

Figure 3.5: Causal Graph in a randomized data collection experiment. A denotes action (or treatment), X context, Y is the loss (or outcome). The causal influence of the contexts on actions is done through the model θ.

In the sequential counterfactual risk minimization (SCRM) framework, if we unfold the causal graph, the following representation can be given in Figure 3.6.

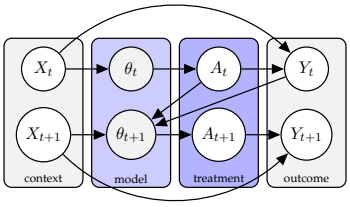

Figure 3.6: Causal Graph in a sequential randomized data collection experiment. A denotes action (or treatment), X context, Y is the loss (or outcome). The contextual treatments are taken through the models θ_t.

Therefore, it is clear that in general, $\theta_t \not\perp\!\!\!\perp \theta_{t+1}$. However, from d-separation and faithfullness (Peters et al., 2017), we have for $t' < t$:

$$\theta_t \perp\!\!\!\perp \theta_{t'} | \theta_{t-1}.$$

Therefore, given that all the dependencies are observed and that we can condition on the direct parents of a given model θ_t, sequential randomized data collection are possible.

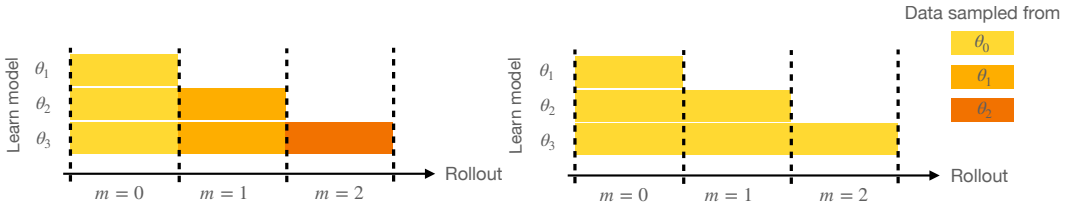

Figure 3.7: Graphical illustration of SCRM setup (left) and CRM (right), learned with same amount of data after each batch m. The training data are displayed with color block and the policy used to sample actions in these block are either adaptive (SCRM) or using the loggind model θ_0 (CRM).

More importantly, in the analysis, to ensure that no additional bias is introduced, we condition on the set of observed realizations $s_0, \dots, s_{t-1}$ that were collected to learn θ_t and apply a tower rule in the expectation as shown in the next section with the multiple importance sampling estimator.

We eventually provide in Figure 3.7 an illustration of SCRM and CRM.

3.9.2. Multiple Importance Sampling Estimators

Note that in order to avoid introducing dependencies in the excess risk bounds we analyzed, the theoretical algorithm we have studied uses geometric sample sizes to discard previous samples. However, using all past samples is effective in practice and developing guarantees for this case would be an interesting area for future research. We present in this section a estimators using aggregation of all previous information. In particular, we can use Multiple Importance Sampling (MIS) (Owen, 2013) over all previous samples. Consider in particular a partition of unity with $m > 1$ weight functions $\omega_t(a) > 0$ which satisfies $\sum_{t=0}^{m} \omega_{t,m}(a) = 1$ for all a and $m \in \{0, \dots M\}$. The MIS estimator writes:

$$\hat{L}_m^{\text{MIS}}(\theta) = \sum_{t=0}^{m} \frac{1}{n_t} \sum_{i=1}^{n_t} \omega_{t,m}(a_{t,i}) y_{t,i} w_{t,i}^{\theta}, \quad w_{t,i}^{\theta} = \frac{\pi_\theta(a_{t,i}|x_{t,i})}{\pi_{t,i}}. \tag{3.6}$$

In multiple importance sampling we usually assume that the behavior distributions are independent. In our case, when we optimize θ_t based on the models $\theta_{t-1}, \dots, \theta_0$, we break this assumption. However, as we will see, we can still have the unbiasedness property and derive an estimator for the variance of the estimator.

Proposition 3.9.1 (Unbiasedness)**.** *The MIS estimator* (3.6) *is unbiased when the loss y is fixed (its distribution $\mathcal{P}_Y(\cdot|x, a)$ does not depend on time rollout m).*

Proof. Let $m \in \{1, \dots M\}$. We recall that at all rounds $t < m$, models $\theta_t \in \Theta$ were deployed and sets s_t of n_t observations $s_t = (x_{t,i}, a_{t,i}, l_{t,i}, \pi_{t,i})_{i=1,\dots,n_t}$ were collected thereof, with propensities $\pi_{t,i} = \pi_{\theta_t}(a_{t,i}|x_{t,i})$ to learn the next model θ_{t+1}. To prove the unbiasedness we

use the tower rule on the expectation and condition on previous observations $s_1, \dots s_{t-1}$:

$$\begin{aligned}
\mathbb{E}[\hat{L}_m^{\text{MIS}}(\theta)] &= \mathbb{E}\left[\sum_{t=0}^{m} \frac{1}{n_t} \sum_{i=1}^{n_t} \mathbb{E}_{x,\theta_m,y}\left[\omega_t(a) y w_t^\theta\right]\right] \\
&= \mathbb{E}\left[\sum_{t=0}^{m} \mathbb{E}_{x,\theta_m,y}\left[\omega_t(a) y w_t^\theta\right]\right] \\
&= \sum_{t=0}^{m} \mathbb{E}_{s_1 \dots s_{t-1}}\left[\mathbb{E}_{x,\theta_m,y}\left[\omega_t(a) y w_t^\theta \mid s_1 \dots s_{t-1}\right]\right] \\
&= \sum_{t=0}^{m} \mathbb{E}_{s_1 \dots s_{t-1}}\left[\mathbb{E}_{x,\theta,y}\left[\omega_t(a) y \mid s_1 \dots s_{t-1}\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}_{x,\theta,y}\left[\left(\sum_{t=0}^{m} \omega_t(a)\right) y\right]\right] \\
&= \mathbb{E}_{x,\theta,y}\left[y\right] \\
&= L(\theta),
\end{aligned}$$

where the second last line is true only when the distribution of y does not change over time roll-outs m. □

Among the proposals for functions $\omega_t(a)$, the most 'naive' and natural heuristic is to choose

$$\omega_t(a) = \frac{n_t}{\sum_{l=1}^{m} n_l}, \tag{3.7}$$

which gives the naive concatenation of all IPS estimators

$$\hat{L}_m^{\text{n-MIS}}(\theta) = \frac{1}{n} \sum_{t=0}^{m} \sum_{i=1}^{n_t} y_{t,i} \frac{\pi_\theta(a_{t,i}|x_{t,i})}{\pi_{\theta_t}(a_{t,i}|x_{t,i})}, \tag{3.8}$$

where $n = \sum_{t=0}^{m} n_t$.

With the previous definition of the empirical mean estimator, we can now derive an empirical variance estimator, starting with the naive multi importance sampling estimator. We write the random variable $r^m = (\pi_\theta / \pi_{\theta_m}) y$. We note that for inside a batch m each realization of $r_i^m = (\pi_\theta(a_{m,i}|x_{m,i})/\pi_{m,i}) y_{m,i}$ and r_j^m are independent. But the realizations of the random variables r^m and $r^{m'}$ are dependent. Writing $n = \sum_{t=0}^{m} n_t$

$$\begin{aligned}
\text{Var}\left[\frac{1}{n} \sum_{t=0}^{m} \sum_{i=1}^{n_m} r_i^m\right] &= \sum_{t=0}^{m} \text{Var}\left[\frac{1}{n} \sum_{i=1}^{n_m} r_i^m\right] + 2 \sum_{1 \le p < q \le m} \text{Cov}\left[\frac{1}{n} \sum_{i=1}^{n_p} r_i^p, \frac{1}{n} \sum_{j=1}^{n_q} r_j^q\right] \\
&= \frac{1}{n^2} \sum_{t=0}^{m} \text{Var}\left[\sum_{i=1}^{n_m} r^m\right] + 2 \frac{1}{n^2} \sum_{1 \le p < q \le m} \sum_{i=1}^{n_p} \sum_{j=1}^{n_q} \text{Cov}\left[r^p, r^q\right] \\
&= \frac{1}{n^2} \left[\sum_{t=0}^{m} \text{Var}\left[\sum_{i=1}^{n_m} r^m\right] + 2 \sum_{1 \le p < q \le m} n_p n_q \text{Cov}\left[r^p, r^q\right]\right],
\end{aligned}$$

where the second last equality is obtained with the bilinearity of the covariance. Given the latter expression of the variance, we propose the following estimator and with a linear sampling where all $n_p = n_q$ for $p, q \in \{1, \dots, M\}$:

$$\hat{V}_m^{\text{n-MIPS}}(\theta) = \frac{1}{n^2}\left[\sum_{t=0}^{m} \hat{V}(r^t) + 2 \sum_{1 \le p < q \le m} n_p n_q \left(\frac{1}{n_p} \sum_{k=1}^{n_p} \left(r_k^p - \bar{r}_p\right)\left(r_k^q - \bar{r}_q\right)\right)\right], \tag{3.9}$$

where $\hat{V}(r^m) = \frac{1}{n_m(n_m-1)} \sum_{i=1}^{n_m} \left(r_i^m - \bar{r}^m\right)^2$ and $\bar{r}^m = \frac{1}{n_m}\sum_{j=1}^{n_m} r_j^m$.

Note also that for other functions $\omega_t(a)$, the most studied one is the balance heuristic with $\omega_t \propto n_t \pi_{\theta_t}(a)$, that is:

$$\omega_t^{BH}(a) = \frac{n_t \pi_{\theta_t}(a)}{\sum_{l=1}^{m} n_l \pi_{\theta_l}(a)}. \tag{3.10}$$

The latter heuristic has been studied for its low variance (Owen, 2013) but these properties have been studied under an i.i.d assumption that is broken in our adaptive data collection strategy. Eventually, note that controlling the variance of this estimator with an implicit exploration estimator as we do in the i.i.d case would make a an interesting research direction.

3.10. Analysis details

In this section, we provide the details of our analysis by starting with essential definitions, then our proofs of variance dependent excess risk bounds and finally our regret analysis.

3.10.1. Definitions

$C_m(\Theta)$ is a complexity measure that will be upper-bounded by the metric entropy in sup-norm at level $\varepsilon = 1/n_m$ of the following function set,

$$\mathcal{F}_{m,\Theta} := \left\{ f_\theta : (x,a,y) \in \mathcal{X}\times\mathcal{A}\times\mathcal{Y} \mapsto \frac{1}{W} + \frac{1}{W} y \left(\frac{\pi_\theta(a|x)}{\pi_{\theta_m}(a|x) + \alpha \pi_\theta(a|x)} - 1 \right) \quad \text{for } \theta \in \Theta \right\}. \tag{3.11}$$

The latter corresponds to clipped prediction errors of policies π_θ normalized into $[0,1]$. More precisely, to define rigorously $C_m(\Theta)$, we denote for any $n_m \ge 1$ and $\varepsilon > 0$, the complexity of a class $\mathcal{F}$ by

$$\mathcal{H}_\infty(\varepsilon, \mathcal{F}, n) = \sup_{(x_i,a_i,y_i)\in(\mathcal{X}\times\mathcal{A}\times\mathcal{Y})^n} \mathcal{H}(\varepsilon, \mathcal{F}\big(\{x_i,a_i,y_i\}\big), \|\cdot\|_\infty), \tag{3.12}$$

where $\mathcal{F}\big(\{x_i,a_i,y_i\}\big) = \big\{\big(f(x_1,a_1,y_1),\dots,f(x_n,a_n,y_n)\big), f \in \mathcal{F}\big\} \subseteq \mathbb{R}^n$ and the number $\mathcal{H}(\varepsilon, A, \|\cdot\|_\infty)$ is the smallest cardinality $|A_0|$ of a set $A_0 \subseteq A$ such that A is contained in the finite union of ε-balls centered at points in A_0 in the metric induced by $\|\cdot\|_\infty$). Then, $C_m(\Theta)$ is defined by

$$C_m(\Theta) = \log \mathcal{H}_\infty(1/n_m, \mathcal{F}_{m,\Theta}, 2n_m). \tag{3.13}$$

3.10.2. Variance-dependent excess risk bounds

We will denote by $\mathbb{E}_m[\cdot] = \mathbb{E}[\cdot|s_0, \dots s_m]$ the conditional expectation given the set of observation samples $s_m = (x_{m,i}, a_{m,i}, y_{m,i}, \pi_{m,i})_{i=1,\dots,n_m}$ up to the rollout m. Here, we recall that $x_{m,i} \sim \mathcal{P}_\mathcal{X}$, $a_{m,i} \sim \pi_{\theta_m}(\cdot|x_{m,i})$, $y_{m,i} \sim \mathcal{P}_\mathcal{Y}(\cdot|x_{m,i}, a_{m,i})$, and $\pi_{m,i} = \pi_{\theta_m}(a_{m,i}|x_{m,i})$. Furthermore, throughout the document, $\mathbb{E}_{x,\theta_m,y}\left[\cdot\right]$ (resp. $\mathrm{Var}_{x,\theta_m,y}\left[\cdot\right]$) denotes the expectation (resp. variance) in (x, a, y) where $x \sim \mathcal{P}_\mathcal{X}$, $a \sim \pi_{\theta_m}(\cdot|x)$, and $y \sim \mathcal{P}_\mathcal{Y}(\cdot|x, a)$.

Proposition 3.4.1 (Generalization Error Bound)**.** *Let $\hat{L}_m^{\text{IPS-IX}}$ and $\hat{V}_m^{\text{IPS-IX}}$ be the empirical estimators defined respectively in Eq.* (3.3) *and Eq.* (3.4)*. Let $\delta \in (0, 1)$, $\theta \in \Theta$, and $n_m \geq 2$ the number of samples associated to the logged dataset at round m. Then, with probability at least $1 - \delta$,*

$$L(\theta) \leq \hat{L}_m^{\text{IPS-IX}}(\theta) + \lambda\sqrt{\frac{\hat{V}_m^{\text{IPS-IX}}(\theta)}{n_m}} + \frac{2\lambda^2 W}{n_m} + \sqrt{\frac{\log(2/\delta)}{2n_m}}\,, \tag{3.14}$$

where $\lambda = \sqrt{18(C_m(\Theta) + \log(2/\delta))}$.

Proof. Let $\delta \in (0, 1)$ and $\theta \in \Theta$. Since all functions in $\mathcal{F}_{m,\Theta}$ defined in Eq. (3.11) take values in $[0, 1]$, we can apply the concentration bound of Maurer and Pontil (2009, Theorem 6) to the set $\mathcal{F}_{m,\Theta}$. This yields, with probability at least $1 - \delta/2$,

$$\begin{aligned}\mathbb{E}_{x,\theta_m,y}[f_\theta(x, a, y)] - \frac{1}{n_m}\sum_{i=1}^{n_m} f_\theta(x_{m,i}, a_{m,i}, y_{m,i}) \\ \leq \sqrt{\frac{18\hat{V}_{n_m}(f_\theta)(C_m(\Theta) + \log(2/\delta))}{n_m}} + \frac{15(C_m(\Theta) + \log(1/\delta))}{(n_m - 1)}\,,\end{aligned} \tag{3.15}$$

where

$$\hat{V}_{n_m}(f_\theta) = \frac{1}{n_m - 1}\sum_{i=1}^{n_m}\left(f_\theta(x_{m,i}, a_{m,i}, y_{m,i}) - \frac{1}{n_m}\sum_{j=1}^{n_m} f_\theta(x_{m,j}, a_{m,j}, y_{m,j})\right)^2$$

is an estimation of the sample variance. Let $\alpha > 0$ and define the following biased estimator of the excess risk:

$$L_m^\alpha(\theta) = \mathbb{E}_{x,\theta_m,y}\left[y\left(\frac{\pi_\theta(a|x)}{\pi_{\theta_m}(a|x) + \alpha\pi_\theta(a|x)} - 1\right)\right] \qquad \forall\theta \in \Theta. \tag{3.16}$$

We recall that $\mathbb{E}_{x,\theta_m,y}\left[\,\cdot\,\right]$ denotes the expectation in (x, a, y) where $x \sim \mathcal{P}_\mathcal{X}$, $a \sim \pi_{\theta_m}(\cdot|x)$, and $y \sim \mathcal{P}_\mathcal{Y}(\cdot|x, a)$. By construction of f_θ (see Eq. (3.11)),

$$\begin{aligned}\mathbb{E}_{x,\theta_m,y}[f_\theta(x, a, y)] &= \frac{1}{W} + \frac{1}{W}L_m^\alpha(\theta) \\ \frac{1}{n_m}\sum_{i=1}^{n_m} f_\theta(x_{m,i}, a_{m,i}, y_{m,i}) &= \frac{1}{W} + \frac{1}{W}\hat{L}_m^{\text{IPS-IX}}(\theta) - \frac{1}{Wn_m}\sum_{i=1}^{n_m} y_{m,i} \\ \hat{V}_{n_m}(f_\theta) &= \frac{1}{W^2}\hat{V}_m^{\text{IPS-IX}}(\theta)\,,\end{aligned}$$

where $\hat{L}_m^{\text{IPS-IX}}$ and $\hat{V}_m^{\text{IPS-IX}}$ are defined respectively in Eq. (3.3) and Eq. (3.4). Thus, multiplying (3.15) by W, substituting the above terms, and using $\lambda = \sqrt{18(C_m(\Theta) + \log(2/\delta))}$, yields

$$L_m^\alpha(\theta) - \hat{L}_m^{\text{IPS-IX}}(\theta) + \frac{1}{n_m}\sum_{i=1}^{n_m} y_{m,i} \leq \lambda\sqrt{\frac{\hat{V}_m^{\text{IPS-IX}}(\theta)}{n_m}} + \frac{15\lambda^2 W}{18(n_m - 1)}, \tag{3.17}$$

with probability $1 - \delta/2$. Now, let us decompose

$$L_m^\alpha(\theta) = \mathbb{E}_{x,\theta_m,y}\left[y\left(\frac{\pi_\theta(a|x)}{\pi_{\theta_m}(a|x) + \alpha\pi_\theta(a|x)} - 1\right)\right] = \mathbb{E}_{x,\theta_m,y}\left[y\frac{\pi_\theta(a|x)}{\pi_{\theta_m}(a|x) + \alpha\pi_\theta(a|x)}\right] - L(\theta_m).$$

But, since the losses y are bounded in $[-1, 0]$ almost surely,

$$\mathbb{E}_{x,\theta_m,y}\left[y\frac{\pi_\theta(a|x)}{\pi_{\theta_m}(a|x) + \alpha\pi_\theta(a|x)}\right] \geq \mathbb{E}_{x,\theta_m,y}\left[y\frac{\pi_\theta(a|x)}{\pi_{\theta_m}(a|x)}\right] = L(\theta),$$

which, substituted into the previous equation, entails,

$$L_m^\alpha(\theta) \geq L(\theta) - L(\theta_m). \tag{3.18}$$

Lower-bounding the left-hand side of (3.17), we thus get w.p $1 - \delta/2$,

$$L(\theta) - \hat{L}_m^{\text{IPS-IX}}(\theta) \leq \lambda\sqrt{\frac{\hat{V}_m^{\text{IPS-IX}}(\theta)}{n_m}} + \frac{15\lambda^2 W}{18(n_m - 1)} + L(\theta_m) - \frac{1}{n_m}\sum_{i=1}^{n_m} y_{m,i}.$$

Using $\mathbb{E}_{m-1}[y_{m,i}] = L(\theta_m)$ and applying Hoeffding's inequality, this further yields w.p. $1 - \delta$

$$L(\theta) \leq \hat{L}_m^{\text{IPS-IX}}(\theta) + \lambda\sqrt{\frac{\hat{V}_m^{\text{IPS-IX}}(\theta)}{n_m}} + \frac{15\lambda^2 W}{18(n_m - 1)} + \sqrt{\frac{\log(2/\delta)}{2n_m}}. \tag{3.19}$$

Eventually, note that $(n_m - 1)^{-1} \leq (2/n_m)$ since $n_m \geq 2$. Thus,

$$L(\theta) \leq \hat{L}_m^{\text{IPS-IX}}(\theta) + \lambda\sqrt{\frac{\hat{V}_m^{\text{IPS-IX}}(\theta)}{n_m}} + \frac{2\lambda^2 W}{n_m} + \sqrt{\frac{\log(2/\delta)}{2n_m}}, \tag{3.20}$$

which concludes the proof.

□

Proposition 3.4.2 (Conservative Excess Risk)**.** *Let $m \geq 0$ and $\theta_m \in \Theta$. Let s_m be a set of samples collected with $a_{m,i} \sim \pi_{\theta_m}(\cdot|x_{m,i})$. Then, under Assumptions 3.4.1 and 3.4.2, the solution θ_{m+1} of Problem* (SCRM) *with the IPS-IX estimator in Eq.* (3.5) *on the samples s_m satisfies the excess risk upper-bound*

$$\Delta_{m+1} = L(\theta_{m+1}) - L(\theta^*) \lesssim \sqrt{\frac{d\log(n_m) + \log(1/\delta)}{n_m}\nu_m^2} + \frac{W^2 + W(d\log(n_m) + \log(1/\delta))}{n_m}, \tag{3.21}$$

where $\nu_m^2 = \mathrm{Var}_{x,\theta_m}\left(\frac{\pi_{\theta^*}(a|x)}{\pi_{\theta_m}(a|x)}\right)$.

Proof. We consider the notations of the proof of Proposition 3.4.1. Fix $\theta^* \in \Theta$. Applying, Theorem 15 of Maurer and Pontil (2009)[2] to the function set $\mathcal{F}_{m,\Theta}$ defined in (3.11), we get with probability $1-\delta$

$$\mathbb{E}_{x,\theta_m,y}[f_{\theta_{m+1}}(x,a,y)] - \mathbb{E}_{x,\theta_m,y}[f_{\theta^*}(x,a,y)] \\ \leq \sqrt{\frac{32\mathrm{Var}_{x,\theta_m,y}\big[f_{\theta^*}(x,a,y)\big]\big(C_m(\Theta)+\log\frac{30}{\delta}\big)}{n_m}} + \frac{22\big(C_m(\Theta)+\log\frac{30}{\delta}\big)}{n_m-1}\,.$$

This can be written as:

$$\Delta_m^* \leq U_m^*, \tag{3.22}$$

with the following definitions:

$$\Delta_m^* = \mathbb{E}_{x,\theta_m,y}[f_{\theta_{m+1}}(x,a,y)] - \mathbb{E}_{x,\theta_m,y}[f_{\theta^*}(x,a,y)] \\ U_m^* = \sqrt{\frac{32\mathrm{Var}_{x,\theta_m,y}\big[f_{\theta^*}(x,a,y)\big]\big(C_m(\Theta)+\log\frac{30}{\delta}\big)}{n_m}} + \frac{22\big(C_m(\Theta)+\log\frac{30}{\delta}\big)}{n_m-1}. \tag{3.23}$$

Step: Lower bounding Δ_m^* Using the definition of $f_\theta(x,a,y)$ in (3.11) and that of L_m^α in Eq. (3.16), we have

$$\mathbb{E}_{x,\theta_m,y}[f_{\theta_{m+1}}(x,a,y)] = \frac{1}{W} + \frac{1}{W}L_m^\alpha(\theta_{m+1}).$$

Thus, Δ_m^* can be re-written as

$$\Delta_m^* = \frac{1}{W}\left(L_m^\alpha(\theta_{m+1}) - L_m^\alpha(\theta^*)\right),$$

which we now lower-bound. To do so, we begin by upper-bounding $L_m^\alpha(\theta^*)$. It can be expressed as

$$L_m^\alpha(\theta^*) = \mathbb{E}_{x,\theta_m,y}\left[y\frac{\pi_{\theta^*}(a|x)}{\pi_{\theta_m}(a|x)+\alpha\pi_{\theta^*}(a|x)}\right] - L(\theta_m). \tag{3.24}$$

To shorten notation, from now on and throughout this proof, we write π_θ instead of $\pi_\theta(a|x)$, omitting the dependence on a and x. Using the inequality $(1+x)^{-1} \geq 1-x$ for $x \geq 0$, we have

$$\mathbb{E}_{x,\theta_m,y}\left[y\frac{\pi_{\theta^*}}{\pi_{\theta_m}+\alpha\pi_{\theta^*}}\right] = \mathbb{E}_{x,\theta_m,y}\left[y\frac{\pi_{\theta^*}}{\pi_{\theta_m}}\frac{1}{1+\alpha\frac{\pi_{\theta^*}}{\pi_{\theta_m}}}\right] \tag{3.25}$$

$$\leq \mathbb{E}_{x,\theta_m,y}\left[y\frac{\pi_{\theta^*}}{\pi_{\theta_m}}\right] - \alpha\mathbb{E}_{x,\theta_m,y}\left[y\left(\frac{\pi_{\theta^*}}{\pi_{\theta_m}}\right)^2\right] \\ = L(\theta^*) - \alpha\mathbb{E}_{x,\theta_m,y}\left[y\left(\frac{\pi_{\theta^*}}{\pi_{\theta_m}}\right)^2\right] \\ \leq L(\theta^*) + \alpha W^2\,, \tag{3.26}$$

[2]Note that in their notation, $\log\mathcal{M}_n(\pi)$ equals $C_m(\Theta)+\log(10)$, $\mathbf{X}$ is the dataset $\{(x_i,a_i,y_i)\}_{1\leq i\leq n}$ where $(x_i,a_i,y_i) \overset{i.i.d.}{\sim} \mathcal{P}_\mathcal{X}\times\pi_{\theta_m}(\cdot|x)\times\mathcal{P}_\mathcal{Y}(\cdot|a,x)$, and $P(\cdot,\mu)$ is the expectation with respect to one test sample $\mathbb{E}_{x,\theta_m,y}[\,\cdot\,]$.

where the last inequality is by Assumption 3.4.1 and because $y \in [-1, 0]$. Together with (3.24), we get

$$L_m^\alpha(\theta^*) \leq L(\theta^*) + \alpha W^2 - L(\theta_m).$$

We recall that $L(\theta_{m+1}) - L(\theta_m) \leq L_m^\alpha(\theta_{m+1})$ by Eq.(3.18). Therefore,

$$\frac{1}{W}(L(\theta_{m+1}) - L(\theta^*) - \alpha W^2) \leq \frac{1}{W}\left(L_m^\alpha(\theta_{m+1}) - L_m^\alpha(\theta^*)\right),$$

which finally gives

$$\frac{1}{W}(L(\theta_{m+1}) - L(\theta^*) - \alpha W^2) \leq \Delta_m^*. \tag{3.27}$$

Step: Upper bound U_m^* By definition of $f_\theta(x, a, y)$ in (3.11), we have

$$\begin{aligned}\mathrm{Var}_{x,\theta_m,y}\big[f_{\theta^*}(x,a,y)\big] &= \frac{1}{W^2}\mathrm{Var}_{x,\theta_m,y}\left[y\left(\frac{\pi_{\theta^*}}{\pi_{\theta_m}+\alpha\pi_{\theta^*}} - 1\right)\right] \\ &\leq \frac{1}{W^2}\mathbb{E}_{x,\theta_m,y}\left[y^2\left(\frac{\pi_{\theta^*}}{\pi_{\theta_m}+\alpha\pi_{\theta^*}} - 1\right)^2\right] \leq \frac{1}{W^2}\mathbb{E}_{x,\theta_m}\left[\left(\frac{\pi_{\theta^*}}{\pi_{\theta_m}+\alpha\pi_{\theta^*}} - 1\right)^2\right].\end{aligned}$$

Then, using the inequality $(x+y)^2 \leq 2x^2 + 2y^2$, for $x, y \in \mathbb{R}$, this may be upper-bounded as

$$\begin{aligned}&\mathrm{Var}_{x,\theta_m,y}\big[f_{\theta^*}(x,a,y)\big] \\ &\leq \frac{2}{W^2}\mathbb{E}_{x,\theta_m}\left[\left(\frac{\pi_{\theta^*}}{\pi_{\theta_m}+\alpha\pi_{\theta^*}} - \mathbb{E}_{x,\theta_m}\left[\frac{\pi_{\theta^*}}{\pi_{\theta_m}+\alpha\pi_{\theta^*}}\right]\right)^2\right] + \frac{2}{W^2}\left(\mathbb{E}_{x,\theta_m}\left[\frac{\pi_{\theta^*}}{\pi_{\theta_m}+\alpha\pi_{\theta^*}}\right] - 1\right)^2.\end{aligned} \tag{3.28}$$

On the one hand, the first term of the right-hand side may be upper-bounded as

$$\mathbb{E}_{x,\theta_m}\left[\left(\frac{\pi_{\theta^*}}{\pi_{\theta_m}+\alpha\pi_{\theta^*}} - \mathbb{E}_{x,\theta_m}\left[\frac{\pi_{\theta^*}}{\pi_{\theta_m}+\alpha\pi_{\theta^*}}\right]\right)^2\right] = \mathrm{Var}_{x,\theta_m}\left[\frac{\pi_{\theta^*}}{\pi_{\theta_m}+\alpha\pi_{\theta^*}}\right] \leq \nu_m^2,$$

where $\nu_m^2 = \mathrm{Var}_{x,\theta_m}\left[\frac{\pi_{\theta^*}}{\pi_{\theta_m}}\right]$. On the other hand, for the second term, we use the same factorization as in Eq. (3.25) to get

$$-\alpha\,\mathbb{E}_{x,\theta_m}\left[\left(\frac{\pi_{\theta^*}}{\pi_{\theta_m}}\right)^2\right] \leq \mathbb{E}_{x,\theta_m}\left[\frac{\pi_{\theta^*}}{\pi_{\theta_m}+\alpha\pi_{\theta^*}}\right] - 1 \leq 0\,,$$

which yields the upper-bound

$$\left(\mathbb{E}_{x,\theta_m}\left[\frac{\pi_{\theta^*}}{\pi_{\theta_m}+\alpha\pi_{\theta^*}}\right] - 1\right)^2 \leq \alpha^2\,\mathbb{E}_{x,\theta_m}\left[\left(\frac{\pi_{\theta^*}}{\pi_{\theta_m}}\right)^2\right] \leq \alpha^2 W^2.$$

Therefore, substituting the last two upper-bounds into (3.28) entails

$$\mathrm{Var}_{x,\theta_m,y}\big[f_{\theta^*}(x,a,y)\big] \leq \frac{2}{W^2}\big(\nu_m^2 + \alpha^2 W^2\big)\,.$$

Then, replacing this upper-bound into the definition of U_m^* in (3.23) and using Assumption 3.4.2 to upper bound the terms in $C_m(\Theta) \leq d\log(n_m)$, we obtain the following upper-bound

$$\begin{aligned} U_m^* &\leq \frac{1}{W}\sqrt{\frac{64(\nu_m^2 + \alpha^2 W)\big(d\log(n_m) + \log\frac{30}{\delta}\big)}{n_m}} + \frac{22\big(d\log(n_m) + \log\frac{30}{\delta}\big)}{n_m - 1} \\ &\leq \frac{1}{W}\sqrt{\frac{64(\nu_m^2 + \alpha^2 W)\big(d\log(n_m) + \log\frac{30}{\delta}\big)}{n_m}} + \frac{44\big(d\log(n_m) + \log\frac{30}{\delta}\big)}{n_m}, \end{aligned} \tag{3.29}$$

where the last inequality is because $n_m \geq 2$.

Step: excess risk upper bound Setting $\alpha = \frac{1}{n_m}$ and using the two previous bounds (3.27) and (3.29) respectively on Δ_m^* and on U_m^* into (3.22), we get

$$L(\theta_{m+1}) - L(\theta^*) \leq \sqrt{\frac{64\big(d\log(n_m) + \log\frac{30}{\delta}\big)}{n_m}\big(\nu_m^2 + \frac{1}{n_m^2}W^2\big)} + W\frac{44\big(d\log(n_m) + \log\frac{30}{\delta}\big)}{n_m} + \frac{1}{n_m}W^2. \tag{3.30}$$

Using that $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$, we have that

$$\begin{aligned} &\sqrt{\frac{64\big(d\log(n_m) + \log\frac{30}{\delta}\big)}{n_m}\big(\nu_m^2 + \frac{1}{n_m^2}W^2\big)} \\ &\qquad \leq \sqrt{\frac{64\big(d\log(n_m) + \log\frac{30}{\delta}\big)}{n_m}\nu_m^2} + \frac{W}{n_m}\sqrt{64\big(d\log(n_m) + \log\frac{30}{\delta}\big)}. \end{aligned}$$

Then, since $n_m \geq 2$ and $\delta < 1$, we have $d\log(n_m) + \log(30/\delta) \geq \log(2) + \log(30) \geq 4$, which yields

$$\frac{1}{n_m}\sqrt{\frac{64\big(d\log(n_m) + \log\frac{30}{\delta}\big)}{n_m}} \leq \frac{\sqrt{32\big(d\log(n_m) + \log\frac{30}{\delta}\big)}}{n_m} \leq \frac{\sqrt{8}\big(d\log(n_m) + \log\frac{30}{\delta}\big)}{n_m}.$$

Substituting the last two inequalities into (3.30) finally entails

$$L(\theta_{m+1}) - L(\theta^*) \leq 8\sqrt{\frac{d\log(n_m) + \log\frac{30}{\delta}}{n_m}\nu_m^2} + 47W\frac{d\log(n_m) + \log\frac{30}{\delta}}{n_m} + \frac{W^2}{n_m}, \tag{3.31}$$

which concludes the proof. □

3.10.3. Regret analysis

Proposition 3.5.1 (Regret upper-bound). *Let $n_0, n \geq 2$ and $\theta^* \in \arg\min_\theta L(\theta)$. Let $n_m = n_0 2^m$ for $m = 0, \dots, M = \big\lfloor \log_2(1 + \frac{n}{n_0})\big\rfloor$. Then, under Assumptions 3.4.1, 3.4.2 and 3.5.1, the SCRM procedure (Alg. 6) satisfies the excess risk upper-bound*

$$L(\theta_M) - L(\theta^*) \leq O\Big(n^{-\frac{1}{2-\beta}}\log n\Big).$$

Moreover, the expected regret is upper-bounded as follows:

$$R_n = \mathbb{E}\left[\sum_{m=0}^{M} n_{m+1}\big(L(\theta_m) - L(\theta^*)\big)\right] \leq O\Big(n^{\frac{1-\beta}{2-\beta}}\log(n)^2\Big).$$

Proof. First, note that for $n_m = n_0 2^m$ and $M = \left\lfloor \log_2(1 + \frac{n}{n_0}) \right\rfloor$, we have $\sum_{m=0}^{M-1} n_m = n_0(2^M - 1) \leq n$. Hence, Alg. 6 has collected at most n samples to design the estimator θ_M. For $m \geq 0$, we recall $\Delta_m = L(\theta_m) - L(\theta^*)$ and use Eq. (3.31) to write

$$\begin{aligned}
\Delta_{m+1} &\leq 8\sqrt{\frac{d\log(n_m) + \log\frac{30}{\delta}}{n_m}\nu_m^2} + 47W\frac{d\log(n_m) + \log\frac{30}{\delta}}{n_m} + \frac{W^2}{n_m} \\
&\leq 8\sqrt{\frac{d\log(n) + \log\frac{30}{\delta}}{n_m}\nu_m^2} + 47W\frac{d\log(n) + \log\frac{30}{\delta}}{n_m} + \frac{W^2}{n_m} \\
&= C\sqrt{\frac{\nu_m^2}{n_m}} + \frac{B}{n_m}\,, \qquad (3.32)
\end{aligned}$$

where $C = 8\sqrt{d\log(n) + \log\frac{30}{\delta}}$ and $B = W^2 + 47W(d\log(n) + \log\frac{30}{\delta})$ are independent of m.

Step: Obtaining a recurrence relation for Δ_{m+1} By Assumption 3.5.1, there exist $\gamma > 0$ and $\beta \in [0, 1]$ such that

$$\nu_m^2 = \mathrm{Var}_{x,\theta_m}\left(\frac{\pi_{\theta^*}}{\pi_{\theta_m}}\right) \leq \frac{1}{\gamma}\big(L(\theta_m) - L(\theta^*)\big)^\beta = \frac{\Delta_m^\beta}{\gamma}\,.$$

Replacing ν_m^2 in Eq. (3.32) thus entails

$$\begin{aligned}
\Delta_{m+1} &\leq C\sqrt{\frac{1}{\gamma}\frac{\Delta_m^\beta}{n_m}} + \frac{B}{n_m} \\
&\leq C2^{-\frac{m}{2}}\sqrt{\frac{n_0}{\gamma}\Delta_m^\beta} + B2^{-m}n_0 \qquad \leftarrow n_m = n_0 2^m \\
&= C\sqrt{\frac{n_0}{\gamma}}2^{-\frac{m}{2}}\Delta_m^{\beta/2} + B2^{-m}n_0\,. \qquad (3.33)
\end{aligned}$$

Step: Solving the recurrence relation for Δ_m We then insure by induction that Δ_m satisfies

$$\Delta_m \leq c_0 2^{\frac{-m}{2-\beta}}\,, \qquad (3.34)$$

for some $c_0 > 0$ that will be specified by the analysis.

Base step Since losses take values in $[-1, 0]$, $\Delta_0 = L(\theta_0) - L(\theta^*) \leq 1$. Equation (3.34) is thus satisfied for $m = 0$ as soon as $c_0 \geq 1$.

Induction step Let $m \geq 0$. We assume that $\Delta_m \leq c_0 2^{\frac{-m}{2-\beta}}$ and prove Equation (3.34) for Δ_{m+1}. Using Eq. (3.33), we have

$$\begin{aligned}
\Delta_{m+1} &\leq C\sqrt{\frac{n_0}{\gamma}}2^{-\frac{m}{2}}\Delta_m^{\beta/2} + B2^{-m}n_0 \\
&\leq C\sqrt{\frac{n_0}{\gamma}}2^{-\frac{m}{2}}{c_0}^{\beta/2}2^{-\frac{m\beta}{\beta(2-\beta)}} + B2^{-m}n_0 \qquad \leftarrow \quad \text{by induction} \\
&\leq \max\left\{2C\sqrt{\frac{n_0}{\gamma}}{c_0}^{\frac{\beta}{2}}2^{-\frac{m}{2}-\frac{m}{2-\beta}}, 2B2^{-m}n_0\right\}. \qquad (3.35)
\end{aligned}$$

Now, we show that both terms inside the maximum can be upper-bounded by $c_0 2^{-(m+1)/(2-\beta)}$ as soon as c_0 is large enough. On the one hand, if $c_0 \geq 4Bn_0$, we have

$$2B2^{-m}n_0 \leq c_0 2^{-(m+1)} \leq c_0 2^{-\frac{m+1}{2-\beta}} .$$

On the other hand, if $c_0 \geq (4C^2 n_0/\gamma)^{1/(2-\beta)}$, we also have

$$2C\sqrt{\frac{n_0}{\gamma}}c_0^{\frac{\beta}{2}}2^{-\frac{m}{2}-\frac{m}{2-\beta}} \leq 2C\sqrt{\frac{n_0}{\gamma}}c_0^{\frac{\beta}{2}}2^{-\frac{m+1}{2-\beta}} \leq c_0 2^{-\frac{m+1}{2-\beta}} .$$

Combining the above two upper-bounds with (3.35) concludes the induction step under the condition

$$c_0 \geq \max\left\{1, \Big(\frac{4C^2 n_0}{\gamma}\Big)^{\frac{1}{2-\beta}}, 4Bn_0\right\}.$$

Step: conclusion Finally, setting the above value for c_0 we proved that for all $m \geq 0$, we have

$$\begin{aligned}
\Delta_m &\leq \max\left\{1, \Big(\frac{4C^2 n_0}{\gamma}\Big)^{\frac{1}{2-\beta}}, 4Bn_0\right\} 2^{-\frac{m}{2-\beta}} \\
&\leq \left(1 + \Big(\frac{4C^2 n_0}{\gamma}\Big)^{\frac{1}{2-\beta}} + 4Bn_0\right) 2^{-\frac{m}{2-\beta}} \\
&= \left(1 + \left(\frac{256(d\log n + \log\frac{30}{\delta})n_0}{\gamma}\right)^{\frac{1}{2-\beta}} + W^2 n_0 + 47Wn_0\Big(d\log n + \log\frac{30}{\delta}\Big)\right) 2^{-\frac{m}{2-\beta}} ,
\end{aligned} \tag{3.36}$$

where the last equality is by substituting the values of B and C from (3.32). For the final step $M = \lfloor \log_2(\frac{n}{n_0} + 1)\rfloor$, this yields

$$\begin{aligned}
\Delta_M &\leq \left(1 + \left(\frac{256(d\log n + \log\frac{30}{\delta})n_0}{\gamma}\right)^{\frac{1}{2-\beta}} + W^2 n_0 + 47Wn_0\Big(d\log n + \log\frac{30}{\delta}\Big)\right) 2^{-\frac{M}{2-\beta}} \\
&\leq 2\left(1 + \left(\frac{256(d\log n + \log\frac{30}{\delta})n_0}{\gamma}\right)^{\frac{1}{2-\beta}} + W^2 n_0 + 47Wn_0\Big(d\log n + \log\frac{30}{\delta}\Big)\right) \times \Big(\frac{n_0}{n}\Big)^{\frac{1}{2-\beta}} \\
&= O\Big(n^{-\frac{1}{2-\beta}}\log n\Big) .
\end{aligned}$$

This concludes the first part of the proof.

Regret upper-bound To upper bound the cumulative regret, using $n_{m+1} = n_0 2^{m+1}$, we write

$$R_n = \sum_{m=0}^{M} \Delta_m n_{m+1} \overset{(3.36)}{\leq} D\sum_{m=0}^{M} 2^{-\frac{m}{2-\beta}} n_{m+1} = 2Dn_0 \sum_{m=0}^{M} 2^{\left(\frac{1-\beta}{2-\beta}\right)m} ,$$

where

$$D = 1 + \left(\frac{256(d\log n + \log\frac{30}{\delta})n_0}{\gamma}\right)^{\frac{1}{2-\beta}} + W^2 n_0 + 47Wn_0\Big(d\log n + \log\frac{30}{\delta}\Big) .$$

Then, computing the sum for $M = \lfloor \log_2(\frac{n}{n_0}+1) \rfloor$, we have

$$R_n \leq 2Dn_0 \sum_{m=0}^{M} 2^{\left(\frac{1-\beta}{2-\beta}\right)m} \leq 2Dn_0(M+1)2^{\left(\frac{1-\beta}{2-\beta}\right)M} \leq 2Dn_0\Big(1+\log_2\Big(\frac{n}{n_0}+1\Big)\Big) \times \Big(1+\frac{n}{n_0}\Big)^{\frac{1-\beta}{2-\beta}}.$$

Using that $D = O(\log n)$, we finally obtain

$$R_n \leq O\Big(n^{\frac{1-\beta}{2-\beta}} \log(n)^2\Big).$$

□

3.11. Additional discussions on the Hölderian Bound Assumption 3.5.1

In this appendix, we discuss Assumption 3.5.1 on different particular examples.

3.11.1. Verification of the assumption on a toy example with Gaussian families

We consider the setting of Example 3.3.1. In the latter, the policies are Gaussian of the form $\pi_\theta = \mathcal{N}(\theta, \sigma^2)$ and the loss is defined by $l_t(a) = (a - y_t)^2 - 1$ where $y_t \sim \mathcal{N}(\theta^*, \sigma^2)$. There is no loss in generality in assuming $\sigma^2 = 1$. Then, we can compute

$$L(\theta) - L(\theta^*) = (\theta - \theta^*)^2 \qquad \text{and} \qquad \mathrm{Var}_\theta\left[\frac{\pi_{\theta^*}(a)}{\pi_\theta(a)}\right] = \exp\big((\theta^* - \theta)^2\big) - 1\,.$$

We recall that we are interested in verifying the existence of $\gamma > 0$ and $\beta > 0$ for which Assumption 3.5.1 holds, that is in this case for any $\theta \in \Theta$:

$$\gamma \mathrm{Var}_\theta\left[\frac{\pi_{\theta^*}(a)}{\pi_\theta(a)}\right] \leq \big(L(\theta) - L(\theta^*)\big)^\beta\,, \tag{3.37}$$

which may be re-written here as

$$\gamma\big(\exp\big((\theta^* - \theta)^2\big) - 1\big) \leq (\theta - \theta^*)^{2\beta}\,.$$

The latter is satisfied for any $\beta \leq 1$ as soon as Θ is a bounded interval. Note that the constant γ may decrease exponentially fast as the diameter of Θ increases. To illustrate, the existence of such couples (β, γ), we plot in Fig. 3.8 different values of the following ratio

$$R(\theta, \beta) = \frac{\mathrm{Var}_\theta\left[\frac{\pi_{\theta^*}(a)}{\pi_\theta(a)}\right]}{\big(L(\theta) - L(\theta^*)\big)^\beta} = \frac{\exp\big((\theta^* - \theta)^2\big) - 1}{\big(\|\theta - \theta^*\|^2\big)^\beta}. \tag{3.38}$$

The value of γ can be found for different values of β in Fig. 3.8 by taking $\frac{1}{\gamma} = \max_\theta R(\theta, \beta)$. Higher values of β induce faster rates and lower values of γ induce worst constant terms in the excess risk and regret bounds. Eventually, note that SCRM does not need those parameters to run and those two parameters γ, β are automatically calibrated by SCRM to find the best trade-off.

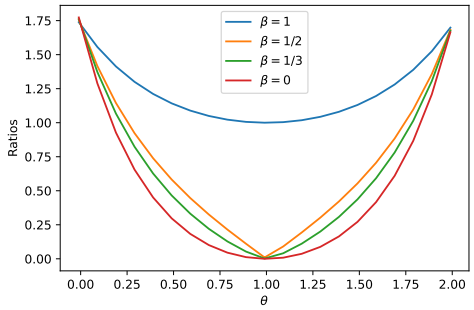

Figure 3.8: Ratio R defined in (3.38) with different values of β.

3.11.2. Discussion of Assumption 3.5.1 for Exponential Families

In this section, we consider a more realistic example in which policies belong to an exponential family. That is, we assume that the policies are parameterized by a parameter $\eta \in \mathbb{R}^q$ and can be written in the form:

$$\forall a \in \mathcal{A}, \qquad \pi_\eta(a) = e^{\eta \cdot t(a) - A(\eta)} h(a),$$

for some known function $h : \mathcal{A} \to \mathbb{R}_+$ and sufficient statistic $t : \mathcal{A} \to \mathbb{R}^q$. Here, $A(\eta)$ is a normalization constant, so that $e^{A(\eta)} = \int_a e^{\eta \cdot t(a)} h(a)\, \mathrm{d}a$. We provide in Example 3.11.1 a concrete example considered by (Swaminathan and Joachims, 2015a; Faury et al., 2020). To ease the notation, we removed here the dependency on contexts, but the generalization to contextual policies can be made similarly. The importance weight ratio may be written as,

$$\frac{\pi_\eta(a)}{\pi_{\eta_m}(a)} = e^{(\eta - \eta_m)t(a) - (A(\eta) - A(\eta_m))}. \tag{3.39}$$

To verify Assumption 3.5.1, we need to upper bound their variance, which we shall write as,

$$\mathrm{Var}_{a \sim \pi_{\eta_m}} \left[\frac{\pi_\eta(a)}{\pi_{\eta_m}(a)} \right] = e^{2(A(\eta_m) - A(\eta))} \mathrm{Var}_{a \sim \pi_{\eta_m}} \left[e^{(\eta - \eta_m)t(a)} \right].$$

Now, computing the moment generating function (MGF) of the statistic $t(a) \in \mathbb{R}^q$

$$\begin{aligned} M_t(s) = \mathbb{E}\left[e^{s \cdot t(a)} \right] &= \int_a e^{s \cdot t(a)} e^{\eta_m \cdot t(a) - A(\eta_m)} h(a)\, \mathrm{d}a \\ &= e^{-A(\eta_m)} \int_a e^{(\eta_m + s) \cdot t(a)} e^{\eta_m \cdot t(a)} h(a)\, \mathrm{d}a \\ &= e^{A(\eta_m + s) - A(\eta_m)}, \end{aligned}$$

the variance term may be written as

$$\mathrm{Var}_{a \sim \pi_{\eta_m}} \left[e^{(\eta - \eta_m)t(a)} \right] = M_t(2(\eta - \eta_m)) - M_t^2(\eta - \eta_m) = e^{A(2\eta - \eta_m) - A(\eta_m)} - e^{2(A(\eta) - A(\eta_m))}.$$

This eventually leads us to

$$\mathrm{Var}_{a \sim \pi_{\eta_m}} \left[\frac{\pi_\eta(a)}{\pi_{\eta_m}(a)} \right] = e^{A(2\eta - \eta_m) + A(\eta_m) - 2A(\eta)} - 1. \tag{3.40}$$

We now discuss two cases that are used for discrete actions (Swaminathan and Joachims, 2015a) and continuous actions (Kallus and Zhou, 2018; Zenati et al., 2020a).

Bounded sufficient statistic Supposing that there exists an upper bound A such that $\|t(a)\| \leq A$, Cauchy-Schwartz inequality states that $|(\eta - \eta_m) \cdot t(a)| \leq \|\eta - \eta_m\| A$, which entails

$$\begin{aligned}
\mathrm{Var}_{a\sim\pi_{\eta_m}}\left[\frac{\pi_\eta(a)}{\pi_{\eta_m}(a)}\right] &= e^{A(2\eta-\eta_m)+A(\eta_m)-2A(\eta)} - 1 \\
&= \frac{\int_a e^{(2\eta-\eta_m)\cdot t(a)} h(a)\,\mathrm{d}a \int_a e^{\eta_m \cdot t(a)} h(a)\,\mathrm{d}a}{\left(\int_a e^{\eta\cdot t(a)} h(a)\,\mathrm{d}a\right)^2} - 1 \\
&= \frac{\int_a e^{(\eta-\eta_m)\cdot t(a)} e^{\eta\cdot t(a)} h(a)\,\mathrm{d}a \int_a e^{(\eta_m-\eta)\cdot t(a)} e^{\eta\cdot t(a)} h(a)\,\mathrm{d}a}{\left(\int_a e^{\eta\cdot t(a)} h(a)\,\mathrm{d}a\right)^2} - 1 \\
&\leq e^{\|\eta-\eta_m\|A} - 1\,.
\end{aligned}$$

Assuming that the parameter space is compact, i.e, $\max_{\eta,\eta'} \|\eta - \eta'\| \leq D$, there exists a constant C that depends on A and D such that, this may be further upper-bounded as

$$\mathrm{Var}_{a\sim\pi_{\eta_m}}\left[\frac{\pi_\eta(a)}{\pi_{\eta_m}(a)}\right] \leq C\|\eta - \eta_m\|\,.$$

Therefore, Assumption 3.5.1 is implied by

$$\gamma C\|\eta - \eta_m\|^2 \leq (L(\theta) - L(\theta^*))^{2\beta}\,.$$

The latter is implied by a local version of strong convexity for $\beta = 1/2$ (d'Aspremont et al., 2021), and holds with $\gamma = C^{-1}D^{-2}$ for $\beta = 0$.

Example 3.11.1. *For discrete actions $\mathcal{A} = \{a_1, \ldots, a_K\}$, we consider, as in (Swaminathan and Joachims, 2015a) and (Faury et al., 2020), policies where given a context x, probabilities $p_i(x)$ of sampling an action a_i are given by*

$$p_i(x) = \frac{\exp(\theta^\top \phi(x, a_i))}{\sum_{j=1}^K \exp(\theta^\top \phi(x, a_j))}. \tag{3.41}$$

The function ϕ is typically a feature map associated to a kernel in a RKHS. In this case, the natural parameter η and the sufficient statistic $t(a)$ may be written as

$$\eta = \begin{bmatrix} \log(\frac{p_1}{p_K}) \\ \vdots \\ \log(\frac{p_{K-1}}{p_K}) \\ 0 \end{bmatrix} \quad t(a) = \begin{bmatrix} \mathbb{1}\{a = a_1\} \\ \vdots \\ \mathbb{1}\{a = a_K\} \end{bmatrix}. \tag{3.42}$$

Lognormal and Normal distributions For normal $\mathcal{N}(\mu, \sigma^2)$ and lognormal Lognormal(μ, σ^2) distributions with fixed variance σ^2 as considered by (Kallus and Zhou, 2018; Zenati et al., 2020a), the normalizing constant writes $A(\eta) = \frac{\eta^2}{2}$, and we then obtain that:

$$A(2\eta - \eta_m) + A(\eta_m) - 2A(\eta) = (\eta - \eta_m)^2\,,$$

which gives:

$$\mathrm{Var}_{a\sim\pi_{\eta_m}}\left[\frac{\pi_\eta(a)}{\pi_{\eta_m}(a)}\right] = e^{\|\eta-\eta_m\|^2} - 1\,.$$

In that case, it is again possible for a bounded parameter space to linearize $e^{\|\eta-\eta_m\|^2} - 1 \lesssim \|\eta-\eta_m\|^2$, consider losses that verify: for all η, there exists an optimal η^* such that

$$\gamma\|\eta_m - \eta^*\|^2 \leq \big(L(\eta_m) - L(\eta^*)\big)^\beta. \tag{3.43}$$

Again, this holds generally for $\beta = 0$ and for locally strongly convex losses for $\beta = 1$.

3.12. Experiment details

3.12.1. Code

All the code to reproduce figures and tables is available in the following repository: `https://github.com/criteo-research/sequential-conterfactual-risk-minimizatio` n.

3.12.2. Empirical settings details

Pricing The pricing application in (Demirer et al., 2019) considers a "personalized pricing" setting where given contexts x, prices p (which are the actions) need to be predicted to maximize the revenue:

$$r(x,p) = p(a(x) - b(x)p + \varepsilon)$$

where $\varepsilon \sim \mathcal{N}(0,1)$ and $d = a(x) + b(x)p + \varepsilon$ is akin to an unknown context-specifidemand function. The data generating process uses contexts $x \in [1,2]^k$ for $k > 1$ a positive integer. Only $l < k$ dimensions however affect the demand, that is if we write $\bar{x} = \frac{1}{l}(z_1, \ldots, z_l)$. The price p is generated from a Gaussian logging policy $p \sim \mathcal{N}(\bar{x}, 1)$ centered in $\bar{x}$. We consider in our example the quadratic functionnal $a(x) = 2x^2$ and $b(x) = 0.6x$ as in the original paper.

Advertising The advertising simulation in (Zenati et al., 2020a) consists in predicting the potential $p \in]0, +\infty[$ of a user that may be compared to their a priori responsiveness to a treatment. The potential is caused by an unobserved random group variable g in G (groups of "high" or "low" potential users in their responsiveness) that influences context x of users. The goal is then to find a policy $\pi(a|x)$ that maximizes reward by adapting to an unobserved potential. The potentials are normally distributed conditionally on the group index, $p|g \sim \mathcal{N}(\mu_g, \sigma_g^2)$ where $\sigma_g = 0.5$ and $\mu_g = 1$ or 3 for two groups. The observed reward $-y$ is then a function of the action a and the context x through the associated potential p_x of the user x. The reward function mimics reward over the offline continuous bidding dataset in (Zenati et al., 2020a) with the form:

$$r_l(p_x, a) = \begin{cases} \frac{a}{p_x} & \text{if } a < p_x \\ \frac{1}{2}(p_x - a) + 1 & \text{else} \end{cases}$$
$$r(p_x, a) = \max(r_l(p_x, a), -0.1)$$

The logging policy is a lognormal distribution as it is common in advertising applications (Bottou et al., 2013). In particular, as in (Zenati et al., 2020a), $\pi_{\theta_0} = \text{Lognormal}(\mu, \sigma^2)$ where the mean $\exp(\mu + \sigma^2/2) = 2$ and the variance $(\exp(\sigma^2) - 1)\exp(2\mu + \sigma^2) = 1$.

Yeast, Scene, TMC2007 We follow Swaminathan and Joachims (2015a). We now recall briefly the setup. The problem is a binary multilabel classification with $|\mathcal{A}| = 2^K$ potential labels. All models are parametrized by $\pi_\theta(a|x) \propto \exp(\theta^\top(x \bigotimes a))$. The baseline (resp. skyline) is a supervised, full information model with identical parameter space than CRM methods trained on 5% (resp. 100%) of the training data. Our main modification it to consider the class of probabilistic policies that satisfy Assumption 3.5.1 by predicting actions in an Epsilon Greedy fashion Sutton and Barto (1998)): $\pi_\theta^\varepsilon(a, x) = (1 - \varepsilon)\pi_\theta(a, x) + \varepsilon/|\mathcal{A}|$ where $\varepsilon = .1$. The loss is the Hamming loss (number of incorrectly assigned labels - both false positives and false negatives in the action vector):

$$L(\theta) = \frac{1}{nK}\sum_{i=1}^{n}\sum_{j=1}^{K}\mathbb{1}_{[y_i^j = a_i^j]} \tag{3.44}$$

where y_i^j (resp. a_i^j) is the j-th component of the label vector (resp. action vector) of line i. A uniform policy will thus evaluate at a loss of .5.

3.12.3. Implementation details

Counterfactual methods In this paragraph we start by detailing the non adaptive counterfactual risk minimization that we compare to in this work.

Algorithm 7: Counterfactual Risk Minimization

Input: Logged observations $(x_{0,i}, a_{0,i}, y_{0,i}, \pi_{0,i})_{i=1,\dots,n_0}$, parameter $\lambda > 0$
for $m = 1$ *to* M **do**
  Build $\mathcal{L}_m$ from observations s_m using Eq. (3.5)
  Learn θ using Eq. (SCRM)
  Re-deploy the logging model θ_0 and collect observations
  $s_{m+1} = (x_{m+1,i}, a_{m+1,i}, l_{m+1,i}, \pi_{m+1,i})_{i=1,\dots,n_{m+1}}$;
end

We also provide the grid of hyperparameters for the λ evaluated in CRM and SCRM methods $\lambda \in [1e-5, 1e-4, 1e-3, 1e-2, 1e-1]$.

Batch Bandits Let $k : (\mathcal{X} \times \mathcal{A}) \times (\mathcal{X} \times \mathcal{A}) \to \mathbb{R}$ be a bounded positive definite Kernel associated to a RKHS $\mathcal{H}$, $\phi : \mathcal{X} \times \mathcal{A} \to \mathcal{H}$ is the feature map such that $k(s, s') = \langle \phi(s), \phi(s') \rangle$

for any $s, s' \in \mathcal{X} \times \mathcal{A}$. Context-actions pairs are written as $s_{m,i} := (x_{m,i}, a_{m,i}) \in \mathcal{X} \times \mathcal{A}$ and $\mathcal{S}_m := \{s_{1,0}, \dots, s_{n_m,m}\}$ denoting the history of all context-actions pairs seen up until the end of batch m. K_m is the kernel matrix of all context-actions seen until the end of the batch $m \geq 1$. Eventually, $K_{\mathcal{S}}(s')$ is the kernel column vector $[k(s_1, s'), \dots, k(s_l, s')]^\top$ of size $|\mathcal{S}| = l$. $Y_m = [-y_{0,1}, \cdots - y_{0,n_0}, \cdots - y_{m,1}, \cdots - y_{m,n_m}]$ denotes the vector of concatenated rewards observed up until the end of the batch m.

At a batch m, a context $x_{m,i}$ is sampled for $i \in \{1, n_m\}$, and then to sample an action a, the following decision rule is applied:

$$a \in \arg\max_{a \in \mathcal{A}} \hat{q}_{m,i,a}. \tag{3.45}$$

In batch Kernel UCB, $\hat{q}_{m,i,a}$ is defined as

$$\hat{q}_{m,i,a} = \hat{\mu}_{m,i,a} + \beta_m \hat{\sigma}_{m,i,a}, \tag{3.46}$$

where

$$\hat{\mu}_{m,i,a} = K_{\mathcal{S}_{t-1}}\big((x_{m,i}, a)\big)^\top K_{m-1}^{-1} Y_{m-1}$$
$$\hat{\sigma}^2_{m,i,a} = \frac{1}{\lambda} k\big((x_{m,i}, a), (x_{m,i}, a)\big) - \frac{1}{\lambda} K_{\mathcal{S}_{m-1}}\big((x_{m,i}, a)\big)^\top K_{m-1}^{-1} K_{\mathcal{S}_{m-1}}\big((x_{m,i}, a)\big),$$

and β_m is a theoretical parameter that is set to $\beta_m = \frac{1}{\sqrt{m}}$ in practical heuristics (Lattimore and Szepesvári, 2020). In SBPE (Han et al., 2020), $\hat{q}_{m,i,a}$ is defined directly as

$$\hat{q}_{m,i,a} = K_{\mathcal{S}_{t-1}}\big((x_{m,i}, a)\big)^\top K_{m-1}^{-1} Y_{m-1}. \tag{3.47}$$

Algorithm 8: Batch bandit - SBPE (Han et al., 2020) and Kernel UCB (Valko et al., 2013)

Input: Logged observations $(x_{0,i}, a_{0,i}, y_{0,i}, \pi_{0,i})_{i=1,\dots,n_0}$, λ regularization and exploration parameters, k the kernel function
initialization
$K_\lambda = [k(s_{0,i}, s_{0,j})]_{1 \leq i,j \leq n_0} + \lambda I, Y_0 = [-y_{0,i}]_{1 \leq i \leq n_0}$
for $m = 1$ *to* M **do**
  for $i = 1$ *to* n_m **do**
    Observe context $x_{i,m}$
    Choose $a_{i,m} \leftarrow \arg\max_{a \in \mathcal{A}} \hat{q}_{m,i,a}$ using Eq. (3.47) or (3.46)
  end
  Observe losses $y_{i,m}$ for all i in past batch $\{1, \dots, n_m\}$
  Update $Y_m \leftarrow [-y_{0,1}, \cdots - y_{0,n_0}, \cdots - y_{m,1}, \cdots - y_{m,n_m}]$
  Update the translated gram matrix $K_\lambda \leftarrow [k(s_{i,p}, s_{j,p})]_{1 \leq i,j \leq n_p, 1 \leq p \leq m} + \lambda I$
end

SBPE (Han et al., 2020) uses a linear modelling, therefore we used a linear kernel. For the Kernel UCB (Valko et al., 2013) method, we used Gaussian and Polynomial kernels in our

experiments. Note also that no regularization parameter λ is used in SBPE so we set $\lambda = 0$ in our experiments, and for K-UCB we chose λ in the grid $[1e0, 1e1, 1e2]$.

Note in particular that we adapted the batch bandit baselines to the CRM setting by benefiting the initialization with the logged dataset to set the gram matrix K_λ as well as the reward vector Y_0 with information from the logging data. This modification changes the original methods which take random actions at initializations.

Eventually, the baselines were carefully optimized using the Jax library (`https://github.com/google/jax`) to allow for just in time compilations of algebraic blocks in both methods and to maximize their scaling capacity.

RL baselines In order to compare our method to the two known off-policy online RL algorithm `PPO` (Schulman et al., 2017) and `TRPO` (Schulman et al., 2015), we do the following:

1. we use the `stable_baselines3`(Raffin et al., 2021) library for the implementation. When necessary we call multiple times the model `PPO` or `TRPO`, to have buffer size of geometrical increase.
2. we initialize the `ActorCriticPolicy` with a simpler MLP model having only one layer with output dimension of 1, (with argument `net_arch= [1]`, that is mathematically the same modelling as in CRM and SCRM baselines).
3. At the initial step only and to enable a fair comparison with counterfactual methods using a logging dataset, we pretrain the RL policies to imitate the actions sampled from the logging policy: we process by multiple step of the Adam optimizer, minimizing a loss being the sum of 2 terms:
   - a MSE term between the sampled action of the `ActorCriticPolicy` for the contexts in the n_0 instances, and the actions sampled by the logging policy.
   - the ENTROPY term guaranteeing to keep a minimum of exploration in order to initialize the RL algorithm ($-\sum p_i \log(p_i)$)
4. we combine the 2 last terms with a linear combinaison with hyperparameters being tuned a posteriori, i.e. LOSS $=$ MSE $+ \lambda$ ENTROPY with the hyperparam $\lambda \in \{.5, 1, 2, 5, 10\}$

3.13. Additional empirical results

3.13.1. SCRM compared to CRM

We provide here the additional plot in the *Pricing* setting.

3.13.2. Evaluation of IPS-IX

We provide here the plots for the whole setting considered in policy evaluation with IPS-IX.

3.13.3. Exploration/Exploitation tradeoff

In this part we give the details used for the experiment described in Section 3.6.3. We consider again Example 3.3.1 with the Gaussian parametrized policies $\pi_\theta = \mathcal{N}(\theta, \sigma^2)$ and a

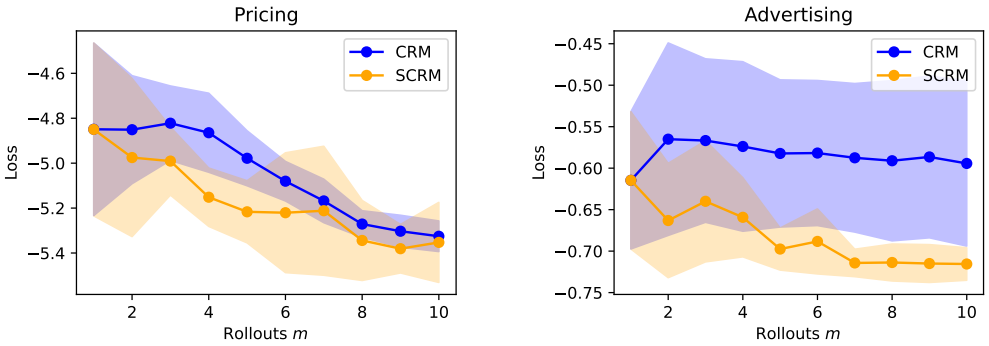

Figure 3.9: Test loss as a function of sample size on *Pricing*, *Advertising* (from left to right).

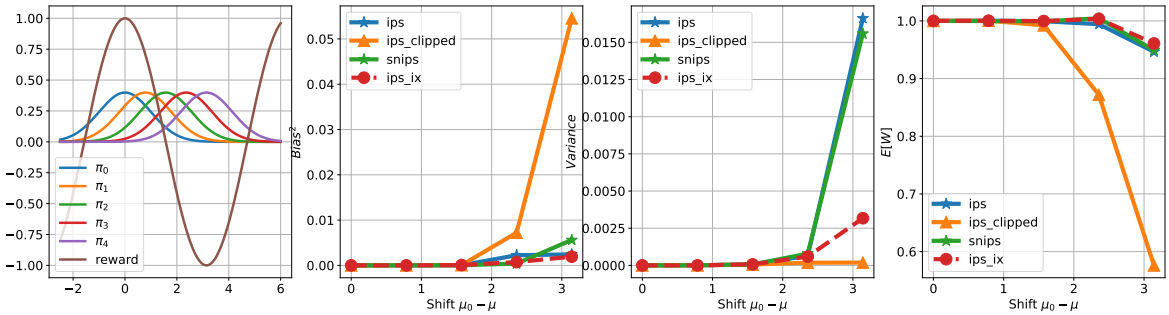

Figure 3.10: Comparison of IPS estimators on a Cosine reward and series of shifted Gaussian policies. Setup (left), Bias (middle left), Variance (middle right), Average IPS weight (right). IPS-IX shows a low bias and compares favorably to IPS and SNIPS in terms of variance.

loss $l_t(a) = (a - y_t)^2 - 1$ where $y_t \sim \mathcal{N}(\theta^*, \sigma^{*2})$ with $\sigma^* = 0.3$. Recall that $\pi_{\theta_0} = \mathcal{N}(\theta_0, \sigma)$. We consider a grid of $\sigma \in [0.1, 0.3, 1, 3]$ and consider $\theta^* = 1$. Our experiment aims at illustrating the influence of sequential exploration that is an important detail of the SCRM and CRM principles.

Part II

Efficient learning in Sequential Bandit Problems

4

Contextual Bandits: an efficient algorithm for Kernel UCB

In this chapter, we tackle the computational efficiency of kernelized UCB algorithms in contextual bandits. While standard methods require a $\mathcal{O}(CT^3)$ complexity where T is the horizon and the constant C is related to optimizing the UCB rule, we propose an efficient contextual algorithm for large-scale problems. Specifically, our method relies on incremental Nyström approximations of the joint kernel embedding of contexts and actions. This allows us to achieve a complexity of $\mathcal{O}(CTm^2)$ where m is the number of Nyström points. To recover the same regret as the standard kernelized UCB algorithm, m needs to be of order of the effective dimension of the problem, which is at most $\mathcal{O}(\sqrt{T})$ and nearly constant in some cases.

This chapter is based on the following material:

H. Zenati, A. Bietti, E. Diemert, J. Mairal, M. Martin, and P. Gaillard. Efficient kernelized ucb for contextual bandits. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2022

4.1. Introduction

Contextual bandits for sequential decision making have become ubiquitous in many applications such as online recommendation systems (Li et al., 2010). At each round, an agent observes a *context* vector and chooses an *action*; then, the *environment* generates a *reward* based on the chosen action. The goal of the agent is to maximize the cumulative reward over time, which requires a careful balancing between exploitation (maximizing reward using past observations) and exploration (increasing the diversity of observations).

In this paper, we consider a kernelized contextual bandit framework, where the rewards are modeled by a function in a reproducing kernel Hilbert space (RKHS). In other words, we assume the expected reward to be linear with respect to a joint context-action feature map of possibly infinite dimension. This setup provides flexible modeling choices through the feature map for both discrete and continuous action sets, and exploration algorithms typically rely on constructing confidence sets for the parameter vector and exploring using upper confidence bound (UCB) rules (Li et al., 2010). The extensions to infinite-dimensional feature maps we consider has been introduced by Krause and Ong (2011); Valko et al. (2013) using kernelized variants of UCB, which allow effective exploration even for rich non-parametric reward functions lying in a RKHS, such as smooth functions over contexts and/or actions.

Despite the rich modeling capabilities of such kernelized UCB algorithms, they lack scalability since standard algorithms scale at best as $\mathcal{O}(CT^3)$ where T is the horizon (total number of rounds) and the constant C is the cost of selecting an action according to the UCB optimization rule. This large cost is due to the need to solve linear systems involving a $t \times t$ kernel matrix at each round t, and motivates developing efficient versions of these algorithms for large problems. In supervised learning, a common technique for reducing computation cost is to leverage the fact that the kernel matrix is often approximately low-rank, and to use Nyström approximations (Williams and Seeger, 2001; Rudi et al., 2015). We extend such approximations to the contextual bandit setting, by relying on incremental updates of a dictionary of Nyström anchor points, which allows us to reduce the complexity to $\mathcal{O}(CTm^2)$, where m is the final number of dictionary elements. In order to preserve a small regret comparable to the vanilla kernel UCB method, m is of the order of an *effective dimension* quantity, which is typically much smaller than T, and at most $\sqrt{T}$.

Closely related to our work, Calandriello et al. (2019, 2020) recently considered Nyström approximations in the non-contextual setting with finite actions, corresponding to a Bayesian optimization problem. Whereas their algorithm is effective when there are no contexts, a direct extension to the contextual setting yields a complexity of $\mathcal{O}(Tm^3)$, which may be $\mathcal{O}(T^{2.5})$ in the worst case, despite a batching strategy allowing to recompute a new dictionary only about m times. In contrast, our incremental strategy reduces the previous complexity to $\mathcal{O}(Tm^2)$, and thus at most $\mathcal{O}(T^2)$.

Even though adopting an incremental strategy for updating the Nyström dictionary may seem to be a simple idea, achieving the previously-mentioned complexity while preserving a regret that is comparable to the original kernel UCB approach is non-trivial. Nyström approximations cause dependencies in the projected kernel matrix that makes it difficult to use martingale arguments, which led Calandriello et al. (2020) to use other mathematical tools that are compatible with updates resampling a new Nyström dictionary. In contrast, we

manage to use martingale arguments for an incremental strategy that is less computationally expensive. For that, we extend the standard analysis of the OFUL algorithm for linear bandits (Abbasi-yadkori et al., 2011; Chowdhury and Gopalan, 2017) to the kernel setting with Nyström approximations. In particular, this requires non-trivial extensions of concentration bounds to infinite-dimensional objects. Our analysis also uses the incremental structure of the projections that Calandriello et al. (2019) do not have. This allows us to prove the complexity of our algorithm. Moreover, unlike previous works, we explicit the regret-complexity trade-off under the capacity condition assumption. Finally, we also provide numerical experiments showing that our theoretical gains are also observed in practice.

Algorithm	Regret	Space	Time Complexity
CGP-UCB (Krause and Ong, 2011)	$\mathcal{O}(\sqrt{T}d_{\text{eff}}(\lambda, T))$	$\mathcal{O}(T^2)$	$\mathcal{O}(CT^3)$
SupKernelUCB (Valko et al., 2013)	$\mathcal{O}(\sqrt{Td_{\text{eff}}(\lambda, T)\log(C)})$	$\mathcal{O}(T^2)$	$\mathcal{O}(CT^3)$
C-BKB (Calandriello et al., 2019)	$\mathcal{O}(\sqrt{T}(\sqrt{\lambda d_{\text{eff}}(\lambda, T)} + d_{\text{eff}}(\lambda, T))$	$\mathcal{O}\left(Td_{\text{eff}}\right)$	$\mathcal{O}\left(T^2 d_{\text{eff}}^2 + CTd_{\text{eff}}^2\right)$
C-BBKB (Calandriello et al., 2020)	$\mathcal{O}(\sqrt{T}(\sqrt{\lambda d_{\text{eff}}(\lambda, T)} + d_{\text{eff}}(\lambda, T))$	$\mathcal{O}\left(Td_{\text{eff}}\right)$	$\mathcal{O}\left(Td_{\text{eff}}^3 + CTd_{\text{eff}}^2\right)$
K-UCB (ours)	$\mathcal{O}(\sqrt{T}(\sqrt{\lambda d_{\text{eff}}(\lambda, T)} + d_{\text{eff}}(\lambda, T))$	$\mathcal{O}(T^2)$	$\mathcal{O}(CT^3)$
EK-UCB (ours)	$\mathcal{O}(\sqrt{T}(\sqrt{\lambda d_{\text{eff}}(\lambda, T)} + d_{\text{eff}}(\lambda, T))$	$\mathcal{O}\left(Td_{\text{eff}}\right)$	$\mathcal{O}(CTd_{\text{eff}}^2)$

Table 4.1: Comparison of regret bounds (up to logarithmic factors in T) and total time complexity. When the action space is finite, for e.g in SupKernelUCB, we write $C = |\mathcal{A}|$ its cardinality and note that the argmax is obtained in C computations of the UCB rule. Note that the reported regret of CGP-UCB, SupKernel UCB and CBBKB use here the definition of the effective dimension $d_{\text{eff}}(\lambda, T)$ in Eq. (4.7) which depends on the horizon T and the parameter λ (i.e the inverse of the GP noise in CGP-UCB, BKB and BBKB). This effective dimension d_{eff} is equivalent, up to logarithmic factors, to the information gain used by Srinivas et al. (2010); Calandriello et al. (2020) and the definition used by Valko et al. (2013) (see Appendix 4.7). Moreover, we report the complexities of the contextualized versions of BKB and BBKBs, noting that the non-contextual versions may benefit from certain optimizations when the action space is discrete (Calandriello et al., 2019, 2020).

4.2. Related Work

UCB algorithms are commonly used in the bandit literature to carefully balance exploration and exploitation by defining confidence sets on unknown reward functions (Lattimore and Szepesvári, 2020). For stochastic linear contextual bandits, the OFUL algorithm (Abbasi-yadkori et al., 2011) obtains improved guarantees compared to previous analyses (e.g.,, Li et al., 2010) by providing tighter confidence bounds based on self-normalized tail inequalities.

Extensions of linear contextual bandits and UCB algorithms to infinite-dimensional representations of contexts or actions have been studied by Krause and Ong (2011) and Valko et al. (2013) by using kernels and Gaussian processes. While their analyses involve different concepts of effective dimension, it can be shown that these are closely related (see Section 4.3.3). Valko et al. (2013) notably achieves a better scaling in the horizon in the regret, but requires a finite action space. Chowdhury and Gopalan (2017) improves the analysis of GP-UCB using tools inspired by Abbasi-yadkori et al. (2011) and similar to our analysis of kernel-UCB, though it considers the non-contextual setting. Tirinzoni et al. (2020) in the contextual linear bandit problem use a primal-dual algorithm to achieve an optimal asymptotical regret bound but does not address the issue of computational complexity nor the kernelized setting. Likewise, Camilleri et al. (2021) propose a new estimator in the non-contextual kernelized

bandit problem to achieve a tighter regret bound using an elimination algorithm but does no focus on computational efficiency neither.

In the Bayesian experimental design literature Derezinski et al. (2020) propose an efficient sampling scheme using determinant point processes in the non-kernel case and a non-contextual framework. For improving the computational complexity of kernelized UCB procedures in a non-contextual setting as well, Calandriello et al. (2019) use a Nyström approximation of the kernel matrix which is recomputed at each step. Because the corresponding algorithm is not practical when a large number of steps are needed, Calandriello et al. (2020) consider a batched version, which significantly improves its computation and complexity.

In contrast, we use an incremental construction based on the KORS method (Calandriello et al., 2017a), which has been used previously with full information feedback (see also Jézéquel et al., 2019), allowing us to significantly improve the computational complexity of the contextual GP-UCB algorithm, for the same regret guarantee. Such an incremental approach appears to be a key to achieve better complexity than a natural contextual variant of the algorithm of Calandriello et al. (2020), see Table 4.1, both in theory and in practice (see Section 4.5). Such an extension is unfortunately non-trivial and requires a different regret analysis, as discussed earlier.

Mutnỳ and Krause (2019) also study kernel approximations for efficient variants of GP-UCB, focusing on random feature expansions. Nevertheless, the number of random features may need to be very large–often exponential in the dimension–in order to achieve good regret, due to a misspecification error which requires stronger, uniform approximation guarantees. Finally, Kuzborskij et al. (2019) also considers leverage score sampling for computational efficiency, but focuses on linear bandits in finite dimension.

4.3. Warm-up: Kernel-UCB for Contextual Bandits

In this section, we introduce stochastic contextual bandits with reward functions lying in a RKHS, and provide an analysis of the Kernel-UCB algorithm (similar to GP-UCB) which will be a starting point for studying the computationally efficient version in Section 4.4.

Notations. We define here basic notations. Given a vector $v \in \mathbb{R}^d$ we write its entries $[v_i]_{1 \leq i \leq d}$ and we will write $v^\top w$ or $\langle v, w \rangle_{\mathcal{H}}$ the dot product for elements in $\mathbb{R}^d$ and in the Hilbert space $\mathcal{H}$. We denote by $\| \cdot \|$ the Euclidean norm and the norm in $\mathcal{H}$. The conjugate transpose for a linear operator L on $\mathcal{H}$ is denoted by L^*. For two operators L, L' on $\mathcal{H}$, we write $L \preccurlyeq L'$ when $L - L'$ is positive semi-definite and we use $\lesssim$ for approximate inequalities up to logarithmic multiplicative or additive terms. A summary of the notations is provided in Appendix 4.7.

4.3.1. Setup

In the contextual bandit problem, at each time t in $1, \dots, T$, where T is the horizon, for each context x_t in $\mathcal{X}$, an action a_t in $\mathcal{A}$ is chosen by an agent and induces a reward r_t in $\mathbb{R}$.

The input and action spaces $\mathcal{X}$ and $\mathcal{A}$ can be arbitrary (e.g., finite or included in $\mathbb{R}^d$ for some $d \geq 1$). Note that $\mathcal{A}$ may change over time, but we keep it fixed here for simplicity.

In this paper, we focus on stochastic kernel contextual bandits and assume that there exists a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ such that

$$r_t = \langle \theta^*, \phi(x_t, a_t) \rangle_{\mathcal{H}} + \varepsilon_t \,,$$

where ε_t are i.i.d. centered subGaussian noise, $\theta^* \in \mathcal{H}$ is an unknown parameter, and $\phi : \mathcal{X} \times \mathcal{A} \to \mathcal{H}$ is a known feature map associated to $\mathcal{H}$. It satisfies

$$\langle \phi(x, a), \phi(x', a') \rangle_{\mathcal{H}} = K\big((x, a), (x', a')\big) \,,$$

where K is a positive definite kernel associated to $\mathcal{H}$. We assume K to be bounded, i.e., there exists $\kappa > 0$ such that $K(s, s) \leq \kappa^2$ for any $s \in \mathcal{X} \times \mathcal{A}$.

Thus, the goal of the agent is, given the previously observed contexts, actions and rewards $(x_s, a_s, r_s)_{s=1\ldots t-1}$ and the current context x_t, to choose an action a_t in order to minimize the following regret after T rounds

$$R_T := \mathbb{E}\left[\sum_{t=1}^{T} \max_{a \in \mathcal{A}} \langle \theta^*, \phi(x_t, a) \rangle_{\mathcal{H}} - \sum_{t=1}^{T} r_t \right]. \tag{4.1}$$

4.3.2. Algorithm: Kernel-UCB

Upper confidence algorithm (UCB) algorithms maintain for each possible action an estimate of the mean reward as well as a confidence interval around that mean, and then chooses at each time the highest upper confidence bound. Formally, if we have a confidence set $\mathcal{C}_t \subset \mathcal{H}$ based on samples $(x_{t'}, a_{t'}, y_{t'})$, for $t' \in \{1, \ldots, t-1\}$ that contains the unknown parameter vector θ^* with high probability, we may define

$$\text{K-UCB}_t(a) = \max_{\theta \in \mathcal{C}_t} \langle \theta, \phi(x_t, a) \rangle_{\mathcal{H}} \tag{4.2}$$

as an upper bound on the mean pay-off $\langle \theta^*, \phi(x_t, a) \rangle_{\mathcal{H}}$ of a. To choose the highest upper confidence bound from the confidence set at time t, the algorithm then selects:

$$a_t \in \arg\max_{a \in \mathcal{A}} \text{K-UCB}_t(a). \tag{4.3}$$

We then build an empirical estimate of the unknown quantity θ^* using regression. More precisely in the kernelized setting, we use the regularized least square estimator with

$$\hat{\theta}_t \in \arg\min_{\theta \in \mathcal{H}} \left\{ \sum_{s=1}^{t} \left(\langle \theta, \phi(x_s, a_s) \rangle_{\mathcal{H}} - r_s \right)^2 + \lambda \|\theta\|^2 \right\}. \tag{4.4}$$

Rearranging the terms $\varphi_s = \phi(x_s, a_s)$ and writing $V_t = \sum_{s=1}^{T} \varphi_s \otimes \varphi_s + \lambda I$, we obtain that the analytical solution for Eq. (4.4) is $\hat{\theta}_t = V_t^{-1} \sum_{s=1}^{t} \varphi_s r_s$. The previous solution from time $t-1$ then defines the center of the ellipsoidal confidence set

$$\mathcal{C}_t = \{\theta \in \mathcal{H} : \|\theta - \hat{\theta}_{t-1}\|_{V_{t-1}} \leq \beta_t(\delta)\}. \tag{4.5}$$

where $\|\theta\|_V^2 = \theta^\top V\theta$, and $\beta_t(\delta)$ is its radius (see Lemma 4.3.1). With $\mathcal{C}_t$ in that form, we can write the solution of Eq. (4.2) as

$$\text{K-UCB}_t(a) = \langle\hat{\theta}_{t-1}, \phi(x_t, a)\rangle_{\mathcal{H}} + \beta_t(\delta)^{1/2}\|\phi(x_t, a)\|_{V_{t-1}^{-1}}. \tag{4.6}$$

Indeed, by defining $B_2 = \{x \in \mathbb{R}^d : \|x\|_2 \leq 1\}$ the unit ball with the Euclidean norm, it is easy to see that $\mathcal{C}_t = \hat{\theta}_t + \beta_t(\delta)^{1/2}V_{t-1}^{-1/2}B_2$. Then, for $\theta \in B_2$ maximising the quantity $\langle\theta, \phi(x_t, a)\rangle_{\mathcal{H}} = \phi(x_t, a)^\top\hat{\theta}_{t-1} + \beta_t(\delta)^{1/2}\phi(x_t, a)^\top V_{t-1}^{-1/2}\theta$ gives Eq. (4.6).

4.3.3. Regret analysis

We provide an analysis of the regret of the kernelized UCB rule in Eq. (4.6) using standard statistical analysis definitions of the effective dimension.

Let us write the operator $\Phi_t : \mathcal{H} \to \mathbb{R}^t$ such that $\Phi_t^* = [\varphi_1, \dots \varphi_t]$, where $\varphi_i = \phi(x_i, a_i)$ for $i \in [1, t]$. Let us define K_t the kernel matrix associated to kernel K and the set of pairs $(x_1, a_1), \dots, (x_t, a_t)$, $K_t = \Phi_t\Phi_t^*$ is a $t \times t$ matrix. We define the effective dimension of a kernel matrix as in Hastie et al. (2001) and will use the following in our work.

Definition 4.3.1. *The effective dimension of the matrix K_T is defined as,*

$$d_{\text{eff}}(\lambda, T) := \text{Tr}(K_T(K_T + \lambda I_T)^{-1}). \tag{4.7}$$

In what follows, for simplicity of notation, we abbreviate $d_{\text{eff}}(\lambda, T)$ to d_{eff} unless we use different parameters on d_{eff}. To extend the analysis of OFUL (Abbasi-yadkori et al., 2011) to the contextual kernel UCB algorithm, we will use the following proposition that has been proved and used by Jézéquel et al. (2019).

Proposition 4.3.1. *For any horizon $T \geq 1, \lambda > 0$ and all input sequences $(x_1, a_1), \dots, (x_T, a_T)$*

$$\sum_{k=1}^{T} \log\left(1 + \frac{\lambda_k(K_T)}{\lambda}\right) \leq \log\left(e + \frac{eT\kappa^2}{\lambda}\right) d_{\text{eff}},$$

where $\lambda_k(K_T)$ denotes the k-th largest eigenvalue of K_T.

We now provide a regret bound extending the analysis of Abbasi-yadkori et al. (2011) to the kernel setting. In particular, we start by providing an upper bound on the ellipsoid greater axis.

Lemma 4.3.1. *Let $\delta \in (0, 1)$ and define $\beta_{t+1}(\delta)$ by*

$$\sqrt{\lambda}\|\theta^*\| + \sqrt{2\log\frac{1}{\delta} + \log\left(e + \frac{et\kappa^2}{\lambda}\right) d_{\text{eff}}}.$$

Then, with probability at least $1 - T\delta$, for all $t \in [T]$

$$\left\|\hat{\theta}_t - \theta^*\right\|_{V_t} \leq \beta_{t+1}(\delta). \tag{4.8}$$

We use this lemma (which relies on Proposition 4.3.1 whose proof is in Appendix 4.7) to bound the distance between the estimated parameter $\hat{\theta}_t$ at each round t and the true parameter θ^*. By combining this result with Proposition 4.3.1, we then prove the following theorem that extends the LinUCB upper bound result from Lattimore and Szepesvári (2020).

Theorem 4.3.1. *Let $T \geq 2$ and $\theta^* \in \mathcal{H}$. Assume that $|\langle \phi(x,a), \theta^* \rangle_{\mathcal{H}}| \leq 1$ for all $a \in \bigcup_{t=1}^{T} \mathcal{A}_t \subset \mathcal{A}$ and $x \in \mathcal{X}$. Then, the K-UCB rule defined in Eq.* (4.3) *for the choice $\mathcal{C}_t$ as in* (4.5) *with parameter $\lambda > 0$, and $\delta = 1/T^2$, satisfies the pseudo-regret bound*

$$R_T \lesssim \sqrt{T}\Big(\|\theta^*\| \sqrt{\lambda d_{\text{eff}}} + d_{\text{eff}} \Big) ,$$

where $\lesssim$ hides logarithmic factors in T.

The proof of Theorem 4.3.1 and the precise statement of the regret bound are given in Appendix 4.7.

In particular, assuming the norm of the true parameter $\theta^\star$ to be bounded, we obtain the following corollary with a capacity condition on the effective dimension.

Corollary 4.3.1. *Assuming the capacity condition $d_{\text{eff}} \leq (T/\lambda)^\alpha$ for $0 \leq \alpha \leq 1$, the regret of K-UCB is bounded as $R_T \lesssim T^{\frac{1+3\alpha}{2+2\alpha}}$ with an optimal $\lambda \approx T^{\frac{\alpha}{1+\alpha}}$.*

As an example, if we consider a kernel that is a tensor product between a linear kernel on contexts and a Sobolev-type kernel (e.g.,, a Matern kernel) of order s on actions, with $s > d/2$ (where d is the dimension of the continuous action space), then we may consider that the kernel eigenvalues decay as $i^{-2s/d}$, leading to an effective dimension as above with $\alpha = d/2s$, and a regret of $T^{\frac{1}{2}\frac{2s+3d}{2s+d}}$.

Discussion. We note that this regret is not optimal for such problems, but matches the regret of most other kernel or Gaussian process optimization algorithms (see, e.g.,, Scarlett et al., 2017). More precisely, our analysis recovers classical rates of the GP-UCB algorithm (Srinivas et al., 2010; Chowdhury and Gopalan, 2017), and extends them to the contextual bandit setting. We note that the analysis of Chowdhury and Gopalan (2017) further removes some logarithmic factors, and similar improvements may be obtained in our setting since it is based on similar tools. The SupKernelUCB algorithm by Valko et al. (2013) obtains improved dependencies on T in the regret bounds, but requires a finite set of actions, and therefore is not directly comparable to ours. The CGP-UCB algorithm by Krause and Ong (2011) obtains similar results to ours in the contextual setting, but uses a different analysis. Our result is therefore not new, and our analysis is meant as a starting point for the efficient variant based on incremental Nyström approximations, which will be introduced in the sequel.

We note that these works use different notions than our effective dimension d_{eff} to characterize complexity, namely the information gain

$$\gamma(\lambda, t) = \frac{1}{2} \log \left(\det \left(I + \frac{1}{\lambda} K_t \right) \right)$$

used by Krause and Ong (2011) as well as the different effective dimension definition in (Valko et al., 2013)

$$\tilde{d}(\lambda, t) \;=\; \min\{ j : j\lambda \log T \geq \sum_{k>j} \lambda_K(K_t) \}.$$

It can be shown that these are equivalent up to logarithmic factors to our definition of the effective dimension d_{eff} (see Appendix 4.7). This allows us to compare up to logarithmic factors the algorithm regrets, as shown in Table 4.1.

4.4. Efficient Kernel-UCB

In this section, we introduce our efficient kernelized UCB (EK-UCB) algorithm based on incremental Nyström projections. We begin by extending the ellipsoidal confidence bounds from the previous section to the case with projections on finite-dimensional linear subspaces of the RKHS. Then, we present our main algorithm and analyze its complexity and regret.

4.4.1. Upper confidence bounds with projections

In this section, we study the UCB updates and corresponding high-probability confidence bounds for our EK-UCB algorithm. Because these steps do not depend on a specific choice of projections, we consider generic projection operators onto subspaces of the RKHS, noting that the next sections will consider specific choices based on Nyström approximations.

At round $t \geq 1$, we consider a generic subspace $\tilde{\mathcal{H}}_t$ of $\mathcal{H}$, and let $P_t : \mathcal{H} \to \tilde{\mathcal{H}}_t$ be the orthogonal projection operator on $\tilde{\mathcal{H}}_t$, so that $P_t\mathcal{H} = \tilde{\mathcal{H}}_t$. For a fixed regularization parameter $\lambda > 0$, we consider the following regularized estimator restricted to $\tilde{\mathcal{H}}_t$:

$$\tilde{\theta}_t \in \operatorname*{arg\,min}_{\theta \in \tilde{\mathcal{H}}_t} \left\{ \sum_{s=1}^{t} \left(\langle \theta, \phi(x_s, a_s)\rangle_{\mathcal{H}} - r_s\right)^2 + \lambda \|\theta\|^2 \right\}. \tag{4.9}$$

Define $\tilde{V}_t = \sum_{s=1}^{t} P_t\varphi_s \otimes P_t\varphi_s + \lambda I$, which may be written $\tilde{V}_t = P_t F_t P_t + \lambda I$ where $F_t = \Phi_t^* \Phi_t : \mathcal{H} \to \mathcal{H}$ is the covariance operator. Recalling the notation $Y_t = (r_1, \dots, r_t)^\top$, we obtain that $\tilde{\theta}_t = \tilde{V}_t^{-1} P_t \Phi_t^* Y_t$. We may then define the following ellipsoidal confidence set:

$$\tilde{\mathcal{C}}_t := \left\{ \theta \in \mathcal{H} : \|\theta - \tilde{\theta}_{t-1}\|_{\tilde{V}_{t-1}} \leq \tilde{\beta}_t(\delta) \right\}, \tag{4.10}$$

for some radius $\tilde{\beta}_t(\delta)$ to be specified later. Note that the ellipsoid is not necessarily contained inside the projected space $\tilde{\mathcal{H}}_t$, and may in fact include θ^* even if $\theta^* \notin \tilde{\mathcal{H}}_t$. This is a crucial difference with random feature kernel approximations (Mutnỳ and Krause, 2019), for which a standard confidence set would be finite dimensional, and thus generally does not include θ^*; this leads to larger regret due to misspecification, unless the number of random features is very large in order to ensure good *uniform* approximation. We may then define the following upper confidence bounds, which still rely on the original feature map ϕ:

$$\text{EK-UCB}_t(a) := \max_{\theta \in \tilde{\mathcal{C}}_t} \langle \theta, \phi(x_t, a)\rangle_{\mathcal{H}}. \tag{4.11}$$

This may again be written in closed form as

$$\text{EK-UCB}_t(a) = \langle \hat{\theta}_{t-1}, \phi(x_t, a)\rangle_{\mathcal{H}} + \tilde{\beta}_t(\delta)^{1/2} \|\phi(x_t, a)\|_{\tilde{V}_{t-1}^{-1}}.$$

We note that for appropriate choices of $\tilde{\mathcal{H}}_t$, such a quantity can be explicitly computed using the kernel trick, as we discuss in Section 4.4.3. The following lemma shows that $\tilde{\mathcal{C}}_t$ is a

Algorithm 9: Incremental KORS subroutine (Calandriello et al., 2017a)

Input: Time t, past dictionary $\mathcal{Z}$, context-action s_t, regularization μ, accuracy ε, budget γ
Compute the leverage score $\tilde{\tau}_t$ from $\mathcal{Z}, s_t, \mu, \varepsilon$;
Compute $\tilde{p}_t = \min\{\gamma\tilde{\tau}_t, 1\}$;
Draw $z_t \sim \mathcal{B}(\tilde{p}_t)$ and if $z_t = 1$, add s_t to $\mathcal{Z}$;
Result: Dictionary $\mathcal{Z}$

valid confidence set, which contains θ^* with high probability, provided that the projection captures well the dominating directions in the covariance operator.

Lemma 4.4.1. *Let $\delta \in (0,1)$. Define $\tilde{\beta}_{t+1}(\delta)$ as*

$$\left(\sqrt{\lambda} + \sqrt{\mu_t}\right) \|\theta^*\| + \sqrt{4\log\frac{1}{\delta} + 2\log\left(e + \frac{et\kappa^2}{\lambda}\right) d_{\text{eff}}},$$

where $\mu_t := \|(I - P_t)F_t^{1/2}\|^2$. Then, with probability at least $1 - T\delta$, for all $t \in [T]$

$$\left\|\tilde{\theta}_t - \theta^*\right\|_{\tilde{V}_t} \leq \tilde{\beta}_{t+1}(\delta). \tag{4.12}$$

The quantity μ_t controls how well the projection operator P_t captures the dominating eigen-directions of the covariance operator, and should be at most of order λ in order for the confidence bounds to be nearly as tight as for the vanilla K-UCB algorithm. The next section further discusses how this quantity is controlled with incremental Nyström projections.

4.4.2. Learning with incremental Nyström projections

We now consider specific choices of the projections P_t and subspaces $\tilde{\mathcal{H}}_t$ obtained by Nyström approximation (Williams and Seeger, 2001; Rudi et al., 2015). In particular, the spaces $\tilde{\mathcal{H}}_t$ now take the form

$$\tilde{\mathcal{H}}_t = \text{Span}\big\{\phi(s), s \in \mathcal{Z}_t\big\}, \tag{4.13}$$

where $\mathcal{Z}_t \subset \{(x_1, a_1), \dots (x_t, a_t)\}$ is a dictionary of anchor points taken from the previously observed data. Our approach consists of constructing the dictionaries $\mathcal{Z}_t$ *incrementally*, by adding new observed examples (x_t, a_t) on the fly when deemed important, so that we have $\mathcal{Z}_1 \subset \mathcal{Z}_2 \cdots \subset \mathcal{Z}_t$. We achieve this using the Kernel Online Row Sampling (KORS) algorithm of Calandriello et al. (2017a), shown in Algorithm 9, which decides whether to include a new sample $s_t = (x_t, a_t)$ by flipping a coin with probability proportional to its *leverage score* (Mahoney and Drineas, 2009). More precisely, an estimate $\tilde{\tau}$ of the leverage score that uses the state feature φ_t and parameters μ, ε is used to assess how a given state is useful to characterize the dataset. More details on the KORS algorithm are given in Appendix 4.7.

We state the following proposition of Calandriello et al. (2017a, Theorem 1, with $\varepsilon = 1/2$), which will be useful for our regret and complexity analyses.

Proposition 4.4.1. *Let $\delta > 0$, $n \geq 1$, $\mu > 0$. Then the sequence of dictionaries $\mathcal{Z}_1 \subset \mathcal{Z}_2 \subset \dots \mathcal{Z}_T$ learned by KORS with parameters $\mu > 0, \varepsilon = 1/2$ and $\gamma = 12\log(T/\delta)$ satisfies with probability $1-\delta$, $\forall t \geq 1$*

$$\|(I - P_t)F_t^{1/2}\|^2 \leq \mu \text{ and } |\mathcal{Z}_t| \leq 9d_{\text{eff}}(\mu, T)\log(2T/\delta)^2 .$$

Additionally, the algorithm runs in $\mathcal{O}(d_{\text{eff}}(\mu, T)^2)$ time and $\mathcal{O}(d_{\text{eff}}(\mu, T)^2 \log(T)^4)$ space per iteration.

This result shows that when choosing $\mu \approx \lambda$, then KORS will maintain dictionaries of size at most d_{eff} (up to log factors), while guaranteeing that the confidence bounds studied in Section 4.4.1 are nearly as good as for the case of K-UCB.

4.4.3. Implementation and complexity analysis

Here, we analyze the complexity of the algorithm and describe its practical implementation. Recall that at each round t the agent chooses an action a that maximises the UCB rule $\mu_{t,a} + \tilde{\beta}_t \sigma_{t,a}$ where we use Eq. (4.11) to reformulate the mean term $\mu_{t,a} = \langle \hat{\theta}_{t-1}, \phi(x_t, a)\rangle_{\mathcal{H}}$ and the variance term $\sigma^2_{t,a} = \|\phi(x_t, a)\|^2_{\tilde{V}_{t-1}^{-1}}$. We use the representer theorem on the projection space $\mathcal{H}_t$ to derive efficient computations of the latter two terms instead of using a kernel trick with $t \times t$ gram matrices. Indeed, in the next proposition, we prove that the two terms can be expressed with $m_t \times m_t$ matrices instead, where $m_t = |\mathcal{Z}_t|$ is the size of the dictionary at time t. We use the notations $K_{\mathcal{S}_t}(s')$ for the kernel column vector $[K(s_1, s'), \dots, K(s_t, s')]^\top$, where $\mathcal{S}_t = \{s_i\}_{i=1\dots t}$ are the past states, and $K_{\mathcal{A},\mathcal{B}}$ for the matrix of kernel evaluations $[K(s, s')]_{s\in\mathcal{A}, s'\in\mathcal{B}}$.

Proposition 4.4.2. *At any round t, by considering $s_{t,a} = (x_t, a)$, the mean and variance term of the EK-UCB rule can be expressed with:*

$$\begin{aligned}
\Gamma_t &= K_{\mathcal{Z}_{t-1}\mathcal{S}_{t-1}} Y_{t-1} \\
\Lambda_t &= \left(K_{\mathcal{Z}_{t-1}\mathcal{S}_{t-1}} K_{\mathcal{S}_{t-1}\mathcal{Z}_{t-1}} + \lambda K_{\mathcal{Z}_{t-1}\mathcal{Z}_{t-1}}\right)^{-1} \\
\tilde{\mu}_{t,a} &= K_{\mathcal{Z}_{t-1}}(s_{t,a})^\top \Lambda_t \Gamma_t \\
\Delta_{t,a} &= K_{\mathcal{Z}_{t-1}}(s_{t,a})^\top \left(\Lambda_t - \frac{1}{\lambda} K^{-1}_{\mathcal{Z}_{t-1}\mathcal{Z}_{t-1}}\right) K_{\mathcal{Z}_{t-1}}(s_{t,a}) \\
\tilde{\sigma}^2_{t,a} &= \frac{1}{\lambda} K(s_{t,a}, s_{t,a}) + \Delta_{t,a}.
\end{aligned}$$

The algorithm then runs in a space complexity of $\mathcal{O}(Tm)$ and a time complexity of $\mathcal{O}(CTm^2)$.

In our algorithm, the incremental updates of the projections allow us to derive rank-one updates of the expressions $\Lambda_t, \Gamma_t, K^{-1}_{\mathcal{Z}_t\mathcal{Z}_t}$ in all cases. First, when the dictionary does not change (i.e $P_t = P_{t-1}$), the update of the $m_t \times m_t$ matrix Λ_t can be performed with Sherman-Morrison updates, and the term $\Gamma_t = K_{\mathcal{Z}_{t-1}\mathcal{S}_{t-1}} Y_{t-1}$ can also benefit from a rank-one update given the latest reward and state. Both updates are performed in no more than $\mathcal{O}(m_t^2)$ time and space. Second, when the dictionary changes (i.e $P_t \succ P_{t-1}$), the matrix Λ_t can be updated in two stages with a rank-one update using Sherman-Morrison on the states as if the dictionary did not change, in $\mathcal{O}(m_t^2)$ time and space, and second rank-one update on the dictionary using the Schur complement in $\mathcal{O}(tm_t + m_t^2)$ time and space. Similarly, we can update $\Gamma_t = K_{\mathcal{Z}_t\mathcal{S}_{t-1}} Y_{t-1}$ with a first update on the states and stacking a block of size $1 \times t$ in $\mathcal{O}(tm_t)$

Algorithm 10: Efficient Kernel UCB

Input: T the horizon, λ regularization and exploration parameters, K(the kernel function, $\varepsilon > 0, \gamma > 0$

Initialization;

Context x_0, a_0 chosen randomly and reward r_0 ;

$\mathcal{S} = \{(x_0, a_0)\}, Y_{\mathcal{S}} = [r_0]$ $\mathcal{Z} = \{(x_0, a_0)\}$;

$\Lambda_t = \left(K_{\mathcal{Z}\mathcal{S}}K_{\mathcal{S}\mathcal{Z}} + \lambda K_{\mathcal{Z}\mathcal{Z}}\right)^{-1}$ $\Gamma_t = K_{\mathcal{Z}\mathcal{S}}Y_{\mathcal{S}}$;

for $t = 1$ *to* T **do**

  Observe context x_t ;

  Choose $\tilde{\beta}_t$ (e.g as in Lem. 4.4.1, and $\delta = \frac{1}{T^2}$) ;

  Choose $a_t \leftarrow \arg\max_{a \in \mathcal{A}} \tilde{\mu}_{t,a} + \tilde{\beta}_t \tilde{\sigma}_{t,a}$;

    $\tilde{\mu}_{t,a} \leftarrow K_{\mathcal{Z}}(s_{t,a})^\top \Lambda_t \Gamma_t$;

    $\Delta_{t,a} = K_{\mathcal{Z}}(s_{t,a})^\top \left(\Lambda_t - \frac{1}{\lambda}K^{-1}_{\mathcal{Z}\mathcal{Z}}\right) K_{\mathcal{Z}}(s_{t,a})$ $\tilde{\sigma}^2_{t,a} \leftarrow \frac{1}{\lambda}K(s_{t,a}, s_{t,a}) + \Delta_{t,a}$;

  Observe reward r_t and $s_t \leftarrow (x_t, a_t)$;

  $Y_{\mathcal{S}} \leftarrow [Y_{\mathcal{S}}, r_t]^\top, \mathcal{S} \leftarrow \mathcal{S} \cup \{s_t\}$;

  $\mathcal{Z}' \leftarrow \text{KORS}(t, \mathcal{Z}, K_{\mathcal{Z}}(s_t), \lambda, \varepsilon, \gamma)$;

  if $\mathcal{Z}' = \mathcal{Z}$ **then**

    Incremental inverse update Λ_t with s_t;

    $\Gamma_{t+1} \leftarrow \Gamma_t + r_t K_{\mathcal{Z}}(s_t)$;

  end

  else

    $z = \mathcal{Z}' \setminus \mathcal{Z}$;

    Incremental inverse update Λ_t with s_t, z ;

    Incremental inverse update $K^{-1}_{\mathcal{Z}\mathcal{Z}}$ with z;

    $\Gamma_{t+1} \leftarrow [\Gamma_t + r_t K_{\mathcal{Z}}(s_t),\ K_{\mathcal{S}}(z)^\top Y_{\mathcal{S}}]^\top$

  end

end

space and time. Eventually, the inverse of the dictionary gram matrix $K^{-1}_{\mathcal{Z}_t\mathcal{Z}_t}$ is updated with Schur complement in $\mathcal{O}(m_t^2)$. Besides, the second case when the projection is updated occurs at most m times and the first case at most T times. When the UCB rule is computed on C discrete actions or when we assume that it can be optimized using $\mathcal{O}(C)$ evaluations, given that the KORS algorithm runs in $\mathcal{O}(m^2)$ time and space, our algorithm has a total complexity of $\mathcal{O}(CTd_{\text{eff}}^2)$ in time and $\mathcal{O}(Td_{\text{eff}})$ in space, using that $m \approx d_{\text{eff}}$. Note that, as in all UCB algorithms, including ours, the theoretical value for $\tilde{\beta}_t$ in Lemma 4.4.1 is hard to estimate and often too pessimistic and leads to over-exploration, as discussed by Calandriello et al. (2020). In practice, choosing a fixed value has shown to perform well in our experiments.

In contrast, the non-incremental approach of Calandriello et al. (2020) in the BBKB algorithm needs to recompute a new dictionary about d_{eff} times. Each update involves the computation of a new covariance matrix $K_{\mathcal{Z}\mathcal{S}}K_{\mathcal{S}\mathcal{Z}}$ which costs $\mathcal{O}(tm_t^2)$ operations for its contextual variant[1], yielding an overall $\mathcal{O}(Td_{\text{eff}}^3)$ with $m \approx d_{\text{eff}}$, as illustrated in Table 4.1.

[1]The original BBKB algorithm does not involve contexts and consider a finite set of actions, allowing to compute the covariance matrix in $\mathcal{O}(\min(t, |\mathcal{A}|)m_t^2)$.

4.4.4. Regret analysis

We now analyze the regret of the EK-UCB algorithm, using Proposition 4.4.1 as well as Lemma 4.4.1.

Theorem 4.4.1. *Let $T \geq 1$ and $\theta^* \in \mathcal{H}$. Assume that $|\langle \phi(x,a), \theta^* \rangle_{\mathcal{H}}| \leq 1$ for all $a \in \bigcup_{t=1}^{T} \mathcal{A}_t \subset \mathcal{A}$ and $x \in \mathcal{X}$. Then, the EK-UCB algorithm with regularization λ along with KORS updates with parameter μ satisfies the regret bound*

$$R_T \lesssim \sqrt{T} \left(\sqrt{\frac{\mu m}{\lambda}} + \sqrt{d_{\text{eff}}} \right) \left(\|\theta^*\| (\sqrt{\lambda} + \sqrt{\mu}) + \sqrt{d_{\text{eff}}} \right),$$

where $m := |\mathcal{Z}_T|$. In particular, the choice $\mu = \lambda$ yields $m \lesssim d_{\text{eff}}$ and the bound

$$R_T \lesssim \sqrt{T} \big(\|\theta^*\| \sqrt{\lambda d_{\text{eff}}} + d_{\text{eff}} \big).$$

Furthermore, the algorithm runs in $O(Tm)$ space complexity and $O(CTm^2)$ time complexity.

The regret bound is again given up to logarithmic factors and we detail the proof as well as the precise bound in Appendix 4.7. As for K-UCB, one may analyze the resulting regret under a capacity condition, and when $\mu \approx \lambda$, we obtain the same guarantees as in Corollary 4.3.1. Note that our analysis leverages the fact that the dictionary is constructed incrementally, in particular using a condition $P_t \succeq P_{t-1}$, which yields the approximation term $\sqrt{\mu m/\lambda}$. Had we used fixed projections with some operator P, this approximation term would instead be $\sqrt{\mu/\lambda}$ with $\mu = \|(I-P)F_T^{1/2}\|^2$.

As a consequence of this theorem, the following corollary analyzes when the approximation terms dominate the regret, i.e when the dictionary size does not suffice to recover the original regret bound.

Corollary 4.4.1. *Assuming the capacity condition $d_{\text{eff}} \leq (T/\lambda)^\alpha$ for $0 \leq \alpha \leq 1$. Let $m \geq 1$, under the assumptions of Thm. 4.4.1, the regret of EK-UCB satisfies*

$$R_T \lesssim \begin{cases} Tm^{\frac{\alpha-1}{2\alpha}} \text{ if } m \leq T^{\frac{\alpha}{1+\alpha}} \\ T^{\frac{1+3\alpha}{2+2\alpha}} \text{ otherwise} \end{cases}$$

for the choice $\lambda = \mu = Tm^{-1/\alpha}$.

The proof is postponed to Appendix 4.7. In a practical setting, the dictionary size is controlled by the choice of the projection parameter μ. When μ is too high, it induces a smaller dictionary size m but thus linear regret as indicated in the previous corollary. However, by choosing a low μ, we still recover the original regret but increase the size of the dictionary and thus pay a higher computation time. To recover the original regret, the regularization parameter λ must be set to μ in all cases to recover the original regret, and both values have a theoretical optimal value which depends on the horizon to recover the best convergence rate under the capacity condition assumption.

4.5. Numerical Experiments

We now evaluate our proposed EK-UCB approach empirically on a synthetic scenario, in order to illustrate its performance in practice. All algorithms have been carefully optimized

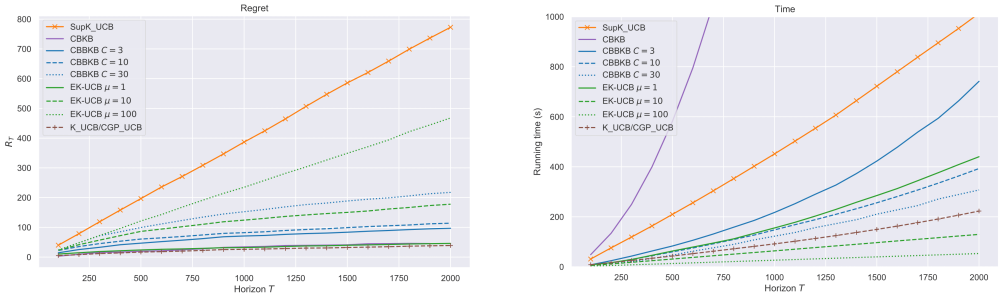

Figure 4.1: 'Bump' setting: Regret and running times of EK-UCB, CBBKB, CBKB, SupKUCB and K-UCB, with $T = 1000$ and $\lambda = 10$ (see Corollary 4.3.1 and 4.4.1). EK-UCB matches the best theoretical regret-time compromise when the projection parameter $\mu = \lambda$. We show other values of μ: higher μ ($\mu = 100$) leads to faster computational time but worse regret, and reciprocally ($\mu = 1$) leads to worse computational time and better regret. Additional results where λ and μ change simultaneously are available in the Appendix 4.7.

for fair comparisons.[2] More experimental details, discussions, and additional experimental results are provided in Appendix 4.7.

Experimental setup. We consider a 'Bump' synthetic environment with contexts uniformly distributed in $[0, 1]^p$, with $p = 5$, and actions in $[0, 1]$. The rewards are generated using the function $r(x, a) = \max(0, 1 - \|a - a^*\|_1 - \langle w^*, x - x^* \rangle_{\mathcal{H}})$ for some a^*, w^* and x^* picked randomly and fixed. We also consider additional 2D synthetic settings 'Chessboard' and 'Step Diagonal' presented in Appendix 4.7. We use a Gaussian kernel in this setting. We run our algorithms for $T = 2000$ steps and average our results over different 3 random runs.

Baselines. In our experiments, we chose to compare to K-UCB, SupK-UCB and to works which focus on improving the $\mathcal{O}(T^3)$ time-complexity for the kernel case. We implemented K-UCB, SupK-UCB (SupKernelUCB, Valko et al. (2013)), EK-UCB (our efficient version of the K-UCB algorithm) as well as our contextual adaptation of the BKB (Calandriello et al., 2019) and BBKB (Calandriello et al., 2020) algorithms; we will refer to these respectively as CBKB and CBBKB. Specifically, we use the same accumulation criteria as Calandriello et al. (2020) for the "resparsification" strategies (i.e., the resampling of the dictionary) with a threshold parameter C. We also proceed to the same sampling and equation updates as the original algorithms while using our joint kernel on context-action pairs. Note also that CGP-UCB/K-UCB only differ from their parameter β_t and match the same algorithm in our implementation (see second last paragraph in Sec. 4.3).

Results. We report the average regret and running times of the algorithms over different runs in Fig. 4.1 and Fig. 4.2 to analyze how the the different algorithms perform. In particular, our algorithm (EK-UCB) achieves low regret while running in low computational time.

In the first example for the 'Bump' environment in Fig. 4.1, for $T = 2000$, we have set $\lambda = 10$ (of the order of $\sqrt{T}$) and see that the value of $\mu = \lambda$ indeed achieves a good

[2]The code with open-source implementations for experimental reproducibility is available at `https://github.com/criteo-research/Efficient-Kernel-UCB`.

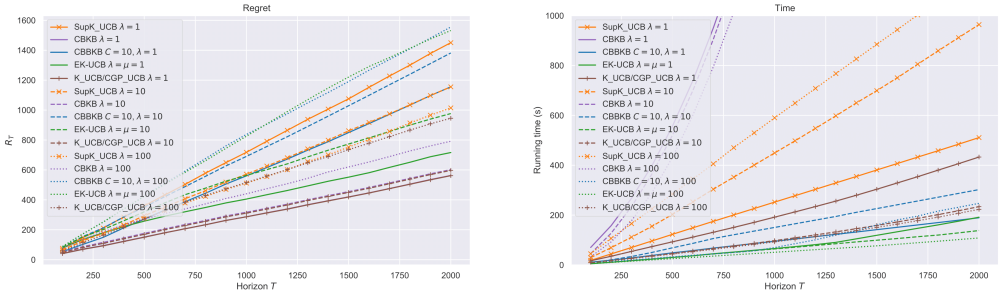

Figure 4.2: 'Chessboard' setting: Regret and running times of EK-UCB ($\lambda = \mu$), CBBKB ($C = 10$) and CBKB, with $T = 2000$ and with varying λ. We notice that low λ values have better regrets but higher computatinal times. Overall EK-UCB achieves the best regret-time compromise for all parameters of λ while CBBKB sometimes improves upon the K-UCB complexity but has both higher regret than EK-UCB and higher computational time.

tradeoff between regret and time. The parameter μ determines the quality of the projection required in the algorithm. Thus, for a smaller μ, the algorithm achieves a better regret but pays a higher time complexity. We note that a similar role is played by the parameter C in the BBKB algorithm. The smaller C, the more frequent the dictionary updates, and thus the slower is the algorithm. While the CGP-UCB/K-UCB obtains the best regret, we note also, that EK-UCB ($\mu = 1$), CBKB (which is CBBKB with $C = 1$) essentially take the full dictionary $m \approx T$ and thus also match K-UCB, but with dictionary building computational overheads which make them more computationally intensive than K-UCB itself. In the Appendix 4.7 we provide additional results that show that consistently EK-UCB provides the best time-regret compromise with regards to K-UCB.

Second, in Fig. 4.2 we show for the 'Chessboard' setting the influence of varying λ for all methods (fixing $\mu = \lambda$ for EK-UCB). Both CBBKB and EK-UCB improve upon the K-UCB computational time in this case, but EK-UCB achieves lower computational times while also having lower regrets than CBBKB for all settings. We also notice that the CBKB algorithm runs much slower than the CBBKB algorithm in all experiments, as expected due to its costly dictionary update at every round which requires processing all previous points. The computational overheads of its dictionary building therefore makes it not practical despite its theoretical guarantees. Note also that CBBKB uses scores based on the variance estimates on past states for its "resparfication" strategy and EK-UCB uses leverage scores to build its dictionary thus looking for directions that are orthogonal to the previous anchor points; both approaches are more effective than updating the dictionary at each round. Eventually, recall that our incremental projection scheme allows us to perform rank-one updates of the dictionary. This also contributes to the practical speedup of our EK-UCB algorithm, as compared to the CBBKB strategy.

Moreover, SupK-UCB performs poorly in our experiments due to its over-exploring elimination strategy that might be beneficial only for large T and makes it unpractical in its current time-complexity. Note that the main author of SupK-UCB co-authored Calandriello et al. (2019) where it is mentioned that it indeed has "tighter analysis than GP-UCB [but] does not work well in practice".

4.6. Discussions

In this work, we proposed a method for contextual kernel UCB algorithms in large-scale problems. The EK-UCB algorithm runs in $\mathcal{O}(Td_{\text{eff}})$ space and $\mathcal{O}(CTd_{\text{eff}}^2)$ time complexity, which significantly improves over the standard contextual kernel UCB method. Note that while previous efficient Gaussian process algorithms allow to scale up the learning problems in non contextual and discrete action environments, we have shown how the incremental projection updates were crucial to perform efficient approximations in the joint context-action space, providing the same regret guarantees for a smaller computational cost. We note that the batching strategy of BBKB may still be useful even under our incremental updates, and thus provides an interesting avenue for future work. Another natural question is whether we may obtain algorithms with better regret guarantees similar to Valko et al. (2013) in the finite action case, while also achieving gains in computational efficiency as in our work.

4.7. Appendices

This appendix is organized as follows:

- Appendix 4.7: notations for the analysis
- Appendix 4.7: proofs of Section 4.3 – Kernel-UCB
- Appendix 4.7: proofs of Section 4.4 – Efficient Kernel-UCB
- Appendix 4.7: details on the implementation of the algorithms
- Appendix 4.7: additional experiment details, discussions and results

4.8. Notations

Below are notations related to the sequential setting. Here, $t \in [T]$ denotes the index of the round:

- $s_t := (x_t, a_t) \in \mathcal{X} \times \mathcal{A}$ is a state at round t
- $\mathcal{S}_t := \{s_1, \dots, s_t\}$ denotes the history
- $\varepsilon_1, \dots, \varepsilon_T$ are independent centered sub-Gaussian noise
- $H_t := (\varepsilon_1, \dots, \varepsilon_t)^\top$ is the vector of noises up to round t
- $\mathcal{F}_t := \sigma(\varepsilon_1, \dots, \varepsilon_t)$ is the natural filtration with respect to $(\varepsilon_i)_{i \geq 1}$
- $r_t := \langle \theta^*, \phi(x_t, a_t) \rangle_{\mathcal{H}} + \varepsilon_t$ is the reward
- $Y_t := (r_1, \dots, r_t)^\top \in \mathbb{R}^t$ is the vector of rewards
- $\varphi_t := \phi(x_t, a_t) \in \mathcal{H}$

Below are notations related to the RKHS. Here, $t \in [T]$ denotes the index of the round:

- $F_t := \sum_{s=1}^t \varphi_s \otimes \varphi_s$ is the covariance operator
- $V_t := \sum_{s=1}^t \varphi_s \otimes \varphi_s + \lambda I : \mathcal{H} \to \mathcal{H}$ is the regularized covariance operator
- $\Phi_t : \mathcal{H} \to \mathbb{R}^t$ is the operator such that $[\Phi_t \varphi]_i = \varphi(x_i, a_i) = \langle \varphi, \phi(x_i, a_i) \rangle_{\mathcal{H}}$ for any $\varphi \in \mathcal{H}$ and $i \in [t]$
- Φ^* denotes the conjugate transpose of a linear operator Φ on $\mathcal{H}$
- $K_t := \Phi_t \Phi_t^* : \mathbb{R}^t \to \mathbb{R}^t$ is the kernel matrix at time $t \geq 1$. Note that $[K_t]_{ij} = K((x_i, a_i), (x_j, a_j))$.
- $\lambda_i(K_t)$ is the i-th largest eigenvalue of K_t
- $d_{\text{eff}}(\lambda, t) := \operatorname{Tr}(K_t(K_t + \lambda I_t)^{-1})$ is the effective dimension of the matrix K_t

Below are notations related to the Kernel-UCB algorithm without projections:

- $\hat{\theta}_t := V_t^{-1} \Phi_t^* Y_t$ is the estimator of the algorithm
- $\delta > 0$ is the confidence level
- $\beta_t(\delta)$ is the radius of the confidence ellipsoid of the algorithm
- $\mathcal{C}_t := \left\{ \theta \in \mathcal{H} : \left\| \theta - \hat{\theta}_{t-1} \right\|_{V_{t-1}} \leq \beta_t(\delta) \right\}$ is the confidence ellipsoid played by the algorithm

Below are notations related to the Kernel-UCB algorithm with projections. All along the analysis, the notation $\tilde{x}$ corresponds to the projected version of the object x.

- $\mathcal{Z}_t \subset \left\{ (x_1, a_1), \dots, (x_t, a_t) \right\}$ is a dictionary
- $\tilde{\mathcal{H}}_t := \operatorname{Span}\left\{ \phi(s), s \in \mathcal{Z}_t \right\}$ is a linear subspace of $\mathcal{H}$ and is used at round t.

- $P_t : \mathcal{H} \to \tilde{\mathcal{H}}_t$ is the Euclidean projection onto $\mathcal{H}$ so that $\tilde{\mathcal{H}}_t = \{P_t\varphi, \varphi \in \mathcal{H}\}$
- $\tilde{V}_t := \sum_{s=1}^{t}(P_t\varphi_s) \otimes (P_t\varphi_s) + \lambda I = P_t F_t P_t + \lambda I$ is the regularized projected covariance operator
- $\tilde{\theta}_t := P_t\tilde{V}_t^{-1}P_t\Phi_t^* Y_t$ is the projected estimator of the algorithm
- $\tilde{C}_t := \left\{\theta \in \mathcal{H} : \|\theta - \tilde{\theta}_{t-1}\|_{\tilde{V}_{t-1}} \leq \tilde{\beta}_t(\delta)\right\}$ is the confidence ellipsoid related to the projected estimator
- $\mu_t := \left\|(I - P_t)F_t^{1/2}\right\|^2$ is the approximation error of the projection

Eventually, we provide notations related to the kernel matrix computations when we write the update rules of the efficient algorithm.

- $K_{\mathcal{S}}(s')$ is the kernel column vector $[K(s_1, s'), \ldots, K(s_l, s')]^\top$ of size $|\mathcal{S}| = l$. Note that $K_{\mathcal{S}_t}(s) = \Phi_t\phi(s)$.
- $K_{\mathcal{Z}\mathcal{S}}$ is the kernel matrix vector $[K(z, s)]_{z\in\mathcal{Z}, s\in\mathcal{S}}$ of size $|\mathcal{Z}| \times |\mathcal{S}|$.
- $s_{t,a} = (x_t, a)$ refers to the pair of context x_t and any action $a \in \mathcal{A}_t$ that can be chosen in the UCB rule.

4.9. Proofs of Section 4.3: Kernel UCB

In this appendix we prove of Lemma 4.3.1 and Theorem 4.3.1.

4.9.1. Proof of Lemma 4.3.1

We first prove Lemma 4.3.1, which controls the size of the confidence intervals considered by the algorithm. It states that with probability $1 - \delta$, for all $t \geq 1$:

$$\theta^* \in C_t, \quad \text{where} \quad C_t = \left\{\theta \in \mathbb{R}^d, \|\theta - \hat{\theta}_{t-1}\|_{V_{t-1}} \leq \beta_t(\delta)\right\}. \tag{4.14}$$

Lemma 4.3.1. *Let $\delta \in (0, 1)$. Assume $\kappa^2 \geq \sup_{s\in\mathcal{X}\times\mathcal{A}} K(s, s)$. Then with probability at least $1 - T\delta$, for all $t \in [T]$*

$$\begin{aligned}\|\hat{\theta}_t - \theta^*\|_{V_t} &\leq \sqrt{\lambda}\|\theta^*\| + \sqrt{2\log\frac{1}{\delta} + \log\left(\det\left(\frac{1}{\lambda}(K_t + \lambda I)\right)\right)} \\ &\leq \sqrt{\lambda}\|\theta^*\| + \sqrt{2\log\frac{1}{\delta} + \log\left(e + \frac{et\kappa^2}{\lambda}\right) d_{\text{eff}}(\lambda, T)} \quad =: \quad \beta_{t+1}(\delta).\end{aligned}$$

Proof. The analysis is inspired by the one of Abbasi-yadkori et al. (2011) for linear bandits and uses inequality tails on vector valued martingales. We introduce $M_t = \sum_{s=1}^{t}\varphi_s\varepsilon_s \in \mathcal{H}$, which is a martingale with regards to the natural filtration $\mathcal{F}_t := \sigma(\varepsilon_1, \ldots, \varepsilon_t)$. Solving the least-square optimization problem (4.4), $\hat{\theta}_t$ equals

$$\hat{\theta}_t = V_t^{-1}\sum_{s=1}^{t}\varphi_s Y_s = V_t^{-1}\sum_{s=1}^{t}\varphi_s(\varphi_s^\top\theta^* + \varepsilon_s) = V_t^{-1}((V_t - \lambda I_d)\theta^* + M_t) = \theta^* - \lambda V_t^{-1}\theta^* + V_t^{-1}M_t\,.$$

Multiplying by the square root of V_t and using the triangle inequality

$$\left\|V_t^{1/2}(\hat{\theta}_t - \theta^*)\right\| = \left\| - \lambda V_t^{-1/2}\theta^* + V_t^{-1/2}M_t\right\| \leq \lambda\left\|V_t^{-1/2}\theta^*\right\| + \left\|V_t^{-1/2}M_t\right\|.$$

On the other hand, given that $V_t = F_t + \lambda I$ where F_t is positive semi-definite, $V_t^{-1/2} \preccurlyeq \lambda^{-1/2} I$ and thus

$$\lambda \|V_t^{-1/2}\theta^*\| \leq \lambda \frac{1}{\sqrt{\lambda}} \|\theta^*\| = \sqrt{\lambda}\|\theta^*\| \,.$$

We now prove for the other term that with probability at least $1-\delta$

$$\|V_t^{-1/2} M_t\| \leq \sqrt{2\log\frac{1}{\delta} + \log\det\frac{1}{\lambda}(K_t + \lambda I)} \,.$$

Step 1: Martingales For all $\nu \in \mathcal{H}$, we define the random-variable

$$S_{t,\nu} = \exp\Big(\nu^\top M_t - \frac{1}{2}\nu^\top V_t \nu\Big)$$

and now show that it is a $\mathcal{F}_t$-super-martingale. First, note that the common distribution of the $\varepsilon_1, \ldots, \varepsilon_t$ is 1-sub Gaussian, i.e., for all $\mathcal{F}_{t-1}$-measurable real-valued random variable ν_{t-1}, we have

$$\mathbb{E}\left[\exp(\nu_{t-1}\varepsilon_t)|\mathcal{F}_{t-1}\right] \leq \exp\Big(\frac{\nu_{t-1}^2}{2}\Big) \,. \tag{4.15}$$

Thus, using that $M_t = M_{t-1} + \varphi_t \varepsilon_t$ and $V_t = V_{t-1} + \varphi_t \otimes \varphi_t$,

$$\begin{aligned}
\mathbb{E}\left[S_{t,\nu}|\mathcal{F}_{t-1}\right] &= \mathbb{E}\left[\exp\big(\nu^\top M_t - \frac{1}{2}\nu^\top V_t\nu\big)|\mathcal{F}_{t-1}\right] \\
&= \mathbb{E}\left[S_{t-1,\nu}\exp\big(\nu^\top\varphi_t\varepsilon_t - \frac{1}{2}\nu^\top(\varphi_t\otimes\varphi_t)\nu\big)|\mathcal{F}_{t-1}\right] \\
&= S_{t-1,\nu}\,\mathbb{E}\left[\exp\big(\nu^\top\varphi_t\varepsilon_t - \frac{1}{2}(\nu^\top\varphi_t)^2\big)\big|\mathcal{F}_{t-1}\right] \leq S_{t-1,\nu} \,,
\end{aligned}$$

where the last inequality is by applying (4.15) with $\nu_{t-1} = \nu^\top\varphi_t$ since $\varphi_t = \phi(x_t, a_t)$ is $\mathcal{F}_{t-1}$-measurable. Therefore, $S_{t,\nu}$ is a $\mathcal{F}_t$-super-martingale for any $\nu \in \mathcal{H}$, and

$$\mathbb{E}\left[S_{t,\nu}\right] \leq \mathbb{E}\left[S_{0,\nu}\right] = \exp\Big(-\frac{\lambda}{2}\|\nu\|^2\Big) \,. \tag{4.16}$$

Rewriting $S_{t,\nu}$ in its vertex form with $m = V_{t-1}M_t$ yields

$$S_{t,\nu} = \exp\left(-\frac{1}{2}(\nu-m)^\top V_t(\nu-m)\right) \times \exp\Big(\frac{1}{2}\big\|V_t^{-1/2}M_t\big\|^2\Big) \,,$$

which substituted into (4.16) entails

$$\mathbb{E}\left[\exp\Big(-\frac{1}{2}(\nu-m)^\top V_t(\nu-m)\Big) \times \exp\Big(\frac{1}{2}\big\|V_t^{-1/2}M_t\big\|^2\Big)\right] \leq \exp\Big(-\frac{\lambda}{2}\|\nu\|^2\Big), \qquad \forall \nu \in \mathcal{H} \,. \tag{4.17}$$

Step 2: Laplace's method integrating

Now, following Laplace's method which is standard for the proof of LinUCB, the goal is to integrate both sides of the above expression. Let us first rewrite it in order to consider finite dimensional objects thanks to the Kernel trick.

Recalling $V_t := \Phi_t^*\Phi_t + \lambda I$ and $K_t := \Phi_t\Phi_t^*$, following (Valko et al., 2013), we will use the following identities:

$$(\Phi_t^*\Phi_t + \lambda I)\Phi_t^* = \Phi_t^*(\Phi_t\Phi_t^* + \lambda I) \tag{4.18}$$

$$\Rightarrow \quad V_t\Phi_t^* = \Phi_t^*(K_t + \lambda I) \tag{4.19}$$

$$\Rightarrow \quad \Phi_t^*(K_t + \lambda I)^{-1} = V_t^{-1}\Phi_t^*\,. \tag{4.20}$$

Let $x \in \mathbb{R}^t$ and write $\nu = V_t^{-1}\Phi_t^* x \in \mathcal{H}$ and recall that $m = V_t^{-1}M_t = V_t^{-1}\Phi_t^* H_t$, where $H_t = (\varepsilon_1, \dots, \varepsilon_t)^\top$. We have

$$\begin{aligned}
\exp\Big(-\frac{1}{2}(\nu - m)^\top V_t(\nu - m)\Big) &= \exp\Big(-\frac{1}{2}(x - H_t)^\top \Phi_t V_t^{-1} V_t V_t^{-1}\Phi_t^*(x - H_t)\Big) \\
&= \exp\Big(-\frac{1}{2}(x - H_t)^\top \Phi_t\Phi_t^*(K_t + \lambda I)^{-1}(x - H_t)\Big) \qquad \leftarrow \text{ by (4.20)} \\
&= \exp\Big(-\frac{1}{2}(x - H_t)^\top K_t(K_t + \lambda I)^{-1}(x - H_t)\Big) \qquad \leftarrow K_t = \Phi_t\Phi_t^* \\
&= \exp\Big(-\frac{1}{2}(x - H_t)^\top K_t^{1/2}(K_t + \lambda I)^{-1}K_t^{1/2}(x - H_t)\Big)\,, \qquad (4.21)
\end{aligned}$$

where the last equality is because $(K_t + \lambda I)^{-1}$ and $K_t^{1/2}$ commute. Similarly,

$$\exp\Big(-\frac{\lambda}{2}\|\nu\|^2\Big) = \exp\Big(-\frac{\lambda}{2}x^\top K_t^{1/2}(K_t + \lambda I)^{-2}K_t^{1/2}x\Big)\,.$$

Combining with (4.16) and (4.21) thus gives for any $x \in \mathbb{R}^t$,

$$\begin{aligned}
\mathbb{E}\left[\exp\Big(-\frac{1}{2}(x - H_t)^\top K_t^{1/2}(K_t + \lambda I)^{-1}K_t^{1/2}(x - H_t)\Big) \times \exp\Big(\frac{1}{2}\big\|V_t^{-1/2}M_t\big\|^2\Big)\right] \\
\leq \exp\Big(-\frac{\lambda}{2}x^\top K_t^{1/2}(K_t + \lambda I)^{-2}K_t^{1/2}x\Big)\,. \quad (4.22)
\end{aligned}$$

Now, that we are back to finite dimensional space, the idea would consists in integrating both parts over $x \in \mathbb{R}^t$. But the matrix K_t may be non-invertible, we thus need a few more steps to integrate over $\mathrm{Im}(K_t)$ only.

Let $d_t = \mathrm{rank}(K_t)$ and $Q_t \in \mathbb{R}^{t\times d_t}$ the matrix formed by the orthonormal eigenvectors of K_t with non-zero eigenvalues. Let $u \in \mathbb{R}^{d_t}$ then $Q_t u \in \mathrm{Im}(K_t)$ and there exists $x \in \mathbb{R}^t$ such that $K_t^{1/2}x = Q_t u$. Defining $z \in \mathbb{R}^{d_t}$ such that $Q_t z = K_t^{1/2}H_t$ and substituting into Inequality (4.22) yields, for any $u \in \mathbb{R}^{d_t}$

$$\begin{aligned}
\mathbb{E}\left[\exp\Big(-\frac{1}{2}(u - z)^\top Q_t^\top(K_t + \lambda I)^{-1}Q_t(u - z)\Big) \times \exp\Big(\frac{1}{2}\big\|V_t^{-1/2}M_t\big\|^2\Big)\right] \\
\leq \exp\Big(-\frac{\lambda}{2}u^\top Q_t^\top(K_t + \lambda I)^{-2}Q_t u\Big)\,. \quad (4.23)
\end{aligned}$$

Now, we integrate both sides over $u \in \mathbb{R}^{d_t}$, recognizing a multidimensional Gaussian density,

we have

$$\int_{\mathbb{R}^d} \exp\Big(-\frac{1}{2}(u-z)^\top Q_t^\top (K_t+\lambda I)^{-1} Q_t (u-z)\Big) d\mu(u) = \sqrt{\det\big(2\pi (Q_t^\top (K_t+\lambda I)^{-1} Q_t)^{-1}\big)} \\ = \sqrt{(2\pi)^{d_t} \prod_{i=1}^{d_t} \big(\lambda_i(K_t)+\lambda\big)}\,,$$

where $\lambda_i(K_t)$ is the i-th largest eigenvalue of K_t. Similarly

$$\int_{\mathbb{R}^d} \exp\Big(-\frac{\lambda}{2} u^\top Q_t^\top (K_t+\lambda I)^{-2} Q_t u\Big) d\mu(u) = \sqrt{\det\Big(2\pi\lambda^{-1}\big(Q_t^\top (K_t+\lambda I)^{-2} Q_t\big)^{-1}\Big)} \\ = \sqrt{\Big(\frac{2\pi}{\lambda}\Big)^{d_t} \prod_{i=1}^{d_t} \big(\lambda_i(K_t)+\lambda\big)^2}.$$

Therefore, by the Fubini-Tonelli theorem, plugging the last two equations into Inequality (4.23) entails

$$\sqrt{(2\pi)^{d_t} \prod_{i=1}^{d_t} \big(\lambda_i(K_t)+\lambda\big)}\, \mathbb{E}\left[\exp\Big(\frac{1}{2}\|V_t^{-1/2} M_t\|^2\Big)\right] \le \sqrt{\Big(\frac{2\pi}{\lambda}\Big)^{d_t} \prod_{i=1}^{d_t} \big(\lambda_i(K_t)+\lambda\big)^2}\,,$$

which, after reorganizing the terms, yields

$$\mathbb{E}\left[\exp\Big(\frac{1}{2}\|V_t^{-1/2} M_t\|^2\Big)\right] \le \sqrt{\prod_{i=1}^{d_t} \Big(1+\frac{\lambda_i(K_t)}{\lambda}\Big)} = \sqrt{\frac{\det(K_t+\lambda I)}{\lambda^t}}\,.$$

Step 3: Markov-Chernov bound. It remains to upper-bound the above expectation using concentration inequalities. For $u > 0$,

$$P\Big(\|V_t^{-1/2} M_t\| > u\Big) = P\left(\frac{\|V_t^{-1/2} M_t\|^2}{2} > \frac{u^2}{2}\right) \le \exp\Big(-\frac{1}{2}u^2\Big)\, \mathbb{E}\left[\exp\Big(\frac{1}{2}\|V_t^{-1/2} M_t\|^2\Big)\right] \\ \le \exp\left(-\frac{u^2}{2} + \frac{1}{2}\log\frac{\det(K_t+\lambda I)}{\lambda^t}\right) = \delta \tag{4.24}$$

for the claimed choice

$$u = \sqrt{2\log\frac{1}{\delta} + \log\det\frac{1}{\lambda}(K_t+\lambda I)}\,.$$

The proof then concludes by using Prop. 4.3.1 on the $\log\det\frac{1}{\lambda}(K_t+\lambda I)$ term and by applying a union bound. □

4.9.2. Proof of Theorem 4.3.1

We are now ready to prove Theorem 4.3.1, which upper-bounds the regret of K-UCB.

Theorem 4.3.1. *Let $T \geq 2$ and $\theta^* \in \mathcal{H}$. Assume that $|\langle \phi(x,a), \theta^* \rangle_{\mathcal{H}}| \leq 1$ and $\|\phi(x,a)\| \leq \kappa$ for all $a \in \bigcup_{t=1}^{T} \mathcal{A}_t \subset \mathcal{A}$ and $x \in \mathcal{X}$. Then, the K-UCB rule defined in Eq. (4.3) for the choice $\mathcal{C}_t$ as in (4.5) satisfies the pseudo-regret bound*

$$R_T \leq 2 + 2\sqrt{T\left(\log\left(e + \frac{eT\kappa^2}{\lambda}\right) d_{\mathrm{eff}}(\lambda, T)\right)} \left[\sqrt{\lambda}\|\theta^*\| + \sqrt{2\log(T) + \log\left(e + \frac{eT\kappa^2}{\lambda}\right) d_{\mathrm{eff}}(\lambda)}\right]$$
$$\lesssim \sqrt{T}\Big(\|\theta^*\|\sqrt{\lambda d_{\mathrm{eff}}(\lambda, T)} + d_{\mathrm{eff}}(\lambda, T)\Big)\,.$$

Proof. Let $\delta \in (0, 1/2)$. By Lemma 4.3.1, with probability $1 - T\delta$,

$$\forall t \in [T], \quad \theta^* \in C_t\,. \tag{4.25}$$

Step 1: Small instantaneous regrets under the event (4.25). Assume that (4.25) holds. Let

$$a_t^* := \max_{a \in \mathcal{A}_t} \langle \phi(x_t, a), \theta^* \rangle_{\mathcal{H}} \quad \text{and} \quad \Delta_t := \langle \phi(x_t, a_t^*) - \phi(x_t, a_t), \theta^* \rangle_{\mathcal{H}}$$

be respectively the optimal decision and the instantaneous regret at round t. We also define

$$\rho_t \in \arg\max_{\theta \in C_t} \left\{ \langle \phi(x_t, a_t), \theta \rangle_{\mathcal{H}} \right\}.$$

Since $\theta^* \in C_t$, we have

$$\langle \phi(x_t, a_t^*), \theta^* \rangle_{\mathcal{H}} \leq \max_{\theta \in C_t} \{\langle \phi(x_t, a_t^*), \theta \rangle_{\mathcal{H}}\} = \text{K-UCB}_t(a_t^*)$$
$$\leq \text{K-UCB}_t(a_t) = \max_{\theta \in C_t} \{\langle \phi(x_t, a_t), \theta \rangle_{\mathcal{H}}\} = \langle \phi(x_t, a_t), \rho_t \rangle_{\mathcal{H}}\,,$$

which entails because θ^* and $\tilde{\theta}_{t-1}$ belong to C_t,

$$\Delta_t = \langle \phi(x_t, a_t^*) - \phi(x_t, a_t), \theta^* \rangle_{\mathcal{H}} \leq \langle \phi(x_t, a_t), \rho_t - \theta^* \rangle_{\mathcal{H}}$$
$$\leq \|\phi(x_t, a_t)\|_{V_{t-1}^{-1}} \|\rho_t - \theta^*\|_{V_{t-1}} \leq 2\|\phi(x_t, a_t)\|_{V_{t-1}^{-1}} \beta_t(\delta)\,.$$

Recall that $\varphi_t := \phi(x_t, a_t)$. Then, summing over $t = 1, \ldots, T$ and using that by assumption

$$|\Delta_t| \leq \left|\langle \phi(x_t, a_t^*), \theta^* \rangle_{\mathcal{H}}\right| + \left|\langle \phi(x_t, a_t), \theta^* \rangle_{\mathcal{H}}\right| \leq 2 \sup_{x \in \mathcal{X}, a \in \mathcal{A}_t} |\langle \phi(x, a), \theta^* \rangle_{\mathcal{H}}| \leq 2\,,$$

we can write the cumulative regret as

$$\begin{aligned}
\sum_{t=1}^{T} \Delta_t &\leq \sqrt{T \sum_{t=1}^{T} \Delta_t^2} && \leftarrow \quad \text{Cauchy-Schwartz's inequality} \\
&\leq 2\sqrt{T \sum_{t=1}^{T} \min\{\|\varphi_t\|^2_{V_{t-1}^{-1}} \beta_t(\delta)^2, 1\}} \\
&\leq 2\beta_T(\delta)\sqrt{T \sum_{t=1}^{T} \min\{\|\varphi_t\|^2_{V_{t-1}^{-1}}, 1\}} && \leftarrow \quad 1 \leq \beta_t(\delta) \leq \beta_T(\delta) \\
&\leq 2\beta_T(\delta)\sqrt{T \sum_{t=1}^{T} \log\left(1 + \|\varphi_t\|^2_{V_{t-1}^{-1}}\right)} && \leftarrow \quad \min(u, 1) \leq 2\log(1+u),\ \forall u > 0\,.
\end{aligned} \tag{4.26}$$

Now we will use the kernel trick to obtain a formulation of $\varphi_t^\top V_{t-1}^{-1}\varphi_t$ using gram matrices. Define $s_t := (x_t, a_t)$ and $\mathcal{S}_t := (s_i)_{1\leq i\leq t}$ the historical data. For any $l \geq 1$ and $\mathcal{S} \in (\mathcal{X}\times\mathcal{A})^l$, we also denote by $K_{\mathcal{S}}(s')$ the kernel column vector $[K(s_1, s'), \ldots, K(s_l, s')]^\top$ of size $|\mathcal{S}| = l$. Specifically, we have $K_{\mathcal{S}_{t-1}}(s_t) := [K(s_1, s_t), \ldots, K(s_{t-1}, s_t)]^\top = \Phi_{t-1}\varphi_t \in \mathbb{R}^t$. When multiplying $V_{t-1} := \Phi_{t-1}^*\Phi_{t-1} + \lambda I$ by φ_t on the right, we can express

$$\begin{aligned}
& V_{t-1}\varphi_t = \Phi_{t-1}^* K_{\mathcal{S}_{t-1}}(s_t) + \lambda\varphi_t, \\
\Rightarrow \quad & \varphi_t = V_{t-1}^{-1}\Phi_{t-1}^* K_{\mathcal{S}_{t-1}}(s_t) + \lambda V_{t-1}^{-1}\varphi_t \\
\Rightarrow \quad & \varphi_t = \Phi_{t-1}^*(K_{t-1} + \lambda I)^{-1} K_{\mathcal{S}_{t-1}}(s_t) + \lambda V_{t-1}^{-1}\varphi_t\,,
\end{aligned}$$

where the last equation is by Eq. (4.20). Thus, multiplying now by $\varphi_t^\top$ on the left and using $\varphi_t^\top\Phi_{t-1}^* = K_{\mathcal{S}_{t-1}}(s_t)$ entails

$$\varphi_t^\top\varphi_t = K_{\mathcal{S}_{t-1}}(s_t)^\top (K_{t-1} + \lambda I)^{-1} K_{\mathcal{S}_{t-1}}(s_t) + \lambda\varphi_t^\top V_{t-1}^{-1}\varphi_t\,.$$

Therefore, reorganizing the terms ans recognizing $\|\varphi_t\|^2_{V_{t-1}^{-1}} = \varphi_t^\top V_{t-1}^{-1}\varphi_t$ and $K(s_t, s_t) = \varphi_t^\top\varphi_t$, we can write

$$\begin{aligned}
1+\|\varphi_t\|^2_{V_{t-1}^{-1}} &= 1 + \varphi_t^\top V_{t-1}^{-1}\varphi_t \\
&= \frac{\lambda + K(s_t,s_t)}{\lambda} - \frac{1}{\lambda} K_{\mathcal{S}_{t-1}}(s_t)^\top (K_{t-1} + \lambda I)^{-1} K_{\mathcal{S}_{t-1}}(s_t) \\
&= \frac{\lambda + K(s_t,s_t)}{\lambda}\left(1 - K_{\mathcal{S}_{t-1}}(s_t)^\top (K_{t-1} + \lambda I)^{-1} K_{\mathcal{S}_{t-1}}(s_t)\big(\lambda + K(s_t,s_t)\big)^{-1}\right) \\
&= \frac{\lambda + K(s_t,s_t)}{\lambda}\det\left(1 - K_{\mathcal{S}_{t-1}}(s_t)^\top (K_{t-1} + \lambda I)^{-1} K_{\mathcal{S}_{t-1}}(s_t)\big(\lambda + K(s_t,s_t)\big)^{-1}\right) \\
&= \frac{\lambda + K(s_t,s_t)}{\lambda}\det\left(I - (K_{t-1} + \lambda I)^{-1/2} K_{\mathcal{S}_{t-1}}(s_t)\big(\lambda + K(s_t,s_t)\big)^{-1} K_{\mathcal{S}_{t-1}}(s_t)^\top (K_{t-1} + \lambda I)^{-1/2}\right),
\end{aligned}$$

where the last equality follows by the matrix determinant lemma $\det(I + AB^\top) = \det(I + B^\top A)$

if A and B are n-by-m matrices. Then, $1 + \|\varphi_t\|^2_{V_{t-1}^{-1}}$ equals

$$\frac{\lambda + K(s_t,s_t)}{\lambda} \det\left((K_{t-1}+\lambda I)^{-1/2}\Big(K_{t-1}+\lambda I - K_{\mathcal{S}_{t-1}}(s_t)\big(\lambda + K(s_t,s_t)\big)^{-1}K_{\mathcal{S}_{t-1}}(s_t)^\top\Big)(K_{t-1}+\lambda I)^{-1/2}\right)$$
$$= \frac{\lambda + K(s_t,s_t)}{\lambda} \frac{\det\left(K_{t-1} + \lambda I - K_{\mathcal{S}_{t-1}}(s_t)\big(\lambda + K(s_t,s_t)\big)^{-1}K_{\mathcal{S}_{t-1}}(s_t)^\top\right)}{\det(K_{t-1}+\lambda I)}\,.$$

Now, using that

$$K_t + \lambda I = \begin{bmatrix} K_{t-1} + \lambda I & K_{S_{t-1}}(s_t) \\ K_{S_{t-1}}(s_t)^\top & K(s_t,s_t) + \lambda \end{bmatrix},$$

by the block matrix determinant formula

$$\det\big(K_t + \lambda I\big) = (K(s_t,s_t)+\lambda)\det\Big(K_{t-1} + \lambda I - K_{S_{t-1}}(s_t)(K(s_t,s_t)+\lambda)^{-1}K_{S_{t-1}}(s_t)^\top\Big)$$

we finally get

$$1 + \|\varphi_t\|^2_{V_{t-1}^{-1}} = \frac{1}{\lambda}\frac{\det(K_t+\lambda I)}{\det(K_{t-1}+\lambda I)}\,. \tag{4.27}$$

Note here that contrary to the proof in Lattimore and Szepesvári (2020), we used here computations using the gram matrix K_t instead of the V_t which lives in the feature space that can be infinite dimensional.

Taking the $\log$ and summing over $t = 1, \dots, T$ telescopes

$$\sum_{t=1}^{T} \log\left(1 + \|\varphi_t\|^2_{V_{t-1}^{-1}}\right) = \log\left(\det\left(\frac{1}{\lambda}(K_t + \lambda I)\right)\right) \le \log\left(e + \frac{eT\kappa^2}{\lambda}\right) d_{\text{eff}}(\lambda, T)\,,$$

where we used the Proposition 4.3.1 for the last inequality and that . Substituting into the regret bound (4.26) together with $\beta_T(\delta) \le \beta_{T+1}(\delta)$ entails with probability at least $1 - T\delta$

$$\sum_{t=1}^{T} \Delta_t \le 2\beta_{T+1}(\delta)\sqrt{T\left(e + \frac{eT\kappa^2}{\lambda}\right) d_{\text{eff}}(\lambda, T)}\,.$$

Choosing $\delta = 1/T^2$, taking the expectation $R_T = \mathbb{E}\left[\sum_{t=1}^{T}\Delta_t\right]$ and using $|\Delta_t| \le 2$ concludes. □

We now provide a proof for the Corollary that gives out the convergence speed of the K-UCB algorithm with the capacity condition assumption.

4.9.3. Proof of Corollary 4.3.1

Corollary 4.3.1. *Assuming the capacity condition $d_{\text{eff}} \le (T/\lambda)^\alpha$ for $0 \le \alpha \le 1$, the regret of K-UCB is bounded as $R_T \lesssim T^{\frac{1+3\alpha}{2+2\alpha}}$ with an optimal $\lambda \approx T^{\frac{\alpha}{1+\alpha}}$.*

Proof. Starting from $R_T \lesssim \sqrt{T}\Big(\sqrt{\lambda d_{\text{eff}}(\lambda)} + d_{\text{eff}}(\lambda)\Big)$ and assuming the capacity condition $d_{\text{eff}}(\lambda) \lesssim \left(\frac{T}{\lambda}\right)^{\alpha}$ for some $\alpha \in (0,1)$,

$$R_T \lesssim \sqrt{T}\Big(\sqrt{T^{\alpha}\lambda^{1-\alpha}} + T^{\alpha}\lambda^{-\alpha}\Big)\,.$$

Minimizing in $\lambda > 0$ entails

$$\sqrt{T^{\alpha}\lambda^{1-\alpha}} = T^{\alpha}\lambda^{-\alpha} \quad \Rightarrow \quad \lambda^* = T^{\frac{\alpha}{1+\alpha}}\,,$$

which yields for $\lambda = \lambda^*$

$$R_T \lesssim T^{\frac{1}{2}+\alpha-\frac{\alpha^2}{1+\alpha}} = T^{\frac{1+3\alpha}{2+2\alpha}}\,.$$

□

4.10. Proofs of Section 4.4: Efficient Kernel-UCB

Let us start by recalling the setting and the notation of this section. Let $\mathcal{Z}_t \subseteq \mathcal{S}_t$, $\tilde{\mathcal{H}}_t := \mathrm{Span}\big\{\phi(s), s \in \mathcal{Z}_t\big\}$ be the corresponding linear subspace of $\mathcal{H}$, and $P_t : \mathcal{H} \to \tilde{\mathcal{H}}_t$ be the Euclidean projection onto $\mathcal{H}$ so that $\tilde{\mathcal{H}}_t = \{P_t\varphi, \varphi \in \mathcal{H}\}$. The EK-UCB algorithm also builds an estimator

$$\tilde{\theta}_{t-1} \in \operatorname*{arg\,min}_{\theta \in \tilde{\mathcal{H}}_{t-1}} \left\{ \sum_{s=1}^{t-1} \left(\langle \theta, \phi(x_s, a_s)\rangle_{\mathcal{H}} - r_s\right)^2 + \lambda\|\theta\|^2 \right\} \in \tilde{\mathcal{H}}_{t-1}\,, \tag{4.28}$$

and uses the confidence set $\tilde{C}_t := \big\{\theta \in \mathcal{H} : \|\theta - \tilde{\theta}_{t-1}\|_{\tilde{V}_{t-1}} \leq \tilde{\beta}_t(\delta)\big\}$. We define

$$\tilde{V}_t := \sum_{s=1}^{t} (P_t\varphi_s) \otimes (P_t\varphi_s) + \lambda I$$

,

that we rewrite $\tilde{V}_t = P_t F_t P_t + \lambda I$ where $\Phi_t^* = [\varphi_1, \ldots, \varphi_t]$ and $F_t = \Phi_t^*\Phi_t$. Recalling the notation, $Y_t := (r_1, \ldots, r_t)^\top$, we then obtain that $\tilde{\theta}_t = P_t\tilde{V}_t^{-1}P_t\Phi_t^*Y_t$. We recall the definition $\mu_t := \left\|(I - P_t)F_t^{1/2}\right\|^2$.

4.10.1. Proof of Lemma 4.4.1

The following lemma serves to compute the distance of the center $\tilde{\theta}_t$ to any point in the ellipsoid in the projected space $\tilde{\mathcal{H}}_t$. Note that the norm uses the geometry induced by the direction matrix $\tilde{V}_t$.

Lemma 4.4.1. *Let $\delta \in (0,1)$. Assume that $\sup_{s \in \mathcal{X}\times\mathcal{A}} K(s,s) \leq \kappa^2$. Then, with probability $1-\delta$, for all $t \geq 1$*

$$\begin{aligned}\|\tilde{\theta}_t - \theta^*\|_{\tilde{V}_t} &\leq \left(\sqrt{\lambda} + \sqrt{\mu_t}\right)\|\theta^*\| + \sqrt{4\log\frac{1}{\delta} + 2\log\det\left(\frac{K_t + \lambda I}{\lambda}\right)} \\ &\leq \left(\sqrt{\lambda} + \sqrt{\mu_t}\right)\|\theta^*\| + \sqrt{4\log\frac{1}{\delta} + 2\log\left(e + \frac{et\kappa^2}{\lambda}\right) d_{\text{eff}}(\lambda, T)} \quad := \quad \tilde{\beta}_{t+1}(\delta)\,,\end{aligned}$$

where $\|\theta\|_V^2 = \theta^\top V\theta$.

Proof. Let $t \geq 1$. Note that $P_t V_t P_t = P_t(F_t + \lambda I)P_t = P_t \tilde{V}_t = \tilde{V}_t P_t$ and consequently as well $P_t \tilde{V}_t^{-1} = \tilde{V}_t^{-1} P_t$. We can write with $H_t := (\varepsilon_1, \dots, \varepsilon_t)^\top$,

$$\begin{aligned}
\tilde{\theta}_t &= P_t \tilde{V}_t^{-1} P_t \Phi_t^* Y_t \\
&= \tilde{V}_t^{-1} P_t \Phi_t^* Y_t && \leftarrow \quad P_t \tilde{V}_t^{-1} = \tilde{V}_t^{-1} P_t \\
&= \tilde{V}_t^{-1} P_t \Phi_t^* (\Phi_t \theta^* + H_t) \\
&= \tilde{V}_t^{-1} P_t F_t P_t \theta^* + \tilde{V}_t^{-1} P_t F_t (I - P_t)\theta^* + \tilde{V}_t^{-1} P_t \Phi_t^* H_t \\
&= \theta^* - \lambda \tilde{V}_t^{-1} \theta^* + \tilde{V}_t^{-1} P_t F_t (I - P_t)\theta^* + \tilde{V}_t^{-1} P_t \Phi_t^* H_t \,.
\end{aligned}$$

To obtain later on the norm $\|\tilde{\theta}_t - \theta^*\|_{\tilde{V}_t}$, we multiply by $\tilde{V}_t^{1/2}$ on the left

$$\tilde{V}_t^{1/2}(\tilde{\theta}_t - \theta^*) = -\underbrace{\lambda \tilde{V}_t^{-1/2}\theta^*}_{(i)} + \underbrace{\tilde{V}_t^{-1/2} P_t F_t (I - P_t)\theta^*}_{(ii)} + \underbrace{\tilde{V}_t^{-1/2} P_t \Phi_t^* H_t}_{(iii)} \,. \tag{4.29}$$

We then compute each norm separately.

(i) Since $\tilde{V}_t = P_t F_t P_t + \lambda I$, all its eigenvalues are larger than λ. Thus, $\tilde{V}_t^{-1/2} \preccurlyeq \lambda^{-1/2} I$, which implies

$$\big\|\lambda \tilde{V}_t^{-1/2}\theta^*\big\| \leq \sqrt{\lambda}\|\theta^*\| \,. \tag{4.30}$$

(ii) We write $\big\|\tilde{V}_t^{-1/2} P_t F_t (I - P_t)\theta^*\big\| = \big\|\tilde{V}_t^{-1/2} P_t F_t^{1/2} F_t^{1/2} (I - P_t)\theta^*\big\|$ and recall $\tilde{V}_t = P_t F_t^{1/2} F_t^{1/2} P_t + \lambda I$ therefore $\tilde{V}_t^{1/2} \succcurlyeq P_t F_t^{1/2}$, which entails

$$\big\|\tilde{V}_t^{-1/2} P_t F_t (I - P_t)\theta^*\big\| \leq \big\|F_t^{1/2}(I - P_t)\theta^*\big\| \leq \sqrt{\mu_t}\|\theta^*\| \,, \tag{4.31}$$

where we recall that $\mu_t := \big\|(I - P_t)F_t^{1/2}\big\|^2$.

(iii) Let us upper-bound the norm of the last term

$$\begin{aligned}
\big\|\tilde{V}_t^{-1/2} P_t \Phi_t^* H_t\big\| &\leq \big\|\tilde{V}_t^{-1/2} P_t V_t^{1/2}\big\| \big\|V_t^{-1/2} \Phi_t^* H_t\big\| \\
&\leq \big\|\tilde{V}_t^{-1/2} P_t V_t^{1/2}\big\| \sqrt{2\log\frac{1}{\delta} + \log\det\Big(\frac{1}{\lambda}(K_t + \lambda I)\Big)} \,,
\end{aligned} \tag{4.32}$$

with probability at least $1 - \delta$, where the last inequality follows from the same analysis as (4.24). Then, using that $P_t V_t P_t = P_t F_t P_t + \lambda P_t = \tilde{V}_t + \lambda(P_t - I)$, we have

$$\begin{aligned}
\big\|\tilde{V}_t^{-1/2} P_t V_t^{1/2}\big\|^2 = \big\|\tilde{V}_t^{-1/2} P_t V_t P_t \tilde{V}_t^{-1/2}\big\| &= \big\|\tilde{V}_t^{-1/2}\big(\tilde{V}_t + \lambda(P_t - I)\big)\tilde{V}_t^{-1/2}\big\| \\
&= \big\|I + \lambda \tilde{V}_t^{-1/2}(P_t - I)\tilde{V}_t^{-1/2}\big\| \leq 1 + \lambda\big\|\tilde{V}_t^{-1/2}\big\|^2 \big\|P_t - I\big\| \leq 2 \,,
\end{aligned}$$

where the last inequality is because $\big\|P_t - I\big\| \leq 1$ and $\big\|\tilde{V}_t^{-1/2}\big\| \leq \lambda^{-1/2}$. Therefore, substituting into Inequality (4.32) yields

$$\big\|\tilde{V}_t^{-1/2} P_t \Phi_t^* H_t\big\| \leq \sqrt{4\log\frac{1}{\delta} + 2\log\det\Big(\frac{1}{\lambda}(K_t + \lambda I)\Big)} \,, \tag{4.33}$$

with probability at least $1 - \delta$.

Finally, combining (4.30), (4.31), and (4.33) with Equation (4.29) concludes

$$\begin{aligned}\left\|\tilde{\theta}_t - \theta\right\|_{\tilde{V}_t} &\leq \lambda\left\|\tilde{V}_t^{-1/2}\theta^*\right\| + \left\|\tilde{V}_t^{-1/2}P_tF_t(I-P_t)\theta^*\right\| + \left\|\tilde{V}_t^{-1/2}P_t\Phi_t^*H_t\right\| \\ &\leq \left(\sqrt{\lambda}+\sqrt{\mu_t}\right)\left\|\theta^*\right\| + \sqrt{4\log\frac{1}{\delta} + 2\log\det\left(\frac{1}{\lambda}(K_t+\lambda I)\right)}\,.\end{aligned}$$

The second line of the statement follows from Proposition 4.3.1. □

4.10.2. Proof of Theorem 4.4.1

Theorem 4.4.1. *Let $T \geq 1$ and $\theta^* \in \mathcal{H}$. Assume that $|\langle\phi(x,a),\theta^*\rangle_{\mathcal{H}}| \leq 1$ for all $a \in \bigcup_{t=1}^T \mathcal{A}_t \subset \mathcal{A}$ and $x \in \mathcal{X}$ then the EK-UCB rule in Eq.* (4.4.1) *with $\tilde{\mathcal{C}}_t$ defined in Eq.* (4.10)*, with $m = |\mathcal{Z}_t|$ dictionary updates, satisfies the pseudo-regret bound*

$$R_T \lesssim \sqrt{T}\left(\sqrt{\frac{\mu m}{\lambda}} + \sqrt{d_{\text{eff}}}\right)\left(\sqrt{\lambda}+\sqrt{\mu}+\sqrt{d_{\text{eff}}}\right).$$

In particular, the choice $\mu = \lambda$ yields $m \lesssim d_{\text{eff}}$ and

$$R_T \lesssim \sqrt{T}\big(\|\theta^*\|\sqrt{\lambda d_{\text{eff}}} + d_{\text{eff}}\big)\,.$$

Proof. Let $\delta > 0$. By Lemma 4.4.1, with probability $1-\delta$,

$$\forall t \geq 1, \quad \theta^* \in \tilde{C}_t\,. \tag{4.34}$$

Let us recall and start from the definition of the regret

$$R_T := \mathbb{E}\left[\sum_{t=1}^T \Delta_t\right], \qquad \text{where} \quad \Delta_t := \langle\phi(x_t,a_t^*) - \phi(x_t,a_t),\theta^*\rangle_{\mathcal{H}} \quad \text{and} \quad a_t^* := \max_{a\in\mathcal{A}_t}\langle\phi(x_t,a),\theta^*\rangle_{\mathcal{H}}\,.$$

Step 1: Small instantaneous regrets under the event (4.34). Assume that (4.34) holds and define

$$\tilde{\rho}_t \in \arg\max_{\theta\in\tilde{C}_t}\{\langle\phi(x_t,a_t),\theta\rangle_{\mathcal{H}}\}\,.$$

Note here that the use of the original feature map allows us to not have any misspecified term that would have been incurred if the projected feature map was used instead instead with $\langle\phi(x_t,a_t^*),\theta^*\rangle_{\mathcal{H}} = \langle P_t\phi(x_t,a_t^*),\theta^*\rangle_{\mathcal{H}} + \langle(I-P_t)\phi(x_t,a_t^*),\theta^*\rangle_{\mathcal{H}})$ in the upper bound expression.

Now given that $\theta^* \in \tilde{C}_t$ and $a_t \in \arg\max_{a\in\mathcal{A}}$ EK-UCB$_t(a)$, we have

$$\begin{aligned}\langle\phi(x_t,a_t^*),\theta^*\rangle_{\mathcal{H}} \leq \max_{\theta\in\tilde{C}_t}\{\langle\phi(x_t,a_t^*),\theta\rangle_{\mathcal{H}}\} = \text{EK-UCB}_t(a_t^*) &\leq \text{EK-UCB}_t(a_t) \\ &= \max_{\theta\in\tilde{C}_t}\{\langle\phi(x_t,a_t),\theta\rangle_{\mathcal{H}}\} \\ &= \langle\phi(x_t,a_t),\tilde{\rho}_t\rangle_{\mathcal{H}}\,.\end{aligned}$$

Therefore,

$$\Delta_t := \langle \phi(x_t, a_t^*) - \phi(x_t, a_t), \theta^* \rangle_{\mathcal{H}} \leq \langle \phi(x_t, a_t), \tilde{\rho}_t - \theta^* \rangle_{\mathcal{H}} \\ \leq \|\varphi_t\|_{\tilde{V}_{t-1}^{-1}} \|\tilde{\rho}_t - \theta^*\|_{\tilde{V}_{t-1}} \leq 2\|\varphi_t\|_{\tilde{V}_{t-1}^{-1}} \tilde{\beta}_t(\delta)\,. \quad (4.35)$$

Then, summing over $t = 1, \dots, T$ and using $|\Delta_t| \leq 2$ and $\tilde{\beta}_T(\delta) \geq \beta_t(\delta) \geq 1$, we get

$$\sum_{t=1}^{T} \Delta_t \leq \sqrt{T \sum_{t=1}^{T} \Delta_t^2} \qquad \leftarrow \quad \text{Cauchy-Schwartz's inequality}$$

$$\leq 2\sqrt{T \sum_{t=1}^{T} \min\left\{ \|\varphi_t\|^2_{\tilde{V}_{t-1}^{-1}} \tilde{\beta}_t(\delta)^2, 1 \right\}} \quad \leftarrow \quad |\Delta_t| \leq 2 \text{ and (4.35)} \quad (4.36)$$

$$\leq 2\tilde{\beta}_T(\delta)\sqrt{T \sum_{t=1}^{T} \min\left\{ \|\varphi_t\|^2_{\tilde{V}_{t-1}^{-1}}, 1 \right\}} \quad \leftarrow \quad 1 \leq \beta_t(\delta) \leq \beta_T(\delta)\,. \quad (4.37)$$

Note now that

$$\min\left\{ \|\varphi_t\|^2_{\tilde{V}_{t-1}^{-1}}, 1 \right\} \leq 2\min\left\{ \|P_t\varphi_t\|^2_{\tilde{V}_{t-1}^{-1}} + \|(I - P_t)\varphi_t\|^2_{\tilde{V}_{t-1}^{-1}}, 1 \right\} \\ \leq 2\min\left\{ \|P_t\varphi_t\|^2_{\tilde{V}_{t-1}^{-1}}, 1 \right\} + 2\|(I - P_t)\varphi_t\|^2_{\tilde{V}_{t-1}^{-1}} \\ \leq 4\log\left(1 + \|P_t\varphi_t\|^2_{\tilde{V}_{t-1}^{-1}}\right) + 2\|(I - P_t)\varphi_t\|^2_{\tilde{V}_{t-1}^{-1}}. \quad (4.38)$$

The first term can be upper-bounded similarly to (4.27). First, note that since $P_s = P_s P_{t-1}$ for any $1 \leq s \leq t-1$,

$$\tilde{V}_{t-1} := \sum_{s=1}^{t-1} (P_{t-1}\varphi_s) \otimes (P_{t-1}\varphi_s) + \lambda I \succcurlyeq \sum_{s=1}^{t-1} (P_s\varphi_s) \otimes (P_s\varphi_s) + \lambda I =: \tilde{W}_{t-1}$$

which implies $\tilde{V}_{t-1}^{-1} \preccurlyeq \tilde{W}_{t-1}^{-1}$ and thus

$$\log\left(1 + \|P_t\varphi_t\|^2_{\tilde{V}_{t-1}^{-1}}\right) \leq \log\left(1 + \|P_t\varphi_t\|^2_{\tilde{W}_{t-1}^{-1}}\right). \quad (4.39)$$

Now, recalling that $V_{t-1} := \sum_{s=1}^{t-1} \varphi_s \otimes \varphi_s + \lambda I$, following the same analysis as for (4.27), replacing φ_s with $P_s\varphi_s$ for all $s = 1, \dots, t$, we get

$$1 + \|P_t\varphi_t\|^2_{\tilde{W}_{t-1}^{-1}} = \frac{1}{\lambda} \frac{\det(\tilde{K}_t + \lambda I)}{\det(\tilde{K}_{t-1} + \lambda I)}\,,$$

where $\tilde{K}_t \in \mathbb{R}^{t \times t}$ is the kernel matrix such that $\left[\tilde{K}_t\right]_{ij} = \langle P_i\varphi_i, P_j\varphi_j \rangle_{\mathcal{H}}$ for all $1 \leq i, j \leq t$. Together with Inequalities (4.38) and (4.39), and summing over $t = 1, \dots, T$, it yields

$$\sum_{t=1}^{T} \min\left\{ \|\varphi_t\|^2_{\tilde{V}_{t-1}^{-1}}, 1 \right\} \leq 4\sum_{t=1}^{T} \log\left(\frac{1}{\lambda} \frac{\det(\tilde{K}_t + \lambda I)}{\det(\tilde{K}_{t-1} + \lambda I)} \right) + 2\sum_{t=1}^{T} \|(I - P_t)\varphi_t\|^2_{\tilde{V}_{t-1}^{-1}} \\ \leq 4\log\left(\frac{\det(\tilde{K}_T + \lambda I)}{\lambda^t} \right) + 2\sum_{t=1}^{T} \|(I - P_t)\varphi_t\|^2_{\tilde{V}_{t-1}^{-1}}. \quad (4.40)$$

We now upper-bound the second term in the right-hand-side. Denoting by $1 = \tau_1 < \tau_2 < \cdots < \tau_m \leq T$ the indexes in time when the projection is updated, i.e., $P_t = P_{\tau_i}$ for all $t \in \{\tau_i, \ldots, \tau_{i+1} - 1\}$, we can write

$$\begin{aligned}
\sum_{t=1}^{T} \big\|(I - P_t)\varphi_t\big\|^2 &= \sum_{i=1}^{m} \sum_{t=\tau_i}^{\tau_{i+1}-1} \big\|(I - P_{\tau_i})\varphi_t\big\|^2 \\
&= \sum_{i=1}^{m} \sum_{t=\tau_i}^{\tau_{i+1}-1} \mathrm{Tr}\left((I - P_{\tau_i})\varphi_t \otimes \varphi_t (I - P_{\tau_i})\right) \\
&= \sum_{i=1}^{m} \mathrm{Tr}\left((I - P_{\tau_i})\Big(\sum_{t=\tau_i}^{\tau_{i+1}-1} \varphi_t \otimes \varphi_t\Big)(I - P_{\tau_i})\right) \\
&= \sum_{i=1}^{m} \mathrm{Tr}\left((I - P_{\tau_{i+1}-1})\Big(\sum_{t=\tau_i}^{\tau_{i+1}-1} \varphi_t \otimes \varphi_t\Big)(I - P_{\tau_{i+1}-1})\right) \\
&\leq \sum_{i=1}^{m} \mathrm{Tr}\left((I - P_{\tau_{i+1}-1})\Big(\sum_{t=1}^{\tau_{i+1}-1} \varphi_t \otimes \varphi_t\Big)(I - P_{\tau_{i+1}-1})\right) \\
&\leq \sum_{l=1}^{m} \mu_{\tau_{i+1}-1} \leq m\mu\,,
\end{aligned}$$

where the last inequality follows from Prop. 4.4.1. Therefore, using that $\tilde{V}_{t-1}^{-1} \preccurlyeq \lambda^{-1} I$, from (4.38) we get

$$\sum_{t=1}^{T} \min\left\{\|\varphi_t\|^2_{\tilde{V}_{t-1}^{-1}}, 1\right\} \leq 4 \log\left(\frac{\det(\tilde{K}_T + \lambda I)}{\lambda^t}\right) + \frac{2m\mu}{\lambda}\,.$$

Substituting into Inequality (4.37) entails

$$\begin{aligned}
\sum_{t=1}^{T} \Delta_t &\leq 2\tilde{\beta}_T(\delta)\sqrt{T\left(4 \log\left(\frac{\det(\tilde{K}_T + \lambda I)}{\lambda^t}\right) + \frac{2m\mu}{\lambda}\right)} \\
&\leq 2\tilde{\beta}_T(\delta)\sqrt{T\left(4 \log\det\left(\frac{K_T + \lambda I}{\lambda}\right) + \frac{2m\mu}{\lambda}\right)} \\
&\leq 2\tilde{\beta}_T(\delta)\sqrt{T\left(4 \log\left(e + \frac{eT\kappa^2}{\lambda}\right) d_{\text{eff}} + \frac{2m\mu}{\lambda}\right)}
\end{aligned}$$

where the last inequality is by Prop. 4.3.1 and where we recall

$$\tilde{\beta}_T(\delta) \leq \left(\sqrt{\lambda} + \sqrt{\mu}\right)\|\theta^*\| + \sqrt{4 \log\frac{1}{\delta} + 2 \log\left(e + \frac{eT\kappa^2}{\lambda}\right) d_{\text{eff}}}\,.$$

Choosing $\delta = 1/T$ and taking the expectation concludes

$$R_T \leq 2 + 2\tilde{\beta}_T(1/T)\sqrt{T\left(4\log\left(e + \frac{eT\kappa^2}{\lambda}\right)d_{\text{eff}} + \frac{2m\mu}{\lambda}\right)}$$
$$\lesssim \left((\sqrt{\lambda} + \sqrt{\mu})\|\theta^*\| + \sqrt{d_{\text{eff}}}\right)\sqrt{T\left(d_{\text{eff}} + \frac{m\mu}{\lambda}\right)}\,.$$

In particular, for the choice $\mu = \lambda$, by Prop. 4.4.1, the dictionary is at most of size $m \lesssim d_{\text{eff}}$ with high probability. □

4.10.3. Proof of Cor. 4.4.1

Corollary 4.4.1. *Assuming the capacity condition $d_{\text{eff}} \leq (T/\lambda)^\alpha$ for $0 \leq \alpha \leq 1$. Let $1 \leq m \leq T^{\alpha/(1+\alpha)}$, under the assumptions of Thm. 4.4.1, the regret of EK-UCB satisfies*

$$R_T \lesssim \begin{cases} Tm^{\frac{\alpha-1}{2\alpha}} \text{ if } m \leq T^{\frac{\alpha}{1+\alpha}} \\ T^{\frac{1+3\alpha}{2+2\alpha}} \text{ otherwise} \end{cases}$$

for the choice $\lambda = \mu = Tm^{-1/\alpha}$. Furthermore, the algorithm runs in $O(Tm)$ space complexity and $O(CTm^2)$ time complexity.

We start from the regret bound of Theorem 4.4.1, which, forgetting all dependencies that do not depend on T, for the choice $\mu = \lambda$ yields

$$R_T \lesssim \sqrt{T}\left(\sqrt{\lambda d_{\text{eff}}} + d_{\text{eff}}\right).$$

Under the capacity condition $d_{\text{eff}}(\lambda, T) \leq (T/\lambda)^\alpha$, it entails

$$R_T \lesssim \sqrt{T}\left(\lambda^{\frac{1-\alpha}{2}}T^{\frac{\alpha}{2}} + \lambda^{-\alpha}T^\alpha\right) = T^{\frac{1}{2}}\left(T^{1/2}m^{\frac{\alpha-1}{2\alpha}} + m\right) = Tm^{\frac{\alpha-1}{2\alpha}} + \sqrt{T}m\,,$$

where we replaced $\lambda = Tm^{-1/\alpha}$. Optimizing in m, we retrieve the original rate $R_T \lesssim T^{\frac{1+3\alpha}{2+2\alpha}}$ for a dictionary of size $m = T^{\frac{\alpha}{1+\alpha}} \ll T$. Note that a larger dictionary is not necessary in theory since it only hurts both the theoretical rate and the computational complexity. For a smaller dictionary, the first term is predominant and yields a regret of order $\mathcal{O}\left(Tm^{\frac{\alpha-1}{2\alpha}}\right)$, highlighting a trade-off between the complexity which increases with m and the regret which decreases.

4.11. Details on the comparison of the regret bounds of CGP-UCB, SupKernelUCB, and K-UCB

In this appendix, we first detail why we can compare the regrets of CGP-UCB (Krause and Ong, 2011), SupKernelUCB (Valko et al., 2013) and K-UCB as shown in Table 4.1. We compare the quantities $\tilde{d}, \gamma$ and d_{eff} that appear in the regret bound of the literature (Valko et al., 2013;

Calandriello et al., 2019; Krause and Ong, 2011). We show that they are essentially equivalent up to logarithmic factors. We recall first their definitions: for any $t \geq 0$ and $\lambda > 0$

$$\begin{aligned}\gamma(\lambda, t) &= \frac{1}{2}\log\left(\det\left(I + \frac{1}{\lambda}K_t\right)\right)\\ \tilde{d}(\lambda, t) &= \min\{j : j\lambda \log T \geq \sum_{k>j} \lambda_K(K_t)\}\\ d_{\text{eff}}(\lambda, T) &= \mathrm{Tr}(K_T(K_T + \lambda I_T)^{-1})\,.\end{aligned}$$

We start by proving the first equality (up to logarithmic factors) $d_{\text{eff}}(\lambda, t) \lesssim \gamma(\lambda, t) \lesssim d_{\text{eff}}(\lambda, t)$. We first obtain that $\gamma(\lambda, t) \lesssim d_{\text{eff}}$ with Proposition 4.3.1. Next, to prove that $d_{\text{eff}} \lesssim \gamma(\lambda, t)$, we prove that for all $x > -1$ $\dfrac{x}{x+1} \leq \log(1+x)$ by writing for $x > -1$, $h(x) = \dfrac{x}{x+1} - \log(1+x)$ studying h' and $h(0)$. Therefore,

$$d_{\text{eff}}(\lambda, t) = \mathrm{Tr}(K_t(Kt + \lambda I)^{-1}) = \sum_{k=1}^{t} \frac{\frac{\lambda_k}{\lambda}}{\frac{\lambda_k}{\lambda} + 1} \leq \sum_{k=1}^{t} \log(1 + \frac{\lambda_k}{\lambda}) = \gamma(\lambda, t)$$

Next we detail that $\tilde{d}(\lambda, t) \lesssim \gamma(\lambda, t) \lesssim \tilde{d}(\lambda, t)$. First, Valko et al. (2013) shows that $\tilde{d}(\lambda, t) \lesssim \gamma(\lambda, t)$. Second, to prove $\gamma(\lambda, t) \lesssim \tilde{d}(\lambda, t)$, we write

$$\sum_{k=1}^{t} \log\left(1 + \frac{\lambda_k}{\lambda}\right) \leq \sum_{k>\tilde{d}} \frac{\lambda_k}{\lambda} + \sum_{k\leq\tilde{d}} \log(1 + \frac{\lambda_k}{\lambda}) \leq \tilde{d}(\lambda, t)\log(t) + \tilde{d}(\lambda, t)\log\left(\frac{\lambda_1}{\lambda}\right),$$

where we used $\log(1+x) \leq x$ on the first term of the sum decomposition, and λ_1 the first and larger eigenvalue of the matrix K_t. Then, using $\lambda_1(K_t) \leq \mathrm{Tr}(K_t) = \sum_{k=1}^{t} \|\varphi_k\|^2 \leq t\kappa^2$, we subsequently obtain $\gamma(\lambda, t) \leq \tilde{d}\left(\log(T) + \log\left(\dfrac{t\kappa^2}{\lambda}\right)\right)$ which concludes the inequality.

4.12. Algorithm Implementations

Here we give details on the implementations of the contextual kernel UCB algorithms as well as our EK-UCB.

4.12.1. Kernel UCB algorithm – Implementation details

Let us write $s_{t,a} := (x_t, a)$ and by abbreviation $s_i := (x_i, a_i)$, let us write the historical data $\mathcal{S}_t = (s_i)_{1\leq i\leq t}$. Let us recall $\Phi_t^* = [\varphi_1, \ldots, \varphi_t]$ where $\varphi_i = \phi(x_i, a_i) = \phi(s_i)$ and $K_{\mathcal{S}_t}(s) = \Phi_t\phi(s) = [K(s_1, s), \ldots K(s_t, s)]^\top$. We write $F_t = \Phi_t^*\Phi_t$ and the gram matrix $K_t = \Phi_t\Phi_t^*$. As in (Valko et al., 2013):

$$\begin{aligned}(\Phi_t^*\Phi_t + \lambda I)\Phi_t^* &= \Phi_t^*(\Phi_t\Phi_t^* + \lambda I)\\ (F_t + \lambda I)\Phi_t^* &= \Phi_t^*(K_t + \lambda I)\\ \Phi_t^*(K_t + \lambda I)^{-1} &= (F_t + \lambda I)^{-1}\Phi_t^*.\end{aligned}$$

Expression of the mean $\hat{\mu}_{t,a} = \langle\hat{\theta}_t, \varphi_{t,a}\rangle_{\mathcal{H}}$. For the mean expression recall that we have: $\hat{\mu}_{t,a} = \langle\hat{\theta}_{t-1}, \varphi_{t,a}\rangle_{\mathcal{H}} = \varphi_{t,a}^\top\hat{\theta}_{t-1}$ and $\hat{\theta}_t = V_t^{-1}\Phi_t^*Y_t$. Therefore,

$$\hat{\mu}_{t,a} = \varphi_{t,a}^\top\hat{\theta}_{t-1} = \varphi_{t,a}^\top\Phi_{t-1}^*(K_{t-1} + \lambda I)^{-1}Y_{t-1} = K_{\mathcal{S}_{t-1}}(s_{t,a})^\top(K_{t-1} + \lambda I)^{-1}Y_{t-1}.$$

Expression of the standard deviation $\hat{\sigma}_{t,a} = \|\varphi_{t,a}\|_{V_{t-1}^{-1}}$. When multiplying by $\varphi_{t,a} := \phi(x_t, a)$ on the right and then by $\varphi_{t,a}^\top$ on the left

$$\begin{aligned}
(\Phi_{t-1}^* \Phi_{t-1} + \lambda I)\varphi_{t,a} &= \Phi_{t-1}^* K_{\mathcal{S}_{t-1}}(s_{t,a}) + \lambda \varphi_{t,a} \\
\varphi_{t,a} &= \Phi_{t-1}^* (K_{t-1} + \lambda I)^{-1} K_{\mathcal{S}_{t-1}}(s_{t,a}) + \lambda (\Phi_{t-1}^* \Phi_{t-1} + \lambda I)^{-1} \varphi_{t,a} \\
\varphi_{t,a}^\top \varphi_{t,a} &= K_{\mathcal{S}_{t-1}}(s_{t,a})^\top (K_{t-1} + \lambda I)^{-1} K_{\mathcal{S}_{t-1}}(s_{t,a}) + \lambda \varphi_{t,a}^\top V_{t-1}^{-1} \varphi_{t,a} \\
\hat{\sigma}_{t,a} = \|\varphi_{t,a}\|_{V_{t-1}^{-1}} &= \frac{1}{\lambda} k(s_{t,a}, s_{t,a}) - \frac{1}{\lambda} K_{\mathcal{S}_{t-1}}(s_{t,a})^\top (K_{t-1} + \lambda I)^{-1} K_{\mathcal{S}_{t-1}}(s_{t,a})
\end{aligned}$$

This allows to compute the UCB rule with kernel representations as illustrated in Alg. 11.

Algorithm 11: Kernel UCB

Input: T the horizon, λ regularization and exploration parameters, K(the kernel function

initialization;

$K_\lambda = \lambda, Y_0 = [r_0]$ where $r_0 = r(x_0, a_0)$ and a_0 is chosen randomly ;

for $t = 1$ *to* T **do**

  Observe context x_t ;

  Compute β_t ;

  Choose $a_t \leftarrow \arg\max_{a \in \mathcal{A}} \hat{\mu}_{t,a} + \beta_t \hat{\sigma}_{t,a}$;

    $\hat{\mu}_{t,a} \leftarrow K_{\mathcal{S}_{t-1}}(s_{t,a})^\top K_\lambda^{-1} Y_{t-1}$;

    $\hat{\sigma}_{t,a}^2 \leftarrow \frac{1}{\lambda} k(s_{t,a}, s_{t,a}) - \frac{1}{\lambda} K_{\mathcal{S}_{t-1}}(s_{t,a})^\top K_\lambda^{-1} K_{\mathcal{S}_{t-1}}(s_{t,a})$;

  Observe reward r_t and update $Y_t \leftarrow [r_1, \dots r_t]$;

  Update the translated gram matrix $K_\lambda \leftarrow [K(s_i, s_j)]_{1 \le i,j \le t} + \lambda I$;

end

Since the kernel matrices are used instead of estimating and computing directly $\hat{\theta}_t$ and $\phi(x_t, a)$, we can use first-rank updates of the matrices K_t, since:

$$K_t = \begin{bmatrix} K_{t-1} & K_{\mathcal{S}_{t-1}}(s_{t,a}) \\ K_{\mathcal{S}_{t-1}}(s_{t,a})^\top & K(s_{t,a}, s_{t,a}) \end{bmatrix}.$$

It is then easy to use the Schur complement on the inverse K_λ^{-1}. Specifically, the update is performed as the following, with

$$\begin{aligned}
s &\leftarrow k(s_{t,a}, s_{t,a}) + \lambda - K_{\mathcal{S}_{t-1}}(s_{t,a})^\top K_\lambda^{-1} K_{\mathcal{S}_{t-1}}(s_{t,a}) \\
Z_{12} &\leftarrow -\frac{1}{s} K_{\mathcal{S}_{t-1}}(s_{t,a})^\top K_\lambda^{-1} \\
Z_{21} &\leftarrow -\frac{1}{s} K_\lambda^{-1} K_{\mathcal{S}_{t-1}}(s_{t,a}) \\
Z_{11} &\leftarrow K_\lambda^{-1} + \frac{1}{s} K_\lambda^{-1} K_{\mathcal{S}_{t-1}}(s_{t,a}) K_{\mathcal{S}_{t-1}}(s_{t,a})^\top K_\lambda^{-1} \\
K_\lambda^{-1} &\leftarrow [Z_{11}, Z_{12}, Z_{21}, \frac{1}{s}].
\end{aligned}$$

Therefore, while inverting the full matrices would induce as full cost of $\mathcal{O}(CT^4)$, using first order updates with Schur complement allows to run the algorithm in $\mathcal{O}(CT^3)$, while using $\mathcal{O}(T^2)$ in space.

4.12.2. Efficient Kernel UCB algorithm – Implementation details

Instead of using the kernel trick as in the standard algorithm, the efficient Kernel UCB algorithm uses computations in the projected feature space. The key high-level idea is to use as much as possible computations in the projected space $\mathcal{H}_t = \text{span}\{\phi(z)\}_{z\in\mathcal{Z}_t}$ which is of dimension m_t and does not use implicit kernel representation of the whole data which are of size $t \times t$. Here, we detail the computations of the predicted mean and variance bound in the projected space.

At the time t we define the dictionary $\mathcal{Z}_t = \{z_1, \dots, z_{m_t}\}$ of size $|\mathcal{Z}_t| = m_t$ and the $m_t \times m_t$ kernel matrix $K_{\mathcal{Z}_t} = [K(z_i, z_j)]_{1\le i,j\le m_t}$, we also write $K_{\mathcal{Z}_t\mathcal{S}_t} = [K(z_i, s_j)]_{1\le i\le m_t, 1\le j\le t}$ the $m_t \times t$ matrix on anchor points and historical data $\mathcal{S}_t = \{s_i\}_{1\le i\le t}$.

The following proposition provides closed-form formulas to implement EK-UCB (Alg. 10).

Proposition 4.4.2. *At any round t, by considering $s_{t,a} = (x_t, a)$, the mean and variance term of the EK-UCB rule (Alg 10) can be expressed as*[3]

$$\begin{aligned}
\Gamma_t &= K_{\mathcal{Z}_{t-1}\mathcal{S}_{t-1}} Y_{t-1} \\
\Lambda_t &= \left(K_{\mathcal{Z}_{t-1}\mathcal{S}_{t-1}} K_{\mathcal{S}_{t-1}\mathcal{Z}_{t-1}} + \lambda K_{\mathcal{Z}_{t-1}\mathcal{Z}_{t-1}}\right)^{-1} \\
\tilde{\mu}_{t,a} &= K_{\mathcal{Z}_{t-1}}(s_{t,a})^\top \Lambda_t \Gamma_t \\
\Delta_{t,a} &= K_{\mathcal{Z}_{t-1}}(s_{t,a})^\top \left(\Lambda_t - \frac{1}{\lambda} K^{-1}_{\mathcal{Z}_{t-1}\mathcal{Z}_{t-1}}\right) K_{\mathcal{Z}_{t-1}}(s_{t,a}) \\
\tilde{\sigma}^2_{t,a} &= \frac{1}{\lambda} K(s_{t,a}, s_{t,a}) + \Delta_{t,a}.
\end{aligned}$$

The algorithm then runs in a space complexity of $\mathcal{O}(Tm)$ and a time complexity of $\mathcal{O}(CTm^2)$.

Expression of the mean $\tilde{\mu}_{t+1,a} = \langle \tilde{\theta}_t, \varphi_{t+1,a}\rangle_{\mathcal{H}}$. At a time $t+1$, we look for $\tilde{\theta} \in \tilde{\mathcal{C}}_{t+1}$ that we write $\tilde{\theta} = \alpha^\top K_{\mathcal{Z}_t}$ where $\alpha \in \mathbb{R}^{m_t}$. We can rewrite the optimization process in Eq. (4.9) as

$$\arg\min_{\alpha\in\mathbb{R}^{m_t}} \left\{ (K_{\mathcal{S}_t\mathcal{Z}_t}\alpha - Y_t)^\top (K_{\mathcal{S}_t\mathcal{Z}_t}\alpha - Y_t) + \lambda \alpha^\top K_{\mathcal{Z}_t\mathcal{Z}_t}\alpha \right\}$$

which can be rewritten as

$$\arg\min_{\alpha\in\mathbb{R}^{m_t}} \left\{ \alpha^\top K_{\mathcal{Z}_t\mathcal{S}_t} K_{\mathcal{S}_t\mathcal{Z}_t}\alpha - 2\alpha^\top K_{\mathcal{Z}_t\mathcal{S}_t} Y_t + \lambda \alpha^\top K_{\mathcal{Z}_t\mathcal{Z}_t}\alpha \right\},$$

and can be solved in closed-form as

$$\alpha^* = (K_{\mathcal{Z}_t\mathcal{S}_t} K_{\mathcal{S}_t\mathcal{Z}_t} + \lambda K_{\mathcal{Z}_t\mathcal{Z}})^{-1} K_{\mathcal{Z}\mathcal{S}_t} Y_t.$$

This eventually gives the expression $\tilde{\mu}_{t+1,a} = \alpha^\top K_{\mathcal{Z}_t s_{t+1}}$

$$\tilde{\mu}_{t+1,a} = K_{\mathcal{Z}_t}(s_{t+1,a})^\top (K_{\mathcal{Z}_t\mathcal{S}_t} K_{\mathcal{S}_t\mathcal{Z}_t} + \lambda K_{\mathcal{Z}_t\mathcal{Z}_t})^{-1} K_{\mathcal{Z}_t\mathcal{S}_t} Y_t\,.$$

[3]Erratum: Note that the proposition slightly differs from the original one in the main document due to typos in the indexes that will be corrected in the final version of the manuscript.

Expression of the standard deviation $\tilde{\sigma}_{t+1,a} = \|\varphi_{t+1,a}\|_{\tilde{V}_t^{-1}}$. When we look for the value of EK-UCB in Eq. (4.11), it is equivalent to have:

$$\text{EK-UCB}_{t+1}(a) = \max_{\theta\in\mathcal{H},\ \text{s.t}\ \|\theta-\tilde{\theta}_t\|_{\tilde{V}_t}\leq\beta} \langle\theta, \phi(s_{t+1,a})\rangle_{\mathcal{H}} = \tilde{\mu}_{t+1,a} + \beta\tilde{\sigma}_{t+1,a}.$$

where the variance term $\tilde{\sigma}_{t+1,a}$ is solution to

$$\max_{\theta\in\mathcal{H}} \theta^\top\phi(s_{t+1,a})\,.$$
$$\text{s.t}\ \|\theta\|_{\tilde{V}_t} \leq 1.$$

Below, we abbreviate $s = s_{t+1,a} := (x_{t+1}, a)$ for simplicity of notation. We advocate that at each time t when we solve this maximization problem, θ lives in the finite dimensional space

$$\theta \in \mathcal{H}_{t+1,s} =: \text{Span}\left(K_{z_1}, \dots K_{z_{m_t}}, K_s\right)\,,$$

where $K_z, K_s \in \mathcal{H}$ such that $K_z(z') = K(z,z')$ and $K_s(s') = K(s,s')$. To prove the above statement, following the Representer theorem proof, and $\mathcal{H}_{t+1,s}$ be the linear span of $K_{z_1}, \dots K_{z_{m_t}}, K_s \in \mathcal{H}$. $\mathcal{H}_{t+1,s}$ is a finite dimensional subspace of $\mathcal{H}$, therefore any $\theta \in \mathcal{H}$ can be uniquely decomposed as

$$\theta = \theta_{\mathcal{H}_{t+1,s}} + \theta_\perp$$

with $\theta_{\mathcal{H}_{t+1,s}} \in \mathcal{H}_{t+1,s}$ and $\theta_\perp \perp \mathcal{H}_{t+1,s}$. $\mathcal{H}$ being a RKHS it holds that $\langle\theta_\perp, \phi(s)\rangle_{\mathcal{H}} = \langle\theta_\perp, K_s\rangle_{\mathcal{H}} = 0$ because $K_s \in \mathcal{H}_{t+1,s}$. Therefore, $\langle\theta, K_s\rangle_{\mathcal{H}} = \langle\theta_{\mathcal{H}_{t+1,s}}, K_s\rangle_{\mathcal{H}}$.

Now writing $\tilde{V}_t = P_tV_tP_t + \lambda(I - P_t)$, we have that $\|\theta\|_{\tilde{V}_t}$ can be written as

$$\|\theta\|_{\tilde{V}_t} = \theta_{\mathcal{H}_{t+1,s}}^\top P_tV_tP_t\theta_{\mathcal{H}_{t+1,s}} + \lambda\theta_{\mathcal{H}_{t+1,s}}^\top(I-P_t)\theta_{\mathcal{H}_{t+1,s}} + \lambda\theta_\perp^\top(I-P_t)\theta_\perp.$$

Therefore, $\|\theta_{\mathcal{H}_{t+1,s}}\|_{\tilde{V}_t} \leq \|\theta\|_{\tilde{V}_t} \leq 1$. The maximization domain $\{\theta \in \mathcal{H}\ \text{s.t}\ \|\theta\|_{\tilde{V}_t} \leq 1\}$ is thus included in $\{\theta \in \mathcal{H}_{t+1,s}\ \text{s.t}\ \|\theta_{\mathcal{H}_{t+1,s}}\|_{\tilde{V}_t} \leq 1\}$, while $\langle\theta, K_s\rangle_{\mathcal{H}} = \langle\theta_{\mathcal{H}_{t+1,s}}, K_s\rangle_{\mathcal{H}}$. Therefore, $\max_{\theta\in\mathcal{H}}\langle\theta, K_s\rangle_{\mathcal{H}} = \max_{\theta\in\mathcal{H}_{t+1,s}}\langle\theta_{\mathcal{H}_{t+1,s}}, K_s\rangle_{\mathcal{H}}$. Hence we can write the solution of the problem from Eq. (4.4.1) as

$$\theta_{\mathcal{H}_{t+1,s}} = \sum_{i=1}^{m_t}\alpha_iK_{z_i} + \alpha_{m_t+1}K_s,\ \alpha \in \mathbb{R}^{m_t}, \quad \alpha_{m_t+1} \in \mathbb{R}.$$

We will write $\bar{K}_{\mathcal{Z}_t}\alpha = \sum_{i=1}^{m_t}\alpha_iK_{z_i}$ and therefore $\bar{K}_{\mathcal{Z}_t}^\top\bar{K}_{\mathcal{Z}_t} = K_{\mathcal{Z}_t,\mathcal{Z}_t}$ or even $\bar{K}_{\mathcal{Z}_t}^\top K_s = K_{\mathcal{Z}_ts} \in \mathbb{R}^{m_t}$.

Using this notation allows us to write $P_t\varphi_{t+1} = \sum_{i=1}^{m_t}\beta_i(s_{t+1,a})K_{z_i} = \bar{K}_{\mathcal{Z}_t}(K_{\mathcal{Z}_t,\mathcal{Z}_t}^{-1}K_{\mathcal{Z}_t}(s_{t+1,a}))$ where the β coefficient is obtained by solving with the minimization problem defined in the Nyström projection. Therefore when taking the projection $P_t : \mathcal{H} \to \mathbb{R}^{m_t}$ and the operator $\Phi_t : \mathbb{R}^t \to \mathcal{H}$ we can write $P_t\Phi_t = \bar{K}_{\mathcal{Z}_t}(K_{\mathcal{Z}_t,\mathcal{Z}_t}^{-1})K_{\mathcal{Z}_t\mathcal{S}_t}$.

Therefore when writing $\tilde{V}_t = P_tF_tP_t + \lambda I$ we can express $\|\theta\|_{\tilde{V}_t}$ as

$$\|\theta\|_{\tilde{V}_t} = [\bar{K}_{\mathcal{Z}_t}\alpha + \alpha_{m_t+1}K_{s_t}]^\top[\bar{K}_{\mathcal{Z}_t}K_{\mathcal{Z}_t,\mathcal{Z}_t}^{-1}K_{\mathcal{Z}_t,\mathcal{S}_t}K_{\mathcal{S}_t,\mathcal{Z}_t}K_{\mathcal{Z}_t,\mathcal{Z}_t}^{-1}\bar{K}_{\mathcal{Z}_t}^\top + \lambda I][\bar{K}_{\mathcal{Z}_t}\alpha + \alpha_{m_t+1}K_s].$$

This can be reformulated as

$$\begin{bmatrix}\alpha & \alpha_{m_t+1}\end{bmatrix} Q_t \begin{bmatrix}\alpha \\ \alpha_{m_t+1}\end{bmatrix},$$

where $Q_t = \begin{bmatrix}A_t & b_t \\ b_t^\top & c_t\end{bmatrix}$ and for which we have $A_t = K_{\mathcal{Z}_t\mathcal{S}_t}K_{\mathcal{S}_t\mathcal{Z}_t} + \lambda K_{\mathcal{Z}_t\mathcal{Z}_t}$, $b_t^\top = K_{s\mathcal{Z}_t}K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}K_{\mathcal{Z}_t\mathcal{S}_t}K_{\mathcal{S}_t\mathcal{Z}_t} + \lambda K_{s\mathcal{Z}_t}$ and eventually $c_t = K_{s\mathcal{Z}_t}K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}K_{\mathcal{Z}_t\mathcal{S}_t}K_{\mathcal{S}_t\mathcal{Z}_t}K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}K_{\mathcal{Z}_ts} + \lambda K_{ss}$

Next to find the variance term, we note $q_t = [K_{\mathcal{Z}_ts}, K_{ss}]^\top$ and reformulate the optimization process above as

$$\max_{\alpha\in\mathbb{R}^{m_t+1}} \alpha^\top q_t$$
$$\text{s.t } \alpha^\top Q_t \alpha \leq 1$$

gives the solution $\alpha' = \dfrac{Q_t^{-1/2}q_t}{\|Q_t^{-1/2}q_t\|}$ which gives $\sigma_{t,a}$ the maximum value: $\sqrt{q_t^\top Q_t^{-1}q_t}$. We will now express the squared maximum $\sigma_{t+1,a}^2 = q_t^\top Q_t^{-1}q_t$ using the Schur complement on the Q_t matrix.

Defining $A_t = K_{\mathcal{Z}_t\mathcal{S}_t}K_{\mathcal{S}_t\mathcal{Z}_t} + \lambda K_{\mathcal{Z}_t\mathcal{Z}_t}$ and the Schur complement $l_t = c_t - b_t^\top A_t^{-1}b_t$.

We start by simplifying the expression of the Schur complement. For this we reformulate

$$\begin{aligned}
A_t &= K_{\mathcal{Z}_t\mathcal{S}_t}K_{\mathcal{S}_t\mathcal{Z}_t} + \lambda K_{\mathcal{Z}_t\mathcal{Z}_t} \\
b_t^\top &= K_{s\mathcal{Z}_t}K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}(A_t - \lambda K_{\mathcal{Z}_t\mathcal{Z}_t}) + \lambda K_{s\mathcal{Z}_t} \\
&= K_{s\mathcal{Z}_t}K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}A_t \\
c_t &= K_{s\mathcal{Z}_t}K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}(A_t - \lambda K_{\mathcal{Z}_t\mathcal{Z}_t})K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}K_{\mathcal{Z}_ts} + \lambda K_{ss} \\
&= K_{s\mathcal{Z}_t}K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}A_tK_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}K_{\mathcal{Z}_ts} - \lambda K_{s\mathcal{Z}_t}K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}K_{\mathcal{Z}_ts} + \lambda K_{ss}.
\end{aligned}$$

Thus we obtain:

$$\begin{aligned}
l_t &= K_{s\mathcal{Z}_t}K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}A_tK_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}K_{\mathcal{Z}_ts} - \lambda K_{s\mathcal{Z}_t}K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}K_{\mathcal{Z}_ts} + \lambda K_{ss} - K_{s\mathcal{Z}_t}K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}A_tA_t^{-1}A_tK_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}K_{\mathcal{Z}_ts} \\
&= \lambda(K_{ss} - K_{s\mathcal{Z}_t}K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}K_{\mathcal{Z}_ts}).
\end{aligned}$$

Then we write the product between Q_t^{-1} and q_t as:

$$\begin{aligned}
\tilde{\sigma}^2_{t+1,a} &= \begin{bmatrix} K_{s\mathcal{Z}_t} & K_{ss} \end{bmatrix} \begin{bmatrix} A_t^{-1} + \frac{1}{l} A_t^{-1} b_t b_t^\top A_t^{-1} & -\frac{1}{l} A_t^{-1} b_t \\ -\frac{1}{l} b_t^\top A_t^{-1} & \frac{1}{l} \end{bmatrix} \begin{bmatrix} K_{\mathcal{Z}_t s} \\ K_{ss} \end{bmatrix} \\
&= \begin{bmatrix} K_{s\mathcal{Z}_t} A_t^{-1} + \frac{1}{l} K_{s\mathcal{Z}_t} A_t^{-1} b_t b_t^\top A_t^{-1} - \frac{1}{l} K_{ss} b_t^\top A_t^{-1} & -\frac{1}{l} \lambda K_{s\mathcal{Z}_t} A_t^{-1} b_t + \frac{1}{l} K_{ss} \end{bmatrix} \begin{bmatrix} K_{\mathcal{Z}_t s} \\ K_{ss} \end{bmatrix} \\
&= K_{s\mathcal{Z}_t} A_t^{-1} K_{\mathcal{Z}_t s} + \frac{1}{l} K_{s\mathcal{Z}_t} A_t^{-1} b_t b_t^\top A_t^{-1} K_{\mathcal{Z}_t s} - \frac{1}{l} K_{ss} b_t^\top A_t^{-1} K_{\mathcal{Z}_t s} - \frac{1}{l} K_{s\mathcal{Z}_t} A_t^{-1} b_t K_{ss} + \frac{1}{l} K_{ss}^2 \\
&= K_{s\mathcal{Z}_t} A_t^{-1} K_{\mathcal{Z}_t s} + \frac{1}{l} \left(K_{s\mathcal{Z}_t} A_t^{-1} b_t - K_{ss} \right)^2 \\
&= K_{s\mathcal{Z}_t} A_t^{-1} K_{\mathcal{Z}_t s} + \frac{1}{l} \left(K_{s\mathcal{Z}_t} K_{\mathcal{Z}_t \mathcal{Z}_t}^{-1} K_{\mathcal{Z}_t s} - K_{ss} \right)^2 \\
&= K_{s\mathcal{Z}_t} A_t^{-1} K_{\mathcal{Z}_t s} + \frac{1}{\lambda} K_{ss} - \frac{1}{\lambda} K_{s\mathcal{Z}_t} K_{\mathcal{Z}_t \mathcal{Z}_t}^{-1} K_{\mathcal{Z}_t s} \\
&= \frac{1}{\lambda} K(s_{t,a}, s_{t,a}) + \Delta_{t+1,a} \,,
\end{aligned}$$

where $\Delta_{t+1,a} := K_{\mathcal{Z}_t}(s_{t+1,a})^\top \left(\Lambda_{t+1} - \frac{1}{\lambda} K_{\mathcal{Z}_t \mathcal{Z}_t}^{-1} \right) K_{\mathcal{Z}_t}(s_{t+1,a})$ and $\Lambda_{t+1} := A_{t+1}^{-1}$.

This proves the first of Prop. 4.4.2.

Discussion on practical implementation and time and space complexities The efficient implementation of the algorithm requires to perform efficient updates of the quantities (defined in Prop 4.4.2) $\Lambda_t = (K_{\mathcal{Z}_{t-1}\mathcal{S}_{t-1}} K_{\mathcal{S}_{t-1}\mathcal{Z}_{t-1}} + \lambda K_{\mathcal{Z}_{t-1}\mathcal{Z}_{t-1}})^{-1}$ and $\Gamma_t = K_{\mathcal{Z}_{t-1}\mathcal{S}_{t-1}} Y_{t-1}$.

(i) When the dictionary is not updated $\mathcal{Z}_t = \mathcal{Z}_{t-1}$. For the matrix Γ_t we can perform the update $\Gamma_{t+1} \leftarrow \Gamma_t + r_t K_{\mathcal{Z}_t}(s_t)$ which requires m_t kernel evaluations. As for the matrix Λ_t we can use the first rank Shermann-Morrison formula on it by adding updates on s_t in $\mathcal{O}(m_t^2)$ operations where $\Lambda_{t+1} = (K_{\mathcal{Z}_t\mathcal{S}_t} K_{\mathcal{S}_t\mathcal{Z}_t} + \lambda K_{\mathcal{Z}_t\mathcal{Z}_t})^{-1}$. Here we only store $K_{\mathcal{Z}_t\mathcal{Z}_t}^{-1}$ and do not update it.

(ii) When the dictionary is updated $\mathcal{Z}_t \neq \mathcal{Z}_{t-1}$ and we can write $\mathcal{Z}_t = \mathcal{Z}_{t-1} \cup \{z_{m_t}\}$,

Regarding Γ_t, we do two updates, one on the state s_t by adding $r_t K_{\mathcal{Z}_{t-1}}(s_t)$ and a second on the new anchor point z_{m_t} so that we have

$$\Gamma_{t+1} \leftarrow [\Gamma_t + r_t K_{\mathcal{Z}_{t-1}}(s_t), K_{\mathcal{S}_t}(z_{m_t})^\top Y_t]^\top \,.$$

The first update is performed in $\mathcal{O}(m_t)$ kernel evaluations as in the (i) case, and the second update requires $\mathcal{O}(t)$ kernel evaluations and then $\mathcal{O}(t)$ computations. Note that the (ii) is only visited at most m times which is the size of the dictionary at $t = T$.

Regarding Λ_t, we note that we can write $K_{\mathcal{Z}_t\mathcal{S}_t} K_{\mathcal{S}_t\mathcal{Z}_t} + \lambda K_{\mathcal{Z}_t\mathcal{Z}_t}$ as

$$\begin{bmatrix} K_{\mathcal{Z}_{t-1}\mathcal{S}_t} K_{\mathcal{S}_t\mathcal{Z}_{t-1}} + \lambda K_{\mathcal{Z}_{t-1}\mathcal{Z}_{t-1}} & K_{\mathcal{Z}_{t-1}\mathcal{S}_t} K_{\mathcal{S}_t}(z) + \lambda K_{\mathcal{Z}_{t-1}}(z) \\ K_{\mathcal{S}_t}(z)^\top K_{\mathcal{S}_t\mathcal{Z}_{t-1}} + \lambda K_{\mathcal{Z}_{t-1}}(z)^\top & K_{\mathcal{S}_t}(z)^\top K_{\mathcal{S}_t}(z) + \lambda K(z,z) \end{bmatrix} .$$

We perform the update in two stages by first computing the inverse $(K_{\mathcal{Z}_{t-1}\mathcal{S}_t} K_{\mathcal{S}_t\mathcal{Z}_{t-1}} + \lambda K_{\mathcal{Z}_{t-1}\mathcal{Z}_{t-1}})^{-1}$ by using a first-rank Sherman Morrison on the state update s_t, as if the

Algorithm 12: Efficient Kernel UCB

Input: T the horizon, λ regularization and exploration parameters, K(the kernel function, $\varepsilon > 0, \gamma > 0$

Initialization;

Context x_0, a_0 chosen randomly and reward r_0 ;

$\mathcal{S} = \{(x_0, a_0)\}, Y_{\mathcal{S}} = [r_0]$;

$\mathcal{Z} = \{(x_0, a_0)\}$;

$\Lambda_t = (K_{\mathcal{Z}\mathcal{S}} K_{\mathcal{S}\mathcal{Z}} + \lambda K_{\mathcal{Z}\mathcal{Z}})^{-1}\, \Gamma_t = K_{\mathcal{Z}\mathcal{S}} Y_{\mathcal{S}}$;

for $t = 1$ *to* T **do**

  Observe context x_t ;

  Choose $\tilde{\beta}_t$ (e.g as in Lem. 4.4.1, and $\delta = \frac{1}{T^2}$) ;

  Choose $a_t \leftarrow \arg\max_{a \in \mathcal{A}} \tilde{\mu}_{t,a} + \tilde{\beta}_t \tilde{\sigma}_{t,a}$;

    $\tilde{\mu}_{t,a} \leftarrow K_{\mathcal{Z}}(s_{t,a})^\top \Lambda_t \Gamma_t$;

    $\Delta_{t,a} = K_{\mathcal{Z}}(s_{t,a})^\top \left(\Lambda_t - \frac{1}{\lambda} K_{\mathcal{Z}\mathcal{Z}}^{-1}\right) K_{\mathcal{Z}}(s_{t,a})$;

    $\tilde{\sigma}^2_{t,a} \leftarrow \frac{1}{\lambda} K(s_{t,a}, s_{t,a}) + \Delta_{t,a}$;

  Observe reward r_t and $s_t \leftarrow (x_t, a_t)$;

  $Y_{\mathcal{S}} \leftarrow [Y_{\mathcal{S}}, r_t]^\top, \mathcal{S} \leftarrow \mathcal{S} \cup \{s_t\}$;

  $\mathcal{Z}' \leftarrow \text{KORS}(t, \mathcal{Z}, K_{\mathcal{Z}}(s_t), \lambda, \varepsilon, \gamma)$;

  if $\mathcal{Z}' = \mathcal{Z}$ **then**

    Incremental inverse update Λ_t with s_t;

    $\Gamma_{t+1} \leftarrow \Gamma_t + r_t K_{\mathcal{Z}}(s_t)$;

  end

  else

    $z = \mathcal{Z}' \setminus \mathcal{Z}$;

    Incremental inverse update Λ_t with s_t, z ;

    Incremental inverse update $K_{\mathcal{Z}\mathcal{Z}}^{-1}$ with z;

    Update $\Gamma_{t+1} \leftarrow [\Gamma_t + r_t K_{\mathcal{Z}}(s_t),\ K_{\mathcal{S}}(z)^\top Y_{\mathcal{S}}]^\top$

  end

end

dictionary did not change, and we then perform a Schur complement update using the latter inverse. Both updates are done in $\mathcal{O}(m_t^2)$ operations.

As for the inverse of the projection gram matrix, we use a Schur complement update in $\mathcal{O}(m_t^2)$ operations that we detail here for $K^{-1}_{\mathcal{Z}_{t+1}\mathcal{Z}_{t+1}}$:

$$K^{-1}_{\mathcal{Z}_t\mathcal{Z}_t} = \begin{bmatrix} K^{-1}_{\mathcal{Z}_{t-1}\mathcal{Z}_{t-1}} + \frac{1}{\omega} w_t w_t^\top & -\frac{1}{\omega} w_t \\ -\frac{1}{\omega} w_t^\top & \frac{1}{\omega} \end{bmatrix}$$

where $\omega = K(z_{m_t}, z_{m_t}) - K_{\mathcal{Z}_{t-1}}(z_{m_t})^\top K^{-1}_{\mathcal{Z}_{t-1}\mathcal{Z}_{t-1}} k_{\mathcal{Z}_{t-1}}(z_{m_t})$ and with $w_t = K^{-1}_{\mathcal{Z}_{t-1}\mathcal{Z}_{t-1}} K_{\mathcal{Z}_{t-1}}(z_{m_t})$.

4.12.3. Kernel Online Row Sampling (KORS) Subroutine

As in Calandriello et al. (2017b), let us define a projection dictionary $\mathcal{Z}_t$ as a collection of

indexed anchor points $\{(z_{t_i}\}_{1\leq i\leq m_t}$ where $m_t = |\mathcal{Z}_t|$ as well as the rescaling diagonal matrix $S_{\mathcal{Z}_t}$ with $1/\sqrt{\tilde{p}_{z_s}}$ corresponding to the past sampling probabilities of points $z \in \mathcal{Z}_t$, this matrix is of size $m_t \times m_t$. At each time step, KORS temporarily adds t with weight 1 to the temporary dictionary $\mathcal{Z}_t^*$ and accordingly augments the corresponding matrix $S_{\mathcal{Z}_t^*}$. The augmented dictionary is then used to compute the ridge leverage score (RLS) estimator:

$$\tilde{\tau}_t = \frac{1+\varepsilon}{\mu}\left(K(s_t,s_t) - K_{\mathcal{Z}_t^*}(s_t)^\top S_{\mathcal{Z}_t^*}(S_{\mathcal{Z}_t^*}^\top K_{\mathcal{Z}_t^*\mathcal{Z}_t^*} S_{\mathcal{Z}_t^*} + \mu I)^{-1} S_{\mathcal{Z}_t^*}^\top K_{\mathcal{Z}_t^*}(s_t)\right) . \tag{4.41}$$

Afterward, it draws a Bernoulli random variable z_t proportionally to $\tilde{\tau}_t$, if it succeeds, ($z_t = 1$) the point is deemed relevant and added to the dictionary, otherwise it is discarded and never added.

Algorithm 13: Incremental Kernel Online Row Sampling (KORS) subroutine

Input: Time t, past dictionary $\mathcal{Z}$, context-action s_t, regularization μ, accuracy ε, budget γ

Compute the leverage score $\tilde{\tau}_t$ from $\mathcal{Z}, s_t, \mu, \varepsilon$;

Compute $\tilde{p}_t = \min\{\gamma\tilde{\tau}_t, 1\}$;

Draw $z_t \sim \mathcal{B}(\tilde{p}_t)$ and if $z_t = 1$, add s_t to $\mathcal{Z}$;

Result: Dictionary $\mathcal{Z}$

Here, all rows and columns for which $S_{t,*}$ is zero (all points outside the temporary dictionary $\mathcal{I}_{t,*}$) do not influence the estimator, so they can be excluded from the computation. As a consequence, the RLS score $\tilde{\tau}_t$ can be computed efficiently in $\mathcal{O}((m_t+1)^2)$ space and $\mathcal{O}((m_t+1)^2)$ time, using an incremental update in Eq. (4.41).

As a side note, the quantity $\tilde{\tau}$ is an estimator of the exact RLS quantity τ_t (see Calandriello et al. (2017b)):

$$\tau_t = \varphi_t^\top (K_t + \mu I)^{-1}\varphi_t \, . \tag{4.42}$$

Here, leverage scores are used to measure the correlation between the new point φ_t w.r.t. the previous $t-1$ points $\{\varphi_i\}_{i\leq t-1}$, and therefore how essential it is in characterizing the dataset. In particular, if φ_t is completely orthogonal to the other points, its RLS is maximized, while in the opposite case it would be minimal. In the incremental strategy of the Nyström dictionary building, we use the RLS estimates to add anchor points that are as informative as possible.

4.13. Experiment details

In this section we provide further details as well as additional discussions and numerical results on our proposed method.

4.13.1. Reproducibility and Implementations

We provide code that is accessible at the link `https://github.com/criteo-research/Efficient-Kernel-UCB`. All experiments were run on a single CPU core (2 x Intel(R) Xeon(R)

Gold 6146 CPU@ 3.20GHz).

Baseline implementations We implemented the BKB and BBKB algorithms in (Calandriello et al., 2019) and (Calandriello et al., 2020) by introducing modifications in their implementation to handle contextual information. For both methods, in the contextual variant, each update involves the computation of a new covariance matrix $K_{\mathcal{Z}\mathcal{S}}K_{\mathcal{S}\mathcal{Z}}$ while the original algorithms do not involve contexts and consider a finite set of actions, allowing to compute the covariance matrix on the finite set of actions (which is done for computational efficiency and is impossible in the joint context-action space). The baselines were carefully optimized using the Jax library (`https://github.com/google/jax`) to allow for just in time compilations of similar blocks in every methods.

Empirical setting In our empirical setting we aimed at showing the regret/computational complexity compromise that is achieved by each method. In particular, both the CBBKB method (Calandriello et al., 2020) and our EK-UCB algorithm use additional hyperparameters than the CBKB. As a matter of fact, CBBKB uses an accumulation threshold C and is used for the 'resparsification' step, with dictionary updates based on all historical states. EK-UCB also uses the hyperparameter μ in KORS that is set to λ for optimal regret-time compromise (see Theorem 4.4.1). The KORS algorithm uses a budget parameter γ, for which we found empirically good performances when $\gamma \approx \lambda$. We tried our method with a grid on hyperparameters and discuss their influence in the next subsection.

4.13.2. Additional Results

In this section we provide additional numerical experiment discussions.

Additional discussions on the setting of Section 4.5

We present additional results on the synthetic setting presented in Section 4.5 that we call 'Bump' in Figures 4.3, 4.4, 4.5. Here we fix $\lambda = \mu$ for EK-UCB and report the performances of the baselines with the same hyperparameters and make the accumulation threshold C of CBBKB vary through the Figures 4.3, 4.4, 4.5. We provide more discussion on the methods we evaluated.

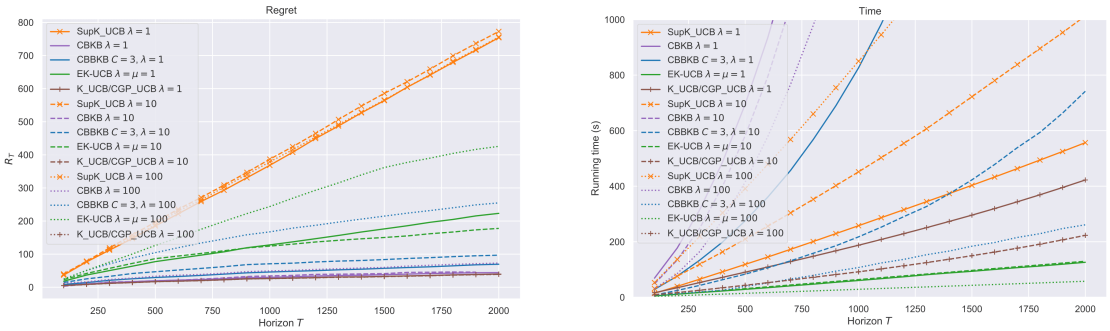

Figure 4.3: 'Bump' setting: Regret and running times of EK-UCB, CBBKB and CBKB, with $T = 2000$ and $\lambda = \mu$ (see Corollary 4.3.1 and 4.4.1) with varying λ and $C = 3$ for CBBKB. EK-UCB matches the best regret-time compromise.

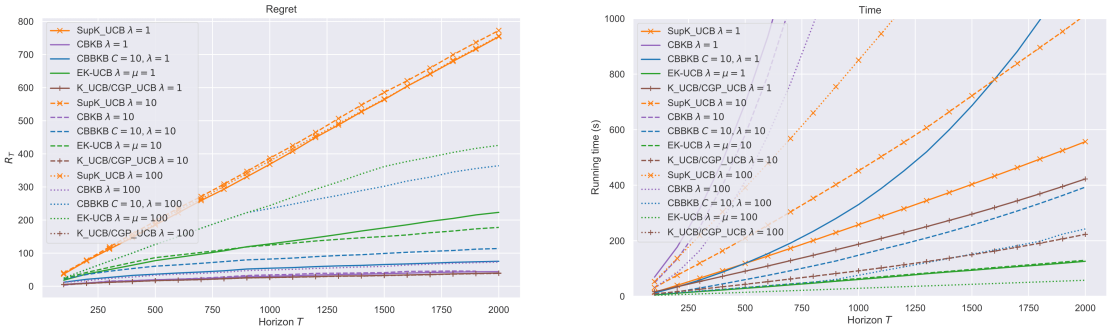

Figure 4.4: 'Bump' setting: Regret and running times of EK-UCB, CBBKB and CBKB, with $T = 2000$ and $\lambda = \mu$ with varying λ and $C = 10$ for CBBKB. EK-UCB matches the best regret-time compromise.

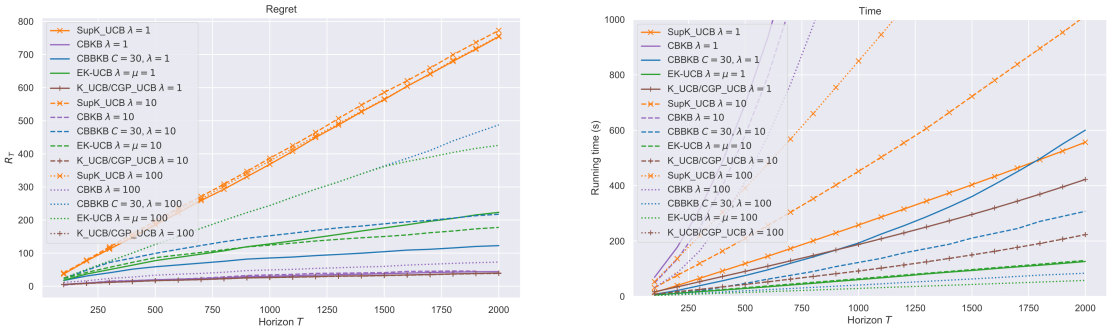

Figure 4.5: 'Bump' setting: Regret and running times of EK-UCB, CBBKB and CBKB, with $T = 2000$ and $\lambda = \mu$ with varying λ and $C = 30$ for CBBKB. EK-UCB matches the best regret-time compromise

More dictionary updates lead to better regret but a higher computational complexity We note that the CBKB baseline achieves satisfactory regret but with a drastically higher computational time. This is due to the fact that it resamples the dictionary at each step and therefore resamples a dictionary at the price of a higher time complexity. As for CBBKB, throughout the Figures 4.3, 4.4, 4.5, we can see that the accumulation threshold C that controls the anchor point update frequency determines the regret-time compromise. The lower C, the better is the regret but the higher is the computational time. We can see through the figures that for all values of C, our EK-UCB method achieves similar or better (especially when $C = 30$) regret than CBBKB while always being both faster than CBBKB but more importantly faster than K-UCB. Overall, EK-UCB proposes the most satisfactory regret-time compromise. Moreover, we see that the SupK-UCB method also performs poorly even with different parameters λ and that the optimized K-UCB method also performs better than efficient strategies when the computational overheads of dictionary buildings overtake the efficient kernel approximations.

The regularization parameter controls the regret-time comprise in EK-UCB In our method, we can see that the higher λ (with $\lambda = \mu$) the faster the algorithm is but the worse is its regret. As discussed in Corollary 4.3.1 and 4.4.1, we use the heuristic to take $\lambda \approx \sqrt{T}$ and set $\mu = \lambda$ afterwards to enjoy the optimal guarantees of our algorithm.

Additional synthetic settings

In this section we introduce additional settings that we call the 'Chessboard' setting as well as the 'Step Diagonal' setting. The two settings lead to similar numerical conclusions as the previous one. We provide more discussions here.

Chessboard and Step Diagonal synthetic setups. The 'Chessboard' synthetic setup is a contextual environment with a piecewise reward function over the joint context-action space $\mathcal{X} \times \mathcal{A} = [0, 1] \times [0, 1]$. More precisely, the joint 2D space is cut into a grid where the values are either 1, 0.5 or 0 according to the part of the grid. Results are shown in Figures 4.7, 4.8, 4.9. The 'Step diagonal' synthetic setup is a contextual environment with a diagonal reward function over the joint context-action space $\mathcal{X} \times \mathcal{A} = [0, 1] \times [0, 1]$. More precisely, the joint 2D space has values of 0 everywhere except along two bands along the diagonal where the action and context values are identical with values 0.5 and 1 respectively on the sub diagonal and the above diagonal. Results are shown in Figures 4.10, 4.11, 4.12. See the code for more details and an illustration of the settings in Fig 4.6.

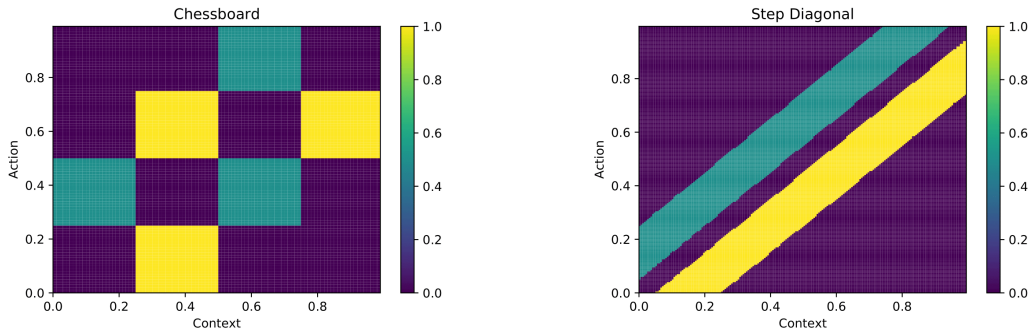

Figure 4.6: Chessboard (left) and Step Diagonal (right) synthetic setups.

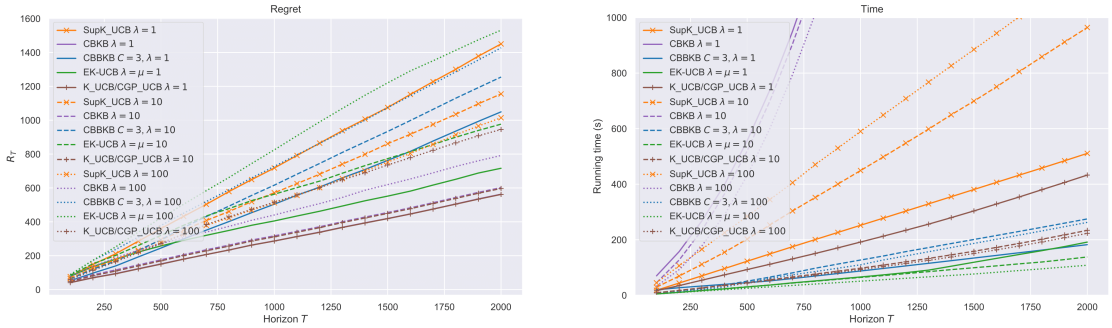

Figure 4.7: 'Chessboard' setting: Regret and running times of EK-UCB, CBBKB and CBKB, with $T = 2000$ and $\lambda = \mu$ with varying λ and $C = 3$ for CBBKB.

Regret-time compromise for CBBKB and EK-UCB. The two settings show what both algorithms CBBKB and EK-UCB achieve as a regret-time compromise. In cases where C is lower (note that CBKB corresponds to CBBKB with $C = 1$) the regret often decreases at the price of higher computational time complexity. Similarly, we can notice that our method has better regrets when λ is low, but with higher computational times, while still

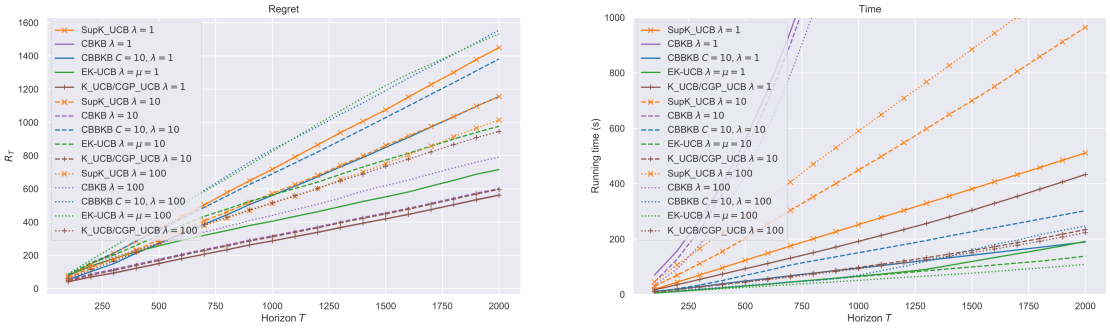

Figure 4.8: 'Chessboard' setting: Regret and running times of EK-UCB, CBBKB and CBKB, with $T = 2000$ and $\lambda = \mu$ with varying λ and $C = 10$ for CBBKB.

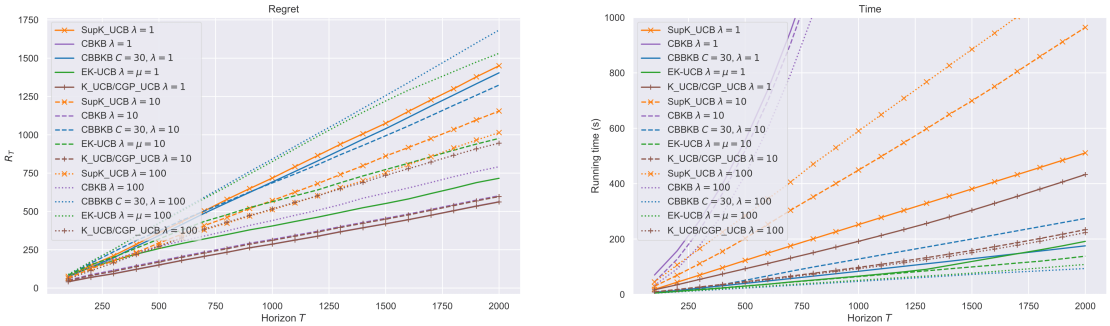

Figure 4.9: 'Chessboard' setting: Regret and running times of EK-UCB, CBBKB and CBKB, with $T = 2000$ and $\lambda = \mu$ with varying λ and $C = 30$ for CBBKB.

providing a benefit over to the K-UCB method, unlike CBBKB. We therefore note again that in practice, dictionary building computational overheads may influence the global computational complexity. Overall, our method with its incremental dictionary building strategy achieves the best satisfactory time-regret compromises in the Chessboard and Step Diagonal settings compared to both K-UCB and the efficient algorithms CBKB and CBBKB.

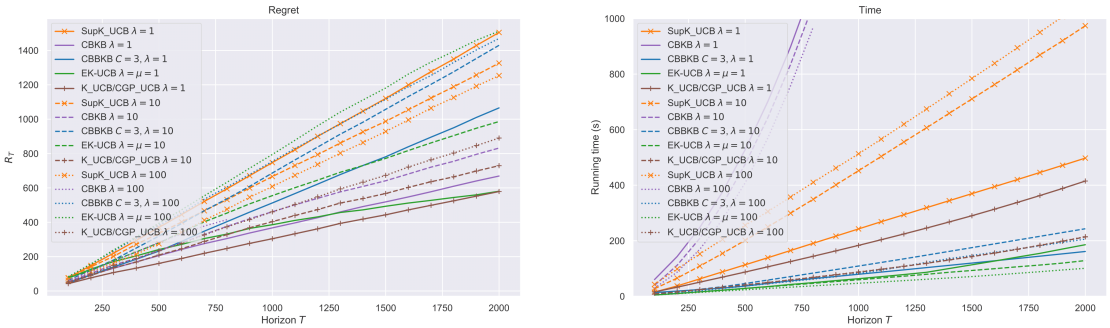

Figure 4.10: 'Step Diagonal' setting: Regret and running times of EK-UCB, CBBKB and CBKB, with $T = 2000$ and $\lambda = \mu$ with varying λ and $C = 3$ for CBBKB.

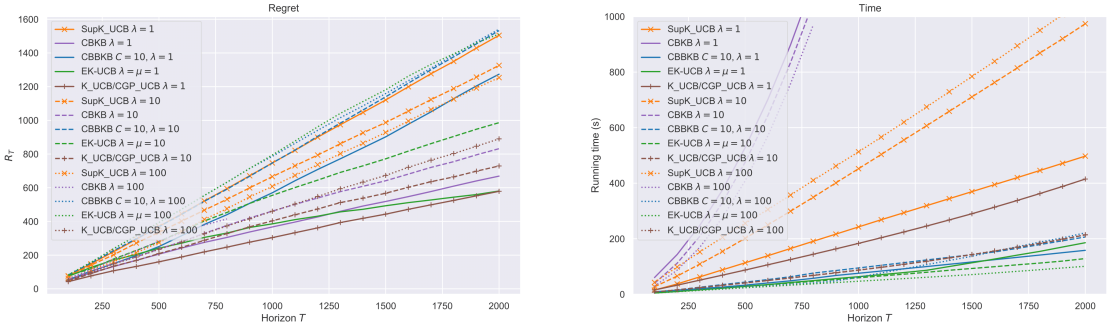

Figure 4.11: 'Step Diagonal' setting: Regret and running times of EK-UCB, CBBKB and CBKB, with $T = 2000$ and $\lambda = \mu$ with varying λ and $C = 10$ for CBBKB.

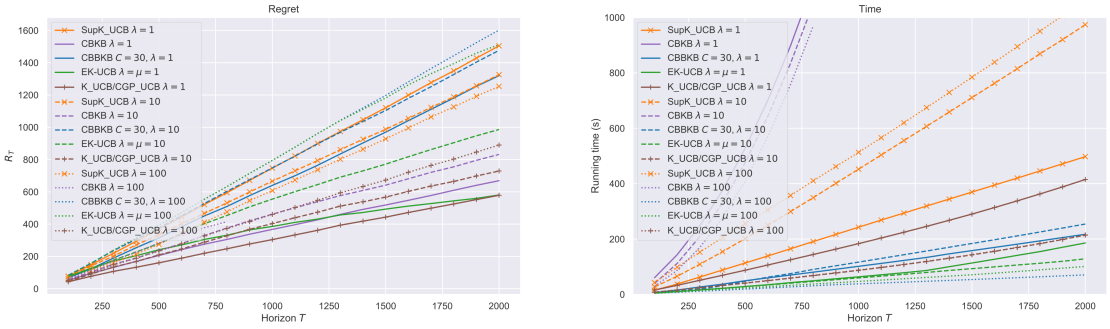

Figure 4.12: 'Step Diagonal' setting: Regret and running times of EK-UCB, CBBKB and CBKB, with $T = 2000$ and $\lambda = \mu$ with varying λ and $C = 30$ for CBBKB.

5

Nested Bandits

In various sequential decision-making processes, the learning agent often faces the challenge of choosing from a high number of alternatives that share many similarities. These similarities can result in closely correlated losses, which can make the use of traditional discrete choice models and bandit algorithms less efficient. In the presence of a similarity structure with a hierarchy of embedded (non-combinatorial) similarities, we tackle the problem of adversarial multi-armed bandits where the learner seeks to minimize their regret. In that context, well-known optimal algorithms based on exponential weights (like Hedge, EXP3, and their variants) may incur unnecessary regret because they tend to spend excessive time exploring irrelevant alternatives that have similar but suboptimal costs. To address this problem, we propose in this chapter the *nested exponential weights* (NEW) algorithm and the *exponential weights with experts and nesting* (EWEN) algorithm with expert advice. Both algorithms employ a layered exploration of the learner's set of alternatives based on a nested, step-by-step selection method where we assume that the learner observes the "intra-class" losses of their chosen alternative. We then establish a series of tight regret bounds for the learner's regret, demonstrating that online learning problems characterized by a high degree of similarity between alternatives can be efficiently resolved with improved dependencies on the alternative set.

This chapter is based on the following material and the manuscript in preparation:

M. Martin, P. Mertikopoulos, T. Rahier, and H. Zenati. Nested bandits. *International Conference on Machine Learning (ICML)*, 2022

H. Zenati, T. Rahier, M. Martin, and P. Mertikopoulos. Sequential Decision Processes with Outcome Similarities

5.1. Introduction

Sequential decision making methods has ubiquitous in many applications such as online recommendation systems (Li et al., 2010). At each round, an agent chooses an alternative; then, the environment generates a cost based on it. The goal of the agent is to minimize the cumulative regret over time, which requires a careful balancing between exploitation (minimizing costs using past observations) and exploration (increasing the diversity of observations). Typically, when the alternative set is large and has an inherent structure which result in outcome similarities, decision models can incur pointless regret.

Consider for example the following case (known as the "red bus / blue bus paradox" in the context of transportation economics). A commuter has a choice between taking a car or a bus to work: commuting by car takes on average half an hour modulo random fluctuations, whereas commuting by bus takes an hour, again modulo random fluctuations (it's a long commute). Then, under the classical multinomial logit choice model for action selection Luce (1959); McFadden (1974), the commuter's odds for selecting a car over a bus would be $\exp(-1/2)/\exp(-1) \approx 1.6 : 1$. This indicates a very clear preference for taking a car to work and is commensurate with the fact that, on average, commuting by bus takes twice as long.

Consider now the same model but with a twist. The company operating the bus network purchases a fleet of new buses that are otherwise completely identical to the existing ones, except for their color: old buses are red, the new buses are blue. This change has absolutely no effect on the travel time of the bus; however, since the new set of alternatives presented to the commuter is $\{\text{car}, \text{red bus}, \text{blue bus}\}$, the odds of selecting a car over a bus (red or blue, it doesn't matter) now drops to $\exp(-1/2)/[\exp(-1) + \exp(-1)] \approx 0.8 : 1$. Thus, by introducing an *irrelevant* feature (the color of the bus), the odds of selecting the alternative with the highest utility have dropped dramatically, to the extent that commuting by car is no longer the most probable choice in this example.

Of course, the shift in choice probabilities may not always be that dramatic, but the point of this example is that the presence of an irrelevant alternative (the blue bus) would always induce such a shift – which is, of course, absurd. In fact, the red bus / blue bus paradox was originally proposed as a sharp criticism of the independence from irrelevant alternatives (IIA) axiom that underlies the multinomial logit choice model (Luce, 1959) and which makes it unsuitable for choice problems with inherent similarities between different alternatives. In turn, this has led to a vast corpus of literature in social choice and decision theory, with an extensive array of different axioms and models proposed to overcome the failures of the IIA assumption. For an introduction to the topic, we refer the reader to the masterful accounts of McFadden (1974), Ben-Akiva and Lerman (1985) and Anderson et al. (1992).

Perhaps surprisingly, the implications of the red bus / blue bus paradox have not been explored in the context of online learning, despite the fact that similarities between alternatives are prevalent in the field's application domains – for example, in recommender systems with categorized product recommendation catalogues, in the economics of transport and product differentiation, etc. What makes this gap particularly pronounced is the fact that logit choice underlies some of the most widely used algorithmic schemes for learning in multi-armed bandit problems – namely the exponential weights algorithm for exploration

and exploitation (EXP3) (Vovk, 1990; Littlestone and Warmuth, 1994; Auer et al., 1995) as well as its variants, Hedge (Auer et al., 2002), EXP3.P (Auer et al., 2003), EXP3-IX (Kocák et al., 2014), EXP4 (Auer et al., 2003) / EXP4-IX (Neu, 2015), etc. Thus, given the vulnerability of logit choice to irrelevant alternatives, it stands to reason that said algorithms may be suboptimal when faced with a set of alternatives with many inherent similarities.

Contributions. This chapter examines this question in the context of repeated decision problems where a learner seeks to minimize their regret in the presence of a large number of distinct alternatives with a hierarchy of embedded (non-combinatorial) similarities. This similarity structure, which we formalize in section 5.2, is defined in terms of a nested series of attributes – like "type" or "color" – and induces commensurate similarities to the losses of alternatives that lie in the same class (just as the red and blue buses have identical losses in the example described above).

Inspired by the nested logit choice model introduced by McFadden (1974) to resolve the original red bus / blue bus paradox, we develop in section 5.4.2 a *nested exponential weights* (NEW) algorithm for no-regret learning in decision problems of this type. The result for this algorithm is that the regret incurred by NEW is bounded as $\mathcal{O}(\sqrt{k_{\text{eff}} \log k \cdot T})$, where k is the total number of alternatives.

In the presence of expert advice, we introduce in section 5.5 the *exponential weights with experts and nesting* (EWEN) algorithm to learn a strategy on experts. The latter achieves a regret bounded as $\mathcal{O}(\sqrt{k_{\text{eff}} \log M \cdot T})$, where M is the total number of experts and k_{eff} is the "effective" number of alternatives when taking similarities into account (for example, in the standard red bus / blue bus paradox, $k_{\text{eff}} = 2$, cf. section 5.2.2).

For both algorithms, the gap between nested and non-nested algorithms can be quantified by the problem's *price of affinity* (PoAf), defined here as the ratio $\alpha = \sqrt{k/k_{\text{eff}}}$. This ratio measures the worst-case ratio between the regret guarantees of the NEW and the EXP3 algorithms (scaling as $\mathcal{O}(\sqrt{k \log k \cdot T})$ in the problem at hand) as well as between the EWEN and exponential weights algorithm for exploration and exploitation with experts (EXP4) (the latter scaling as $\mathcal{O}(\sqrt{k \log M \cdot T})$).

In practical applications (such as the type of recommendation problems that arise in online advertising), α can be exponential in the number of attributes, indicating that our proposed algorithms could lead to significant performance gains in this context. We verify that this is indeed the case in a range of synthetic experiments in section 5.7.

Related Work. The problem of exploiting the structure of the loss model and/or any side information available to the learner is a staple of the bandit literature. More precisely, in the setting of contextual bandits, the learner is assumed to observe some "context-based" information and tries to learn the "context to reward" mapping underlying the model in order to make better predictions. Bandit algorithms of this type – like EXP4 – are often studied as "expert" models (Auer et al., 2003; Cesa-Bianchi and Lugosi, 2006) or attempt to model the agent's loss function with a semi-parametric contextual dependency in the stochastic setting to derive optimistic action selection rules (Abbasi-yadkori et al., 2011); for a survey, we refer the reader to Lattimore and Szepesvári (2020) and references therein. While the nested

bandit model we study assumes an additional layer of information relative to standard bandit models, there are no experts or a contextual mapping conditioning the action taken, so it is not comparable to the contextual setup.

The type of feedback we consider assumes that the learner observes the "intra-class" losses of their chosen alternative, similar to the semi-bandit in the study of combinatorial bandit algorithms Cesa-Bianchi and Lugosi (2012); György et al. (2007). However, the similarity with combinatorial bandit models ends there: even though the categorization of alternatives gives rise to a tree structure with losses obtained at its leaves, there is no combinatorial structure defining these costs, and modeling this as a combinatorial bandit would lead to the same number of arms and ground elements, thus invalidating the concept.

Besides these major threads in the literature, (Thune and Seldin, 2018) recently showed that the range of losses can be exploited with an additional free observation, while (Cesa-Bianchi and Shamir, 2018) improves the regret guarantees by using effective loss estimates. However, both works are susceptible to the advent of irrelevant alternatives and can incur significant regret when faced with such a problem. Finally, in the Lipschitz bandit setting, (Cesa-Bianchi et al., 2017; Héliou et al., 2021) obtain order-optimal regret bounds by building a hierarchical covering model in the spirit of Bubeck et al. (2011); the correlations induced by a Lipschitz loss model cannot be compared to our model, so there is no overlap of techniques or results.

5.2. Similarity structures: the general model

We begin in this section by defining our general nested choice model. Because the technical details involved can become cumbersome at times, it will help to keep in mind the running example of a music catalogue where songs are classified by, say, genre (classical music, jazz, rock,. . .), artist (Rachmaninov, Miles Davis, Led Zeppelin,. . .), and album. This is a simple – but not simplistic – use case which requires the full capacity of our model, so we will use it as our "go-to" example throughout.

5.2.1. Attributes, classes, and the relations between them

Let $\mathcal{A} = \{a_i : i = 1, \dots, k\}$ be a set of *alternatives* (or *atoms*) indexed by $i = 1, \dots, k$. A *similarity structure* (or *structure of attributes*) on $\mathcal{A}$ is defined as a tower of nested *similarity partitions* (or *attributes*) $\mathcal{S}_\ell$, $\ell = 0, \dots, L$, of $\mathcal{A}$ with $\{\mathcal{A}\} =: \mathcal{S}_0 \succcurlyeq \mathcal{S}_1 \succcurlyeq \cdots \succcurlyeq \mathcal{S}_L := \{\{a\} : a \in \mathcal{A}\}$. As a result of this definition, each partition $\mathcal{S}_\ell$ captures successively finer attributes of the elements of $\mathcal{A}$ (in our music catalogue example, these attributes would correspond to genre, artist, album, etc.).[1] Accordingly, each constituent set A of a partition $\mathcal{S}_\ell$ will be referred to as a *similarity class* and we assume it collects all elements of $\mathcal{A}$ that share the attribute defining $\mathcal{S}_\ell$: for example, a similarity class for the attribute "artist" might consist of all Beethoven symphonies, all songs by Led Zeppelin, etc.

[1]The trivial partitions $\mathcal{S}_0 = \{\mathcal{A}\}$ and $\mathcal{S}_L = \{\{a\} : a \in \mathcal{A}\}$ do not carry much information in themselves, but they are included for completeness and notational convenience later on.

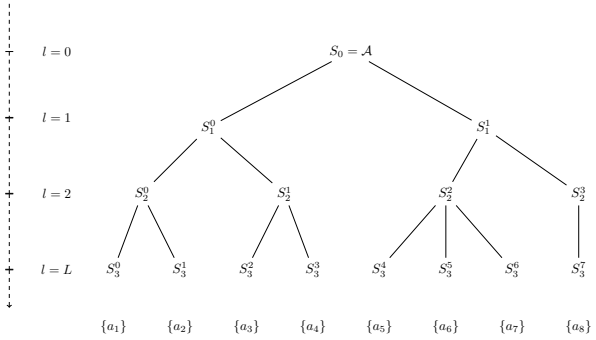

Figure 5.1: A structure with $L = 3$ attributes on the set $\mathcal{A} = \{a_1, \ldots, a_8\}$; for example, the class S_2^1 consists of $\{a_3, a_4\}$.

Collectively, a structure of attributes will be represented by the disjoint union

$$\mathcal{S} := \coprod_{\ell=0}^{L} \mathcal{S}_\ell \equiv \bigcup_{\ell=0}^{L} \{(A, \ell) : A \in \mathcal{S}_\ell\} \tag{5.1}$$

of all class/attribute pairs of the form (A, ℓ) for $A \in \mathcal{S}_\ell$. In a slight abuse of terminology (and when there is no danger of confusion), the pair $S = (A, \ell)$ will also be referred to as a "class", and we will write $S \in \mathcal{S}_\ell$ and $a \in S$ instead of $A \in \mathcal{S}_\ell$ and $a \in A$ respectively. By contrast, when we need to clearly distinguish between a class and its underlying set, we will write $A = \mathrm{elem}S$ for the set of atoms contained in S and $\ell = \mathrm{attr}S$ for the attached attribute label.

Remark 5.2.1. *The reason for including the attribute label ℓ in the definition of $\mathcal{S}$ is that a set of alternatives may appear in different partitions of $\mathcal{A}$ in a different context. For example, if "IV" is the only album by Led Zeppelin in the catalogue, the album's track list represents both the set of "all songs in IV" as well as the set of "all Led Zeppelin songs". However, the focal attribute in each case is different – "artist" in the former versus "album" in the latter – and this additional information would be lost in the non-discriminating union $\bigcup_{\ell=0}^{L} \mathcal{S}_\ell$ (unless, of course, the partitions $\mathcal{S}_\ell$ happen to be mutually disjoint, in which case the distinction between "union" and "disjoint union" becomes set-theoretically superfluous).* ¶

Moving forward, if a class $S \in \mathcal{S}_\ell$ contains the class $S' \in \mathcal{S}_k$ for some $k > \ell$, we will say that S' is a *descendant* of S (resp. S is an *ancestor* of S'), and we will write "$S' \prec S$" (resp. "$S \succ S'$").[2] As a special case of this relation, if $S' \prec S$ and $k = \ell + 1$, we will say that S' is a *child* of S (resp. S is *parent* of S') and we will write "$S' \lhd S$" (resp. "$S \rhd S'$"). For completeness, we will also say that S' and S'' are *siblings* if they are children of the same parent, and we will write $S' \sim S''$ in this case. Finally, when we wish to focus on descendants sharing a certain attribute, we will write "$S' \prec_\ell S$" as shorthand for the predicate "$S' \prec S$ and $\mathrm{attr}S' = \ell$".

Building on this, a similarity structure on $\mathcal{A}$ can also be represented graphically as a rooted directed tree – an *arborescence* – by connecting two classes $S, S' \in \mathcal{S}$ with a directed

[2]More formally, we will write $S' \prec S$ when $\mathrm{elem}S' \subseteq \mathrm{elem}S$ and $\mathrm{attr}S' > \mathrm{attr}S$. The corresponding weak relation "$\preccurlyeq$" is defined in the standard way, i.e., allowing for the case $\mathrm{attr}S' = \mathrm{attr}S$ which in turn implies that $S' = S$.

edge $S \to S'$ whenever $S \rhd S'$. By construction, the root of this tree is $\mathcal{A}$ itself,[3] and the unique directed path $\mathcal{A} \equiv S_0 \rhd S_1 \rhd \cdots \rhd S_\ell \equiv S$ from $\mathcal{A}$ to any class $S \in \mathcal{S}$ will be referred to as the *lineage* of S. For notational simplicity, we will not distinguish between $\mathcal{S}$ and its graphical representation, and we will use the two interchangeably; for an illustration, see figure 5.1.

5.2.2. Loss model

Throughout what follows, we will consider loss models in which alternatives that share a common set of attributes incur similar costs, with the degree of similarity depending on the number of shared attributes. More precisely, given a similarity class $S \in \mathcal{S}$, we will assume that all its immediate subclasses S' share the same base cost c_S (determined by the parent class S) plus an idiosyncratic cost increment $r_{S'}$ (which is specific to the child $S' \lhd S$ in question). Formally, starting with $c_{\mathcal{A}} = 0$ (for the root class $\mathcal{A}$), this boils down to the recursive definition

$$c_{S'} = c_S + r_{S'} \quad \text{for all } S' \lhd S, \tag{5.2}$$

which, when unrolled over the lineage $\mathcal{A} \equiv S_0 \rhd S_1 \rhd \cdots \rhd S_\ell \equiv S$ of a target class $S \in \mathcal{S}_\ell$, yields the expression

$$c_S = \sum\nolimits_{S' \trianglerighteq S} r_{S'} = r_{S_1} + \cdots + r_{S_\ell}. \tag{5.3}$$

Thus, in particular, when $S \leftarrow a \in \mathcal{A}$, the cost assigned to an individual alternative $a \in \mathcal{A}$ will be given by

$$c_a = \sum\nolimits_{\ell=1}^{L} r_{S_\ell} = \sum\nolimits_{S \ni a} r_S \quad \text{for all } a \in \mathcal{A}. \tag{5.4}$$

Finally, to quantify the "intra-class" variability of costs, we will assume throughout that the idiosyncratic cost increments within a given parent class S are bounded as

$$r_{S'} \in [0, \Gamma_S] \quad \text{for all } S' \lhd S. \tag{5.5}$$

This terminology is justified by the fact that, under the loss model (5.2), the costs $c_{S'}, c_{S''}$ to any two *sibling* classes $S', S'' \lhd S$ (i.e., any two classes parented by S) differ by at most Γ_S. Analogously, the costs to any two alternatives $a, a' \in \mathcal{A}$ that share a set of common attributes $S_1, \ldots, S_\ell$ will differ by at most $\sum_{k=\ell+1}^{L} \Gamma_{S_k}$.

Example 5.2.1. *To represent the original red bus /blue bus problem as an instance of the above framework, let $\mathcal{S}_1 = \{\{$red bus, blue bus$\}, \{$car$\}\}$ be the partition of the set $\mathcal{A} = \{$red bus, blue bus, car$\}$ by type ("`bus`" or "`car`"), and let $\mathcal{S}_2$ be the corresponding sub-partition by color ("`red`" or "`blue`" for elements of the class "`bus`"). The fact that color does not affect travel times may then be represented succinctly by taking $\Gamma_{color} = 0$.* ¶

Remark 5.2.2. *We make no distinction here between c_a and $c_{\{a\}}$, i.e., between an alternative a of $\mathcal{A}$ and the (unique) singleton class of $\{a\} \in \mathcal{S}_L$ containing it. This is done purely for reasons of notational convenience.* ¶

[3] Stricto sensu, the root of the tree is $(\mathcal{A}, 0)$, but since there is no danger of confusion, the attribute label "0" will be dropped.

Remark 5.2.3. *For posterity, we also note that the optimizing agent is assumed to be aware of the cost decomposition* (5.4) *after selecting an alternative $a \in \mathcal{A}$. In the context of combinatorial bandits Cesa-Bianchi and Lugosi (2012) this would correspond to the so-called "semi-bandit" setting.* ¶

To align our presentation with standard bandit models with losses in $[0, 1]$, we will assume throughout that for an alternative $a \in \mathcal{A}$, we have $\sum_{S \ni a} \Gamma_S \leq 1$ for all $a \in \mathcal{A}$, meaning in particular that the maximal cost incurred by any alternative $a \in \mathcal{A}$ is upper bounded by 1. Other than this normalization, the sequence of idiosyncratic loss vectors $r_t \in \mathbb{R}^{\mathcal{S}}$, $t = 1, 2, \ldots$, is assumed arbitrary and unknown to the learner as per the standard adversarial setting Cesa-Bianchi and Lugosi (2006); Shalev-Shwartz (2012).

5.2.3. Contrasting with other similarity structure models

In this section, we further discuss our similarity structure model with regards to bandit methods that aim to model similarities.

Combinatorial bandits Consider a class of learning models with an underlying combinatorial structure in the spirit of Cesa-Bianchi and Lugosi (2009, 2012). In this class of problems, a participating agent selects a specific combination $A \subseteq \mathcal{A}$ from a set $\mathcal{A} = \{a_1, \ldots, a_k\}$ of k possible resources (a set of congestible facilities, the edges of a path in a network routing problem, etc.). Then, every agent receives as a reward the aggregate payoff of the utilized resources (which, depending on the context, may be a function of the number of agents employing it or other, exogenous factors). Instead, our method involves a hierarchical decision-making process where the available actions are structured in a nested but *not necessarily combinatorial* manner. Indeed, the choice of a ℓ-level class $S_\ell \in \mathcal{S}_\ell$ determines a specific set of child classes $S_{\ell+1} \lhd S_\ell$ from which the agent must choose next. This set of child classes is in general exclusive to that parent class, as is the case in the genre/artist/album example. For instance, "Dark Side of the Moon" is only available in the graph through the parent class "Pink Floyd", preventing this example to be modeled in a combinatorial manner.

Graph bandits Graph bandits (Valko, 2016) are class of bandit problems where the actions in $\mathcal{A}$ are the vertices of a graph $G = (\mathcal{A}, E)$ which edges symbolize similarities between their corresponding losses. At each round t the player observes losses in the neighbourhood of each arm (Mannor and Shamir, 2011; Alon et al., 2013; Kocák et al., 2014), and thus exploits the structure of the graph by estimating the expected loss of each node. Instead of the regret bound $\mathcal{O}(\sqrt{k \log k \cdot T})$, such methods achieve a regret of $\mathcal{O}(\sqrt{\widetilde{k_{\text{eff}}} \log k \cdot T})$ where $\widetilde{k_{\text{eff}}}(G) \in [k]$ is the independence number of G which is smaller as G becomes denser. Our work first differs from the graph bandit setup because only the leaves from the graph obtained in the similarity structure (such as the one displayed in figure 5.1) are the actions that can be selected. Aside from this difference, our setting also uses incremental losses of parent classes up to the leaves. We note however some analogies in the similarity of the losses that can be seen in the distance between action nodes through common ancestors.

5.3. The Nested Importance Weighted Estimator

Before we move forward to the sequential decision processes in Section 5.4, 5.5, we will define a cost estimation procedure for the decisions that are not made by the agents. In both cases, an alternative $\hat{a}$ is either selected by the learner or via the experts. Such a selection is done step by step in the similarity structure $\mathcal{A} \equiv \hat{S}_0 \rhd \hat{S}_1 \rhd \cdots \rhd \hat{S}_L = \{\hat{a}\}$ that we presented in the previous section. A key component for both methods is then to estimate effectively the costs of alternatives that were not chosen. Actually, when given a cost vector $c \in [0,1]^{\mathcal{A}}$ and a mixed strategy $u \in \Delta(\mathcal{A})$ with full support, a standard way to estimate the unselected alternatives $a \in \mathcal{A}$ is via the importance-weighted estimator (Bubeck and Cesa-Bianchi, 2012; Lattimore and Szepesvári, 2020)

$$\hat{c}_a = \frac{\mathbb{1}\{a = \hat{a}\}}{u_a} c_a \tag{IWE}$$

where $\hat{a} \sim u$ is the (random) element of $\mathcal{A}$ chosen under u and thus defines $\hat{c} = (\hat{c}_a)_{a \in \mathcal{A}}$.

This estimator enjoys the following important properties:

1. It is non-negative.
2. It is *unbiased*, i.e.,
$$\mathbb{E}[\hat{c}_a] = c_a \quad \text{for all } a \in \mathcal{A}. \tag{5.6}$$
3. Its *importance-weighted mean square* is bounded as
$$\mathbb{E}\Big[\sum\nolimits_{a \in \mathcal{A}} u_a \hat{c}_a^2\Big] \leq k \tag{5.7}$$

This trifecta of properties plays a key role in establishing the no-regret guarantees of the vanilla exponential weights algorithm Auer et al. (2002); Littlestone and Warmuth (1994); Vovk (1990) and which can then be adapted to the expert variants Auer et al. (2003); McMahan and Streeter (2009); Lattimore and Szepesvári (2020); at the same time however, (IWE) fails to take into account any side information provided by similarities between different elements of $\mathcal{A}$. This is perhaps most easily seen in the original red bus / blue bus paradox: if the commuter takes a red bus, the observed utility would be immediately translatable to the blue bus (and vice versa). However, (IWE) is treating the red and blue buses as unrelated, so the alternative cost $\hat{c}_{\text{blue bus}}$ is not updated under (IWE), even though $c_{\text{blue bus}} = c_{\text{red bus}}$ by default.

To exploit this type of similarities, we introduce below a layered estimator. To define it, let $u \in \Delta(\mathcal{A})$ be a mixed strategy on $\mathcal{A}$ with full support, and assume that an element $\hat{a} \in \mathcal{A}$ is selected progressively according to u; the conditional probabilities $u_{S'|S}$ may of course differ. First, a similarity class $\hat{S}_1 \in \mathcal{S}_1$ is chosen with probability $\mathbb{P}(\hat{S}_1 = S_1) = u_{S_1}$; subsequently, conditioned on the choice of $\hat{S}_1$, a class $\hat{S}_2 \lhd \hat{S}_1$ is selected with probability $\mathbb{P}(\hat{S}_2 = S_2 | \hat{S}_1) = u_{S_2|\hat{S}_1}$, and the process repeats until reaching a leaf $\hat{S}_L = \{\hat{a}\}$ of $\mathcal{S}$ (at which point the selection procedure terminates and returns $\hat{a}$). Then, given a loss profile $r \in [0, +\infty)^{\mathcal{S}}$ and a mixed strategy $u \in \Delta(\mathcal{A})$, the *nested importance weighted estimator* (NIWE) is defined for all $\ell = 1, \ldots, L$ as

$$\hat{r}_{S_\ell} = \frac{\mathbb{1}\{S_\ell = \hat{S}_\ell, \dots, S_1 = \hat{S}_1\}}{u_{S_\ell|S_{\ell-1}} \cdots u_{S_2|S_1} u_{S_1}} r_{S_\ell} \tag{NIWE}$$

where the chain of categorical random variables $\mathcal{A} \equiv \hat{S}_0 \rhd \hat{S}_1 \rhd \cdots \rhd \hat{S}_L = \{\hat{a}\}$ is drawn according to $u \in \Delta(\mathcal{A})$ as outlined above.[4]

This estimator will play a central part in our analysis, so some remarks are in order. First and foremost, the non-nested estimator (IWE) is recovered as a special case of (NIWE) when there are no similarity attributes on $\mathcal{A}$ (i.e., $L = 1$). Second, in a bona fide nested model, we should note that $\hat{c}_{S_\ell}$ is $\hat{S}_\ell$-measurable but *not* $\hat{S}_{\ell-1}$-measurable: this property has no analogue in (IWE), and it is an intrinsic feature of the step-by-step selection process underlying (NIWE). Third, it is also important to note that (NIWE) concerns the idiosyncratic losses of each chosen class, *not* the base costs c_a of each alternative $a \in \mathcal{A}$. This distinction is again redundant in the non-nested case, but it leads to a distinct estimator for c_a in nested environments, namely

$$\hat{c}_a = \sum\nolimits_{S \ni a} \hat{r}_S \quad \text{for all } a \in \mathcal{A}. \tag{5.8}$$

In particular, in the red bus / blue bus paradox, this means that an observation for the class "bus" automatically updates both $\hat{c}_{\text{red bus}}$ and $\hat{c}_{\text{blue bus}}$, thus overcoming one of the main drawbacks of (IWE) when facing irrelevant alternatives.

To complete the comparison with the non-nested setting, we summarize below the most important property of the layered estimator (NIWE):

Proposition 5.3.1. *Let $\mathcal{S} = \coprod_{\ell=1}^{L} \mathcal{S}_\ell$ be a similarity structure on $\mathcal{A}$. Then, given a mixed strategy $u \in \Delta(\mathcal{A})$ and a vector of cost increments $r \in \mathbb{R}^{\mathcal{S}}$ as per* (5.5)*, the estimator* (NIWE) *satisfies the following:*

1. *It is unbiased:*
$$\mathbb{E}[\hat{r}_S] = r_S \quad \textit{for all } S \in \mathcal{S}. \tag{5.9}$$

2. *It enjoys the importance-weighted mean-square bound*
$$\mathbb{E}\big[u_S \hat{r}_S^2\big] \leq \Gamma_S^2 \quad \textit{for all } S \in \mathcal{S}. \tag{5.10}$$

Accordingly, the loss estimator (5.8) *is itself unbiased and enjoys the bound*

$$\mathbb{E}\Big[\sum\nolimits_{a \in \mathcal{A}} u_a \hat{c}_a^2\Big] \leq k_{\text{eff}} \tag{5.11}$$

where k_{eff} is an "effective number of arms" that we define below. We note that proposition 5.3.1 yields the standard properties of (IWE) as a special case when $L = 1$ (in which case there are no similarities to exploit between alternatives). To streamline our presentation, we prove this result in section 5.9.

[4]The indicator in (NIWE) is assumed to take precedence over $u_{S_k|S_{k-1}}$, i.e., $\hat{c}_{S_\ell} = 0$ if $S_k \neq \hat{S}_k$ for some $k = 1, \dots, \ell$.

Definition 5.3.1 (Effective number of arms)**.** *Let us denote the "root-mean-square" range of all classes in $\mathcal{S}_\ell$ as*

$$\bar{\Gamma}_\ell = \sqrt{\frac{1}{k_\ell} \sum\nolimits_{S_\ell \in \mathcal{S}_\ell} \Gamma_{S_\ell}^2}, \tag{5.12}$$

with $k_\ell = |\mathcal{S}_\ell|$ denoting the number of classes of attribute $\mathcal{S}_\ell$. Then, the effective number of arms k_{eff} is defined as

$$k_{\text{eff}} = \left(\sum\nolimits_{\ell=1}^{L} \sqrt{k_\ell} \bar{\Gamma}_\ell \right)^2. \tag{5.13}$$

Note that the notion of outcome similarity that we defined in the previous Section 5.2 is instrumental to understand this definition. Indeed, suppose for example that we have a red bus / blue bus type of problem with, say, $k_1 = 2$ similarity classes, $k_2 = 100$ alternatives per class, and a negligible intra-class loss differential ($\Gamma_2 \approx 0$). Intuitively, it is tempting to say that only $k_{\text{eff}} = 2$ choices are determinant for this problem given the irrelevance of the colors.

5.4. Nested Exponential Weights

In this section we now present the *nested exponential weights* (NEW) algorithm. We will consider the generic online decision process that unfolds over a set of alternatives $\mathcal{A}$ endowed with a similarity structure $\mathcal{S} = \coprod_\ell \mathcal{S}_\ell$ as follows:

1. At each stage $t = 1, 2, \dots$, the learner chooses an alternative $a_t \in \mathcal{A}$ and its choice is made by selecting attributes from $\mathcal{S}$ one-by-one.
2. Concurrently, nature sets the idiosyncratic, intra-class losses $r_{S,t}$ for each similarity class $S \in \mathcal{S}$.
3. The learner incurs $r_{S,t}$ for each chosen class $S \ni a_t$ for a total cost of $c_t = \sum_{S \ni a_t} r_{S,t}$, and the process repeats.

In order to prevent the vulnerability of deterministic strategies that could be exploited by an adversary, the learner chooses an alternative a_t at time t based on a mixed strategy $u_t \in \Delta(\mathcal{A})$, i.e., $a_t \sim u_t$. The regret of a policy u_t, $t = 1, 2, \dots$, against a benchmark strategy $p \in \Delta(\mathcal{A})$ is then defined again as the cumulative difference between the player's mean cost under p and u_t, that is

$$R_T(p) = \sum_{t=1}^{T} [\mathbb{E}_{u_t}[c_{a_t,t}] - \mathbb{E}_p[c_{a_t,t}]] = \sum_{t=1}^{T} \langle c_t, u_t - p \rangle, \tag{5.14}$$

where $c_t = (c_{a,t})_{a \in \mathcal{A}} \in \mathbb{R}^{\mathcal{A}}$ denotes the vector of costs encountered by the learner at time t, i.e., $c_{a,t} = \sum_{S \ni a} r_{S,t}$ for all $a \in \mathcal{A}$. This definition will now be our figure of merit in this section.

5.4.1. The Nested Logit Choice rule

We begin by introducing the attribute selection scheme that forms the backbone of our proposed policy. Our guiding principle in this is the *nested logit choice* (NLC) rule of McFadden (1974) which selects an alternative $a \in \mathcal{A}$ by traversing $\mathcal{S}$ one attribute at a time and prescribing the corresponding conditional choice probabilities at each level of $\mathcal{S}$.

To set the stage for all this, if $u = (u_1, \dots, u_k) \in \Delta(\mathcal{A})$ is a mixed strategy on $\mathcal{A}$ we will write

$$u_S = \sum_{a \in S} u_a \tag{5.15}$$

for the probability of choosing $S \in \mathcal{S}$ under u, and

$$u_{S'|S} = u_{S'}/u_S \tag{5.16}$$

for the conditional probability of choosing a descendant S' of S assuming that S has already been selected under u.[5] Then the NLC rule proceeds as follows: first, it prescribes choice probabilities u_{S_1} for all classes $S_1 \in \mathcal{S}_1$ (i.e., the coarsest ones); subsequently, once a class $S_1 \in \mathcal{S}_1$ has been selected, NLC prescribes the conditional choice probabilities $u_{S_2|S_1}$ for all children S_2 of S_1 and draws a class from $\mathcal{S}_2$ based on $u_{S_2|S_1}$. The process then continues downwards along $\mathcal{S}$ until reaching the finest partition $\mathcal{S}_L$ and selecting an atom $\{a\} \equiv S_L \lhd S_{L-1} \lhd \cdots \lhd S_0 \equiv \mathcal{A}$.

This step-by-step selection process captures the "nested" part of the nested logit choice rule; the "logit" part refers to the way that the conditional probabilities (5.16) are actually prescribed given the agent's predisposition towards each alternative $a \in \mathcal{A}$. To make this precise, suppose that the learner associates to each element $a \in \mathcal{A}$ a *propensity score* $y_a \in \mathbb{R}$ indicating their tendency – or *propensity* – to select it. The associated propensity score of a similarity class $S_{\ell-1} \in \mathcal{S}_{\ell-1}$, $\ell = 1, \dots, L$, is then defined inductively as

$$y_{S_{\ell-1}} = \mu_\ell \log \sum\nolimits_{S_\ell \lhd S_{\ell-1}} \exp(y_{S_\ell}/\mu_\ell) \tag{5.17}$$

where $\mu_\ell > 0$ is a tunable parameter that reflects the learner's *uncertainty level* regarding the ℓ-th attribute $\mathcal{S}_\ell$ of $\mathcal{S}$. In words, this means that the score of a class is the weighted softmax of the scores of its children; thus, starting with the individual alternatives of $\mathcal{A}$ – that is, the *leaves* of $\mathcal{S}$ – propensity scores are propagated backwards along $\mathcal{S}$, and this is repeated one attribute at a time until reaching the root of $\mathcal{S}$.

Remark 5.4.1. *We should also note that Eq.* (5.17) *assigns a propensity score to* any *similarity class $S \in \mathcal{S}$. However, because the primitives of this assignment are the original scores assigned to each alternative $a \in \mathcal{A}$, we will reserve the notation $y = (y_1, \dots, y_k) \in \mathbb{R}^{\mathcal{A}}$ for the* profile *of propensity scores $(y_a)_{a \in \mathcal{A}}$ that comprises the basis of the recursive definition* (5.17). ¶

With all this in hand, given a propensity score profile $y = (y_1, \dots, y_k) \in \mathbb{R}^{\mathcal{A}}$, the *nested logit choice* (NLC) rule is defined via the family of conditional selection probabilities

$$P_{S_\ell|S_{\ell-1}}(y) = \frac{\exp(y_{S_\ell}/\mu_\ell)}{\exp(y_{S_{\ell-1}}/\mu_\ell)} \tag{NLC}$$

where:

[5] Note here that the joint probability of selecting *both* S and S' under u is simply $u_{S'}$ whenever $S' \preccurlyeq S$.

1. $S_\ell \in \mathcal{S}_\ell$ and $S_{\ell-1} \in \mathcal{S}_{\ell-1}$ is a child / parent pair of similarity classes of $\mathcal{S}$.
2. $\mu_1 \geq \cdots \geq \mu_L > 0$ is a nonincreasing sequence of uncertainty parameters (indicating a higher uncertainty level for coarser attributes; we discuss this later).

In more detail, the choice of an alternative $a \in \mathcal{A}$ under (NLC) proceeds as follows: given a propensity score $y_a \in \mathbb{R}$ for each $a \in \mathcal{A}$, every similarity class $S_{L-1} \in \mathcal{S}_{L-1}$ is assigned a propensity score via the recursive softmax expression (5.17), and the same procedure is applied inductively up to the root $\mathcal{A}$ of $\mathcal{S}$. Then, to select an alternative $a \in \mathcal{A}$, the conditional logit choice rule (NLC) proceeds in a top-down manner, first by selecting a similarity class $S_1 \lhd S_0 \equiv \mathcal{A}$, then by selecting a child $S_2 \lhd S_1$ of S_1, and so on until reaching a leaf $\{a\} \equiv S_L \lhd S_{L-1} \lhd \cdots \lhd S_0 \equiv \mathcal{A}$ of $\mathcal{S}$.

Remark 5.4.2. *The previously defined* (NIWE) *shadows the step-by-step selection process of* (NLC).

Equivalently, unrolling (NLC) over the lineage $\mathcal{A} \equiv S_0 \rhd S_1 \rhd \cdots \rhd S_\ell \equiv S$ of a target class $S \in \mathcal{S}_\ell$, we obtain the expression

$$P_S(y) = \prod_{k=1}^{\ell} \frac{\exp(y_{S_k}/\mu_k)}{\exp(y_{S_{k-1}}/\mu_k)} \tag{5.18}$$

for the total probability of selecting class S under the propensity score profile $y \in \mathbb{R}^{\mathcal{A}}$. Clearly, (NLC) and (5.18) are mathematically equivalent, so we will refer to either one as the definition of the nested logit choice rule.

5.4.2. The nested exponential weights algorithm

We are finally in a position to present the *nested exponential weights* (NEW) algorithm in detail. The main ingredients of our method are a cost estimation rule that we described in section 5.3 and the nested attribute selection that we just detailed. Building on the original exponential weights blueprint Littlestone and Warmuth (1994); Auer et al. (2002); Vovk (1990), the main steps of the NEW algorithm can be summed up as follows:

1. For each stage $t = 1, 2, \ldots$, the learner maintains and updates a propensity score profile $y_t \in \mathbb{R}^{\mathcal{A}}$.
2. The learner selects an action $a_t \in \mathcal{A}$ based on the nested logit choice rule $a_t \sim P(\eta_t y_t)$ where $\eta_t \geq 0$ is the method's *learning rate* and P is given by (NLC).
3. The learner incurs $r_{S,t}$ for each class $S \ni a_t$ and constructs a model $\hat{c}_t$ of the cost vector c_t of stage t via (NIWE).
4. The learner updates their propensity score profile based on $\hat{c}_t$ and the process repeats.

For a presentation of the algorithm in pseudocode form, see Algorithm 14; the tuning of the method's uncertainty parameters $\mu_1 \geq \ldots \geq \mu_L > 0$ and the learning rate η_t is discussed in the nexts section, where we undertake the analysis of the NEW algorithm.

5.4.3. Regret guarantees

We are now in a position to state and discuss our main regret guarantees for the NEW algorithm. These are as follows:

Algorithm 14: Nested exponential weights (NEW)

Require: set of alternatives $\mathcal{A}$, attribute partitions $\mathcal{S}_1 \succcurlyeq \cdots \succcurlyeq \mathcal{S}_L$, attribute structure $\mathcal{S} = \coprod_{\ell=1}^{L} \mathcal{S}_\ell$

Input: sequence of class costs $r_t \in [0,1]^{\mathcal{S}}$, $t = 1, 2, \ldots$, uncertainty levels $\mu_1, \ldots, \mu_L > 0$, learning rate $\eta_t \geq 0$

initialize $y \leftarrow 0 \in \mathbb{R}^{\mathcal{A}}$, $S_0 = \mathcal{A}$;
for $t = 1, 2, \ldots$ **do**
  for $\ell = L-1, \ldots, 0$ *and* ***for all*** $S \in \mathcal{S}_\ell$ **do**
    set $y_S \leftarrow \mu_{\ell+1} \log \sum_{S' \lhd S} \exp(y_{S'}/\mu_{\ell+1})$; //as per (5.17)
    set $\hat{r}_S \leftarrow 0$; //baseline guess
  for $\ell = 1, \ldots, L$ **do**
    select class $S_\ell \lhd S_{\ell-1}$; //class choice, (NLC)

$$S_\ell \sim u_{S_\ell|S_{\ell-1}} = \frac{\exp(\eta_t y_{S_\ell}/\mu_\ell)}{\exp(\eta_t y_{S_{\ell-1}}/\mu_\ell)}$$

    get $r_{S_\ell,t}$; //intra-class cost
    set $\hat{r}_{S_\ell} \leftarrow \hat{r}_{S_\ell} + \dfrac{r_{S_\ell,t}}{u_{S_\ell|S_{\ell-1}} \cdots u_{S_1|S_0}}$; //(NIWE)
  set $\hat{c}_a \leftarrow \sum_{S \ni a} \hat{r}_S$ for all $a \in \mathcal{A}$; //costs
  set $y \leftarrow y - \hat{c}$; //update propensities

Theorem 5.4.1. *Suppose that Algorithm 14 is run with a non-increasing learning rate $\eta_t > 0$ and uncertainty parameters $\mu_1 \geq \cdots \geq \mu_L > 0$ against a sequence of cost vectors $c_t \in [0,1]^{\mathcal{A}}$, $t = 1, 2, \ldots$, as per* (5.4)*. Then, for all $p \in \Delta(\mathcal{A})$, the learner enjoys the regret bound*

$$\mathbb{E}[R_T(p)] \leq \frac{H}{\eta_{T+1}} + \frac{k_{\text{eff}}}{2\mu_L} \sum_{t=1}^{T} \eta_t \tag{5.19}$$

with k_{eff} given by (5.13) *and $H \equiv H(\mu_1, \ldots, \mu_L)$ defined by setting $y = 0$ in* (5.17) *and taking $H = y_{\mathcal{A}}$, i.e.,*

$$H = \log \left[\sum_{S_1 \lhd S_0} \left[\sum_{S_2 \lhd S_1} \cdots \left[\sum_{S_L \lhd S_{L-1}} 1 \right]^{\frac{\mu_L}{\mu_{L-1}}} \cdots \right]^{\frac{\mu_2}{\mu_1}} \right]^{\mu_1} \tag{5.20}$$

In particular, if Algorithm 14 is run with $\mu_1 = \cdots = \mu_L = \sqrt{k_{\text{eff}}/2}$ and $\eta_t = \sqrt{\log k/(2t)}$, we have

$$\mathbb{E}[R_T(p)] \leq 2\sqrt{k_{\text{eff}} \log k \cdot T}. \tag{5.21}$$

Proof outline of theorem 5.4.1. The detailed proof of theorem 5.4.1 is quite lengthy, so we defer it to section 5.9 and only sketch here the main ideas.

The first basic step is to derive a suitable "potential function" that can be used to track the evolution of the NEW policy relative to the benchmark $p \in \Delta(\mathcal{A})$. The main ingredient of

this potential is the "nested" entropy function

$$h(u) = \sum_{k=0}^{L} \delta_k \sum_{S_k \in \mathcal{S}_k} u_{S_k} \log u_{S_k}, \tag{5.22}$$

where $\delta_k = \mu_k - \mu_{k+1}$ for all $k = 1, \dots, L$ (with $\mu_{L+1} = 0$ by convention).[6] As we show in proposition 5.13.1 in section 5.9, the "tiers" of h can be unrolled to give the "non-tiered" recursive representation

$$h(u) = \sum_{S \in \mathcal{S}} h(u|S) \tag{5.23}$$

where $h(u|S) = \mu_{\ell+1} \sum_{S' \lhd S} u_{S'} \log(u_{S'}/u_S)$ denotes the "conditional" entropy of u relative to class $S \in \mathcal{S}_\ell$. Then, by means of this decomposition and a delicate backwards induction argument, we show in proposition 5.13.2 that

1. the recursively defined propensity score $y_{\mathcal{A}}$ of $\mathcal{A}$ can be expressed *non-recursively* as $y_{\mathcal{A}} = \arg\max_{u \in \Delta(\mathcal{A})} \{\langle y, u \rangle - h(u)\}$; and
2. that the choice rule (NLC) can be expressed itself as

$$P_a(y) = \frac{\partial y_{\mathcal{A}}}{\partial y_a} \quad \text{for all } y \in \mathbb{R}^{\mathcal{A}}, a \in \mathcal{A}. \tag{5.24}$$

This representation of (NLC) provides the first building block of our proof because, by Danskin's theorem (Berge, 1997), it allows us to rewrite Algorithm 14 in more concise form as

$$\begin{aligned} y_{t+1} &= y_t - \hat{c}_t \\ u_{t+1} &= \arg\max_{u \in \Delta(\mathcal{A})} \{\langle \eta_{t+1} y_{t+1}, u \rangle - h(u)\} \end{aligned} \tag{NEW}$$

with $\hat{c}_t$ given by (5.8) applied to $u \leftarrow u_t$. Importantly, this shows that the NEW algorithm is an instance of the well-known *"follow the regularized leader"* (FTRL) algorithmic framework (Shalev-Shwartz, 2007, 2012). Albeit interesting, this observation is not particularly helpful in itself because there is no universal, "regularizer-agnostic" analysis giving optimal (or near-optimal) regret rates for FTRL with bandit/partial information.[7] Nonetheless, by adapting a series of techniques that are used in the analysis of FTRL algorithms, we show in section 5.9 that the iterates of (NEW) satisfy the "energy inequality"

$$\begin{aligned} \langle \hat{c}_t, u_t - p \rangle \leq W_t - W_{t+1} + \frac{1}{\eta_t} F(u_t, \eta_t y_{t+1}) \\ + (\eta_{t+1}^{-1} - \eta_t^{-1})[h(p) - \min h] \end{aligned} \tag{5.25}$$

where $\hat{c}_t$ is the nested importance weighted estimator (5.8) for the cost vector encountered c_t, and we have set

$$F(u, y) = h(u) + y_{\mathcal{A}} - \langle y, u \rangle \tag{5.26}$$

and $W_t = \eta_t^{-1} F(p, \eta_t y_t)$.

Then, by proposition 5.3.1, we obtain:

[6] In the non-nested case, (5.22) boils down to the standard (negative) entropy $h(u) = \sum_a u_a \log u_a$. However, the inverse problem of deriving the "correct" form of h in a nested environment involves a technical leap of faith and a fair degree of trial-and-error.

[7] For the analysis of specific versions of FTRL with non-entropic regularizers, cf. (Audibert et al., 2011; Zimmert and Seldin, 2019) and references therein.

Proposition 5.4.1. *The NEW algorithm enjoys the bound*

$$\mathbb{E}[R_T(p)] \leq \frac{H}{\eta_{T+1}} + \sum_{t=1}^{T} \frac{\mathbb{E}[F(u_t, \eta_t y_{t+1})]}{\eta_t}. \tag{5.27}$$

proposition 5.4.1 provides the first half of the bound (5.19), with the precise form of H derived in lemma 5.15.1. The second half of (5.19) revolves around the term $\mathbb{E}[F(u_t, \eta_t y_{t+1})]$ and boils down to estimating how propensity scores are back-propagated along $\mathcal{S}$. In particular, the main difficulty is to bound the difference $y^+_{\mathcal{A}} - y_{\mathcal{A}}$ in the propensity score of the root node $\mathcal{A}$ of $\mathcal{S}$ when the underlying score profile $y \in \mathbb{R}^{\mathcal{A}}$ is incremented to $y^+ = y + w$ for some $w \in \mathbb{R}^{\mathcal{A}}$.

A first bound that can be obtained by convex analysis arguments is $|y^+_{\mathcal{A}} - y_{\mathcal{A}}| \leq \langle y, P(y)\rangle + \|w\|^2_\infty$; however, because the increments of (NEW) are unbounded in norm, this global bound is far too lax for our puposes. A similar issue arises in the analysis of EXP3, and is circumvented by deriving a bound for the log-sum-exp function using the identity $\exp(x) \leq 1 + x + x^2/2$ for $x \leq 0$ and the fact that the estimator (IWE) is non-negative (Lattimore and Szepesvári, 2020; Shalev-Shwartz, 2012; Cesa-Bianchi and Lugosi, 2006). Extending this idea to nested environments is a very delicate affair, because each tier in $\mathcal{S}$ introduces an additional layer of error propagation in the increments $y_{t+1} - y_t$. However, by a series of inductive arguments that traverse $\mathcal{S}$ both forward and backward, we are able to show the bound

$$y^+_{\mathcal{A}} - y_{\mathcal{A}} \leq \langle y, P(y)\rangle + \frac{1}{2\mu_L} \sum_{\ell=1}^{L} \sum_{S_\ell \in \mathcal{S}_\ell} P_{S_\ell}(y) r^2_{S_\ell} \tag{5.28}$$

which, after taking expecations and using the bounds of proposition 5.3.1, finally yields the pseudo-regret bound (5.19).

5.5. Exponential Weights with Experts and Nesting

In this section we introduce the *exponential weights with experts and nesting* (EWEN) algorithm to learn with expert advice.

5.5.1. Expert model

Let $\mathcal{E} = \{e_m : m = 1, \dots, M\}$ be a set of *experts* indexed by $m = 1, \dots, M$. The experts make recommendations to the learner at the beginning of each round t by providing probability recommendations on which alternatives $a \in \mathcal{A}$ that induce less losses. Specifically, the recommendation of the $M = |\mathcal{E}|$ experts are given by a matrix of recommendation $E_t \in [0,1]^{M,k}$ where the m-th row is a mixed strategy $E^m_t \in \Delta(\mathcal{A})$. The probability of sampling the alternative $a \in \mathcal{A}$ is then written $E^m_{t,a}$ so that E^m_t is the vector $E^m_t = \big(E^m_{t,a}\big)_{a\in\mathcal{A}}$.

With all this hand, we consider now the following sequential decision process where the learner selects at each step experts to choose an alternative $a_t \in \mathcal{A}$:

1. At each stage $t = 1, 2, \dots$, the nature sets the recommendations E_t and the learner chooses experts that select an alternative a_t by selecting attributes from $\mathcal{S}$ one-by-one.

2. Concurrently, nature sets the idiosyncratic, intra-class losses $r_{S,t}$ for each similarity class $S \in \mathcal{S}$.

3. The experts incur $r_{S,t}$ for each chosen class $S \ni a_t$ for a total cost of $c_t = \sum_{S \ni a_t} r_{S,t}$, the learner suffers the loss $l_t = E_t c_t$and the process repeats.

Contrary to the previous setting, the learner now employs a strategy to learn a policy on the expert set. Once again, to avoid deterministic strategies that could be exploited by an adversary, we will assume that the learner selects experts at time t based on a mixed strategy $w_t \in \Delta(\mathcal{E})$, i.e., $e_t \sim w_t$. The regret of a policy w_t, $t = 1, 2, \dots$, against a benchmark strategy $q \in \Delta(\mathcal{E})$ is then defined as the cumulative difference between the player's mean cost under q and w_t, that is

$$R_T(q) = \sum_{t=1}^{T} [\mathbb{E}_{w_t}[l_{e_t,t}] - \mathbb{E}_q[l_{e_t,t}]] = \sum_{t=1}^{T} \langle l_t, w_t - q \rangle \tag{5.29}$$

where $l_t = (l_{e,t})_{e \in \mathcal{E}} \in \mathbb{R}^{\mathcal{E}}$ denotes the vector of costs encountered by the learner at time t, i.e., $l_{e,t} = E_t c_t$ where c_t is the cost associated to the alternatives with $c_{a,t} = \sum_{S \ni a} r_{S,t}$ for all $a \in \mathcal{A}$.

In our setting we assume that the similarity structure $\mathcal{S} = \coprod_{\ell=1}^{L} \mathcal{S}_\ell$ is known by the learner and that the experts provide recommendations on alternatives $a \in \mathcal{A}$. In so, it is possible to derive recommendations on similarity classes at levels $\ell = 1 \dots L$. More precisely, given a recommendation matrix E and an expert e_m, the mixed strategy $E^m \in \Delta(\mathcal{A})$ can be used to derive recommendations on any classes $S \in \mathcal{S}$ by considering all of its descendants in $\mathcal{A}$:

$$E^m_S = \sum\nolimits_{a \prec S} E^m_a. \tag{5.30}$$

Then, it is natural to define a conditional probability $E^m_{S'|S}$ on a class S given a parent class S', related to the expert e_m and where $S' \lhd S$, by writing:

$$E^m_{S'|S} = \frac{E^m_{S'}}{E^m_S}. \tag{5.31}$$

This will allow us to define a nested sampling scheme for the EWEN algorithm in the next subsections. Moreover, we note that for a given element $a \in \mathcal{A}$ and its associated lineage $\mathcal{A} \equiv S_0 \rhd S_1 \rhd \cdots \rhd S_L = \{a\}$, the probability E^m_a of an expert e_m to sample the alternative a can be easily recovered with the relation:

$$E^m_a = \prod_{\ell=1}^{L} E^m_{S_\ell|S_{\ell-1}}. \tag{5.32}$$

5.5.2. The exponential weights with experts and nesting algorithm

Before introducing our algorithm, we require a cost estimation rule on the experts. Fortunately, to estimate the costs of experts, it is easy to note that the previous cost estimator $\hat{c}$ on alternatives can be used to estimate losses on experts given a recommendation matrix $E \in [0,1]^{M,k}$:

$$\hat{l} = E\hat{c} \tag{5.33}$$

Then, the properties of the (NIWE) estimator can be extended as follows.

Proposition 5.5.1. *Let $\mathcal{S} = \coprod_{\ell=1}^{L} \mathcal{S}_\ell$ be a similarity structure on $\mathcal{A}$. Then, given a mixed strategy $w \in \Delta(\mathcal{E})$ the expert cost estimator* (5.33) *with the* (NIWE) *estimator in* (5.8) *and a recommendation matrix $E \in [0,1]^{M,k}$ satisfies the following:*

1. *It is unbiased:*
$$\mathbb{E}\left[\hat{l}_e\right] = l_e \quad \textit{for all } e \in \mathcal{E}. \tag{5.34}$$

2. *It enjoys the importance-weighted mean-square bound*
$$\mathbb{E}\left[\sum\nolimits_{e\in\mathcal{E}} w_e \hat{l}_e^2\right] \leq k_{\text{eff}}. \tag{5.35}$$

This Proposition is proven in section 5.9. We are now in position to propose an algorithm that learns to select experts. We call this algorithm the *exponential weights with experts and nesting* (EWEN).

Formally, suppose that the learner associates to each expert $e \in \mathcal{E}$ a *propensity score* $z_e \in \mathbb{R}$ indicating their tendency – or *propensity* – to select it. Given this propensity score profile $z = (z_1, \ldots, z_M) \in \mathbb{R}^{\mathcal{E}}$, the *logit choice* (LC) rule is defined via the selection probabilities:

$$Q_e(z) = \frac{\exp(z_e)}{\sum_{e'\in\mathcal{E}} \exp(z_{e'})} \tag{LC}$$

This (LC) rule thus allows to define a strategy using a propensity score and with the relation $w = Q(z)$.

We are finally in a position to present the *exponential weights with experts and nesting* (EWEN) algorithm in detail. Building on the original exponential weights blueprint (Littlestone and Warmuth, 1994; Auer et al., 2002; Vovk, 1990), the main steps of the EWEN algorithm can be summed up as follows:

1. For each stage $t = 1, 2, \ldots$, the nature sets the recommendation E_t and the learner maintains and updates a propensity score profile $z_t \in \mathbb{R}^{\mathcal{E}}$.
2. At each level $\ell = 1, \ldots, L$, the learner selects an expert based on the logit choice rule $Q(\gamma_t z_t)$ where $\gamma_t \geq 0$ is the method's *learning rate* and Q is given by (LC); the expert then selects a class S_ℓ.

Algorithm 15: Exponential weights with experts and nesting (EWEN)

Require: set of experts $\mathcal{E}$, attribute partitions $\mathcal{S}_1 \succcurlyeq \cdots \succcurlyeq \mathcal{S}_L$, attribute structure $\mathcal{S} = \coprod_{\ell=1}^{L} \mathcal{S}_\ell$
Input: sequence of class costs $r_t \in [0,1]^{\mathcal{S}}$, $t = 1, 2, \ldots$, learning rate $\gamma_t \geq 0$

initialize $z \leftarrow 0 \in \mathbb{R}^{\mathcal{E}}$, $S_0 = \mathcal{A}$;
for $t = 1, 2, \ldots$ **do**
  get experts recommendations E_t ; //expert advices
  set strategy w_t ; //expert strategy, (LC)

$$w_t = \left(\frac{\exp(\gamma_t z_e)}{\sum_{e' \in \mathcal{E}} \exp(\gamma_t z_{e'})} \right)_{e \in \mathcal{E}}$$

  for $S \in \mathcal{S}_\ell$ **do**
    set $\hat{r}_S \leftarrow 0$; //baseline guess
    set $E_{t,S} \leftarrow \sum_{a \prec S} E_{t,a}$ and $E_{t,S'|S}$ for $S' \lhd S$ using (5.31) ;
  for $\ell = 1, \ldots, L$ **do**
    select class $S_\ell \lhd S_{\ell-1}$; //class choice

$$S_\ell \sim u_{S_\ell|S_{\ell-1}} = w_t E_{t,S_\ell|S_{\ell-1}}$$

    get $r_{S_\ell,t}$; //intra-class cost
    set $\hat{r}_{S_\ell} \leftarrow \hat{r}_{S_\ell} + \dfrac{r_{S_\ell,t}}{u_{S_\ell|S_{\ell-1}} \cdots u_{S_1|S_0}}$; //(NIWE)
  set $\hat{c}_a \leftarrow \sum_{S \ni a} \hat{r}_S$ for all $a \in \mathcal{A}$; //costs on alternatives
  set $\hat{l} \leftarrow E_t \hat{c}$; //losses on experts
  set $z \leftarrow z - \hat{l}$; //update propensities

3. The experts incur $r_{S,t}$ for each class $(S_\ell)_{\ell=1,\ldots,L}$; the learner constructs a model $\hat{c}_t$ of the cost vector c_t of stage t via (NIWE) to build $\hat{l}_t = E_t \hat{c}_t$.
4. The learner updates their propensity score profile based on $\hat{l}_t$ and the process repeats.

To view a pseudocode representation of the algorithm, refer to Algorithm 15. The process of adjusting the learning rate γ_t will be discussed in the subsequent section, which is dedicated to analyzing the EWEN algorithm.

5.5.3. Regret guarantees

Having reached this point, we can now articulate and examine the main regret guarantees for the EWEN algorithm. They can be summarized as follows:

Theorem 5.5.1. *Suppose that Algorithm 15 is run with a non-increasing learning rate $\gamma_t > 0$ against a sequence of cost vectors $l_t \in [0,1]^{\mathcal{E}}$, $t = 1, 2, \ldots$, as per* (5.4)*. Then, for all $q \in \Delta(\mathcal{E})$, the learner enjoys the regret bound*

$$\mathbb{E}[R_T(q)] \leq \frac{H_{\mathcal{E}}}{\gamma_{T+1}} + \frac{k_{\text{eff}}}{2} \sum_{t=1}^{T} \gamma_t \tag{5.36}$$

with k_{eff} given by (5.13) *and $H_{\mathcal{E}}$ is defined as the depth over $\Delta(\mathcal{E})$ of the entropic regularizer $h_{\mathcal{E}}$ in*

(5.12.1), *i.e.*,

$$H_{\mathcal{E}} = \max h_{\mathcal{E}} - \min h_{\mathcal{E}} = \log M \tag{5.37}$$

In particular, if Algorithm 15 is run with $\gamma_t = \sqrt{\log M/(2t \cdot k_{\text{eff}})}$*, we have*

$$\mathbb{E}[R_T(q)] \leq 2\sqrt{k_{\text{eff}} \log M \cdot T}. \tag{5.38}$$

We provide all the details of the proof of this Theorem in section 5.9.

5.6. Discussion

In this section we discuss the regret bounds obtained for the NEW and EWEN algorithms.

The first thing of note is the comparison to the corresponding bound for EXP3 and EXP4, respectively of upper bounded by $2\sqrt{k \log k \cdot T}$ and $2\sqrt{k \log M \cdot T}$. This shows our guarantees differ by a factor of[8]

$$\alpha = \sqrt{k/k_{\text{eff}}}, \tag{5.39}$$

which, for reasons that become clear below, we call the *price of affinity* (PoAf).

Since the variabilities of the idiosyncratic losses within each attribute have been normalized to 1 (recall the relevant discussion in section 5.3), Hölder's inequality trivially gives $k_{\text{eff}} \leq k$, no matter the underlying similarity structure. Of course, if there are no similarities to exploit ($L = 1$), we get $k_{\text{eff}} = k$, in which case the two bounds coincide ($\alpha = 1$).

At the other extreme, suppose again we have a red bus / blue bus type of problem with, say, $k_1 = 2$ similarity classes, $k_2 = 100$ alternatives per class, and a negligible intra-class loss differential ($\Gamma_2 \approx 0$). In this case, EXP3 and EXP4 would have to wrestle with $k = k_1 k_2 = 200$ alternatives, while NEW and EWEN would only need to discriminate between $k_{\text{eff}} \approx k_1 = 2$ alternatives, leading to an improvement by a factor of $\alpha \approx 10$ in terms of regret guarantee. Thus, even though the red bus / blue bus paradox could entangle EXP3 or EXP4 and cause the algorithm to accrue significant regret over time, this is no longer the case under the NEW and EWEN methods; we also explore this issue numerically in section 5.7.

As another example, suppose that each non-terminal class in $\mathcal{S}$ has s children and the variability of the idiosyncratic losses likewise scales down by a factor of s per attribute. In this case, a straightforward calculation shows that k_{eff} scales as $\Theta(s)$, so the gain in efficiency would be of the order of $\alpha = \sqrt{k/k_{\text{eff}}} = \Theta(s^{(L-1)/2})$, i.e., polynomial in s and exponential in L. This gain in performance can become especially pronounced when there is a very large number of alternatives organized in categories and subcategories of geometrically decreasing impact on the end cost of each alternative. We explore this issue in practical scenarios in sections 5.7 and 5.9.

[8]Depending on the source, those bounds may differ up to a factor of $\sqrt{2}$, compare for example (Shalev-Shwartz, 2012, Corollary 4.2) and (Lattimore and Szepesvári, 2020, Theorem 11.2). This factor is due to the fact that regret of EXP3 is usually stated for a known horizon T (which saves a factor of $\sqrt{2}$ relative to anytime algorithms). Ceteris paribus, the bound (5.21) can be sharpened by the same factor, but we omit the details.

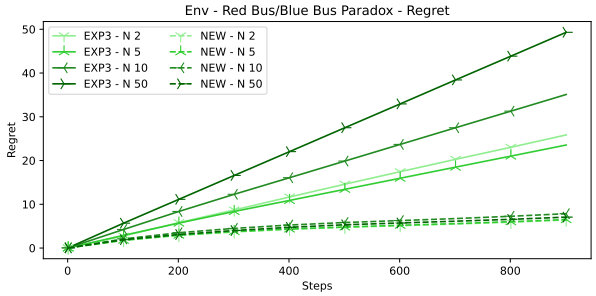

Figure 5.2: Regret of EXP3 and NEW in the red bus / blue bus problem with different numbers of buses.

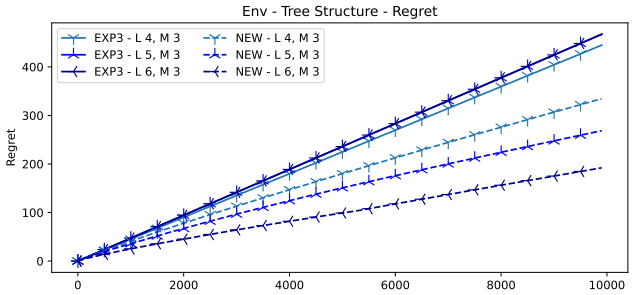

Figure 5.3: Regret of EXP3 and NEW in a tree environment with different values of levels L and classes per level M

Finally, we should also note that the parameters of NEW have been tuned so as to facilitate the comparison with EXP3. This tuning is calibrated for the case where $\mathcal{S}$ is fully symmetric, i.e., all subcategories of a given attribute have the same number of children. Otherwise, in full generality, the tuning of the algorithm's uncertainty levels would boil down to a transcendental equation involving the nested term $H(\mu_1, \dots, \mu_L)$ of (5.19). This can be done efficiently offline via a line search, but since the result would be structure-dependent, we do not undertake this analysis here.

5.7. Numerical experiments

In this section we present a series of numerical experiments designed to test the efficiency of the NEW algorithm compared to EXP3. We use a synthetic environment where we simulate nested similarity partitions with trees. While NEW exploits the similarity structure by making forward/backward passes through the associated tree with its logit choice rule (NLC), EXP3 is simply run over the leaves of the tree, i.e., $\mathcal{A}$. All experiment details (as well as additional results) are presented in section 5.9. For every setting, we report the results of our experiments by plotting the average regret of each algorithm for 20 seeds of randomly drawn losses. The code to reproduce the experiments can be found at `https://github.com/criteo-research/Nested-Exponential-Weights`.

Benefits in the red bus/blue bus problem. We consider here a variant of the red bus/blue bus problem with N different buses (the original paradox has $N = 2$). In this experiment

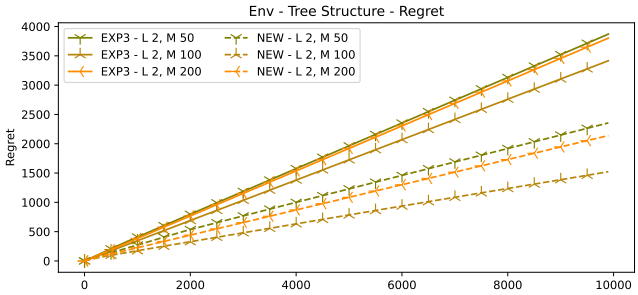

Figure 5.4: Regret of EXP3 and NEW in a tree environment with different values of levels L and classes per level M

(see illustration in figure 5.5, Appendix 5.9) we allow each bus to have non-zero intrinsic losses and illustrate in figure 5.2 how both algorithms perform when N grows. We observe there that for all configurations NEW achieves better regret than EXP3. While both methods achieve sublinear regret, EXP3 requires far more steps to identify the best alternative as N grows and suffers overall from worse regret while NEW achieves similar regret and does not suffer as much from the number of irrelevant alternatives. We provide additional plots in section 5.9 which show that NEW performs consistently better than EXP3 when there exists a similarity structure allowing to efficiently update scores of classes that have very similar losses.

Performance in general nested structures. In this setting we generate symmetric trees and experiment with different values of number of levels L and number of child per nodes $M = |S_\ell|$ for $\ell = 1, \dots, L$. Specifically, in figure 5.3 with a fixed M, we see that NEW obtains better regret than EXP3 even when L increases. We provide variance plots for the experiments that generated the same performance on the plots in 5.9 as well as additional visualisations. Finally, in figure 5.4, we can see that for a shallow tree ($L = 2$) NEW performs always better than EXP3, even for high values of M. Indeed, when the number of children per nodes M increases, the tree loses its "factorized" structure which also affects NEW due to the less "structured" tree. Thus, again, NEW performs consistently better than EXP3 when it is possible to efficiently handle classes with similar losses.

Overall, our experiments confirm that a learning algorithm based on nested logit choice can lead to significant benefits in problems with a high degree of similarity between alternatives. This leaves open the question of whether a similar approach can be applied to structures with *non-nested* attributes; we defer this question to future work.

5.8. Discussions

One limitation of the current framework is that the nested estimator (5.8) requires knowledge of the intra-class cost increments r_S for every chosen similarity class $S \ni a_t$. This is akin to the difference between the "full bandit" and "semi-bandit" setting that arises in combinatorial bandits (Cesa-Bianchi and Lugosi, 2012). While relevant in a number of application domains (e.g., in path-planning or when layering a structured security, such as the tranches of a CDO), treating the fully unobservable case – possibly using an approach in

the spirit of the hierarchical contextual analysis of Sen et al. (2021) – is an important open question for future research.

Finally, it is also interesting to note that our analysis has been carried out in an arbitrarily changing "adversarial" environment. In a stochastic environment, it would be fruitful to consider other, contextual-based approaches such as LinUCB, KernelUCB and their variants Lattimore and Szepesvári (2020). Ideally, one would like to employ a nested variant of the "universal" algorithm of Zimmert and Seldin (2019) that attains optimal regret guarantees in both stochastic and adversarial environments, but this question lies beyond the scope of our work.

5.9. Appendices

This appendix is organized as follows:

- Appendix 5.9: recapitulation of the notations
- Appendix 5.9: auxiliary bounds for the EWEN algorithm
- Appendix 5.9: regret analysis of the EWEN algorithm
- Appendix 5.9: analysis of the nested entropy and related bounds
- Appendix 5.9: auxiliary bounds for the NEW algorithm
- Appendix 5.9: regret analysis of the NEW algorithm
- Appendix 5.9: additional experiment details, discussions and results

5.10. Notations

In this appendix, we recall useful notations that are used throughout the paper.

- T is the horizon or number of rounds

Alternatives, experts, ground sets:

- a denotes a generic alternartive element (a.k.a. arm)
- $\mathcal{A} := \{a_i : i = 1, \dots, k\}$ is the ground set of alternatives ($a \in \mathcal{A}$)
- k is the number of elements in the set of alternatives ($|\mathcal{A}| = k$)
- A is a generic subset of alternatives ($A \subset \mathcal{A}$)
- $\mathcal{E} := \{e_m : m = 1, \dots, M\}$ is the set of experts
- M is the number of experts in the set of experts ($|\mathcal{E}| = M$)

Partitions, nested structure:

- $\mathcal{S}$ denotes the generic partition of the alternative set
- S is a class of the partition set
- $\preccurlyeq, \succcurlyeq$ respectively defines relation for finer and coarser classes in the partition set: $\{\mathcal{A}\} =: \mathcal{S}_0 \succcurlyeq \mathcal{S}_1 \succcurlyeq \cdots \succcurlyeq \mathcal{S}_L := \{\{a\} : a \in \mathcal{A}\}$ tower of nested *similarity partitions* (or *attributes*)
- $\mathcal{S}_\ell$ partition that refer to as a *similarity class*
- $S' \prec S$ means $S' \prec S$ i.e. a class $S \in \mathcal{S}_\ell$ contains the class $S' \in \mathcal{S}_k$ for some $k > \ell$
- $S' \prec S$ and $k = \ell + 1$,= means that S' is a *child* of S and we will write "$S' \lhd S$"
- $S' \sim S''$ means S' and S'' are *siblings* if they are children of the same parent
- *lineage* of S is the unique directed path $\mathcal{A} \equiv S_0 \rhd S_1 \rhd \cdots \rhd S_\ell \equiv S$ from $\mathcal{A}$ to any class $S \in \mathcal{S}$

Payoff structure:

- Γ_S "intra-class" variability of costs $r_{S'} \in [0, \Gamma_S]$ for all $S' \lhd S$
- $r_{S'}$ idiosyncratic cost increment, i.e. $c_{S'} = c_S + r_{S'}$ for all $S' \lhd S$
- c_S cost of the similarity class $S \in \mathcal{S}$
- k_{eff} effective number of alternatives (arms). Could be strictly lower than k under some assumption

– $\alpha = \sqrt{k/k_{\text{eff}}}$ *price of affinity* (PoAf)

Learning with expert advice

– $E \in [0,1]^{M,k}$ are the experts recommendations
– $w \in \Delta(\mathcal{E})$ mixed strategy to select experts by the learner
– $q \in \Delta(\mathcal{E})$ benchmark generic strategy
– $\hat{l}_e$ estimator of l_e
– Q is a choice function that maps score vectors $z \in \mathbb{R}^{\mathcal{E}}$ to mixed strategies via the relation $w = Q(z)$
– γ is the learning rate for the expert choice map Q
– $h_{\mathcal{E}}$ is the total entropy on the expert set $\mathcal{E}$
– $H_{\mathcal{E}}$ is the depth of the previous entropy over the space $\Delta(\mathcal{E})$

Designing an expert

– $u \in \Delta(\mathcal{A})$ mixed strategy to select arms by the learner
– $p \in \Delta(\mathcal{A})$ benchmark generic strategy
– $\hat{c}_a$ estimator of c_a
– P is a choice function that maps score vectors $y \in \mathbb{R}^{\mathcal{A}}$ to mixed strategies via the relation $u = P(y)$
– η is the learning rate for the choice map P
– μ_ℓ is the temperature parameter associated to each level $\ell \in \{1, \dots, L\}$
– δ is the temperature difference
– h is the nested entropy
– H is the depth of the previous entropy over the space $\Delta(\mathcal{A})$

5.11. Auxiliary bounds and results - EWEN algorithm

This part of the Appendix will serve to provide auxiliary bounds and results that will enable the analysis of the exponential weights with experts and nesting algorithm later.

We will now prove the basic property of the NIWE estimator for experts, which we restate below:

Proposition 5.5.1. *Let $\mathcal{S} = \coprod_{\ell=1}^{L} \mathcal{S}_\ell$ be a similarity structure on $\mathcal{A}$. Then, given a mixed strategy $w \in \Delta(\mathcal{E})$ the expert cost estimator* (5.33) *with the* (NIWE) *estimator in* (5.8) *and a recommendation matrix $E \in [0,1]^{M,k}$ satisfies the following:*

1. *It is unbiased:*
$$\mathbb{E}\left[\hat{l}_e\right] = l_e \quad \textit{for all } e \in \mathcal{E}. \tag{5.34}$$

2. *It enjoys the importance-weighted mean-square bound*
$$\mathbb{E}\left[\sum\nolimits_{e\in\mathcal{E}} w_e \hat{l}_e^2\right] \leq k_{\text{eff}}. \tag{5.35}$$

Before proving Proposition 5.5.1, we state and prove the following lemma that we will help in this proof.

Lemma 5.11.1. *Let $\mathcal{S} = \coprod_{\ell=1}^{L} \mathcal{S}_\ell$ be a similarity structure on $\mathcal{A}$. Let $w \in \Delta(\mathcal{E})$, $E \in [0,1]^{M,k}$. Let also $\Gamma_S \in [0,1]$ for all $\ell \in \{1,\dots,L\}$, it holds that:*

$$\sum_{m=1}^{M} w_{e_m} \sum_{S_\ell \in \mathcal{S}_\ell} \mathbb{E}\left[\left(\hat{l}_{e_m}^{S_\ell}\right)^2\right] \leq k_\ell \bar{\Gamma}_\ell^2, \tag{5.11.1}$$

where for all $\ell \in \{1,\dots,L\}$,

$$\bar{\Gamma}_\ell = \sqrt{\frac{1}{k_\ell} \sum_{S_\ell \in \mathcal{S}_\ell} \Gamma_{S_\ell}^2}.$$

In particular, for all $S \in \mathcal{S}$ we have:

$$\sum_{m=1}^{M} w_{e_m} \sum_{S \in \mathcal{S}} \mathbb{E}\left[\left(\hat{l}_{e_m}^{S}\right)^2\right] \leq \sum_{\ell=1}^{L} k_\ell \bar{\Gamma}_\ell^2. \tag{5.11.2}$$

Proof. Proof of lemma 5.11.1

Let $\ell \in \{1,\dots,L\}$. First, we will note that in the similarity structure $\mathcal{S}$, a node S_ℓ is chosen with probability:

$$u_{S_\ell} = w E_{S_\ell} \tag{5.11.3}$$

where E_{S_ℓ} is obtained from (5.30) on E. We then write:

$$\begin{aligned}
\sum_{m=1}^{M} w_{e_m} \sum_{S_\ell \in \mathcal{S}_\ell} \mathbb{E}\left[\left(\hat{l}_{e_m}^{S_\ell}\right)^2\right] &= \sum_{m=1}^{M} w_{e_m} \sum_{S_\ell \in \mathcal{S}_\ell} \mathbb{E}\left[\left(\frac{\mathbb{1}\{S_\ell = \hat{S}_\ell\} r_{S_\ell}}{u_{S_\ell}} E_{S_\ell}^m\right)^2\right] \\
&= \sum_{m=1}^{M} w_{e_m} \sum_{S_\ell \in \mathcal{S}_\ell} \mathbb{E}\left[\mathbb{1}\{S_\ell = \hat{S}_\ell\}\right] \frac{r_{S_\ell}^2}{(u_{S_\ell})^2} \left(E_{S_\ell}^m\right)^2 \\
&= \sum_{m=1}^{M} w_{e_m} \sum_{S_\ell \in \mathcal{S}_\ell} u_{S_\ell} \frac{r_{S_\ell}^2}{(u_{S_\ell})^2} \underbrace{\left(E_{S_\ell}^m\right)^2}_{\leq E_{S_\ell}^m} \\
&\leq \sum_{m=1}^{M} w_{e_m} \sum_{S_\ell \in \mathcal{S}_\ell} \frac{r_{S_\ell}^2}{u_{S_\ell}} E_{S_\ell}^m \\
&= \sum_{S_\ell \in \mathcal{S}_\ell} \underbrace{r_{S_\ell}^2}_{\leq \Gamma_{S_\ell}^2} \underbrace{\frac{\sum_{m=1}^{M} w_{e_m} E_{S_\ell}^m}{u_{S_\ell}}}_{=1} \\
&\leq \underbrace{\sum_{S_\ell \in \mathcal{S}_\ell} \Gamma_{S_\ell}^2}_{k_\ell \bar{\Gamma}_\ell^2} \\
&= k_\ell \bar{\Gamma}_\ell^2.
\end{aligned}$$

This proves both (5.11.1) (directly) and (5.11.2) (summing on all levels $\ell \in \{1,\dots,L\}$ to browse the whole similarity set $\mathcal{S}$). □

We can now prove the proposition on the (NIWE) estimator for the expert costs.

Proof. Proof of proposition 5.5.1

Fix an expert $e \in \mathcal{E}$. We will now prove both properties of the (NIWE) estimator.

Part 1. Fix some $S \in \mathcal{S}$ with $\mathrm{attr}S = \ell \in \{1, \dots, L\}$ and lineage $\mathcal{A} \equiv S_0 \rhd S_1 \rhd \cdots \rhd S_\ell \equiv S$.

We begin by showing that the estimator (NIWE) is unbiased for the idiosyncratic loss term $\hat{r}_S$. Indeed, we have:

$$\begin{aligned}\mathbb{E}[\hat{r}_S] &= \mathbb{E}\left[\frac{\mathbb{1}\{S_\ell = \hat{S}_\ell, \dots, S_1 = \hat{S}_1\}}{u_{S_\ell|S_{\ell-1}} \cdots u_{S_2|S_1} u_{S_1}} r_{S_\ell}\right] = \mathbb{E}\left[\frac{\mathbb{1}\{S = \hat{S}\}}{u_S} r_S\right] \quad \text{\# \textit{Rewriting} (NIWE)} \\ &= \frac{r_S}{u_S} \underbrace{\mathbb{E}\Big[\mathbb{1}\{S = \hat{S}\}\Big]}_{u_S} = r_S. \end{aligned} \tag{5.11.4}$$

This readily shows that the cost estimator in (5.8) then verifies $\mathbb{E}[\hat{c}_a] = c_a$ by summation and then $\mathbb{E}\Big[\hat{l}_e\Big] = l_e$ by linearity of the expectation.

Part 2. We will now proceed to upper bound the weighted sum on the squarred estimator.

Recalling the increment loss decomposition in (5.8), the cost for an expert $e_m \in \mathcal{E}$ with index $m \in \{1, \dots M\}$ then writes:

$$\begin{aligned}\hat{l}_{e_m} &= \hat{c} E^m \\ &= \sum_{a \in \mathcal{A}} \hat{c}_a E_a^m \qquad (5.11.5) \\ &= \sum_{a \in \mathcal{A}} \sum_{S \ni a} \hat{r}_S E_a^m \qquad (5.11.6) \\ &= \sum_{S \in \mathcal{S}} \hat{r}_S E_S^m \qquad (5.11.7)\end{aligned}$$

where the last expression is given using the definition of E_S in (5.30). Thus, we can reorganize the sum as:

$$\hat{l}_{e_m} = \sum_{S \in \mathcal{S}} \hat{l}_{e_m}^S.$$

where for $m \in \{1, \dots, M\}$ and $S, S' \in \mathcal{S}$ such that $S' \rhd S$ we define $\hat{l}_{e_m}^S$, the expert increment cost related to S, as:

$$\hat{l}_{e_m}^S = \hat{r}_S E_S^m.$$

Now, we aim at upper bounding $\mathbb{E}\Big[\sum_{m=1}^M w_{e_m} (\hat{l}_{e_m})^2\Big]$. Using (5.11.7) we decompose this quantity as follows:

$$\mathbb{E}\left[\sum_{m=1}^{M} w_{e_m}(\hat{l}_{e_m})^2\right] = \sum_{m=1}^{M} w_{e_m}\,\mathbb{E}\Big[(\hat{l}_{e_m})^2\Big] = \sum_{m=1}^{M} w_{e_m}\,\mathbb{E}\left[\left(\sum_{S\in\mathcal{S}} \hat{l}_{e_m}^{S}\right)^2\right] \tag{5.11.8}$$

For a given value of $m \in \{1,\dots,M\}$, we can decompose $(\sum_{S\in\mathcal{S}} \hat{l}_{e_m}^{S})^2$ as follows:

$$\left(\sum_{S\in\mathcal{S}} \hat{l}_{e_m}^{S}\right)^2 = \sum_{S\in\mathcal{S}}\Big(\hat{l}_{e_m}^{S}\Big)^2 + 2\sum_{S'\in\mathcal{S}}\sum_{S\succ S'} \hat{l}_{e_m}^{S}\hat{l}_{e_m}^{S'}. \tag{5.11.9}$$

In order to tightly bound the right-hand side of (5.11.9), we use the lemma 5.11.1. Indeed, combining (5.11.8) and (5.11.9) yields:

$$\mathbb{E}\left[\sum_{m=1}^{M} w_{e_m}(\hat{l}_{e_m})^2\right] = \underbrace{\sum_{m=1}^{M} w_{e_m}\sum_{S\in\mathcal{S}}\mathbb{E}\left[\Big(\hat{l}_{e_m}^{S}\Big)^2\right]}_{(1)} + \underbrace{2\sum_{m=1}^{M} w_{e_m}\sum_{S'\in\mathcal{S}}\sum_{S\succ S'}\mathbb{E}\Big[\hat{l}_{e_m}^{S}\hat{l}_{e_m}^{S'}\Big]}_{(2)}. \tag{5.11.10}$$

lemma 5.11.1 directly enables to bound term (1) as:

$$\sum_{m=1}^{M} w_{e_m}\sum_{S\in\mathcal{S}}\mathbb{E}\left[\Big(\hat{l}_{e_m}^{S}\Big)^2\right] \le \sum_{\ell=1}^{L} k_\ell\big(\bar{\Gamma}_\ell\big)^2. \tag{5.11.11}$$

Now we rewrite term (2) by making an explicit sum on the levels:

$$(2) = 2\sum_{m=1}^{M} w_{e_m}\sum_{S'\in\mathcal{S}}\sum_{S\succ S'} u_{S'}\,\mathbb{E}\Big[\hat{l}_{e_m}^{S}\hat{l}_{e_m}^{S'}\Big] = \sum_{m=1}^{M} w_{e_m}\sum_{1\le\ell<\ell'\le L}\sum_{\substack{S_\ell\in\mathcal{S}_\ell\\ S_{\ell'}\prec_{\ell'}\mathcal{S}_\ell}}\mathbb{E}\Big[\Big(2\hat{l}_{e_m}^{S_\ell}\hat{l}_{e_m}^{S_{\ell'}}\Big)\Big]. \tag{5.11.12}$$

Let $\{\varepsilon_{\ell,\ell'}\}_{1\le\ell'<\ell\le L}$ be any fixed sequence of positive numbers. For any fixed expert e_m, any $\ell,\ell' \in \{1,\dots,L\}$ and any $S_\ell \in \mathcal{S}_\ell$ and $S_{\ell'} \in \mathcal{S}_{\ell'}$, the Peter-Paul inequality yields:

$$2\hat{l}_{e_m}^{S_\ell}\hat{l}_{e_m}^{S_{\ell'}} \le \frac{1}{\varepsilon_{\ell,\ell'}}(\hat{l}_{e_m}^{S_\ell})^2 + \varepsilon_{\ell,\ell'}(\hat{l}_{e_m}^{S_{\ell'}})^2. \tag{5.11.13}$$

Injecting (5.11.13) in (5.11.12) enables to proceed with the following series of derivations:

$$
\begin{aligned}
(2) &\leq \sum_{m=1}^{M} w_{e_m} \sum_{1\leq \ell<\ell'\leq L} \sum_{\substack{S_\ell\in\mathcal{S}_\ell \\ S_{\ell'}\prec_{\ell'}\mathcal{S}_\ell}} \left(\frac{1}{\varepsilon_{\ell,\ell'}} \mathbb{E}\Big[(\hat{l}^{S_\ell}_{e_m})^2\Big] + \varepsilon_{\ell,\ell'}\, \mathbb{E}\Big[(\hat{l}^{S_{\ell'}}_{e_m})^2\Big] \right) \\
&\leq \sum_{m=1}^{M} w_{e_m} \sum_{1\leq \ell<\ell'\leq L} \sum_{\substack{S_\ell\in\mathcal{S}_\ell \\ S_{\ell'}\prec_{\ell'}\mathcal{S}_\ell}} \frac{1}{\varepsilon_{\ell,\ell'}} \mathbb{E}\Big[(\hat{l}^{S_\ell}_{e_m})^2\Big] + \sum_{m=1}^{M} w_{e_m} \sum_{1\leq \ell<\ell'\leq L} \sum_{\substack{S_\ell\in\mathcal{S}_\ell \\ S_{\ell'}\prec_{\ell'}\mathcal{S}_\ell}} \varepsilon_{\ell,\ell'}\, \mathbb{E}\Big[(\hat{l}^{S_{\ell'}}_{e_m})^2\Big] \\
&= \sum_{1\leq \ell<\ell'\leq L} \frac{1}{\varepsilon_{\ell,\ell'}} \underbrace{\sum_{m=1}^{M} w_{e_m} \sum_{S_\ell\in\mathcal{S}_\ell} \mathbb{E}\Big[(\hat{l}^{S_\ell}_{e_m})^2\Big]}_{\leq k_\ell(\bar{\Gamma}_\ell)^2 \text{ (by lemma 5.11.1)}} + \sum_{1\leq \ell<\ell'\leq L} \varepsilon_{\ell,\ell'} \underbrace{\sum_{m=1}^{M} w_{e_m} \sum_{S_{\ell'}\in\mathcal{S}_{\ell'}} \mathbb{E}\Big[(\hat{l}^{S_{\ell'}}_{e_m})^2\Big]}_{\leq k_{\ell'}(\bar{\Gamma}_{\ell'})^2 \text{ (by lemma 5.11.1)}} \\
&\leq \sum_{1\leq \ell<\ell'\leq L} \left(\frac{1}{\varepsilon_{\ell,\ell'}} k_\ell(\bar{\Gamma}_\ell)^2 + \varepsilon_{\ell,\ell'} k_{\ell'}(\bar{\Gamma}_{\ell'})^2 \right). \qquad (5.11.14)
\end{aligned}
$$

Now, injecting (5.11.11) and (5.11.14) in (5.11.10) gives:

$$
\mathbb{E}\left[\sum_{m=1}^{M} w_{e_m}(\hat{l}^m)^2\right] \leq \sum_{\ell=1}^{L} k_\ell\big(\bar{\Gamma}_\ell\big)^2 + \sum_{1\leq \ell<\ell'\leq L} \left(\frac{1}{\varepsilon_{\ell,\ell'}} k_\ell(\bar{\Gamma}_\ell)^2 + \varepsilon_{\ell,\ell'} k_{\ell'}(\bar{\Gamma}_{\ell'})^2 \right). \qquad (5.11.15)
$$

For all ℓ, ℓ', choosing $\varepsilon_{\ell,\ell'} = \sqrt{\frac{k_\ell(\bar{\Gamma}_\ell)^2}{k_{\ell'}(\bar{\Gamma}_{\ell'})^2}}$ in (5.11.15) yields:

$$
\begin{aligned}
\mathbb{E}\left[\sum_{m=1}^{M} w_{e_m}(\hat{l}^m)^2\right] &\leq \sum_{\ell=1}^{L} k_\ell(\bar{\Gamma}_\ell)^2 + 2 \sum_{1\leq \ell<\ell'\leq L} \sqrt{k_\ell k_{\ell'}}\bar{\Gamma}_\ell\bar{\Gamma}_{\ell'} \\
&= \sum_{\ell=1}^{L} \Big(\sqrt{k_\ell}\bar{\Gamma}_\ell\Big)^2 + 2 \sum_{1\leq \ell<\ell'\leq L} \Big(\sqrt{k_\ell}\bar{\Gamma}_\ell\Big)\Big(\sqrt{k_{\ell'}}\bar{\Gamma}_{\ell'}\Big) \\
&= \left(\sum_{\ell=1}^{L} \sqrt{k_\ell}\bar{\Gamma}_\ell\right)^2. \qquad (5.11.16)
\end{aligned}
$$

Which concludes the proof.

□

5.12. Regret analysis of the EWEN algorithm

At the core of our analysis lies a "template inequality", that will first require an energy function measuring the disparity between a benchmark strategy $w \in \Delta(\mathcal{E})$ and a propensity score profile $z \in \mathbb{R}^{\mathcal{E}}$. We therefore introduce $h_{\mathcal{E}} : \Delta(\mathcal{E}) \to \mathbb{R}$ as the total entropy function

$$
h_{\mathcal{E}}(w) = \sum_{e\in\mathcal{E}} w_e \log w_e, \qquad \text{for } w \in \Delta(\mathcal{E}), \qquad (5.12.1)
$$

and let

$$h^*_{\mathcal{E}}(z) = \max_{w \in \Delta(\mathcal{E})} \{\langle z, w\rangle - h_{\mathcal{E}}(w)\}, \quad \text{for } z \in \mathbb{R}^{\mathcal{E}}, \tag{5.12.2}$$

denote the convex conjugate of $h_{\mathcal{E}}$.

The *Fenchel coupling* between $w \in \Delta(\mathcal{E})$ and $z \in \mathbb{R}^{\mathcal{E}}$ is then defined as

$$F_{\mathcal{E}}(w, z) = h_{\mathcal{E}}(w) + h^*_{\mathcal{E}}(z) - \langle z, w\rangle \quad \text{for all } w \in \Delta(\mathcal{E}),\, z \in \mathbb{R}^{\mathcal{E}}, \tag{5.12.3}$$

and we have the following first result:

Proposition 5.12.1. *Let $\mathcal{E}$ be an expert set. Then:*

1. *The Fenchel coupling* (5.12.3) *is positive-definite, i.e.,*

$$F_{\mathcal{E}}(w, z) \geq 0 \qquad \textit{for all } w \in \Delta(\mathcal{E}) \textit{ and all } z \in \mathbb{R}^{\mathcal{E}}, \tag{5.12.4}$$

   with equality if and only if w is given by (LC)*, i.e., if and only if $w = Q(z)$.*

2. *For all $w \in \mathcal{E}$, we have*

$$F_{\mathcal{E}}(w, 0) = h_{\mathcal{E}}(w) + h^*_{\mathcal{E}}(0) = h_{\mathcal{E}}(w) - \min h_{\mathcal{E}} \tag{5.12.5}$$

   where $\min h_{\mathcal{E}} \equiv \min_{w' \in \Delta(\mathcal{E})} h_{\mathcal{E}}(w')$ denotes the minimum of $h_{\mathcal{E}}$ over $\Delta(\mathcal{E})$.

Proof. To show our first claim we rewrite the definition of convex conjugate $h^*_{\mathcal{E}}$: for any $z \in \mathbb{R}^{\mathcal{E}}$ we have

$$h^*_{\mathcal{E}}(z) = \max_{w \in \Delta(\mathcal{E})} \{\langle z, w\rangle - h_{\mathcal{E}}(w)\}.$$

This straightforwardly implies that for any $w \in \Delta(\mathcal{E}), z \in \mathbb{R}^{\mathcal{E}}$,

$$h^*_{\mathcal{E}}(z) \geq \langle z, w\rangle - h_{\mathcal{E}}(w),$$

and therefore that

$$F_{\mathcal{E}}(w, z) = h_{\mathcal{E}}(w) + h^*_{\mathcal{E}}(z) - \langle z, w\rangle \geq 0.$$

Moreover, we classically show using Jensen's inequality that for a fixed $z \in \mathbb{R}^{\mathcal{E}}$ and any $w \in \Delta(\mathcal{E})$,

$$\langle z, w\rangle - h_{\mathcal{E}}(w) \leq \langle z, Q(z)\rangle - h_{\mathcal{E}}(Q(z)),$$

where $Q(z) = \exp(z_e)/\big(\sum_{e' \in \mathcal{E}} \exp(z_{e'})\big)$ is the logit choice presented in (LC). The fact that the equality happens only when $w = Q(z)$ comes from the strict concavity of the logarithm function.

As for our second claim, simply note that

$$h^*_{\mathcal{E}}(0) = \max_{w \in \Delta(\mathcal{E})} \{\langle 0, w\rangle - h_{\mathcal{E}}(w)\} = -\min_{w \in \Delta(\mathcal{E})} h_{\mathcal{E}}(w)$$

and set $z \leftarrow 0$ in the definition (5.12.3) of the Fenchel coupling.

□

We next state a property that will help us for our regret analysis by exhibiting the dependency of the convex conjugate $h^*_{\mathcal{E}}$ on the variance of the importance weighted estimators.

Proposition 5.12.2. *For $z \in \mathbb{R}^{\mathcal{E}}$ and $l \in [0, +\infty)^{\mathcal{E}}$, we have:*

$$h^*_{\mathcal{E}}(z - l) - h^*(z) \leq -\langle Q(z), l\rangle + \frac{1}{2}\sum_{e\in\mathcal{E}} Q_e(z) l_e^2. \tag{5.12.6}$$

Proof. As shown in the proof of proposition 5.12.1, for any $z \in \mathbb{R}^{\mathcal{E}}$ we have that

$$F_{\mathcal{E}}(Q(z), z) = 0,$$

which directly implies that

$$h^*_{\mathcal{E}}(z) = \langle z, Q(z)\rangle - h_{\mathcal{E}}(Q(z)).$$

Using the fact that $Q(z) = \exp(z_e)/\big(\sum_{e'\in\mathcal{E}} \exp(z_{e'})\big)$, a series of straightforward derivations yields

$$h^*_{\mathcal{E}}(z) = \log\left(\sum_{e\in\mathcal{E}} \exp(z_e)\right).$$

Now this enables to write, for $z \in \mathbb{R}^{\mathcal{E}}$ and $l \in [0, +\infty)^{\mathcal{E}}$:

$$\begin{aligned}
h^*_{\mathcal{E}}(z - l) &= \log\left(\sum_{e\in\mathcal{E}} \exp(z_e - l_e)\right) \\
&= \log\left(\sum_{e\in\mathcal{E}} \exp(z_e)\exp(-l_e)\right) \\
&= \log\left(\sum_{e\in\mathcal{E}} \underbrace{\frac{\exp(z_e)}{\sum_{e'\in\mathcal{E}}\exp(z_{e'})}}_{=Q_e(z)} \exp(-l_e)\sum\nolimits_{e'\in\mathcal{E}}\exp(z_{e'})\right) \\
&= \log\left(\sum\nolimits_{e'\in\mathcal{E}}\exp(z_{e'})\sum_{e\in\mathcal{E}} Q_e(z)\exp(-l_e)\right) \\
&= \underbrace{\log\Big(\sum\nolimits_{e'\in\mathcal{E}}\exp(z_{e'})\Big)}_{=h^*_{\mathcal{E}}(z)} + \log\left(\sum_{e\in\mathcal{E}} Q_e(z)\exp(-l_e)\right),
\end{aligned}$$

which delivers the following equality

$$h^*_{\mathcal{E}}(z - l) - h^*_{\mathcal{E}}(z) = \log\left(\sum_{e\in\mathcal{E}} Q_e(z)\exp(-l_e)\right). \tag{5.12.7}$$

Finally we write

$$\begin{aligned} h^*_{\mathcal{E}}(z-l) - h^*_{\mathcal{E}}(z) &= \log\left(\sum_{e\in\mathcal{E}} Q_e(z)\exp(-l_e)\right) \\ &\leq \log\left(\sum_{e\in\mathcal{E}} Q_e(z)\left(1 - l_e + \frac{1}{2}(l_e)^2\right)\right) \\ &\leq \sum_{e\in\mathcal{E}} Q_e(z)\left(-l_e + \frac{1}{2}(l_e)^2\right) \\ &= -\sum_{e\in\mathcal{E}} Q_e(z) l_e + \frac{1}{2}\sum_{e\in\mathcal{E}} Q_e(z)(l_e)^2 \\ &= -\langle Q(z), l\rangle + \frac{1}{2}\sum_{e\in\mathcal{E}} Q_e(z)(l_e)^2 \end{aligned}$$

where we used that for any $x \in [0, +\infty)$ we have $\exp(x) \leq 1 - x + x^2/2$ and $log(1-x) \leq x$. This is the wanted result. □

With all this in hand, the specific energy function that we will use for our regret analysis is the "*rate-deflated*" Fenchel coupling

$$W_t = \frac{1}{\gamma_t} F_{\mathcal{E}}(q, \gamma_t z_t) \tag{5.12.8}$$

where $q \in \Delta(\mathcal{E})$ is the regret comparator, γ_t is the algorithm's learning rate at stage t, and z_t is the corresponding propensity score estimate. The learner's mixed strategy at stage t, denoted as w_t, is determined by the expression $w_t = Q(\gamma_t z_t)$. The energy, denoted as W_t, measures the difference between w_t and the desired strategy q (appropriately adjusted by the learning rate of the method).

We now can state the following inequality on the difference of energies:

Proposition 5.12.3. *For all $q \in \Delta(\mathcal{E})$ and all $t = 1, 2, \ldots$, we have:*

$$W_{t+1} \leq W_t + \langle \hat{l}_t, w_t - q\rangle + (\gamma_{t+1}^{-1} - \gamma_t^{-1})[h_{\mathcal{E}}(q) - \min h_{\mathcal{E}}] + \frac{1}{\gamma_t} F_{\mathcal{E}}(w_t, \gamma_t z_{t+1}). \tag{5.12.9}$$

Proof. By the definition of W_t, we have

$$W_{t+1} - W_t = \frac{1}{\gamma_{t+1}} F_{\mathcal{E}}(q, \gamma_{t+1} z_{t+1}) - \frac{1}{\gamma_t} F_{\mathcal{E}}(q, \gamma_t z_t) = \frac{1}{\gamma_{t+1}} F_{\mathcal{E}}(q, \gamma_{t+1} z_{t+1}) - \frac{1}{\gamma_t} F_{\mathcal{E}}(q, \gamma_t z_{t+1}) \tag{5.12.10a}$$

$$+ \frac{1}{\gamma_t} F_{\mathcal{E}}(q, \gamma_t z_{t+1}) - \frac{1}{\gamma_t} F_{\mathcal{E}}(q, \gamma_t z_t). \tag{5.12.10b}$$

We now proceed to upper-bound each of the two terms (5.12.10a) and (5.12.10b) separately.

For the term (5.12.10a), the definition of the Fenchel coupling (5.12.3) readily yields:

$$\text{(5.12.10a)} = \left[\frac{1}{\gamma_{t+1}} - \frac{1}{\gamma_t}\right] h_{\mathcal{E}}(q) + \frac{1}{\gamma_{t+1}} h^*_{\mathcal{E}}(\gamma_{t+1} z_{t+1}) - \frac{1}{\gamma_t} h^*_{\mathcal{E}}(\gamma_t z_{t+1}). \tag{5.12.11}$$

Inspired by a trick of Nesterov (2009), consider the function $\varphi(\gamma) = \gamma^{-1}[h^*_{\mathcal{E}}(\gamma z) + \min h_{\mathcal{E}}]$. Then, by proposition 5.13.2, letting $w = Q(\gamma z)$ and differentiating φ with respect to γ gives

$$\begin{aligned}
\varphi'(\gamma) &= \frac{1}{\gamma}\langle z, Q(\gamma z)\rangle - \frac{1}{\gamma^2}[h^*_{\mathcal{E}}(\gamma z) + \min h_{\mathcal{E}}] \\
&= \frac{1}{\gamma^2}[\langle \gamma z, w\rangle - h^*_{\mathcal{E}}(\gamma z) - \min h_{\mathcal{E}}] \\
&= \frac{1}{\gamma^2}[h_{\mathcal{E}}(w) - \min h_{\mathcal{E}}] \geq 0.
\end{aligned} \tag{5.12.12}$$

Since $\gamma_{t+1} \leq \gamma_t$, the above shows that $\varphi(\gamma_t) \geq \varphi(\gamma_{t+1})$. Accordingly, setting $z \leftarrow z_{t+1}$ in the definition of φ yields

$$\frac{1}{\gamma_{t+1}} h^*_{\mathcal{E}}(\gamma_{t+1} z_{t+1}) - \frac{1}{\gamma_t} h^*_{\mathcal{E}}(\gamma_t z_{t+1}) \leq \left[\frac{1}{\gamma_t} - \frac{1}{\gamma_{t+1}}\right] \min h_{\mathcal{E}} \tag{5.12.13}$$

and hence

$$\text{(5.12.10a)} \leq (\gamma_{t+1}^{-1} - \gamma_t^{-1})[h_{\mathcal{E}}(q) - \min h_{\mathcal{E}}]. \tag{5.12.14}$$

Now, recall that:

$$\begin{aligned}
z_{t+1} &= z_t - \hat{l}_t \\
w_{t+1} &= \operatorname*{arg\,max}_{w \in \Delta(\mathcal{E})} \{\langle \gamma_{t+1} z_{t+1}, w\rangle - h_{\mathcal{E}}(w)\}
\end{aligned} \tag{EWEN}$$

Then, after a straightforward rearrangement, the second term of (5.12.10) becomes

$$\begin{aligned}
\text{(5.12.10b)} &= \frac{1}{\gamma_t}[h_{\mathcal{E}}(q) + h^*_{\mathcal{E}}(\gamma_t z_{t+1}) - \gamma_t\langle z_{t+1}, q\rangle] - \frac{1}{\gamma_t}[h_{\mathcal{E}}(q) + h^*_{\mathcal{E}}(\gamma_t z_t) - \gamma_t\langle z_t, q\rangle] \\
&= \frac{1}{\gamma_t}\Big[h^*_{\mathcal{E}}(\gamma_t z_{t+1}) - h^*_{\mathcal{E}}(\gamma_t z_t) - \gamma_t\langle \hat{l}_t, q\rangle\Big] && \text{\# \textit{by} (EWEN)} \\
&= \frac{1}{\gamma_t}\Big[h^*_{\mathcal{E}}(\gamma_t z_{t+1}) - h^*_{\mathcal{E}}(\gamma_t z_t) - \gamma_t\langle \hat{l}_t, w_t\rangle\Big] + \langle \hat{l}_t, w_t - q\rangle && \text{\# \textit{isolate benchmark}} \\
&= \frac{1}{\gamma_t}\Big[h^*_{\mathcal{E}}(\gamma_t z_{t+1}) - \langle \gamma_t z_t, w_t\rangle + h_{\mathcal{E}}(w_t) - \gamma_t\langle \hat{l}_t, w_t\rangle\Big] + \langle \hat{l}_t, w_t - q\rangle \\
& && \text{\# \textit{by proposition 5.13.2}} \\
&= \frac{1}{\gamma_t} F_{\mathcal{E}}(u_t, \gamma_t z_{t+1}) + \langle \hat{l}_t, w_t - q\rangle
\end{aligned} \tag{5.12.15}$$

Thus, combining the above with (5.12.14), we finally obtain

$$\begin{aligned}
W_{t+1} &= W_t + \text{(5.12.10a)} + \text{(5.12.10b)} \\
&\leq W_t + (\gamma_{t+1}^{-1} - \gamma_t^{-1})[h_{\mathcal{E}}(q) - \min h_{\mathcal{E}}] + \langle \hat{l}_t, w_t - q\rangle + \frac{1}{\gamma_t} F_{\mathcal{E}}(w_t, \gamma_t z_{t+1})
\end{aligned} \tag{5.12.16}$$

and our proof is complete. □

We are now in a position to state and prove the template inequality that provides the scaffolding for our regret bounds:

Proposition 5.12.4. *The EWEN algorithm enjoys the bound*

$$\mathbb{E}[R_T(q)] \leq \frac{H_{\mathcal{E}}}{\gamma_{T+1}} + \sum_{t=1}^{T} \frac{\mathbb{E}[F_{\mathcal{E}}(w_t, \gamma_t z_{t+1})]}{\gamma_t}. \tag{5.12.17}$$

Proof. Let $Z_t = \hat{l}_t - Ev_t$ denote the error in the learner's estimation of the t-th stage payoff vector v_t. Then, by substituting in proposition 5.12.3 and rearranging, we readily get:

$$\langle Ev_t, q - w_t\rangle \leq W_t - W_{t+1} + \langle Z_t, w_t - q\rangle + \left(\gamma_{t+1}^{-1} - \gamma_t^{-1}\right)[h_{\mathcal{E}}(q) - \min h_{\mathcal{E}}] + \gamma_t F_{\mathcal{E}}(q, \gamma_t z_{t+1}) \tag{5.12.18}$$

Thus, telescoping over $t = 1, 2, \ldots, T$, we have

$$\begin{aligned}\mathrm{Reg}_q(T) &\leq W_1 - W_{T+1} + \left(\frac{1}{\gamma_{T+1}} - \frac{1}{\gamma_1}\right)[h_{\mathcal{E}}(q) - \min h_{\mathcal{E}}] + \sum_{t=1}^{T}\langle Z_t, w_t - q\rangle + \sum_{t=1}^{T}\frac{1}{\gamma_t}F_{\mathcal{E}}(w_t, \gamma_t z_{t+1}) \\ &\leq \frac{h_{\mathcal{E}}(q) - \min h_{\mathcal{E}}}{\gamma_{T+1}} + \sum_{t=1}^{T}\langle Z_t, w_t - q\rangle + \sum_{t=1}^{T}\frac{1}{\gamma_t}F_{\mathcal{E}}(w_t, \gamma_t z_{t+1})\end{aligned} \tag{5.12.19}$$

where we used the fact that

1. $W_t \geq 0$ for all t (a consequence of the first part of proposition 5.12.1); and that
2. $W_1 = \gamma_1^{-1}[h_{\mathcal{E}}(q) + h_{\mathcal{E}}^*(0)] = \gamma_1^{-1}[h_{\mathcal{E}}(q) - \min h_{\mathcal{E}}]$

(from the second part of the same proposition). Our claim then follows by taking expectations in (5.12.19) and noting that $\mathbb{E}[Z_t \,|\, \mathcal{F}_t] = 0$ (by proposition 5.5.1). □

Considering this, we can derive our primary regret bound by bounding the two terms within the template inequality (5.12.9). The second term can be bounded using proposition 5.12.2 applied to results derived in section 5.9. On the other hand, the first term is easily manageable and can be bounded as follows:

Proposition 5.12.5. *For all $q \in \Delta(\mathcal{A})$ and all $t = \{1, 2, \ldots\}$, we have:*

$$F_{\mathcal{E}}(w_t, \gamma_t z_{t+1}) + \gamma_t\langle\hat{l}_t, w_t\rangle = h_{\mathcal{E}}^*(\gamma_t z_t + \gamma_t\hat{l}_t) - h_{\mathcal{E}}^*(\gamma_t z_t). \tag{5.12.20}$$

Proof. Let $q \in \Delta(\mathcal{A})$ and $t \in 1, 2, \ldots$, we simply write:

$$\begin{aligned}F_{\mathcal{E}}(w_t, \gamma_t z_{t+1}) &= h_{\mathcal{E}}(w_t) + h_{\mathcal{E}}^*(\gamma_t z_{t+1}) - \gamma_t\langle z_{t+1}, w_t\rangle \\ &= \underbrace{h_{\mathcal{E}}(w_t) + h_{\mathcal{E}}^*(\gamma_t z_t) - \langle\gamma_t z_t, w_t\rangle}_{=F_{\mathcal{E}}(w_t, \gamma_t z_t)} + h_{\mathcal{E}}^*(\gamma_t z_{t+1}) - h_{\mathcal{E}}^*(\gamma_t z_t) - \gamma_t\langle\hat{l}_t, w_t\rangle \\ &= h_{\mathcal{E}}^*(\gamma_t z_t + \gamma_t\hat{l}_t) - h_{\mathcal{E}}^*(z_t) - \gamma_t\langle\hat{l}_t, w_t\rangle \qquad \#\ F_{\mathcal{E}}(w_t, \gamma_t z_t) = 0\end{aligned}$$

and our assertion follows. □

We are finally in a position to prove our main result (which we restate below for convenience):

Theorem 5.5.1. *Suppose that Algorithm 15 is run with a non-increasing learning rate $\gamma_t > 0$ against a sequence of cost vectors $l_t \in [0,1]^{\mathcal{E}}$, $t = 1, 2, \ldots$, as per* (5.4)*. Then, for all $q \in \Delta(\mathcal{E})$, the learner enjoys the regret bound*

$$\mathbb{E}[R_T(q)] \leq \frac{H_{\mathcal{E}}}{\gamma_{T+1}} + \frac{k_{\text{eff}}}{2} \sum_{t=1}^{T} \gamma_t \tag{5.36}$$

with k_{eff} given by (5.13) *and $H_{\mathcal{E}}$ is defined as the depth over $\Delta(\mathcal{E})$ of the entropic regularizer $h_{\mathcal{E}}$ in* (5.12.1)*, i.e.,*

$$H_{\mathcal{E}} = \max h_{\mathcal{E}} - \min h_{\mathcal{E}} = \log M \tag{5.37}$$

In particular, if Algorithm 15 is run with $\gamma_t = \sqrt{\log M/(2t \cdot k_{\text{eff}})}$, we have

$$\mathbb{E}[R_T(q)] \leq 2\sqrt{k_{\text{eff}} \log M \cdot T}. \tag{5.38}$$

Proof. Injecting Eq. (5.12.20) in the result of proposition 5.12.4 and using proposition 5.12.2 and Eq. (5.35) of proposition 5.5.1 directly yields the pseudo-regret bound (5.36).

Then, we write:

$$H = \max h_{\mathcal{E}} - \min h_{\mathcal{E}} = 0 - \sum_{e \in \mathcal{E}} (1/M) \log(1/M) = \log M.$$

Thus, taking $\gamma_t = \sqrt{\log M/(2t \cdot k_{\text{eff}})}$ and substituting in (5.36) along with the latter finally delivers

$$\mathbb{E}[R_T(q)] \leq 2\sqrt{k_{\text{eff}} \log M \cdot T}, \tag{5.12.21}$$

and our claim follows. □

5.13. The nested entropy and its properties - NEW algorithm

Our aim in this appendix is to prove the basic properties of the series of (negative) entropy functions that fuel the regret analysis of the nested exponential weights (NEW) algorithm.

To begin with, given a similarity structure $\mathcal{S}$ on $\mathcal{A}$ and a sequence of uncertainty parameters $\mu_1 \geq \cdots \geq \mu_L > 0$ (with $\mu_{L+1} = 0$ by convention), we define:

1. The *conditional entropy* of $u \in \Delta(\mathcal{A})$ relative to a target class $S \in \mathcal{S}_\ell$:

$$h(u|S) = \mu_{\ell+1} \sum_{S' \lhd S} u_{S'} \log \frac{u_{S'}}{u_S} = \mu_{\ell+1}\, u_S \sum_{S' \lhd S} u_{S'|S} \log u_{S'|S}. \tag{5.13.1}$$

2. The *nested entropy* of $u \in \Delta(\mathcal{A})$ relative to $S \in \mathcal{S}_\ell$:

$$h_S(u) = \sum_{k=\ell}^{L} \delta_k \sum_{S_k \preccurlyeq_k S} u_{S_k} \log u_{S_k} \tag{5.13.2}$$

where $\delta_k = \mu_k - \mu_{k+1}$ for all $k = 1, \ldots, L$.

3. The *restricted entropy* of $u \in \Delta(\mathcal{A})$ relative to $S \in \mathcal{S}_\ell$:

$$h_{|S}(u) = h_S(u) + \chi_{\Delta(S)}(u) = \begin{cases} h_S(u) & \text{if } u \in \Delta(S), \\ \infty & \text{otherwise,} \end{cases} \tag{5.13.3}$$

where $\chi_{\Delta(S)}$ denotes the (convex) characteristic function of $\Delta(S)$, i.e., $\chi_{\Delta(S)}(u) = 0$ if $u \in \Delta(S)$ and $\chi_{\Delta(S)}(u) = \infty$ otherwise. [Obviously, $h_{|S}(u) = h_S(u)$ whenever $u \in \Delta(S)$.]

Remark 5.13.1. *As per our standard conventions, we are treating S interchangeably as a subset of $\mathcal{A}$ or as an element of $\mathcal{S}$; by analogy, to avoid notational inflation, we are also viewing $\Delta(S)$ as a subset of $\Delta(\mathcal{A})$ – more precisely, a face thereof. Finally, in all cases, the functions $h(u|S)$, $h_S(u)$ and $h_{|S}(u)$ are assumed to take the value $+\infty$ for $u \in \mathbb{R}^{\mathcal{A}} \setminus \Delta(\mathcal{A})$.* ¶

Remark 5.13.2. *For posterity, we also note that the nested and restricted entropy functions ($h_S(u)$ and $h_{|S}(u)$ respectively) are both convex – though not necessarily* strictly *convex – over $\Delta(\mathcal{A})$. This is a consequence of the fact that each summand $u_S \log u_S$ in* (5.13.2) *is convex in u and that $\delta_k = \mu_k - \mu_{k+1} \geq 0$ for all $k = 1, \ldots, L$. Of course, any two distributions $u, u' \in \Delta(\mathcal{A})$ that assign the same probabilities to elements of S but not otherwise have $h_S(u) = h_S(u')$, so h_S is* not *strictly convex over $\Delta(\mathcal{A})$ if $S \neq \mathcal{A}$. However, since the function $\sum_{a \in S} u_a \log u_a$ is strictly convex over $\Delta(S)$, it follows that h_S – and hence $h_{|S}$ –* is *strictly convex over $\Delta(S)$.* ¶

Our main goal in the sequel will be to prove the following fundamental properties of the entropy functions defined above:

Proposition 5.13.1. *For all $S \in \mathcal{S}_\ell$, $\ell = 1, \ldots, L$, and for all $u \in \Delta(\mathcal{A})$, we have:*

$$h_S(u) = \sum_{S' \preccurlyeq S} h(u|S') + \mu_\ell \, u_S \log u_S. \tag{5.13.4}$$

Consequently, for all $u \in \Delta(S)$, we have:

$$h_{|S}(u) = \sum_{S' \preccurlyeq S} h(u|S'). \tag{5.13.5}$$

Proposition 5.13.2. *For all $S \in \mathcal{S}$ and all $y \in \mathbb{R}^{\mathcal{A}}$, we have:*

1. *The recursively defined propensity score y_S of S as given by* (5.17) *can be expressed as*

$$y_S = \max_{u \in \Delta(S)} \{\langle y, u \rangle - h_{|S}(u)\} \tag{5.13.6}$$

2. *The conditional probability of choosing $a \in \mathcal{A}$ given that S has already been selected under* (NLC) *is given by*

$$P_{a|S}(y) = \frac{\partial y_S}{\partial y_a} \tag{5.13.7}$$

and the conditional probability vector $P_{|S}(y) = (P_{a|S}(y))_{a \in \mathcal{A}}$ solves the problem (5.13.6)*, viz.*

$$P_{|S}(y) = \arg\max_{u \in \Delta(S)} \{\langle y, u \rangle - h_{|S}(u)\} \tag{5.13.8}$$

These propositions will be the linchpin of the analysis to follow, so some remarks are in order:

Remark 5.13.3. *Note here that the maximum in* (5.13.6) *is taken over the* restricted *entropy function* $h_{|S}$, not *the nested entropy* h_S. *This distinction will play a crucial role in the sequel; in particular, since* $h_{|S}$ *is strictly convex over* $\Delta(S)$, *it implies that the* $\arg\max$ *in* (5.13.8) *is a singleton.* ¶

Remark 5.13.4. *The first part of proposition 5.13.2 can be rephrased more concisely (but otherwise equivalently) as*

$$y_S = h^*_{|S}(y) \tag{5.13.9}$$

where

$$h^*_{|S}(y) = \max_{u \in \Delta(\mathcal{A})} \{\langle y, u\rangle - h_{|S}(u)\} \tag{5.13.10}$$

denotes the convex conjugate of $h_{|S}$. *This interpretation is conceptually important because it spells out the precise functional dependence between the (primitive) propensity score profile* $y \in \mathbb{R}^{\mathcal{A}}$ *and the propensity scores* y_S *that are propagated to higher-tier similarity classes* $S \in \mathcal{S}$ *via the recursive definition* (5.17). *In particular, this observation leads to the recursive rule*

$$\exp\left(\frac{h^*_{|S}(y)}{\mu_{\ell+1}}\right) = \sum_{S' \lhd S} \exp\left(\frac{h^*_{|S'}(y)}{\mu_{\ell+1}}\right) \quad \textit{for all } S \in \mathcal{S}_\ell,\ \ell = 0, 1, \ldots, L-1. \tag{5.13.11}$$

We will we use this representation freely in the sequel. ¶

Remark 5.13.5. *It is also worth noting that the propensity scores* y_{S_ℓ}, $S_\ell \in \mathcal{S}_\ell$, *can also be seen as primitives for the arborescence* $\mathcal{S}' = \coprod_{k=0}^{\ell} \mathcal{S}_k$ *obtained from* $\mathcal{S}$ *by excising all (proper) descendants of* $\mathcal{S}_\ell$. *Under this interpretation, the second part of proposition 5.13.2 readily gives the more general expression*

$$P_{S'|S}(y) = \frac{\partial y_S}{\partial y_{S'}} \quad \textit{for all } S' \preccurlyeq S, \tag{5.13.12}$$

where, in the right-hand side, y_S *is to be construed as a function of* $y_{S'}$, *defined recursively via* (5.17) *applied to the truncated arborescence* $\mathcal{S}'$. *Even though we will not need this specific result, it is instructive to keep it in mind for the sequel.*

The proofs of propositions 5.13.1 and 5.13.2 were made by Martin et al. (2022).

These properties of the nested entropy function (and its restricted variant) will play a key role in deriving a suitable energy function for the nested exponential weights algorithm. We make this precise in section 5.9 below.

5.14. Auxiliary bounds and results - NEW algorithm

Throughout this appendix, we assume the following primitives:

- A fixed sequence of real numbers $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_L > 0$; all entropy-related objects will be defined relative to this sequence as per the previous section.
- A score vector $y \in \mathbb{R}^{\mathcal{A}}$ that defines inductively the score y_S of any class $S \in \mathcal{S}$ using (5.17), as well as the associated nested choice probability $P(y)$ as per (NLC).

- A vector of cost increments $r = (r_S)_{S\in\mathcal{S}} \in \mathbb{R}^{\mathcal{S}}$ that defines an associated *cost vector* $c \in \mathbb{R}^{\mathcal{A}}$ as per (5.4), viz.

$$c_a = \sum_{S\ni a} r_S \quad \text{for all } a \in \mathcal{A}. \tag{5.14.1}$$

Moreover, for all $c, y \in \mathbb{R}^{\mathcal{A}}$, we define the *nested power sum* function $\sigma_{c,y} : \mathcal{S}\backslash\mathcal{S}_L \to \mathbb{R}$ which, to any $S \in \mathcal{S}\backslash\mathcal{S}_L$, associates the real number

$$\sigma_{c,y}(S) = \begin{cases} \sum\limits_{a\lhd S} P_{a|S}(y)\exp(-c_a/\mu_L) & \text{if } \mathrm{attr}S = L-1, \\ \sum\limits_{S'\lhd S} P_{S'|S}(y)\sigma_{c,y}(S')^{\frac{\mu_{\ell+2}}{\mu_{\ell+1}}} & \text{if } \mathrm{attr}S = \ell < L-1. \end{cases} \tag{5.14.2}$$

The following lemma links the increments of the conjugate entropy h^* to the nested power sum defined above:

Lemma 5.14.1. *For all $y \in \mathbb{R}^{\mathcal{A}}$, $c \in \mathbb{R}^{\mathcal{A}}$, we have*

$$h^*(y-c) = h^*(y) + \mu_1 \log(\sigma_{c,y}(\mathcal{A})). \tag{5.14.3}$$

Lemma 5.14.1 will be proved as a corollary of the more general result below:

Lemma 5.14.2. *Fix some $y \in \mathbb{R}^{\mathcal{A}}$ and $c \in \mathbb{R}^{\mathcal{A}}$. Then, for all $S_\ell \in \mathcal{S}_\ell$, $\ell < L$,we have*

$$\exp\left(\frac{h^*_{|S_\ell}(y-c)}{\mu_{\ell+1}}\right) = \exp\left(\frac{h^*_{|S_\ell}(y)}{\mu_{\ell+1}}\right)\sigma_{c,y}(S_\ell) \tag{5.14.4}$$

Proof of lemma 5.14.1. Simply invoke lemma 5.14.2 with $S \leftarrow \mathcal{A}$. □

The proof of lemma 5.14.2 was made by Martin et al. (2022).

The next lemma provides an upper bound for $\sigma_{c,y}(\mathcal{A})$, which will in turn allow us to derive a bound for the increment of h^*.

Lemma 5.14.3. *For $y \in \mathbb{R}^{\mathcal{A}}$ and $c \in [0,+\infty)^{\mathcal{A}}$, we have:*

$$\sigma_{c,y}(\mathcal{A}) \le 1 - \frac{1}{\mu_1}\left[\sum_{a\in\mathcal{A}} P_a(y)c_a - \frac{1}{2\mu_L}\sum_{a\in\mathcal{A}} P_a(y)c_a^2\right]. \tag{5.14.5}$$

As in the case of 5.14.1, lemma 5.14.3 will follow as a special case of the more general, class-based result below:

Lemma 5.14.4. *Fix some $y \in \mathbb{R}^{\mathcal{A}}$ and $c \in \mathbb{R}^{\mathcal{A}}_+$. Then, for all $S_\ell \in \mathcal{S}_\ell$, $\ell < L$,we have*

$$\sigma_{c,y}(S_\ell) \le 1 - \frac{1}{\mu_{\ell+1}}\left[\sum_{a\in S_\ell} P_{a|S_\ell}(y)c_a - \frac{1}{2\mu_L}\sum_{a\in S_\ell} P_{a|S_\ell}(y)c_a^2\right], \tag{5.14.6}$$

Proof of lemma 5.14.3. Simply invoke lemma 5.14.4 with $S \leftarrow \mathcal{A}$. □

Proof of lemma 5.14.4. We proceed again by descending induction on $\ell = \mathrm{attr}S$.

Base step. Fix some $S \in \mathcal{S}$ with $\mathrm{attr}S = L-1$. We then have:

$$\begin{aligned}
\sigma_{c,y}(S) &= \sum_{S' \lhd S} P_{S'|S}(y) \exp(-\frac{c_{S'}}{\mu_L}) \\
&\leq \sum_{S' \lhd S} P_{S'|S}(y)(1 - \frac{c_{S'}}{\mu_L} + \frac{c_{S'}^2}{2\mu_L^2}) \qquad \text{\# } e^{-x} \leq 1 - x + x^2/2 \textit{ for } x \geq 0 \\
&= 1 - \frac{1}{\mu_L}\left[\sum_{S' \lhd S} P_{S'|S}(y) c_{S'} - \frac{1}{2\mu_L}\sum_{S' \lhd S} P_{S'|S}(y) c_{S'}^2\right] \\
&= 1 - \frac{1}{\mu_{(L-1)+1}}\left[\sum_{a \lhd S} P_{a|S}(y) c_a - \frac{1}{2\mu_L}\sum_{a \lhd S} P_{a|S}(y) c_a^2\right] \qquad (5.14.7)
\end{aligned}$$

so the initialization of the induction process is complete.

Induction step. Fix some $S \in \mathcal{S}$ with $\mathrm{attr}S = \ell - 1$, $\ell < L$, and suppose that (5.14.6) holds at level ℓ. We then have:

$$\begin{aligned}
\sigma_{c,y}(S) &= \sum_{S' \lhd S} P_{S'|S}(y) \sigma_{c,y}(S')^{\frac{\mu_{\ell+1}}{\mu_\ell}} \\
&\leq \sum_{S' \lhd S} P_{S'|S}(y) \left[1 + \frac{1}{\mu_{\ell+1}}\left(-\sum_{a \lhd S'} P_{a|S'}(y) c_a + \frac{1}{2\mu_L}\sum_{a \lhd S'} P_{a|S'}(y) c_a^2\right)\right]^{\frac{\mu_{\ell+1}}{\mu_\ell}} \\
&\qquad \text{\# } \textit{inductive hypothesis} \\
&\leq \sum_{S' \lhd S} P_{S'|S}(y) \left[1 + \frac{1}{\mu_\ell}\left(-\sum_{a \lhd S'} P_{a|S'}(y) c_a + \frac{1}{2\mu_L}\sum_{a \lhd S'} P_{a|S'}(y) c_a^2\right)\right] \\
&\qquad \text{\# } (1+x)^\beta \leq 1 + \beta x \textit{ for } \beta \leq 1 \\
&= 1 + \frac{1}{\mu_\ell}\left[-\sum_{S' \lhd S}\sum_{a \lhd S'} P_{a|S'}(y) P_{S'|S}(y) c_a + \frac{1}{2\mu_L}\sum_{S' \lhd S}\sum_{a \lhd S'} P_{a|S'}(y) P_{S'|S}(y) c_a^2\right] \qquad (5.14.8) \\
&= 1 + \frac{1}{\mu_{(\ell-1)+1}}\left[\sum_{a \lhd S} P_{a|S}(y) c_a + \frac{1}{2\mu_L}\sum_{a \lhd S} P_{a|S}(y) c_a^2\right] \qquad (5.14.9)
\end{aligned}$$

This being true for all $S \in \mathcal{S}$ s.t. $\mathrm{attr}S = \ell - 1$, the induction step and the proof of our assertion are complete. □

With all this in hand, we are now in a position to upper bound the increments of the conjugate nested entropy h^*.

Proposition 5.14.1. *For $y \in \mathbb{R}^{\mathcal{A}}$ and $c \in [0, +\infty)^{\mathcal{A}}$, we have:*

$$h^*(y - c) - h^*(y) \leq -\langle P(y), c\rangle + \frac{1}{2\mu_L}\sum_{a \in \mathcal{A}} P_a(y) c_a^2. \qquad (5.14.10)$$

Proof. Using lemmas 5.14.1 and 5.14.3 and the concavity inequality $\log x \leq x - 1$ directly delivers our assertion. □

Remark 5.14.1. *It is useful to note that, given a cost increment vector $r \in \mathbb{R}^{\mathcal{S}}$ with associated aggregate costs given by $c \in \mathbb{R}^{\mathcal{A}}$ we have:*

$$\begin{aligned}\langle P(y), c\rangle &= \sum_{a\in\mathcal{A}} P_a(y)c_a \\ &= \sum_{a\in\mathcal{A}} P_a(y) \sum_{S\ni a} r_S \\ &= \sum_{a\in\mathcal{A}} P_a(y) \sum_{S\in\mathcal{S}} r_S \, \mathbb{1}_{a\in S} \\ &= \sum_{S\in\mathcal{S}} \left[\sum_{a\in\mathcal{A}} P_a(y)\, \mathbb{1}_{a\in S}\right] r_S \\ &= \sum_{S\in\mathcal{S}} P_S(y) r_S.\end{aligned}$$

We are finally in a position to prove the basic properties of the NIWE estimator, which we restate below for convenience:

Proposition 5.3.1. *Let $\mathcal{S} = \coprod_{\ell=1}^{L} \mathcal{S}_\ell$ be a similarity structure on $\mathcal{A}$. Then, given a mixed strategy $u \in \Delta(\mathcal{A})$ and a vector of cost increments $r \in \mathbb{R}^{\mathcal{S}}$ as per* (5.5)*, the estimator* (NIWE) *satisfies the following:*

1. *It is unbiased:*
$$\mathbb{E}[\hat{r}_S] = r_S \quad \textit{for all } S \in \mathcal{S}. \tag{5.9}$$
2. *It enjoys the importance-weighted mean-square bound*
$$\mathbb{E}\big[u_S \hat{r}_S^2\big] \le \Gamma_S^2 \quad \textit{for all } S \in \mathcal{S}. \tag{5.10}$$

Accordingly, the loss estimator (5.8) *is itself unbiased and enjoys the bound*

$$\mathbb{E}\Big[\sum\nolimits_{a\in\mathcal{A}} u_a \hat{c}_a^2\Big] \le k_{\text{eff}} \tag{5.11}$$

Proof. Fix some $S \in \mathcal{S}$ with $\mathrm{attr} S = \ell \in \{1, \dots, L\}$ and lineage $\mathcal{A} \equiv S_0 \rhd S_1 \rhd \cdots \rhd S_\ell \equiv S$. We will now prove both properties of the (NIWE) estimator.

Part 1. We begin by showing that the estimator (NIWE) is unbiased. Indeed, we have:

$$\begin{aligned}\mathbb{E}[\hat{r}_S] &= \mathbb{E}\left[\frac{\mathbb{1}\{S_\ell = \hat{S}_\ell, \dots, S_1 = \hat{S}_1\}}{u_{S_\ell|S_{\ell-1}} \cdots u_{S_2|S_1} u_{S_1}} r_{S_\ell}\right] = \mathbb{E}\left[\frac{\mathbb{1}\{S = \hat{S}\}}{u_S} r_S\right] \quad \text{\# \textit{Rewriting} (NIWE)} \\ &= \frac{r_S}{u_S} \underbrace{\mathbb{E}\Big[\mathbb{1}\{S = \hat{S}\}\Big]}_{u_S} = r_S. \end{aligned} \tag{5.14.11}$$

Part 2. We now turn to the proof of the importance-weighted mean-square bound of the estimator (NIWE). In this case, for any $S \in \mathcal{S}$, we have:

$$\begin{aligned}\mathbb{E}\big[u_S \hat{r}_S^2\big] = u_S\, \mathbb{E}\big[\hat{r}_S^2\big] &= u_S\, \mathbb{E}\left[\left(\frac{\mathbb{1}\{S=\hat{S}\}}{u_S} r_{S_\ell}\right)^2\right] \\ &= u_S \frac{r_{S_\ell}^2}{u_S^2}\, \mathbb{E}\Big[\mathbb{1}\{S=\hat{S}\}\Big] = r_{S_\ell}^2 \qquad \text{\# because } \mathbb{E}\big[\mathbb{1}\{S=\hat{S}\}\big] = u_S \\ &\leq \Gamma_S^2. \end{aligned} \tag{5.14.12}$$

We are left to derive the bound for the aggregate cost estimator (5.8), viz.

$$\hat{c}_a = \sum_{S \ni a} \hat{r}_S. \tag{5.14.13}$$

With this in mind, we can write:

$$\begin{aligned}\sum_{a\in\mathcal{A}} u_a \hat{c}_a^2 &= \sum_{a\in\mathcal{A}} u_a \left(\sum_{S\ni a} \hat{r}_S\right)^2 \\ &= \sum_{a\in\mathcal{A}} u_a \left[\sum_{S\ni a} \hat{r}_S^2 + 2 \sum_{S'\ni a} \sum_{S\succ S'} \hat{r}_S \hat{r}_{S'}\right] \\ &= \sum_{a\in\mathcal{A}} \sum_{S\in\mathcal{S}} u_a \hat{r}_S^2\, \mathbb{1}_{a\in S} + 2 \sum_{a\in\mathcal{A}} \sum_{S'\in\mathcal{S}} \sum_{S\succ S'} u_a \hat{r}_S \hat{r}_{S'}\, \mathbb{1}_{a\in S'} \\ &= \sum_{S\in\mathcal{S}} \hat{r}_S^2 \underbrace{\sum_{a\in\mathcal{A}} u_a\, \mathbb{1}_{a\in S}}_{u_S} + 2 \sum_{S'\in\mathcal{S}} \sum_{S\succ S'} \hat{r}_S \hat{r}_{S'} \underbrace{\sum_{a\in\mathcal{A}} u_a\, \mathbb{1}_{a\in S'}}_{u_{S'}} \\ &= \sum_{S\in\mathcal{S}} u_S \hat{r}_S^2 + 2 \sum_{S'\in\mathcal{S}} \sum_{S\succ S'} u_{S'} \hat{r}_S \hat{r}_{S'}. \end{aligned} \tag{5.14.14}$$

Now, decomposing the above sums attribute-by-attribute and taking expectations in (5.14.14), we get:

$$\mathbb{E}\left[\sum_{a\in\mathcal{A}} u_a \hat{c}_a^2\right] = \sum_{\ell=1}^{L} \sum_{S_\ell\in\mathcal{S}_\ell} u_{S_\ell}\, \mathbb{E}\big[\hat{r}_{S_\ell}^2\big] + 2 \sum_{1\leq\ell<\ell'\leq L} \sum_{\substack{S_\ell\in\mathcal{S}_\ell \\ S_{\ell'}\prec_{\ell'}\mathcal{S}_\ell}} u_{S_{\ell'}}\, \mathbb{E}\big[\hat{r}_{S_\ell} \hat{r}_{S_{\ell'}}\big]. \tag{5.14.15}$$

The first term in (5.14.15) can simply be bounded using (5.14.12). Indeed:

$$\sum_{\ell=1}^{L} \sum_{S_\ell\in\mathcal{S}_\ell} u_{S_\ell}\, \mathbb{E}\big[\hat{r}_{S_\ell}^2\big] \leq \sum_{\ell=1}^{L} \sum_{S_\ell\in\mathcal{S}_\ell} \Gamma_{S_\ell}^2 = \sum_{\ell=1}^{L} k_\ell \bar{\Gamma}_\ell^2. \tag{5.14.16}$$

with $\bar{\Gamma}_\ell = \sqrt{\frac{1}{k_\ell} \sum_{S_\ell\in\mathcal{S}_\ell} \Gamma_{S_\ell}^2}$ for any $\ell = 1, \ldots, L$.

We now turn to the second term in (5.14.15). Let $\{\varepsilon_{\ell,\ell'}\}_{1\leq\ell'<\ell\leq L}$ be any fixed sequence of positive numbers. For any $\ell,\ell'\in\{1,\dots,L\}$ and any $S_\ell\in\mathcal{S}_\ell$ and $S_{\ell'}\in\mathcal{S}_{\ell'}$, the Peter-Paul inequality yields:

$$2\hat{r}_{S_{\ell'}}\hat{r}_{S_\ell}\leq\frac{1}{\varepsilon_{\ell,\ell'}}\hat{r}^2_{S_{\ell'}}+\varepsilon_{\ell,\ell'}\hat{r}^2_{S_\ell} \tag{5.14.17}$$

Injecting (5.14.17) into the second term of (5.14.15) yields:

$$\begin{aligned}
&2\sum_{1\leq\ell<\ell'\leq L}\sum_{\substack{S_\ell\in\mathcal{S}_\ell\\ S_{\ell'}\prec_{\ell'}\mathcal{S}_\ell}}u_{S_{\ell'}}\,\mathbb{E}\big[\hat{r}_{S_\ell}\hat{r}_{S_{\ell'}}\big]\\
&\qquad\leq\sum_{1\leq\ell<\ell'\leq L}\sum_{\substack{S_\ell\in\mathcal{S}_\ell\\ S_{\ell'}\prec_{\ell'}\mathcal{S}_\ell}}u_{S_{\ell'}}\left(\frac{1}{\varepsilon_{\ell,\ell'}}\,\mathbb{E}\Big[\hat{r}^2_{S_{\ell'}}\Big]+\varepsilon_{\ell,\ell'}\,\mathbb{E}\big[\hat{r}^2_{S_\ell}\big]\right)\\
&\qquad=\sum_{1\leq\ell<\ell'\leq L}\frac{1}{\varepsilon_{\ell,\ell'}}\sum_{\substack{S_\ell\in\mathcal{S}_\ell\\ S_{\ell'}\prec_{\ell'}\mathcal{S}_\ell}}u_{S_{\ell'}}\,\mathbb{E}\Big[\hat{r}^2_{S_{\ell'}}\Big]+\sum_{1\leq\ell<\ell'\leq L}\varepsilon_{\ell,\ell'}\sum_{\substack{S_\ell\in\mathcal{S}_\ell\\ S_{\ell'}\prec_{\ell'}\mathcal{S}_\ell}}u_{S_{\ell'}}\,\mathbb{E}\big[\hat{r}^2_{S_\ell}\big]\\
&\qquad=\sum_{1\leq\ell<\ell'\leq L}\frac{1}{\varepsilon_{\ell,\ell'}}\sum_{\substack{S_\ell\in\mathcal{S}_\ell\\ S_{\ell'}\prec_{\ell'}\mathcal{S}_\ell}}u_{S_{\ell'}}\,\mathbb{E}\Big[\hat{r}^2_{S_{\ell'}}\Big]+\sum_{1\leq\ell<\ell'\leq L}\varepsilon_{\ell,\ell'}\sum_{S_\ell\in\mathcal{S}_\ell}\mathbb{E}\big[\hat{r}^2_{S_\ell}\big]\underbrace{\sum_{S_{\ell'}\prec_{\ell'}\mathcal{S}_\ell}u_{S_{\ell'}}}_{u_{S_\ell}}\\
&\qquad=\sum_{1\leq\ell<\ell'\leq L}\frac{1}{\varepsilon_{\ell,\ell'}}\sum_{S_{\ell'}\in\mathcal{S}_{\ell'}}u_{S_{\ell'}}\,\mathbb{E}\Big[\hat{r}^2_{S_{\ell'}}\Big]+\sum_{1\leq\ell<\ell'\leq L}\varepsilon_{\ell,\ell'}\sum_{S_\ell\in\mathcal{S}_\ell}u_{S_\ell}\,\mathbb{E}\big[\hat{r}^2_{S_\ell}\big]\\
&\qquad\leq\sum_{1\leq\ell<\ell'\leq L}\frac{1}{\varepsilon_{\ell,\ell'}}\sum_{S_{\ell'}\in\mathcal{S}_{\ell'}}\Gamma^2_{S_{\ell'}}+\sum_{1\leq\ell<\ell'\leq L}\varepsilon_{\ell,\ell'}\sum_{S_\ell\in\mathcal{S}_\ell}\Gamma^2_{S_\ell}\qquad\#\textit{ by }(5.14.12)\\
&\qquad\leq\sum_{1\leq\ell<\ell'\leq L}\frac{1}{\varepsilon_{\ell,\ell'}}k_{\ell'}\bar{\Gamma}^2_{\ell'}+\sum_{1\leq\ell<\ell'\leq L}\varepsilon_{\ell,\ell'}k_\ell\bar{\Gamma}^2_\ell.
\end{aligned}\tag{5.14.18}$$

Injecting (5.14.16) and (5.14.18) into (5.14.15) ensures that:

$$\mathbb{E}\Big[\sum\nolimits_{a\in\mathcal{A}}u_a\hat{c}_a^2\Big]\leq\sum\nolimits_{\ell=1}^{L}k_\ell\bar{\Gamma}^2_\ell+\sum\nolimits_{1\leq\ell<\ell'\leq L}\left(\frac{1}{\varepsilon_{\ell,\ell'}}k_{\ell'}\bar{\Gamma}^2_{\ell'}+\varepsilon_{\ell,\ell'}k_\ell\bar{\Gamma}^2_\ell\right)$$

holds for any sequence of positive numbers $\{\varepsilon_{\ell,\ell'}\}_{1\leq\ell'<\ell\leq L}$. As a result, taking $\varepsilon_{\ell,\ell'}=\sqrt{\frac{k_{\ell'}}{k_\ell}}\frac{\bar{\Gamma}_{\ell'}}{\bar{\Gamma}_\ell}$ yields the tight bound

$$\mathbb{E}\Big[\sum\nolimits_{a\in\mathcal{A}}u_a\hat{c}_a^2\Big]\leq\sum\nolimits_{\ell=1}^{L}k_\ell\bar{\Gamma}^2_\ell+2\sum\nolimits_{1\leq\ell<\ell'\leq L}\sqrt{k_{\ell'}}\bar{\Gamma}_{\ell'}\sqrt{k_\ell}\bar{\Gamma}^2_\ell=\left(\sum\nolimits_{\ell=1}^{L}\sqrt{k_\ell}\bar{\Gamma}_\ell\right)^2, \tag{5.14.19}$$

which proves our original assertion.

□

5.15. Regret analysis of the NEW algorithm

As we mentioned in the main text, the principal component of our analysis is a recursive inequality which, when telescoped over $t = 1, 2, \dots$, will yield the desired regret bound. To establish this "template inequality", we will also require an energy function measuring the disparity between a benchmark strategy $u \in \Delta(\mathcal{A})$ and a propensity score profile $y \in \mathbb{R}^{\mathcal{A}}$ as we did in section 5.9. To that end, similarly to the notions introduced in section 5.9, let $h\colon\ \Delta(\mathcal{A}) \to \mathbb{R}$ denote the total nested entropy function

$$h(u) = h_{\mathcal{A}}(u) = \sum_{k=0}^{L} \delta_k \sum_{S_k \in \mathcal{S}_k} u_{S_k} \log u_{S_k}, \quad u \in \Delta(\mathcal{A}), \tag{5.15.1}$$

and let

$$h^*(y) = \max_{u \in \Delta(\mathcal{A})} \{\langle y, u\rangle - h(u)\}, \qquad\qquad y \in \mathbb{R}^{\mathcal{A}}, \tag{5.15.2}$$

denote the convex conjugate of h so, by proposition 5.13.2, we have

$$h^*(y) = y_{\mathcal{A}} \quad \text{and} \quad P_a(y) = \frac{\partial h^*}{\partial y_a} \quad \text{for all } y \in \mathbb{R}^{\mathcal{A}}. \tag{5.15.3}$$

The *Fenchel coupling* between $u \in \Delta(\mathcal{A})$ and $y \in \mathbb{R}^{\mathcal{A}}$ is then defined as

$$F(u, y) = h(u) + h^*(y) - \langle y, u\rangle \quad \text{for all } u \in \Delta(\mathcal{A}),\, y \in \mathbb{R}^{\mathcal{A}}, \tag{5.15.4}$$

and we have the following key result:

Proposition 5.15.1. *Let $\mathcal{S} = \coprod_{\ell=0}^{L} \mathcal{S}_\ell$ be a similarity structure on $\mathcal{A}$ with uncertainty parameters $\mu_1 \geq \cdots \geq \mu_L > 0$. Then:*

1. *The Fenchel coupling* (5.15.4) *is positive-definite, i.e.,*

$$F(u, y) \geq 0 \qquad \textit{for all } u \in \Delta(\mathcal{A}) \textit{ and all } y \in \mathbb{R}^{\mathcal{A}}, \tag{5.15.5}$$

   with equality if and only if u is given by (NLC)*, i.e., if and only if $u = P(y)$.*

2. *For all $u \in \mathcal{A}$, we have*

$$F(u, 0) = h(u) + h^*(0) = h(u) - \min h \tag{5.15.6}$$

   where $\min h \equiv \min_{u' \in \Delta(\mathcal{A})} h(u')$ denotes the minimum of h over $\Delta(\mathcal{A})$.

Proof. Our first claim follows by setting $S \leftarrow \mathcal{A}$ in propositions 5.13.1 and 5.13.2 and noting that $h_S = h_{|S}$ when $S = \mathcal{A}$: indeed, by Young's inequality, we have $h(u) + h^*(y) - \langle y, u\rangle \geq 0$ with equality if and only if $y \in \partial h(u)$, so the equality $u = P(y)$ follows from (Martin et al., 2022, Eq. A.37) applied to $S \leftarrow \mathcal{A}$ and the fact that $P_{a|\mathcal{A}}(y) = P_a(y)$. As for our second claim, simply note that $h^*(0) = \max_{u \in \Delta(\mathcal{A})}\{\langle 0, u\rangle - h(u)\} = -\min_{u \in \Delta(\mathcal{A})} h(u)$ and set $y \leftarrow 0$ in the definition (5.15.4) of the Fenchel coupling. □

As in section 5.9, he specific energy function we will utilize for our regret analysis is the "*rate-deflated*" Fenchel coupling:

$$W_t = \frac{1}{\eta_t} F(p, \eta_t y_t) \tag{5.15.7}$$

where $p \in \Delta(\mathcal{A})$ is the genereic benchmark strategy, η_t is the algorithm's learning rate at stage t, and y_t is the associated propensity score estimate. In words, since the mixed strategy employed by the learner at stage t is $u_t = P(\eta_t y_t)$, the energy W_t essentially measures the disparity between u_t and the target strategy p (suitably rescaled by the method's learning rate). We then have the following fundamental estimate:

Proposition 5.15.2. *For all $p \in \Delta(\mathcal{A})$ and all $t = 1, 2, \dots$, we have:*

$$W_{t+1} \leq W_t + \langle \hat{c}_t, u_t - p \rangle + (\eta_{t+1}^{-1} - \eta_t^{-1})[h(p) - \min h] + \frac{1}{\eta_t} F(u_t, \eta_t y_{t+1}). \tag{5.15.8}$$

Proof. The proof follows the same line as the proof of proposition 5.12.3 in section 5.9.

□

Having reached that standpoint, we are now ready to reiterate the template inequality that forms the framework for our regret bounds:

Proposition 5.4.1. *The NEW algorithm enjoys the bound*

$$\mathbb{E}[R_T(p)] \leq \frac{H}{\eta_{T+1}} + \sum_{t=1}^{T} \frac{\mathbb{E}[F(u_t, \eta_t y_{t+1})]}{\eta_t}. \tag{5.27}$$

Proof. Similarly, the proof follows the same line as the proof of proposition 5.12.4 in section 5.9. □

In view of the above, our main regret bound follows by bounding the two terms in the template inequality (5.15.8). The second term is by far the most difficult one to bound, and is where section 5.9 comes in; the first term is easier to handle, and it can be bounded as follows:

Lemma 5.15.1. *Suppose that each class $S \in \mathcal{S}_{\ell-1}$ has at most s_ℓ children, $\ell = 1, \dots, L$. Then, for all $p \in \Delta(\mathcal{A})$, we have*

$$H \leq \sum_{\ell=1}^{L} \mu_\ell \log s_\ell \qquad \textit{with equality iff the tree is symmetric,} \tag{5.15.9}$$

$$H = \mu \log(k) \qquad \textit{if } \mu_1 = \mu_2 = \cdots = \mu_L = \mu. \tag{5.15.10}$$

Proof. The proof was made by Martin et al. (2022). □

Proposition 5.15.3. *For all $p \in \Delta(\mathcal{A})$ and all $t = \{1, 2, \dots\}$, we have:*

$$F(u_t, \eta_t y_{t+1}) + \eta_t \langle \hat{c}_t, u_t \rangle = h^*(\eta_t y_t + \eta_t \hat{c}_t) - h^*(\eta_t y_t). \tag{5.15.11}$$

Proof. Similarly, the proof follows the same steps as the proof of proposition 5.12.5 in section 5.9. □

At long last, we have reached the point where we can provide a proof for our main result. For the sake of convenience, we restate the result below:

Theorem 5.4.1. *Suppose that Algorithm 14 is run with a non-increasing learning rate $\eta_t > 0$ and uncertainty parameters $\mu_1 \geq \cdots \geq \mu_L > 0$ against a sequence of cost vectors $c_t \in [0,1]^{\mathcal{A}}$, $t = 1, 2, \ldots$, as per* (5.4)*. Then, for all $p \in \Delta(\mathcal{A})$, the learner enjoys the regret bound*

$$\mathbb{E}[R_T(p)] \leq \frac{H}{\eta_{T+1}} + \frac{k_{\text{eff}}}{2\mu_L} \sum_{t=1}^{T} \eta_t \tag{5.19}$$

with k_{eff} given by (5.13) *and $H \equiv H(\mu_1, \ldots, \mu_L)$ defined by setting $y = 0$ in* (5.17) *and taking $H = y_{\mathcal{A}}$, i.e.,*

$$H = \log \left[\sum_{S_1 \lhd S_0} \left[\sum_{S_2 \lhd S_1} \cdots \left[\sum_{S_L \lhd S_{L-1}} 1 \right]^{\frac{\mu_L}{\mu_{L-1}}} \cdots \right]^{\frac{\mu_2}{\mu_1}} \right]^{\mu_1} \tag{5.20}$$

In particular, if Algorithm 14 is run with $\mu_1 = \cdots = \mu_L = \sqrt{k_{\text{eff}}/2}$ and $\eta_t = \sqrt{\log k/(2t)}$, we have

$$\mathbb{E}[R_T(p)] \leq 2\sqrt{k_{\text{eff}} \log k \cdot T}. \tag{5.21}$$

Proof. Injecting Eq. (5.15.11) in the result of proposition 5.4.1 and using proposition 5.14.1 and Eq. (5.11) of proposition 5.3.1 directly yields the pseudo-regret bound (5.19).

Finally, if we choose $\mu_1 = \cdots = \mu_L = \sqrt{k_{\text{eff}}/2}$, lemma 5.15.1 gives

$$H = \sqrt{k_{\text{eff}}/2} \log k. \tag{5.15.12}$$

Thus, taking $\eta_t = \sqrt{\log k/(2t)}$ and substituting in (5.19) along with (5.15.12) finally delivers

$$\mathbb{E}[R_T(p)] \leq 2\sqrt{k_{\text{eff}} \log k \cdot T}, \tag{5.15.13}$$

and our claim follows. □

5.16. Additional Experiment Details and Discussions

In this appendix we provide additional details on the experiments as well as further discussions on the settings we presented. The code with the implementation of the algorithms as well as the code to reproduce the figures will be open-sourced and is provided along with the supplementary materials.

5.16.1. Experiment additional details

In the synthetic environment, at each level, the rewards are generated randomly according for each class nodes, through uniform distributions of randomly generated means and fixed bandwidth. From a level ℓ to the next $\ell + 1$, the rewards range are divided by a multiplicative

factor $\Gamma_\ell/\Gamma_{\ell+1} = 10$. The implemented method of NEW uses the reward based IW. Moreover, no model selection was used in this experiment as no hyperparameter was tuned. Indeed, a decaying rate of $\frac{1}{\sqrt{t}}$ was used for the score updates for all methods, as is common in the bandit literature (Lattimore and Szepesvári, 2020).

5.16.2. Blue Bus/ Red Bus environment

We detail in Figure 5.5 a graphical representation of such blue bus/red bus environment, where many colors of the bus item build irrelevant alternatives. In this setting, with few arms, we run the methods up to the horizon $T = 1000$. We provide in Figure 5.6 the average reward of the two methods NEW and EXP3 with varying number of subclasses of the "bus".

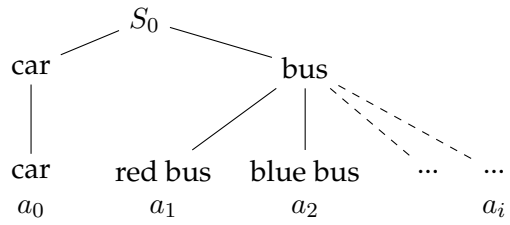

Figure 5.5: Diagram of the blue Bus/Red Bus environment.

While the NEW method ends up selecting the best alternative and having the lowest regret, the EXP3 seems to pick wrong alternative in some experiments, and ends up having higher regret and requiring more iterations to converge to higher average reward. In some of our experiments over the multiple random runs, alternatives of very low sampling probability that were sampled changed the score vector too brutally in the IPS estimator which seemed to hurt the EXP3 method much more than the NEW algorithm.

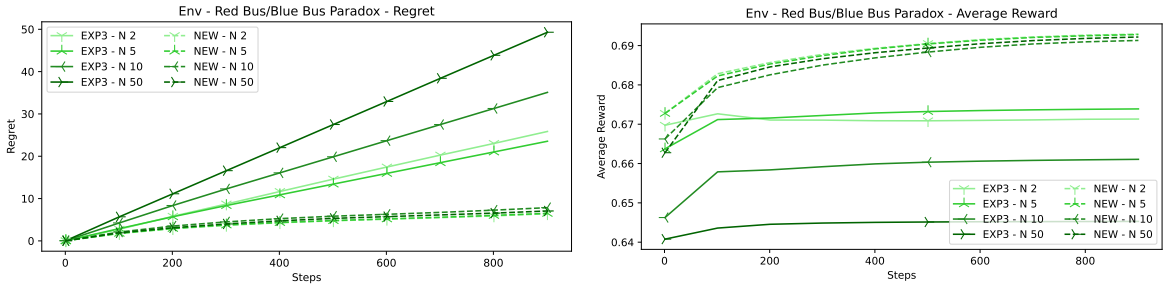

Figure 5.6: Regret and Average Reward of NEW and EXP3 on the Blue Bus/ Red Bus environment.

5.16.3. Tree structures

In this appendix we show additional results and visualisations for the second setting presented in the main paper. We start with discussions on the depth parameter L and follow with the breadth parameter related to the number of child per class $M = |S|$.

Influence of the depth parameter L In Figure 5.7 we show the influence of the depth parameter with a fixed number of child per class. By making the tree deeper, we illustrate the effect of knowing the nested structure compared to running the logit choice to the whole

alternative set. As shown in both the regret and average reward plots, the NEW method outperforms the EXP3 algorithm. While the NEW method also use an IPS estimator, it is less prone to variance issues than the EXP3 method. Indeed, due to the nested structure and the reward decay related to the ratio $\Gamma_{\ell+1}/\Gamma_\ell$, the NEW estimator end up not hurting the regret by still selecting "right" parent classes.

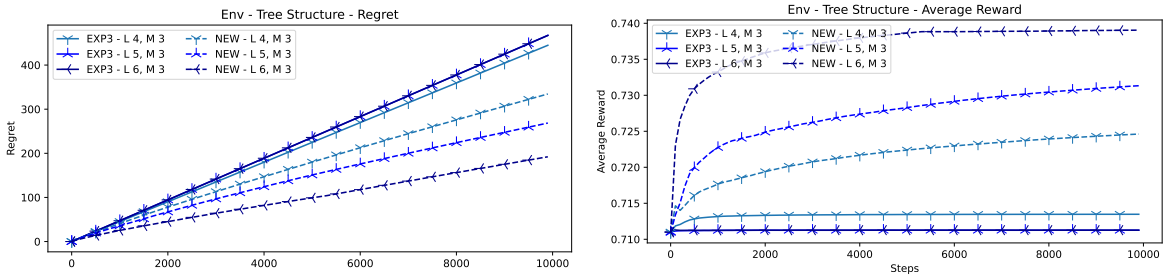

Figure 5.7: Regret and Average Reward of NEW and EXP3 on the synthetic environment with varying number of levels L.

Influence of the number of child per class (wideness) $M = |S|$ In this setting we fix the number of levels L and vary the number of child per classes M. In Figure 5.8 we can see that the NEW method outperforms the EXP3 in terms of regret and average reward. Interestingly, we see that the gap between the two methods shrinks when the number of child per class augments. This is because when the size of a class increase, the NEW method also end up having less knowledge locally and end up having a large number of alternatives to choose among.

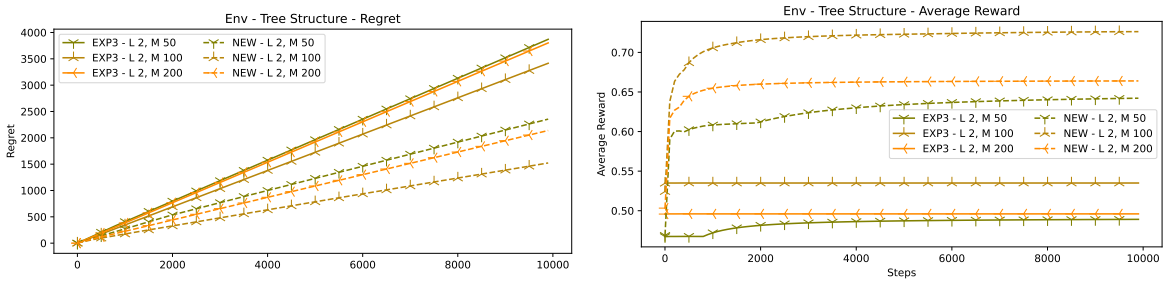

Figure 5.8: Regret and Average Reward of NEW and EXP3 on the synthetic environment with varying number of child per class $M = |S|$.

5.16.4. A visualisation of the effects of NEW

In this appendix we want to show the effects of NEW through the simple setting where we assume a nested structure with $L = 4$ and $M = |S| = 3$. We illustrate in Figure 5.9 the score vectors of the NEW method along the optimal path in the tree (path which nodes have the highest cumulated mean, i.e which generates the highest reward) along with the oracle means of the child nodes. We can see that the algorithm takes advantage of the nested structure and updates the scores vectors optimally with regards to the oracle means of all the nodes. The

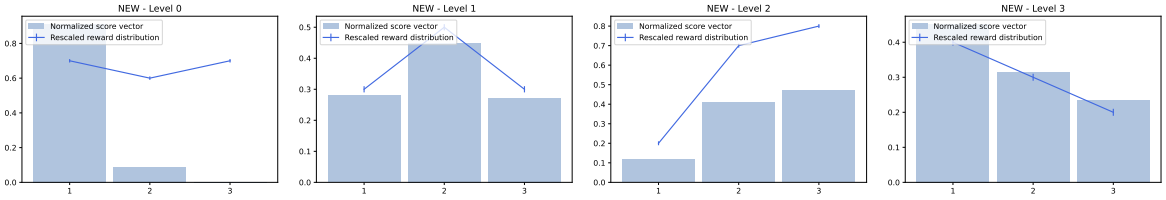

Figure 5.9: Histograms of the score vectors along the optimal path in the nested structure, with visualisation of the mean value of the node.

NEW algorithm therefore estimates correctly the rewards of the environment.

Inversely we see in Figure 5.10 that the EXP3 method has suffered from variance issue and selected a suboptimal alternative among the $|S|^L = 81$ possible ones. The EXP3 did not take advantage of the nested structure and therefore did not learn as correctly as the NEW algorithm the reward values.

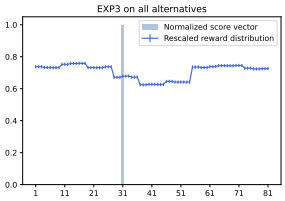

Figure 5.10: Histogram of the score vector of the all alternatives, with a visualisation of the mean value of all nodes.

5.16.5. Cases where both algorithms perform identically

In this appendix we merely show that the implementation of the NEW and EXP3 algorithm match exactly and observe the same behavior when the number of levels L is set to 1. This setting is where we have no knowledge of any nested structure, therefore both algorithms perform identically in Figure 5.11.

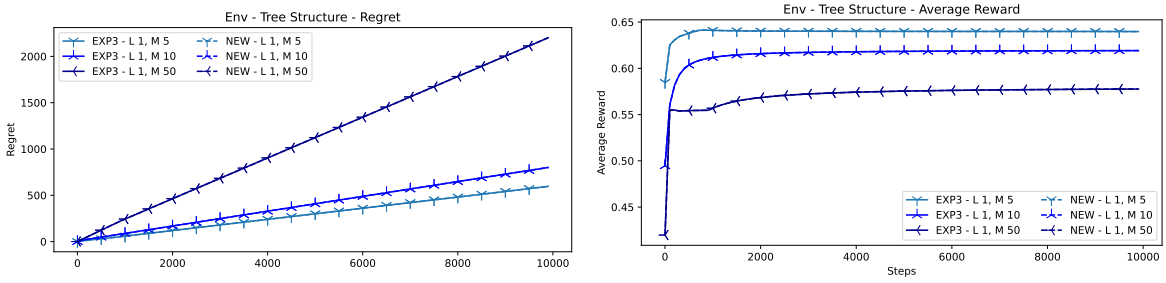

Figure 5.11: Regret and Average Reward of NEW and EXP3 on the synthetic environment where $L = 1$.

5.16.6. Variance plots for the synthetic experiments

We discuss here the variance of the regret at the final timestep $T = 10000$. Indeed, as

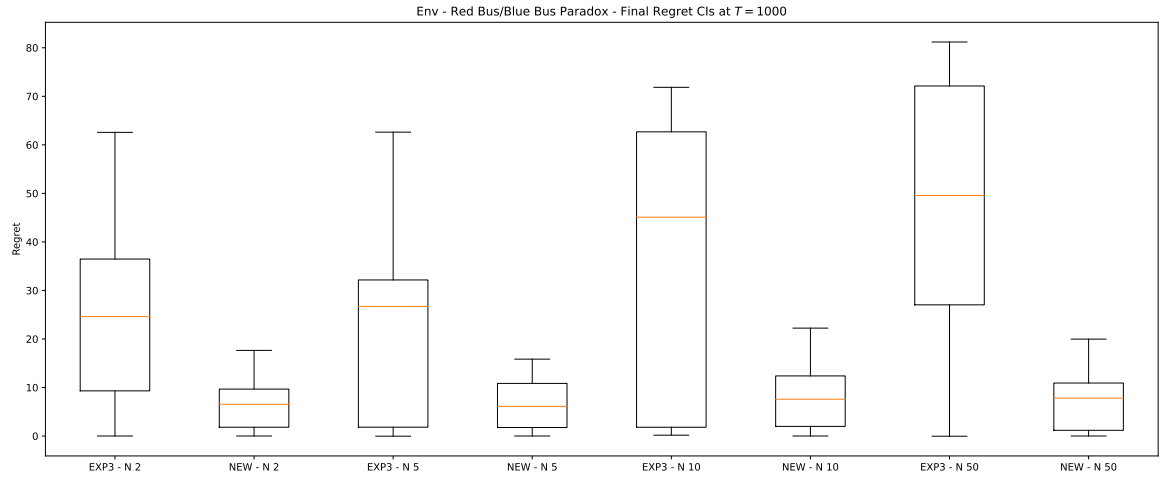

Figure 5.12: Regret distribution at the final stepsize $T = 1000$ for the Red Bus/Blue Bus environment.

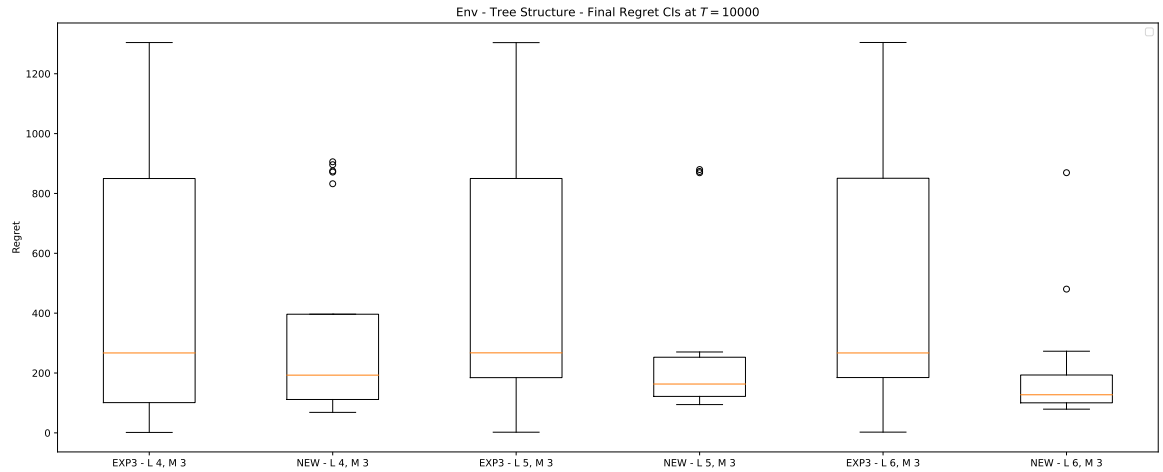

Figure 5.13: Regret distribution at the final stepsize $T = 10000$ when varying the depth parameter L.

shown on Figure 5.6 for the NEW algorithm , on Figure 5.7 for both algorithms EXP3 and NEW, and on Figure 5.8 for EXP3, some of the plots do no exhibit the monotonicity one would have expected when increasing the number of arms through L or M, and are even overlapping on the regret plot. This can be explained on Figures 5.12 for the Red Bus/Blue Bus environment, and in Figures 5.13 and 5.14 respectively for depth and wideness tree experiments. Those plots show the variances (across the 20 random seeds) of the final regret for both methods at the final step-size. In Figure 5.13 we see that the EXP3 arms have similar mean values with large variances, which explains why they are overlapping on the plot in Figure 5.3. In Figure 5.14 when varying M we can also have a closer look on how NEW outperforms EXP3 and how the close values of NEW regrets through different M can be explained by their high variance.

5.16.7. Reproducibility

We provide code for reproducibility of our experiments and plots, in addition to a more

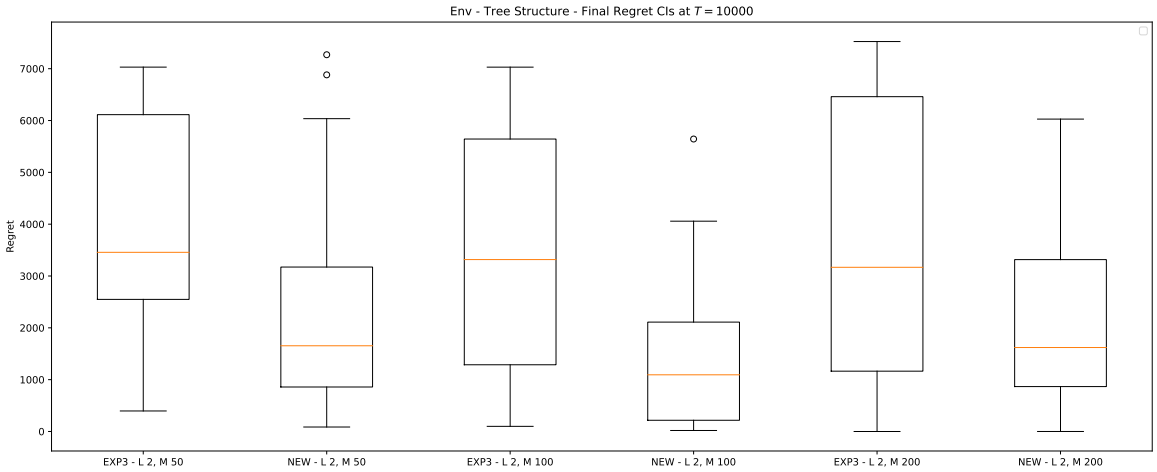

Figure 5.14: Regret distribution at the final stepsize $T = 10000$ when varying the wideness parameter M.

general implementation of both the NEW algorithm and EXP3 baseline. All experiments were run on a Mac book pro laptop, with 1 processor of 6 cores @2.6GHz (6-Core Intel Core i7).

6

Concluding Remarks and Perspectives

This thesis brought various contributions with regard to sequential learning where offline and online settings where subsequently considered. In this work I focused on the effectiveness and the efficiency of those learning methods and their statistical guarantees.

First, in Chapter 2, we addressed the challenge of learning counterfactual stochastic policies from real-world data with continuous actions. This posed obstacles in modelization, optimization, and evaluation within the CRM pipeline. To overcome these challenges, we brought contributions in a novel parametrization technique based on joint kernel embedding of contexts and actions, delivering competitive performance. We highlighted the importance of optimization using techniques like soft-clipping and proximal point methods, along with providing statistical guarantees for our estimator and policy class. Additionally, we introduced an offline evaluation protocol and a large-scale dataset, the first of its kind with real-world logged propensities and continuous actions.

Second, we presented a novel sequential deployment of incrementally optimized CRM policies in Chapter 3 called sequential counterfactual risk minimization (SCRM). Our proposed method introduced a novel counterfactual estimator to enhance variance control in excess risk bounds. By applying these excess risk guarantees sequentially under a weak error bound assumption, we achieved accelerated rates comparable to existing acceleration methods. Notably, our method outperforms CRM in practical applications and is particularly well-suited for this sequential setting. (SCRM) is a significant step towards the realistic learning setting that companies such as Criteo might consider in applications which sits in-between offline and sequential learning.

Third, when online learning is possible, we investigated solutions for the scalability issues of the kernel methods in the optimistic learning algorithms in Chapter 4, where we introduced a method for contextual kernel Upper Confidence Bound (UCB) algorithms in large-scale problems. The proposed EK-UCB algorithm exhibits a space complexity of $\mathcal{O}(Td_{\text{eff}})$ and a time complexity of $\mathcal{O}(CTd_{\text{eff}}^2)$, representing a significant improvement over the standard contextual kernel UCB approach. Notably, while previous efficient Gaussian process algorithms have enabled scalability in learning problems within non-contextual and discrete action environments, we demonstrated the crucial role of incremental projection updates in achieving efficient approximations within the joint context-action space. This

resulted in equivalent regret guarantees at a lower computational cost.

Eventually, note that all previous settings considered stochastic environments, while the arbitrarily changing "adversarial" environment requires less assumptions and might be more suitable for some applications. In the presence of side knowledge on an outcome similarity structure where observed losses also have inherent similarities, we proposed in Chapter 5 an approach that leveraged layered exploration of the set of alternatives based on a nested selection method. This allowed us to improved the regret bounds and benefits were seen in practical implementations of the exponential weights algorithms.

Perspectives Interestingly, when working on the CoCoA large-scale dataset we measured the difficulty to learn policies and assess their statistical significance in real-world settings, even with variance reduction estimators. For future research a promising venue would be to enhance policies as in offline RL (Schulman et al., 2017, 2015)) with distributionally robust methods that constraint the optimized policy to keep closer to the logging and make incremental learning steps. In such a case, incorporating sequential deployments in a CRM procedure would be the most natural learning setting for incremental learning steps, which is what we started with Chapter 3 with (SCRM).

It is important to mention that in our analysis of excess risk bounds in (SCRM) we employed a theoretical algorithm that utilizes geometric sample sizes to discard previous samples, thus avoiding the introduction of dependencies. However, in practice using all past samples has been found to be effective as well and developing guarantees for this scenario would be an interesting area for future research, for example with maximal inequality tools used in (Bibaut et al., 2021a). Furthermore, similarly to online settings that involve an exploration-exploitation tradeoff, exploring in (SCRM) the application of the optimism in the face of uncertainty (OFUL) principle (Abbasi-yadkori et al., 2011) holds promise as a potential avenue for future investigation. Indeed, algorithms in sequential learning that use this optimistic principle prove to be effective in practice.

For the computational efficiency of the kernel contextual bandits, it is worth mentioning that the batching strategy employed by BBKB (Calandriello et al., 2020) can offer benefits even when considering the incremental updates that we used in Chapter 4. This observation presents an intriguing area for future exploration and investigation. Another relevant question is whether it is possible to develop algorithms that not only achieve improved regret guarantees comparable to the elimination algorithms (Valko et al., 2013; Lattimore and Szepesvári, 2020) in the context of finite actions but also provide computational efficiency gains similar to those achieved in our work. In the case where such elimination algorithms would be effective in practice, it would bring significant advancements in the field.

As for the adversarial muti-armed bandit we considered in Chapter 5, a limitation of the framework we proposed is that the nested estimator requires knowledge of the intra-class cost increments; that can be compared to the distinction between the "full bandit" and "semi-bandit" settings found in combinatorial bandits (Cesa-Bianchi and Lugosi, 2012). While this limitation is relevant in various application domains (e.g., path-planning), addressing the fully unobservable case, possibly by adopting an approach similar to the hierarchical contextual analysis proposed by Sen et al. (2021), remains an important open question for future research.

Bibliography

Estimation of the Warfarin dose with clinical and pharmacogenetic data. *New England Journal of Medicine*, 360(8):753–764, 2009.

Y. Abbasi-yadkori, D. Pál, and C. Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

A. Agarwal, D. Hsu, S. Kale, J. Langford, L. Li, and R. Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In *International Conference on Machine Learning (ICML)*, 2014.

Z. Ahmed, N. Le Roux, M. Norouzi, and D. Schuurmans. Understanding the impact of entropy on policy optimization. In *International Conference on Machine Learning (ICML)*, 2019.

A. Alaoui and M. W. Mahoney. Fast randomized kernel ridge regression with statistical guarantees. *Advances in Neural Information Processing Systems (NIPS)*, 2015.

N. Alon, N. Cesa-Bianchi, C. Gentile, and Y. Mansour. From bandits to experts: A tale of domination and independence. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

S. P. Anderson, A. de Palma, and J.-F. Thisse. *Discrete Choice Theory of Product Differentiation*. MIT Press, Cambridge, MA, 1992.

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950.

J.-Y. Audibert and S. Bubeck. Minimax policies for adversarial and stochastic bandits. *Conference on Learning Theory (COLT)*, 2009.

J.-Y. Audibert, R. Munos, and C. Szepesvari. Tuning bandit algorithms in stochastic environments. In *International Conference on Algorithmic Learning Theory*, 2007.

J.-Y. Audibert, S. Bubeck, and G. Lugosi. Minimax policies for combinatorial prediction games. In *Conference on Learning Theory (COLT)*, 2011.

P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Symposium on Foundations of Computer Science*, 1995.

P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256, 05 2002.

P. Auer, N. Cesa-Bianchi, Y. Freund, and R. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32:48–77, 01 2003.

F. Bach. Sharp analysis of low-rank kernel matrix approximations. In *Conference on Learning Theory (COLT)*, 2013.

F. Bach. On the equivalence between kernel quadrature rules and random feature expansions. *Journal of Machine Learning Research*, 18(21):1–38, 2017.

F. R. Bach. Learning theory from first principles. 2023.

F. R. Bach and M. I. Jordan. Predictive low-rank decomposition for kernel methods. In *International Conference on Machine Learning (ICML)*, 2005.

E. Bakshy, L. Dworkin, B. Karrer, K. Kashin, B. Letham, A. Murthy, and S. Singh. Ae: A domain-agnostic platform for adaptive experimentation. 2018.

C. D. Barnes and L. G. Eltherington. *Drug dosage in laboratory animals: a handbook*. Univ of California Press, 1966.

P. L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 2002.

A. Beck. *First-Order Methods in Optimization*. Society for Industrial and Applied Mathematics, 2017.

S. R. Becker, E. J. Candès, and M. C. Grant. Templates for convex cone problems with applications to sparse signal recovery. *Mathematical Programming Computation*, 3(3):165–218, 2011.

M. Ben-Akiva and S. R. Lerman. *Discrete Choice Analysis: Theory and Application to Travel Demand*. MIT Press, Cambridge, 1985.

C. Berge. *Topological Spaces*. Dover, New York, 1997.

A. Berlinet and C. Thomas-Agnan. *Reproducing Kernel Hilbert Space in Probability and Statistics*. Springer US, Kluwer Academic Publishers, 2004.

D. Bertsimas and C. McCord. Optimization over continuous and multi-dimensional decisions with observational data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

A. Bibaut, A. Chambaz, M. Dimakopoulou, N. Kallus, and M. van der Laan. Post-contextual-bandit inference, 2021a. URL `https://arxiv.org/abs/2106.00418`.

A. Bibaut, N. Kallus, M. Dimakopoulou, A. Chambaz, and M. van der Laan. Risk minimization from adaptively collected data: Guarantees for supervised and policy learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021b.

J. Bolte, A. Daniilidis, and A. Lewis. The Łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. *SIAM Journal on Optimization*, 17(4):1205–1223, 2007.

L. Bottou, J. Peters, J. Quiñonero Candela, D. X. Charles, D. M. Chickering, E. Portugaly, D. Ray, P. Simard, and E. Snelson. Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research (JMLR)*, 14(1): 3207–3260, 2013.

S. Boucheron, O. Bousquet, and G. Lugosi. Theory of classification: a survey of some recent advances. *ESAIM: PS*, 9:323–375, 2005.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

S. Bubeck. Introduction to online optimization, December 2011.

S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

S. Bubeck, R. Munos, G. Stoltz, and C. Szepesvári. $\mathcal{X}$-armed bandits. *Journal of Machine Learning Research*, 12:1655–1695, 2011.

D. Calandriello, A. Lazaric, and M. Valko. Efficient second-order online kernel learning with adaptive embedding. In *Advances in Neural Information Processing Systems (NIPS)*, 2017a.

D. Calandriello, A. Lazaric, and M. Valko. Second-order kernel online convex optimization with adaptive sketching. In *International Conference on Machine Learning (ICML)*, 2017b.

D. Calandriello, L. Carratino, A. Lazaric, M. Valko, and L. Rosasco. Gaussian process optimization with adaptive sketching: Scalable and no regret. In *Conference on Learning Theory (COLT)*, 2019.

D. Calandriello, L. Carratino, A. Lazaric, M. Valko, and L. Rosasco. Near-linear time Gaussian process optimization with adaptive batching and resparsification. In *International Conference on Machine Learning (ICML)*, 2020.

R. Camilleri, J. Katz-Samuels, and K. Jamieson. High-dimensional experimental design and kernel bandits. In *International Conference on Machine Learning (ICML)*, 2021.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

N. Cesa-Bianchi and G. Lugosi. Combinatorial bandits. In *Conference on Learning Theory (COLT)*, 2009.

N. Cesa-Bianchi and G. Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78:1404–1422, 2012.

N. Cesa-Bianchi and O. Shamir. Bandit regret scaling with the effective loss range. In *International Conference on Algorithmic Learning Theory (ALT)*, 2018.

N. Cesa-Bianchi, P. Gaillard, C. Gentile, and S. Gerchinovitz. Algorithmic chaining and the role of partial feedback in online nonparametric learning. In *Conference on Learning Theory (COLT)*, 2017.

G. Chen, D. Zeng, and M. R. Kosorok. Personalized dose finding using outcome weighted learning. *Journal of the American Statistical Association*, 111(516):1509–1521, 2016.

S. R. Chowdhury and A. Gopalan. On kernelized multi-armed bandits. In *International Conference on Machine Learning (ICML)*, 2017.

W. Chu, L. Li, L. Reyzin, and R. Schapire. Contextual bandits with linear payoff functions. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

B. Dai, O. Nachum, Y. Chow, L. Li, C. Szepesvári, and D. Schuurmans. Coindice: Off-policy confidence interval estimation. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

M. Demirer, V. Syrgkanis, G. Lewis, and V. Chernozhukov. Semi-parametric efficient policy learning with continuous actions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

M. Derezinski, F. Liang, and M. Mahoney. Bayesian experimental design using regularized determinantal point processes. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

M. Dudik, J. Langford, and L. Li. Doubly robust policy evaluation and learning. In *International Conference on Machine Learning (ICML)*, 2011.

M. Dudík, D. Erhan, J. Langford, and L. Li. Doubly robust policy evaluation and optimization. *Statist. Sci.*, 29(4):485–511, 11 2014.

A. d'Aspremont, D. Scieur, and A. Taylor. Acceleration methods. *Foundations and Trends® in Optimization*, 5(1-2):1–245, 2021.

L. Faury, U. Tanielian, E. Dohmatob, E. Smirnova, and F. Vasile. Distributionally Robust Counterfactual Risk Minimization. *Conference on Artificial Intelligence (AAAI)*, 2020.

S. Fine and K. Scheinberg. Efficient svm training using low-rank kernel representations. 2002.

D. Foster and A. Rakhlin. Beyond UCB: Optimal and efficient contextual bandits with regression oracles. *International Conference on Machine Learning (ICML)*, 2020.

D. Foster and A. Rakhlin. Bridging learning and decision making. *International Conference on Machine Learning, Tutorial*, 2022.

M. Fukushima and H. Mine. A generalized proximal point algorithm for certain non-convex minimization problems. *International Journal of Systems Science*, 12(8):989–1000, 1981.

P. Gaillard. Lecture notes on sequential learning. 2022.

P. Gaillard and O. Wintenberger. Efficient online algorithms for fast-rate regret bounds under sparsity. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Z. Gao, Y. Han, Z. Ren, and Z. Zhou. Batched multi-armed bandits problem. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, ii: Shrinking procedures and optimal algorithms. *SIAM Journal on Optimization*, 23(4):2061–2089, 2013.

A. Goldenshluger and A. Zeevi. A linear response bandit problem. *Stochastic Systems*, 3(1): 230 – 261, 2013.

A. György, T. Linder, G. Lugosi, and G. Ottucsák. The online shortest path problem under partial monitoring. *Journal of Machine Learning Research*, 8:2369–2403, 2007.

Y. Han, Z. Zhou, Z. Zhou, J. Blanchet, P. W. Glynn, and Y. Ye. Sequential batch learning in finite-action linear contextual bandits, 2020.

A. Harutyunyan, M. G. Bellemare, T. Stepleton, and R. Munos. Q(λ) with off-policy corrections. In *Conference on Algorithmic Learning Theory (ALT)*, 2016.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. 2001.

A. Héliou, M. Martin, P. Mertikopoulos, and T. Rahier. Zeroth-order non-convex learning via hierarchical dual averaging. In *International Conference on Machine Learning (ICML)*, 2021.

K. Hirano and G. W. Imbens. The propensity score with continuous treatments. *Applied Bayesian modeling and causal inference from incomplete-data perspectives*, 226164:73–84, 2004.

D. G. Horvitz and D. J. Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47(260):663–685, 1952.

A. Iouditski and Y. Nesterov. Primal-dual subgradient methods for minimizing uniformly convex functions. *arXiv preprint arXiv:1401.1792*, 2014.

R. Jézéquel, P. Gaillard, and A. Rudi. Efficient online learning with kernels for adversarial large scale problems. In *Advances in Neural Information Processing Systems (NeurIPS)*. 2019.

N. Jiang and L. Li. Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2016.

T. Joachims, A. Swaminathan, and M. de Rijke. Deep learning with logged bandit feedback. In *International Conference on Learning Representations (ICLR)*, 2018.

S. Kakade and J. Langford. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2002.

S. M. Kakade, K. Sridharan, and A. Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.

N. Kallus and A. Zhou. Policy evaluation and optimization with continuous treatments. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018.

M. Kasy and A. Sautmann. Adaptive treatment assignment in experiments for policy choice. *Econometrica*, 89(1):113–132, 2021.

T. Kocák, G. Neu, M. Valko, and R. Munos. Efficient learning by implicit exploration in bandit problems with side observations. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

R. Kohavi, R. Longbotham, D. Sommerfield, and R. M. Henne. Controlled experiments on the web: survey and practical guide. *Data mining and knowledge discovery*, 18:140–181, 2009.

A. Krause and C. S. Ong. Contextual gaussian process bandit optimization. In *Advances in neural information processing systems (NIPS)*, 2011.

A. Kulunchakov and J. Mairal. A generic acceleration framework for stochastic composite optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

I. Kuzborskij, L. Cella, and N. Cesa-Bianchi. Efficient linear bandits through matrix sketching. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

T. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

J. Langford and T. Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.

T. Lattimore and C. Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

D. Lefortier, A. Swaminathan, X. Gu, T. Joachims, and M. de Rijke. Large-scale validation of counterfactual learning methods: A test-bed. *arXiv preprint arXiv:1612.00367*, 2016.

L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *International Conference on World Wide Web (WWW)*, 2010.

L. Li, W. Chu, J. Langford, T. Moon, and X. Wang. An unbiased offline evaluation of contextual bandit algorithms with generalized linear models. In *Workshop on On-line Trading of Exploration and Exploitation 2*, 2012.

N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

D. C. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

R. D. Luce. *Individual Choice Behavior: A Theoretical Analysis*. Wiley, New York, 1959.

M. W. Mahoney and P. Drineas. Cur matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences (PNAS)*, 106(3):697–702, 2009.

M. Majzoubi, C. Zhang, R. Chari, A. Krishnamurthy, J. Langford, and A. Slivkins. Efficient contextual bandits with continuous actions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

S. Mannor and O. Shamir. From bandits to experts: On the value of side-observations. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

M. Martin, P. Mertikopoulos, T. Rahier, and H. Zenati. Nested bandits. *International Conference on Machine Learning (ICML)*, 2022.

A. Maurer and M. Pontil. Empirical bernstein bounds and sample variance penalization. In *Conference on Learning Theory (COLT)*, 2009.

D. L. McFadden. Conditional logit analysis of qualitative choice behavior. In P. Zarembka, editor, *Frontiers in Econometrics*, pages 105–142. Academic Press, New York, NY, 1974.

H. McMahan and M. Streeter. Tighter bounds for multi-armed bandits with expert advice. 01 2009.

P. Mertikopoulos. *Online optimization and learning in games: Theory and applications*. Habilitation à diriger des recherches, Grenoble 1 UGA - Université Grenoble Alpes, 2019.

A. M. Metelli, A. Russo, and M. Restelli. Subgaussian and differentiable importance sampling for off-policy evaluation and learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

R. Munos, T. Stepleton, A. Harutyunyan, and M. G. Bellemare. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

M. Mutnỳ and A. Krause. Efficient high dimensional bayesian optimization with additivity and quadrature fourier features. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

T. Nedelec, N. L. Roux, and V. Perchet. A comparative study of counterfactual estimators. *arXiv preprint arXiv:1704.00773*, 2017.

T. Nedelec, C. Calauzènes, N. E. Karoui, and V. Perchet. Learning in repeated auctions. *Foundations and Trends in Machine Learning*, 15, 2022.

Y. Nesterov. A method of solving a convex programming problem with convergence rate o(1/k2. *Soviet Mathematics Doklady.*, 1983.

Y. Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical Programming*, 120:221–259, 04 2009.

Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140:125 – 161, 2012.

G. Neu. Explore no more: Improved high-probability regret bounds for non-stochastic bandits. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, NY, USA, 2e edition, 2006.

A. B. Owen. *Monte Carlo theory, methods and examples*. 2013.

C. Paquette, H. Lin, D. Drusvyatskiy, J. Mairal, and Z. Harchaoui. Catalyst for gradient-based nonconvex optimization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018.

V. Perchet, P. Rigollet, S. Chassang, and E. Snowberg. Batched bandit problems. *The Annals of Statistics*, 44, 05 2015.

J. Peters, D. Janzing, and B. Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2017.

M. J. Powell. Restart procedures for the conjugate gradient method. *Math. Program.*, 12(1): 241–254, dec 1977.

A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22 (268):1–8, 2021.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. *Advances in Neural Information Processing Systems*, 2007.

H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527 – 535, 1952.

J. M. Robins and A. Rotnitzky. Semiparametric efficiency in multivariate regression models with missing data. *Journal of the American Statistical Association*, 90(429):122–129, 1995.

R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM journal on control and optimization*, 14(5):877–898, 1976.

A. Rudi and L. Rosasco. Generalization properties of learning with random features. *Advances in Neural Information Processing Systems (NIPS)*, 2017.

A. Rudi, R. Camoriano, and L. Rosasco. Less is more: Nyström computational regularization. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

N. Sachdeva, Y. Su, and T. Joachims. Off-policy bandits with deficient support. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, 2020.

J. Scarlett, I. Bogunovic, and V. Cevher. Lower bounds on regret for noisy gaussian process bandit optimization. In *Conference on Learning Theory (COLT)*, 2017.

B. Schölkopf and A. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. 2015.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

R. Sen, A. Rakhlin, L. Ying, R. Kidambi, D. Foster, D. Hill, and I. Dhillon. Top-k eXtreme contextual bandits with arm hierarchy. In *International Conference on Machine Learning (ICML)*, 2021.

S. Shalev-Shwartz. Online learning: theory, algorithms and applications (phd thesis.). 2007.

S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

D. Simchi-Levi and Y. Xu. Bypassing the monster: A faster and simpler optimal algorithm for contextual bandits under realizability. *Mathematics of Operations Research*, 2022.

A. Smola and B. Schölkopf. Sparse greedy matrix approximation for machine learning. 2000.

N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning (ICML)*, 2010.

Y. Su, L. Wang, M. Santacatterina, and T. Joachims. CAB: Continuous adaptive blending for policy evaluation and learning. 2019.

R. S. Sutton and A. G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998.

R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems (NIPS)*, 2000.

A. Swaminathan and T. Joachims. Counterfactual risk minimization: Learning from logged bandit feedback. In *International Conference on Machine Learning (ICML)*, 2015a.

A. Swaminathan and T. Joachims. The self-normalized estimator for counterfactual learning. In *Advances in Neural Information Processing Systems (NIPS)*. 2015b.

W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

T. S. Thune and Y. Seldin. Adaptation to easy data in prediction with limited advice. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

A. Tirinzoni, M. Pirotta, M. Restelli, and A. Lazaric. An asymptotically optimal primal-dual incremental algorithm for contextual linear bandits. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

M. Valko. *Bandits on graphs and structures*. Habilitation à diriger des recherches, École normale supérieure de Cachan - ENS Cachan, 2016.

M. Valko, N. Korda, R. Munos, I. Flaounas, and N. Cristianini. Finite-time analysis of kernelised contextual bandits. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2013.

J. P. Vert and J. Mairal. Machine learning with kernel methods. 2020.

V. G. Vovk. Aggregating strategies. In *Workshop on Computational Learning Theory (COLT)*, pages 371–383, 1990.

Y.-X. Wang, A. Agarwal, and M. Dudík. Optimal and adaptive off-policy evaluation in contextual bandits. In *International Conference on Machine Learning (ICML)*, 2017.

C. K. Williams and M. Seeger. Using the nyström method to speed up kernel machines. *Advances in Neural Information Processing Systems (NIPS)*, 2001.

R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

H. Zenati, T. Rahier, M. Martin, and P. Mertikopoulos. Sequential Decision Processes with Outcome Similarities.

H. Zenati, A. Bietti, M. Martin, E. Diemert, P. Gaillard, and J. Mairal. Counterfactual learning of stochastic policies with continuous actions: from models to offline evaluation. *arXiv preprint arXiv:2004.11722*, 2020a.

H. Zenati, A. Bietti, M. Martin, E. Diemert, and J. Mairal. Optimization approaches for counterfactual risk minimization with continuous actions. *International Conference on Learning Representation (ICLR), Causal Learning for Decision Making Workshop*, 2020b.

H. Zenati, A. Bietti, E. Diemert, J. Mairal, M. Martin, and P. Gaillard. Efficient kernelized ucb for contextual bandits. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2022.

H. Zenati, E. Diemert, M. Martin, J. Mairal, and P. Gaillard. Sequential counterfactual risk minimization. *International Conference on Machine Learning (ICML)*, 2023.

R. Zhan, Z. Ren, S. Athey, and Z. Zhou. Policy learning with adaptively collected data. *arXiv preprint arXiv:2105.02344*, 2021.

J. Zimmert and Y. Seldin. An optimal algorithm for stochastic and adversarial bandits. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

M. A. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning (ICML)*, 2003.

S. Łojasiewicz. *Une propriété topologique des sous ensembles analytiques réels*. Les équations aux dérivées partielles, 1963.

S. Łojasiewicz. Sur la géométrie semi et sous analytique. *Annales de l'institut Fourier*, 43(5): 1575–1595, 1993.