

UNIVERSITÀ DEGLI STUDI DI MILANO



PH.D. PROGRAM IN COMPUTER SCIENCE

(XXXVIII CYCLE)

DEPARTMENT OF COMPUTER SCIENCE

A thesis submitted for the degree of

Doctor of Philosophy

Learning under Structure and Uncertainty: Algorithms for Bandit and Online Decision Making

Subject Area: INF/01

Author

Alberto RUMI

Supervisor

Prof. Nicolò CESA-BIANCHI

Co-Supervisor

Dr. Fabio VITALE

PhD Coordinator

Prof. Roberto SASSI

Academic Year 2025–2026

Abstract

This thesis investigates the design and analysis of algorithms for online decision-making under uncertainty, with a particular focus on the multi-armed bandit and online learning frameworks. At the core of these settings lies a simple yet powerful interaction protocol: at each round the learner receives a **question** (information available prior to a decision), provides an **answer** (an action or prediction), and observes a **feedback** signal that may be partial, noisy, or structured. This iterative loop captures a wide range of sequential decision problems and serves as the unifying perspective for the contributions of this thesis. We address this challenge across four distinct, fundamental problems in online learning.

First, we address settings in which the learner receives no additional information prior to making a decision (i.e., the *question* is implicit), and must adapt to a nonstationary environment under bandit feedback. We introduce a parameter-free algorithm achieving optimal dynamic regret guarantees without prior knowledge of the environment’s variability. This result resolves a long-standing open problem in the field.

Next, we enrich the *answer* phase by allowing the learner to abstain. We propose an efficient algorithm, that leverages this option to obtain cumulative reward guarantees that can outperform standard methods, while matching their worst-case performance. This framework is then extended to the adversarial contextual bandit problem with abstention.

We investigate problems with combinatorial *answers*, where the learner selects a subset of actions and the reward *feedback* is defined by a sum-max function, a rich subclass of monotone submodular functions. We design an algorithm for this bandit problem, and prove it achieves significant improvement in regret with respect to standard algorithms for monotone submodular functions.

Finally, we consider the Stochastic Shortest Path problem, where the additional structure is dictated by an underlying Markov Decision Process. We analyze the impact of sparse cost *feedback*, demonstrating that standard methods fail to exploit it. We design a novel family of regularizers that achieves an optimal, sparsity-adaptive regret bound. This result, complemented by a matching lower bound, provides a complete characterization of the benefits of sparsity in this setting.

The contributions in this thesis demonstrate that by designing principled algorithms that recognize and exploit structure within the *question*, *answer*, *feedback* loop, we can achieve strong, adaptive guarantees for learning under uncertainty.

Abstract

Questa tesi indaga la progettazione e l'analisi di algoritmi per il processo decisionale online in condizioni di incertezza, con un focus particolare sui framework del multi-armed bandit e dell'apprendimento online. Al centro di questi scenari si trova un protocollo di interazione semplice ma potente: a ogni round il learner riceve una **domanda** (informazioni disponibili prima di una decisione), fornisce una **risposta** (un'azione o una previsione) e osserva un segnale di **feedback** che può essere parziale, rumoroso o strutturato. Questo ciclo iterativo cattura un'ampia gamma di problemi decisionali e funge da prospettiva unificante per i contributi di questa tesi. Affrontiamo questa sfida attraverso quattro problemi distinti e fondamentali nell'apprendimento online.

In primo luogo, affrontiamo scenari in cui il learner non riceve informazioni aggiuntive prima di prendere una decisione (ovvero, la *domanda* è implicita) e deve adattarsi a un ambiente non stazionario con feedback parziale. Introduciamo un algoritmo parameter-free che ottiene garanzie ottimali di regret dinamico senza alcuna conoscenza a priori delle dinamiche dell'ambiente. Questo risultato risolve un problema aperto da lungo tempo nel settore.

Successivamente, arricchiamo la fase di *risposta* consentendo al learner di astenersi. Proponiamo un algoritmo efficiente che sfrutta questa opzione per ottenere garanzie sulla ricompensa cumulata che possono superare i metodi standard, eguagliando al contempo le loro prestazioni nel caso peggiore. Questo framework viene poi esteso al problema del contextual bandit avversariale con astensione.

Indaghiamo poi problemi con *risposte* combinatorie, in cui il learner seleziona un sottoinsieme di azioni e il *feedback* di ricompensa è definito da una funzione *sum-max*, una ricca sottoclasse di funzioni submodulari monotone. Proponiamo quindi un algoritmo per questo problema e dimostriamo che ottiene un miglioramento significativo nel regret rispetto agli algoritmi standard per funzioni submodulari monotone.

Infine, consideriamo il problema dello Stochastic Shortest Path, dove la struttura aggiuntiva è dettata da un sottostante Processo Decisionale Markoviano. Analizziamo l'impatto del *feedback* nel caso in cui è sparso, dimostrando che i metodi standard non riescono a sfruttare questa proprietà. Progettiamo una nuova famiglia di regolarizzatori che ottiene garanzie sul regret ottimali e adattive alla sparsità. Questo risultato, integrato da un lower bound, fornisce una caratterizzazione completa dei benefici della sparsità in questo contesto.

I contributi di questa tesi dimostrano che, progettando algoritmi rigorosi che riconoscono e sfruttano la struttura all'interno del ciclo *domanda*, *risposta*, *feedback*, possiamo ottenere garanzie forti e adattive per l'apprendimento in condizioni di incertezza.

Acknowledgments

First and foremost, I extend my deepest gratitude to my supervisors, Nicolò Cesa-Bianchi and Fabio Vitale. Their invaluable guidance, support, and inspiration have been the bedrock of this entire journey. I could not have asked for better mentors.

My sincere thanks also go to all my collaborators and the LAILA lab. You made all of this work possible and my journey into online learning an amazing and rewarding experience.

To my all family, my mum Laura, my dad Gianni, and my brother Paolo, thank you for being my constant source of love and support, and the best ohana ever. This gratitude also extends to my extended family — especially my aunt Adriana, uncle Bruno, grandma Cristina, and grandad Pippo —, for their love and support throughout this journey.

I am forever grateful to Flaminia for being by my side through every challenge and for patiently bearing all of my discussions. Your support has meant the world to me, and I look forward to annoying you more in the future.

To my lifelong friends, Berga and Fina, thank you for always reminding me of what truly matters.

Last but certainly not least, I want to thank my second family, the Ezios. Thank you for being my greatest source of fun and distraction, for the countless adventures we've shared, and for the invaluable memories we've made. I look forward to many more to come.

Preface

The contents of this dissertation are based on the following published papers or manuscripts under review, emerged from different collaborations with the respective co-authors during my PhD activities. Here is a comprehensive list divided into chapters:

- Chapter 3 is based on the following paper, appeared at The 18th European Workshop on Reinforcement Learning, whose extended version currently under review:

Alberto Rumi, Andrew Jacobsen, Nicolò Cesa-Bianchi, and Fabio Vitale. Parameter-free dynamic regret for unconstrained linear bandits. In *Eighteenth European Workshop on Reinforcement Learning*, 2025.

- Chapter 4 is based on the published conference paper:

Stephen Pasteris, Alberto Rumi, Maximilian Thiessen, Shota Saito, Atsushi Miyauchi, Fabio Vitale, and Mark Herbster. Bandits with abstention under expert advice. *Advances in Neural Information Processing Systems*, 37:126059–126087, 2024a.

- Chapter 5 is based on the published conference paper:

Stephen Pasteris, Alberto Rumi, Fabio Vitale, and Nicolò Cesa-Bianchi. Sum-max submodular bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 1216–1224. PMLR, 2024b.

- Chapter 6 is based on the following paper whose extended version is to appear in the The Thirty-Ninth Annual Conference on Neural Information Processing Systems, 2025:

Emmeran Johnson, Alberto Rumi, Ciara Pike-Burke, and Patrick Rebeschini. Stochastic shortest path with sparse adversarial costs. In *Eighteenth European Workshop on Reinforcement Learning*, 2025.

The following published conference paper has also been the result of my activities during the PhD program:

Yuko Kuroki, Alberto Rumi, Taira Tsuchiya, Fabio Vitale, and Nicolò Cesa-Bianchi. Best-of-both-worlds algorithms for linear contextual bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 1216–1224. PMLR, 2024.

This work has not been included in the thesis, as my contribution was primarily focused on verification and proof checking, rather than on the design and derivation of the main techniques.

Contents

Acknowledgments	iii
Preface	v
1 Introduction	1
1.1 Definitions and Notations	2
2 Introduction to Online Learning	3
2.1 Prediction with Expert Advice	3
2.1.1 Introduction to the main algorithms	4
2.2 Limited feedback: the bandit setting	6
2.2.1 Bandits with Expert Advice	7
3 Dynamic Regret and Parameter Free Guarantees	9
3.0.1 Our contributions	10
3.0.2 Related works	11
3.1 Combining guarantees with uniform sampling	12
3.2 Linear Bandits	14
3.3 Conclusions and Future Work	17
4 Bandits with Abstention under Expert Advice	19
4.1 Additional related work	20
4.2 Problem formulation and notation	20
4.3 Main result	21
4.4 The CBA algorithm	22
4.5 Adversarial contextual bandits with abstention	25
4.5.1 A lower bound	27
4.5.2 Efficient learning with balls	27
4.6 Experiments	27
5 Sum-Max Submodular Bandits	31
5.1 Sum-Max Functions	33
5.2 Additional Related Work	35
5.3 Main Result	36
5.4 Bandit Facility Location	38
5.5 Analysis	39
5.6 Experiments	42
6 Stochastic Shortest Path under Sparsity	45
6.1 Related works	47
6.2 Preliminaries	48
6.2.1 Problem setting	48

6.2.2	SSP as online linear optimisation and online mirror descent	50
6.3	Failure of negative entropy regularization	51
6.4	The benefits of sparsity	52
6.4.1	Sparse-agnostic parameter-free upper bound	54
6.4.2	Lower bound	55
6.5	Unknown transition setting	56
6.6	Conclusion, limitations and future-work	56

Bibliography	56
---------------------	-----------

Appendices	67
-------------------	-----------

A Appendix - Parameter Free	67
------------------------------------	-----------

A.1	Unconstrained Linear Bandits	67
A.1.1	Switching Regret on the Simplex	69
A.1.2	Optimal Switching Regret without Prior Knowledge	71
A.2	Supporting Lemmas	73

B Appendix - Bandits with Abstention under Expert Advice	75
---	-----------

B.1	CBA analysis	75
B.1.1	Unbounded precision case	77
B.2	Efficient implementation proof	78
B.3	Lower bound proof	80
B.4	Overlapping balls extension	81
B.5	The details of the graph bases	82
B.5.1	p -seminorm balls on graphs	82
B.5.2	Community detection bases	83
B.5.3	Graph convexity bases	83

C Appendix - Sum-Max	85
-----------------------------	-----------

C.1	Analysis Proofs (Proof Of Theorem 5.2)	85
C.1.1	Lemma 5.1	85
C.1.2	Lemma 5.2	86
C.1.3	Lemma 5.3	86
C.1.4	Lemma 5.4	87
C.1.5	Lemma 5.6	88
C.1.6	Lemma 5.7	90
C.1.7	Lemma 5.8	91
C.1.8	Lemma 5.9	92
C.1.9	Theorem 5.2	92
C.2	Proof Of Theorem 5.3	94
C.3	Submodular Monotone Non-Pseudoconcave Functions	96
C.4	Efficient Implementation of COMBAND	97
C.5	Additional Experimental Results	99

D Appendix - Stochastic Shortest Path	101
--	------------

D.1	The benefits of sparsity - upper bounds	101
D.2	Failure of the negative entropy	106
D.3	Efficient implementation of OMD using our regularizer	113
D.4	The benefits of sparsity - lower bound under sparsity	115
D.5	Lower bound under unknown transitions	118

D.6	Lower bound on the maximum of asymmetric zero-mean random walks	120
D.6.1	1st part of the proof:	120
D.6.2	2nd part of the proof:	122

Chapter 1

Introduction

At the heart of this thesis lies a simple yet powerful idea: online learning can be viewed as a sequential process of interaction, where a learner is asked questions, provides answers in the form of actions, and updates its strategy based on feedback. This framework captures a wide range of decision-making problems under uncertainty, from classical online learning to complex, structured environments with dynamic feedback and combinatorial constraints.

In contrast to traditional learning paradigms that rely on a fixed train-test split, online learning unfolds over time, in a continual feedback loop. At each round, the learner receives partial or complete information about the consequences of its actions, and uses this to adapt to the future. In such settings, the central goal is to minimize regret, that is, the difference in cumulative performance between the learner and a suitable benchmark strategy.

This thesis builds upon and extends the online learning literature by exploring how structure and uncertainty appear in this (apparently) simple learning setting. Structure may arise from the action space (e.g., combinatorial or continuous actions), from the input (e.g., contextual features, Markov Decision Process states), or from the objective (e.g., submodular reward functions). Uncertainty can be introduced by partial feedback, nonstationary environments, or unknown dynamics in sequential decision processes.

The overall online learning protocol can be described as:

1. *Question*: the learner receives a question, representing the information available before making a decision. This could be a context, a state, or implicit, e.g., simply the request to choose an action.
2. *Answer*: the learner selects an action, possibly randomized, based on its past interactions and the current question.
3. *Feedback*: the learner observes a feedback signal, which may depend on both the question and the action taken. This signal provides (possibly partial) information about the loss or reward incurred, and is leveraged to update the learner's internal state or strategy.

This general protocol serves as the backbone for the entire thesis. All four of the core contributions developed in this work can be seen as specific instantiations of this learning loop, each characterized by a different kind of question, action space, and feedback model. More precisely:

- In Chapter 3, we address adaptivity in dynamic environments, where the benchmark sequence may shift over time. The question is implicit, but the learner must now compete with time-varying comparators under bandit feedback.
- In Chapter 4, we study learning with abstention in the expert advice setting, where the question is implicit, the action is a choice among confidence-rated predictors, and the feedback is partial due to bandit feedback and abstention.
- In Chapter 5, we explore structured maximization over submodular objectives in a bandit setting, where the question is implicit, the actions are combinatorial, and the learner receives a submodular reward feedback.
- In Chapter 6, we turn to stochastic shortest path problems with sparse cost feedback, where the question is the current Markov Decision Process state, the action affects both cost and transitions, and the feedback reflects structural sparsity in the environment.

Despite their apparent differences, these problems are all governed by a common meta-principle: online learning under structure and uncertainty. By casting them in a unified framework, we expose the common algorithmic challenges and highlight how principled tools like Online Mirror Descent and its variants can be specialized and extended to tackle each setting.

The rest of this thesis is organized to reflect this perspective. We begin with an introduction to the online learning literature and the fundamental algorithmic tools used throughout. Each subsequent chapter builds on this foundation, addressing different types of structure and uncertainty, while staying grounded in the shared learning protocol introduced here.

1.1 Definitions and Notations

We denote by $\mathbb{N}$ the set of natural numbers $\{0, 1, 2, \dots\}$, by $\mathbb{N}^+ := \mathbb{N} \setminus \{0\}$ the set of positive natural numbers, by $\mathbb{R}$ the set of real numbers, and by $\mathbb{R}_{\geq 0}$ the set of non-negative real numbers. If $\mathcal{A}$ is any finite set, the number of elements in $\mathcal{A}$ is denoted by $|\mathcal{A}|$. Fix any positive integer $N \in \mathbb{N}^+$. We denote by $[N] := \{1, \dots, N\}$ the set containing the first N positive integers. We use bold notation to indicate vectors of the kind $\mathbf{x} \in \mathbb{R}^d$, such that $\mathbf{x} = (x(1) \dots x(d))$. The inner product, or dot product, between two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ is defined as $\langle \mathbf{x}, \mathbf{y} \rangle := \sum_{i=1}^d x(i)y(i)$ or, alternatively, as $\mathbf{x}^\top \mathbf{y}$. For any $p \in (0, \infty)$, we denote the p -norm of a vector $\mathbf{x} \in \mathbb{R}^n$ as $\|\mathbf{x}\|_p := (\sum_{i=1}^n |x(i)|^p)^{1/p}$; whenever the subscript is unspecified, $\|\cdot\| = \|\cdot\|_2$ is considered as the Euclidean norm. We let Δ_n be the simplex $\{\mathbf{p} \in [0, 1]^n : \|\mathbf{p}\|_1 = 1\}$. The indicator function $\mathbb{I}[E] \in \{0, 1\}$ for some event E is defined as $\mathbb{I}[E] := 1$ if event E occurs, and $\mathbb{I}[E] := 0$ otherwise.

Throughout this manuscript, we adopt the Bachmann-Landau symbols with the following meaning: for any two functions $f, g: \mathbb{R} \rightarrow \mathbb{R}$ of the same variable $x \in \mathbb{R}$, $f(x) = \mathcal{O}(g(x))$ denotes an upper bound $f(x) \lesssim g(x)$ ignoring positive multiplicative constants, $f(x) = \Omega(g(x))$ is the analogue for the lower bound $f(x) \gtrsim g(x)$, and $f(x) = \Theta(g(x))$ is the analogue for both upper and lower bounds $f(x) \approx g(x)$. We use $\tilde{\mathcal{O}}(\cdot)$, $\tilde{\Omega}(\cdot)$, and $\tilde{\Theta}(\cdot)$ with a tilde to ignore up-to-polylogarithmic factors while preserving the same overall meaning of the notation.

Chapter 2

Introduction to Online Learning

In this chapter we will go through the main concepts in the online learning literature. This introduction has the objective to recall the main online learning algorithms and literature relevant to this thesis, but does not aim to provide a detailed explanation of the tools and concepts at the heart of online learning, as such a job is already satisfied by many excellent works in the literature, such as Cesa-Bianchi and Lugosi (2006), Shalev-Shwartz et al. (2012), Orabona (2019), Lattimore and Szepesvári (2020).

The online learning framework is a continuous interaction between a learning agent and an environment over a finite time horizon $T \in \mathbb{N}^+$. Throughout this thesis, we focus on an **oblivious adversarial environment**. This means that we make no stochastic assumptions about how the environment behaves: the losses for all actions are fixed in advance and do not adapt to the learner's strategy. The environment specifies a sequence of T loss functions $\ell_t : \mathcal{A} \rightarrow \mathbb{R}$, one per round, mapping each action to a real-valued loss.

At each time step, the interaction can be described as follows:

1. *Question*: the learner receives the information for the current round (e.g., a context, a state, or simply an implicit prompt to act).
2. *Answer*: the learner plays an action $a_t \in \mathcal{A}$ from the action set.
3. *Feedback*: the learner suffers the loss $\ell_t(a_t)$ and receives some information about this function.

2.1 Prediction with Expert Advice

One of the most natural and classical settings is that of *prediction with expert advice* (Cesa-Bianchi, Freund, Helmbold, Haussler, Schapire, and Warmuth, 1993, Littlestone and Warmuth, 1994, Freund and Schapire, 1997b), where the learner observes the predictions of $K \geq 2$ fixed experts at each round t and is asked to select one expert, possibly at random. For simplicity, let us denote the experts as the first K positive integers, so that the action set consists of the finite set $[K]$. An equivalent formulation allows the learner to play a probability distribution $\mathbf{p}_t \in \Delta_K$ over the experts and then sample an action $a_t \sim \mathbf{p}_t$ (see Algorithm 2.1).

In this setting, the environment arbitrarily decides a loss vector $\ell_t \in [0, 1]^K$ at each round, where the loss of action $a \in [K]$ is given by the a -th component of the vector. As Feedback, the learner receives the entire vector ℓ_t , making this a **full-information** setting.

Algorithm 2.1: Expert Learning Protocol

Environment decides loss sequence $\ell_t \in [0, 1]^K$ for $t \in [T]$
for $t = 1, \dots, T$ **do**
 Learner selects $\mathbf{p}_t \in \Delta_K$ and plays $a_t \sim \mathbf{p}_t$.
 Learner suffers loss $\ell_t(a_t)$ and observes the loss vector ℓ_t .

A key performance metric in online learning is the **expected regret**, which measures how much the learner regrets not having played the best fixed action in hindsight $a^* = \arg \min_{a \in [K]} \sum_{t=1}^T \ell_t(a)$:

$$R_T(a^*) = \mathbb{E} \left[\sum_{t=1}^T \ell_t(a_t) \right] - \sum_{t=1}^T \ell_t(a^*),$$

where the expectation is taken with respect to the learner's randomization. If we define $\mathbf{p}^*$ as the comparator distribution centered on a^* (i.e., sampling from $\mathbf{p}^*$ always returns a^*), we can equivalently express the regret as

$$R_T(\mathbf{p}^*) = \mathbb{E} \left[\sum_{t=1}^T \langle \mathbf{p}_t - \mathbf{p}^*, \ell_t \rangle \right].$$

2.1.1 Introduction to the main algorithms

The goal is to design algorithms that ensure that the regret grows *sublinearly* with the time horizon T , and ideally also reflects additional problem-specific quantities when possible. Throughout this thesis, we provide both upper and lower bounds on the expected regret to characterize the achievable performance as tightly as possible.

For the expert setting, the most natural strategy is to always play the action that minimizes the cumulative loss observed so far. This strategy, known in the literature as *Follow the Leader* (FTL), corresponds to selecting the probability distribution

$$\mathbf{p}_{t+1} = \arg \min_{\mathbf{p} \in \Delta_K} \langle \mathbf{p}, \sum_{s=1}^t \ell_s \rangle.$$

While this approach can achieve sublinear regret under certain conditions for the loss function, it is known to suffer linear regret in adversarial settings. In fact, even in simple two-action examples, the regret can grow as $\Theta(T)$ (see, e.g., Cesa-Bianchi and Lugosi (2006, Section 4.3)).

The underlying main reason is that the distribution selected is a deterministic one, concentrated on the current empirical best action. Therefore, an adversarial environment can construct a loss sequence that exploits this determinism, forcing linear regret. To address this issue, we introduce regularization into the update rule, introducing *randomness* in the learner's strategy.

This naturally leads to the *Follow The Regularized Leader* (FTRL) algorithm:

$$\begin{aligned} \mathbf{p}_1 &= \arg \min_{\mathbf{p} \in \Delta_K} \phi(\mathbf{p}) \\ \mathbf{p}_{t+1} &= \arg \min_{\mathbf{p} \in \Delta_K} \langle \mathbf{p}, \sum_{s=1}^t \eta \ell_s \rangle + \phi(\mathbf{p}), \end{aligned}$$

where $\phi(\mathbf{p})$ is a convex regularizer that introduces stability and randomness into the updates. When the regularizer is the Shannon entropy, $\phi(\mathbf{p}) = \sum_{i \in [K]} p(i) \ln(p(i))$, we recover the HEDGE (or EXPONENTIAL WEIGHTS) algorithm. The probabilities are then computed as

$$p_{t+1}(a) = \frac{\exp\left(\sum_{s=1}^t -\eta \ell_s(a)\right)}{\sum_{a' \in [K]} \exp\left(\sum_{s=1}^t -\eta \ell_s(a')\right)}.$$

Some manipulation (e.g., Shalev-Shwartz et al. (2012)) shows that for any ϕ :

$$R_T(\mathbf{p}^*) \leq \frac{\phi(\mathbf{p}^*) - \phi(\mathbf{p}_1)}{\eta} + \sum_{t=1}^T \langle \mathbf{p}_t - \mathbf{p}_{t+1}, \boldsymbol{\ell}_t \rangle.$$

This decomposition is often referred to as the *penalty* and *stability* decomposition. This is because the first term describes, with respect to ϕ , how far the initial point and the comparator actually are, while the second term measures the difference between the consecutive updates in the direction of the loss function of the previous update. When the regularizer is the Shannon entropy, which is 1-strongly convex w.r.t. the $\|\cdot\|_1$ over the action set Δ_K , we can guarantee that:

$$\langle \mathbf{p}_t - \mathbf{p}_{t+1}, \boldsymbol{\ell}_t \rangle \leq \eta \|\boldsymbol{\ell}_t\|_\infty^2 \leq \eta.$$

Given that $\phi(\mathbf{p}^*) - \phi(\mathbf{p}_1) = \ln K$, setting $\eta = \sqrt{\frac{\ln K}{T}}$, gives the regret upper bound $\mathcal{O}\left(\sqrt{T \ln K}\right)$, which is known to be minimax optimal.

Another equivalent formulation of this algorithm is given by the well-known *Online Mirror Descent* (OMD) algorithm over the probability simplex. For a given regularizer $\psi(\mathbf{p})$, the algorithm starts with

$$\mathbf{p}_1 = \arg \min_{\mathbf{p} \in \Delta_K} \psi(\mathbf{p})$$

and updates at each round as

$$\mathbf{p}_{t+1} = \arg \min_{\mathbf{p} \in \Delta_K} \eta \langle \mathbf{p}, \boldsymbol{\ell}_t \rangle + D_\psi(\mathbf{p}, \mathbf{p}_t),$$

where $D_\psi(\mathbf{x}, \mathbf{y}) = \psi(\mathbf{x}) - \psi(\mathbf{y}) - \langle \nabla \psi(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle$ is the Bregman divergence induced by the regularizer ψ .

Intuitively, the Bregman divergence in OMD plays the role of a regularizer in FTRL. Just as FTRL adds a convex regularization term to the cumulative loss to stabilize the update, OMD uses the Bregman divergence to penalize moving too far from the previous iterate. This prevents the learner from reacting too aggressively to the current loss vector. Therefore, OMD is a powerful framework where the choice of the regularizer defines the geometry of the parameter space, adjusting the iterates accordingly*.

In particular, when the regularizer is induced the Shannon entropy $\psi(\mathbf{p}) = \sum_{i \in [K]} p(i) \ln p(i)$, the OMD update reproduces the FTRL one (see, e.g., Orabona (2019, Section 6.6)).

Despite their similarity, the equivalence between FTRL and OMD is not universal. In particular, it does not hold in general for time-varying learning rates or non-Legendre regularizers. However, in this setting, the two algorithms coincide with HEDGE and therefore achieve the same optimal regret

*For instance, setting the regularizer to $\psi(\mathbf{p}) = \frac{1}{2} \|\mathbf{p}\|_2^2$ recovers the classic Gradient Descent update (Orabona, 2019).

guarantees.

Standard results (e.g., Bubeck, Cohen, and Li (2018a, Theorem 6)) give also a similar analysis to the one we've seen for the FTRL algorithm, obtaining:

$$R_T(\mathbf{p}^*) \leq \frac{\psi(\mathbf{p}) - \psi(\mathbf{p}_1)}{\eta} + \frac{\eta}{2c} \sum_{t=1}^T \|\ell_t\|_{\nabla^2 \psi(\mathbf{p}_t)}^2.$$

where for a $A \in \mathbb{R}^{K \times K}$, $\|\mathbf{w}\|_A = \mathbf{w} A \mathbf{w}^\top$ and the c constant is given by the condition of the regularizer:

$$\nabla \psi(\mathbf{w}_t) \in [\nabla \psi(\mathbf{p}_t), \nabla \psi(\mathbf{p}_t) - \eta \ell_t] \implies \nabla^2 \psi(\mathbf{w}_t) \succeq c \nabla^2 \psi(\mathbf{p}_t),$$

as a *relative* strong convexity condition of the regularizer. Since we are playing with non-negative bounded losses and the Shannon Entropy we have that $c = 1$ and $\nabla^2 \psi(\mathbf{p})^{-1} = \text{DIAG}(\mathbf{p})$, therefore:

$$\begin{aligned} R_T(\mathbf{p}^*) &\leq \frac{\ln(K)}{\eta} + \frac{\eta}{2} \mathbb{E} \left[\sum_{t=1}^T \sum_{i \in [K]} \ell_t(i)^2 p_t(i) \right] \\ &\leq \frac{\ln(K)}{\eta} + \frac{\eta}{2} T \end{aligned}$$

Setting $\eta = \sqrt{\frac{\ln K}{T}}$, retrieving $\mathcal{O}(\sqrt{T \ln K})$ as expected.

2.2 Limited feedback: the bandit setting

The multi-armed bandit problem (Auer, Cesa-Bianchi, Freund, and Schapire, 2002) is essentially the same as the expert advice framework. The only difference is in the *Feedback* the learner receives. Instead of observing the full loss vector ℓ_t , the learner only observes the loss of the selected action $\ell_t(a_t)$ (i.e. the a_t -th component of the loss vector).

Algorithm 2.2: Bandit Learning Protocol

for $t = 1, \dots, T$ **do**
 Learner picks an action $a_t \in [K]$
 Learner suffers and observes loss $\ell_t(a_t)$.

Despite the limited feedback, it is still possible to design algorithms that achieve sublinear regret over the time horizon. The key idea is that, although the learner no longer has access to the full loss vector required to run any of the previously discussed algorithms, it can instead construct an *unbiased estimator* of it and then proceed as if the full information were available.

The most common unbiased estimator in this setting is the *importance-weighted estimator*, defined as

$$\widehat{\ell}_t(a) = \frac{\ell_t(a)}{p_t(a)} \mathbb{I}\{a_t = a\} \quad \forall a \in [K]. \quad (2.1)$$

The algorithm obtained by running Hedge over the sequence of importance-weighted estimates $\widehat{\ell}_1, \dots, \widehat{\ell}_T$, instead of the true losses, is known as Exp3 (Exponential-weight algorithm for Exploration and Exploitation) first introduced in Auer et al. (2002).

If we let $\mathbb{E}_t[\cdot] = \mathbb{E}[\cdot \mid a_1, \dots, a_{t-1}]$ denote the conditional expectation given the past actions, then

$\widehat{\ell}_t(a)$ is an unbiased estimator of the true loss, since

$$\mathbb{E}_t[\widehat{\ell}_t(a)] = \ell_t(a).$$

The key challenge this estimator introduces is its *variance*:

$$\mathbb{E}_t[\widehat{\ell}_t(a)^2] = \mathbb{E} \left[\frac{\ell_t(a)^2}{p_t(a)^2} \mathbb{I}\{a_t = a\} \right] = \frac{\ell_t(a)^2}{p_t(a)}.$$

This variance scales inversely with the probability of playing an action, which can be difficult to manage. The good news is that if we run OMD with the Shannon Entropy, this additional variance is naturally handled by the stability term of the analysis, as:

$$\begin{aligned} R_T(\mathbf{p}^*) &\leq \frac{\psi(\mathbf{p}) - \psi(\mathbf{p}_1)}{\eta} + \frac{\eta}{2c} \mathbb{E} \left[\sum_{t=1}^T \|\widehat{\ell}_t\|_{\nabla^2 \psi(\mathbf{p}_t)}^2 \right] \\ &\leq \frac{\ln K}{\eta} + \frac{\eta}{2} \mathbb{E} \left[\sum_{t=1}^T \sum_{i \in [K]} \widehat{\ell}_t(i)^2 p_t(i) \right] \\ &\leq \frac{\ln K}{\eta} + \frac{\eta}{2} KT. \end{aligned}$$

Setting $\eta = \sqrt{\frac{\ln K}{KT}}$, yields to a regret bound of order $\mathcal{O}(\sqrt{KT \ln K})$, which is only a factor of $\sqrt{K}$ worse than the minimax regret in the full-information setting and optimal up to logarithmic factors.

2.2.1 Bandits with Expert Advice

So far, we have considered the basic bandit setting where the learner is asked to select an action without any additional information. A natural extension is to enrich the *Question* part of the learning protocol by providing the learner with extra side information before choosing an action. This is often referred to as the *contextual bandit setting*.

One way to formalize this is through the *bandits with expert advice* model. Here, instead of directly receiving the context, the learner is given advice from a finite set of E experts. At each round, each expert $e \in [E]$, recommends a probability distribution $\mathbf{p}_{t,e} \in \Delta_K$ over the K actions. The learner must then aggregate these recommendations to select its own distribution. This formulation can be seen as a contextual setting where the context is encoded in the experts' advice (see Lattimore and Szepesvári (2020)). This connection between the two settings is more powerful than it may initially appear, as the notion of an “expert” is highly abstract. In this framework, an expert can be seen as any fixed policy that maps contexts to actions, from a simple heuristic to a complex pre-trained model.

This abstraction provides a powerful algorithmic tool: one can reduce a general contextual bandit problem to “bandits with expert advice problem” by first defining a set of competing policies. At each round, the advice from this set forms the input for a master algorithm, which then learns how to best

aggregate their recommendations.

The standard algorithm for this setting is EXP4, which extends EXP3 by running it over the set of experts rather than the actions. The resulting regret guarantee with respect to the best expert in hindsight is:

$$R_T = \mathcal{O}(\sqrt{TK \ln E}).$$

This bound shows that the performance penalty for not knowing the best expert in advance grows only logarithmically with the number of experts E , a key feature that makes the approach scalable. Additionally, this logarithmic dependence on the number of experts is known to be necessary by Seldin and Lugosi (2016), showing that at least a factor of $\sqrt{\ln E / \ln K}$ in the regret lower bound.

Chapter 3

Dynamic Regret and Parameter Free Guarantees

In chapter we start introducing a simple linear structure to the action set while looking to a slightly different notion of regret, as the environment is allowed to select a *sequence* of comparators instead of a fixed one. We consider a bandit optimization setting characterized by an *unbounded* action space $\mathbb{R}^d$ and an oblivious adversary privately selecting a sequence of convex loss functions $\ell_t : \mathcal{W} \rightarrow \mathbb{R}$. At each time step $t \in [T]$:

1. *Question*: the learner is asked to choose an action in $\mathbb{R}^d$.
2. *Answer*: The learner plays an action $\mathbf{w}_t \in \mathcal{W}$.
3. *Feedback*: the learner suffers and observes the loss $\langle \ell_t, \mathbf{w}_t \rangle$.

A common special case we study is the linear bandit optimization setting in which the learner can choose actions from $\mathbb{R}^d$. The environment is controlled by an oblivious adversary, preliminary selecting a sequence of linear loss functions $\ell_1, \dots, \ell_T$, where $\ell_t = \langle \ell_t, \cdot \rangle$ and $\ell_t \in \mathbb{R}^d$ for each t . At each round $t \in [T]$, the learner chooses an action $\mathbf{w}_t \in \mathcal{W}$. The learner then receives only partial (bandit) feedback: instead of observing ℓ_t , it only observes the scalar $\ell_t(\mathbf{w}_t)$ corresponding to the loss of the chosen action. The learner's performance is evaluated according to the expected *dynamic* regret

$$\mathbb{E} [R_T(\mathbf{u}_1, \dots, \mathbf{u}_T)] = \mathbb{E} \left[\sum_{t \in [T]} \ell_t(\mathbf{w}_t) \right] - \sum_{t=1}^T \ell_t(\mathbf{u}_t),$$

where $\mathbf{w}_1, \dots, \mathbf{w}_T$ is the sequence of the learner's actions, $\mathbf{u}_1, \dots, \mathbf{u}_T$ is an arbitrary sequence of comparators, and the expectation is taken with respect to the internal randomness of the algorithm. The standard (static) regret is recovered when $\mathbf{u}_1 = \dots = \mathbf{u}_T$.

Notice that the “complexity” of the comparator sequence contributes to characterizing the difficulty of a given problem. If the comparator is allowed to change arbitrarily on each round, there is no hope to achieve sublinear dynamic regret since the comparator sequence can ensure $\sum_{t=1}^T \ell_t(\mathbf{u}_t) = 0$ even against completely unpredictable losses. On the other hand, we know that in the special case $\mathbf{u}_1 = \dots = \mathbf{u}_T$ we can ensure sublinear regret, since this is just the usual linear bandits setting. Thus, we are typically interested in algorithms which make dynamic regret guarantees that gracefully *adapt* to some measure of complexity or *variability* of the comparator sequence.

The classic way to quantify the variability of the comparator sequence is via its *path-length*

$$P_T = \sum_{t=2}^T \|\mathbf{u}_{t-1} - \mathbf{u}_t\|,$$

which measure the distance of consecutive comparators under a chosen norm.

A related classical benchmark is the *switching regret*, which bounds regret with respect to the number of times the comparator sequence changes over time:

$$S_T = \sum_{t=2}^T \mathbb{I}[\mathbf{u}_t \neq \mathbf{u}_{t-1}],$$

capturing a coarser notion of variability. We will consider a generalization of this notion in which the comparator at time t is sampled from a distribution $\mathbf{p}_t^*$ over $\mathcal{W}$, such that its expectation satisfies $\mathbb{E}_{\mathbf{w} \sim \mathbf{p}_t^*}[\mathbf{w}] = \mathbf{u}_t$. This generalizes to situations where the learner is compared against randomized strategies or smooth shifts in behavior. In this case, we define the *distributional path-length* as:

$$P_T^\Delta = \sum_{t=2}^T \|\mathbf{p}_t^* - \mathbf{p}_{t-1}^*\|_1.$$

While dynamic or switching regret provides a more realistic benchmark for non-stationary environments, it introduces significant challenges. Notably, achieving regret bounds that scale optimally with the comparator variability typically requires preliminary knowledge of this quantity or a known upper bound on it (Agarwal, Luo, Neyshabur, and Schapire, 2017, Marinov and Zimmert, 2021, Luo, Zhang, Zhao, and Zhou, 2022). This assumption is often unrealistic, as the comparator sequence is not known in advance.

Parameter-free methods, which in dynamic regret settings eliminate the need for knowledge of the path length, are relatively well understood in the full-information setting. (Zhang, Lu, and Zhou, 2018, Cutkosky, 2020, Campolongo and Orabona, 2021, Jacobsen and Cutkosky, 2022, 2023, Zhang, Cutkosky, and Paschalidis, 2023, Jacobsen and Cutkosky, 2024). In this work, we focus instead on the bandit setting, where the learner only receives point-evaluation feedback at the end of each round. With bandit feedback, the parameter-free techniques developed for the full information setting are either not directly applicable or result in a suboptimal dependence on the time horizon and path length.

3.0.1 Our contributions

We obtain the optimal dynamic regret guarantees of order $\tilde{\mathcal{O}}(\sqrt{(1 + P_T^\Delta)T})$ against an oblivious adversary without prior knowledge of the distributional path length P_T^Δ (which includes the number of switches S_T as a special case), thus resolving a long-standing open problem (Marinov and Zimmert, 2021, Luo et al., 2022). Key to obtaining this result is a technique for combining the guarantees of *comparator-adaptive* base algorithms inspired by a clever result in the full information setting (Cutkosky, 2019), which we adapt to bandit feedback via a sampling trick. This simple and elegant trick enables us to easily combine the outputs of several bandit algorithms to achieve the best of their respective dynamic regret guarantees, effectively enabling us to “tune” hyperparameters

on-the-fly. Such hyperparameter tuning arguments have been attempted by several prior works using sophisticated mixture-of-experts style arguments (Agarwal et al., 2017, Marinov and Zimmert, 2021, Luo et al., 2022), but have only achieved the optimal $\sqrt{P_T}$ dependence by leveraging *a priori* knowledge of P_T . We expect that our approach might also find applications in other settings where bandit-over-bandit ensemble strategies have failed in the past.

3.0.2 Related works

The study of dynamic regret was initiated by Herbster and Warmuth (1998, 2001). In the setting of online convex optimization (OCO) with Lipschitz losses and full information, Zinkevich (2003) showed that online gradient descent achieves a bound of order $\mathcal{O}((1 + P_T)\sqrt{T})$. Yang, Zhang, Jin, and Yi (2016) improved the rate to $\mathcal{O}(\sqrt{(1 + P_T)T})$ by leveraging prior knowledge of P_T . This bound was later shown to be minimax optimal by Zhang et al. (2018), who also provided the first algorithm achieving a matching upper bound without prior knowledge of P_T .

Since then, several works have achieved generalizations of the $\mathcal{O}(\sqrt{(1 + P_T)T})$ guarantee by incorporating more adaptive, data-dependent quantities. These refinements typically replace the dependence on T dependence with terms like the cumulative sum of the squared gradient norms $\sum_t \|\nabla \ell_t(\mathbf{w}_t)\|^2$, or the temporal variation of the loss $\sum_t \sup_{\mathbf{x}} |\ell_t(\mathbf{x}) - \ell_{t-1}(\mathbf{x})|$. Many of these methods also adapts to the maximum norm of the comparator sequence $\max_t \|\mathbf{u}_t\|$ (Cutkosky, 2020, Campolongo and Orabona, 2021, Jacobsen and Cutkosky, 2023, Zhang et al., 2023). Various improvements in adaptivity can also be obtained under additional assumptions on the losses such as smoothness Mokhtari, Shahrampour, Jadbabaie, and Ribeiro (2016), Zhao, Zhang, Zhang, and Zhou (2020, 2024).

Key to our results is the notion of *comparator adaptive* online learning, where the goal is to design algorithms that adapt to the complexity of the comparator sequence (e.g., its norm or a measure of its variability) without requiring prior knowledge about it. In the static comparator setting, this idea has been extensively studied under full-information feedback, where the optimal guarantee $R_T(\mathbf{u}) = \mathcal{O}(\|\mathbf{u}\|\sqrt{T})$, for any $\mathbf{u} \in \mathbb{R}^d$, can be obtained up to logarithmic terms (McMahan and Streeter, 2012, McMahan and Orabona, 2014, Orabona and Pál, 2016, Cutkosky and Orabona, 2018). Notably, Cutkosky (2019) showed that algorithms making comparator-adaptive guarantees can be easily combined, obtaining regret proportional to the best among them; this observation will be crucial to our approach in Section 3.1. In both linear and convex settings with bandit feedback, comparator-adaptive bounds were studied by van der Hoeven, Cutkosky, and Luo (2020) where they consider static regret and propose a black-box reduction approach, taking inspiration from the full-information reduction of Cutkosky and Orabona (2018).

In the bandit setting, the study of the closely-related notion of switching regret was initiated by Auer (2002), where a bound of $\mathcal{O}(\sqrt{(1 + S_T)T})$ was obtained with prior knowledge of the number of switches S_T . The optimal regret bound for the non-stationary *stochastic* bandit setting without *a priori* variational knowledge was first obtained in Auer, Gajane, and Ortner (2018). Interestingly, Marinov and Zimmert (2021) showed the impossibility of obtaining a $\mathcal{O}(\sqrt{(1 + S_T)T})$ bound in *adaptive* adversary settings, leaving the question open (until now) for the oblivious setting.

In the bandit convex optimization (BCO) literature, Zhao, Wang, Zhang, and Zhou (2021) showed that the classical Bandit Gradient Descent algorithm of Flaxman, Kalai, and McMahan (2004) can achieve dynamic regret bound of order $\mathcal{O}(T^{3/4}(1 + P_T^{1/4}))$ when the step-size is optimally tuned using

Algorithm 3.1: Uniform Sampling Interface

Input: Domain $\mathcal{W} \subseteq \mathbb{R}^d$, base algorithms $(\mathcal{A}_n)_{n=1}^N$
for $t = 1, \dots, T$ **do**
 Get $\mathbf{w}_t^{(i)}$ from $\mathcal{A}_i$ for all $i \in [N]$
 Sample $i_t \sim \text{Uniform}(N)$ and play $\mathbf{w}_t = \mathbf{w}_t^{(i_t)}$
 Receive feedback $\phi(\mathbf{w}_t, \ell_t)$
 Send $\phi(\mathbf{w}_t, \ell_t) \mathbb{I}\{i_t = i\}$ to $\mathcal{A}_i$ for $i \in [N]$

knowledge of P_T . They also proposed a parameter-free version using a bandit-over-bandit ensemble strategy, but this leads to a suboptimal regret of $\mathcal{O}(T^{3/4}(1 + P_T^{1/2}))$ due to additional variance from the meta-learning layer. Later, Yan, Zhao, and Zhou (2023) improved the dependence on P_T but picked up an additive $T^{5/6}$ penalty, obtaining $\mathcal{O}(P_T^{1/4}T^{3/4} + T^{5/6})$ overall.

More broadly, ensemble and meta-algorithm strategies have been proposed to adapt to unknown environment parameters in bandits. One notable example is the CORRAL framework introduced by Agarwal et al. (2017), Luo et al. (2022), which maintains a pool of base bandit algorithms to hedge against misspecification. However, to obtain their result, they still need to assume a fixed and known number of switches (path length in our setting). Naive bandit-over-bandit schemes incur significant overhead: for instance, running EXP4 (Auer et al., 2002) as a master over EXP3 bases would yield $\mathcal{O}(T^{2/3})$ regret due to the extra exploration needed (Odalric and Munos, 2011, Cheung, Simchi-Levi, and Zhu, 2019). While these methods are flexible, they often pay a price in terms of worse regret bounds or higher variance, particularly in dynamic or tuning-free scenarios.

Additional Notation. Given a norm $\|\cdot\|$ and $\rho \geq 0$, we denote by $\mathcal{B}_\rho := \{x : \|x\| \leq \rho\}$ the closed ball of radius ρ , or the unit ball when the radius is not specified. We denote with $\mathcal{M}_1$ the set of distributions over $\mathbb{R}^d$. Finally, $\mathcal{O}(\cdot)$ hides constant factors and $\tilde{\mathcal{O}}(\cdot)$ hides constant and logarithmic factors.

3.1 Combining guarantees with uniform sampling

Our approach is inspired by a framework for combining guarantees of *comparator-adaptive* online learning algorithms proposed by Cutkosky (2019) for the simpler Online Convex Optimization (OCO) setting. The framework applies to a class of algorithms known as *comparator-adaptive*, which are characterized by the fundamental feature of having constant regret against the zero comparator, i.e. $R_T(\mathbf{0}) = \mathcal{O}(1)$ (McMahan and Abernethy, 2013, McMahan and Orabona, 2014, Orabona and Pál, 2016, Cutkosky and Orabona, 2018).

The key insight from Cutkosky (2019) is that if we have N such OCO algorithms $\mathcal{A}_1, \dots, \mathcal{A}_N$, we can achieve a regret competitive with the best guarantee among them, $R_T(\mathbf{u}) = \mathcal{O}(\min_i R_T^{\mathcal{A}_i}(\mathbf{u}))$, by simply adding the iterates together. To see this in the case of linear losses $\ell_t = \langle \ell_t, \cdot \rangle$, let $\mathbf{w}_t^{(i)}$ denote the output of $\mathcal{A}_i$ on round t , if we play $\mathbf{w}_t = \sum_{i \in [N]} \mathbf{w}_t^{(i)}$, then for any $j \in [N]$ we have

$$R_T(\mathbf{u}) = \sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u} \rangle$$

$$\begin{aligned}
 &= \sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{w}_t^{(j)} - \mathbf{u} \rangle + \sum_{i \neq j} \sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{w}_t^{(i)} \rangle \\
 &= R_T^{\mathcal{A}_j}(\mathbf{u}) + \sum_{i \neq j} R_T^{\mathcal{A}_i}(\mathbf{0}) \\
 &= \mathcal{O}\left(R_T^{\mathcal{A}_j}(\mathbf{u}) + N\right),
 \end{aligned}$$

where the last step holds because each algorithm guarantees $R_T^{\mathcal{A}_i}(\mathbf{0}) = \mathcal{O}(1)$. Moreover, since this holds for any $j \in [N]$, it must hold for the best among them.

While the iterate-adding approach is elegant in the full-information OCO setting, it is not directly applicable under bandit feedback. The fundamental issue is that when playing the sum $\mathbf{w}_t = \sum_{i \in [N]} \mathbf{w}_t^{(i)}$, the feedback observed is the total feedback $\langle \mathbf{g}_t, \sum_i \mathbf{w}_t^{(i)} \rangle$. As a result, there is no way to precisely assign feedback to any individual learner, as the observed feedback includes contributions from all other learners' decisions. Instead, we make the following observation: if on each round we sample one of the algorithms uniformly at random and play *only its action* on round t , then *in expectation*, this is equivalent to playing $\sum_i \mathbf{w}_t^{(i)} / N$. As a result, we can still apply *nearly* the same iterate-adding argument outlined above, but we can now accurately assign feedback since only one learner's action is played on each round. The main difference from the iterate-adding approach is that we need to rescale the comparator to account for the $1/N$ factor that shows up in $\sum_i \mathbf{w}_t^{(i)} / N$.

The procedure described above is summarized in Algorithm 3.1 for a generic feedback oracle ϕ , which returns a feedback signal $\phi(\mathbf{w}_t, \ell_t)$ given the decision $\mathbf{w}_t$. This captures first-order feedback when $\phi(\mathbf{w}_t, \ell_t) = \mathbf{g}_t$ and bandit feedback when $\phi(\mathbf{w}_t, \ell_t) = \ell_t(\mathbf{w}_t)$. To build intuition, we begin with a warm-up result that analyzes the performance of this strategy in the first-order setting.

Proposition 3.1. *Let $\mathcal{A}_1, \dots, \mathcal{A}_N$ be online learning algorithms and let $\mathbf{w}_t^{(i)}$ denote the output of $\mathcal{A}_i$ on round t . Suppose that for all i , $\mathcal{A}_i$ guarantees $R_T^{\mathcal{A}_i}(\mathbf{0}) = \sum_{t=1}^T \ell_t(\mathbf{w}_t^{(i)}) - \ell_t(\mathbf{0}) \leq G\epsilon$ for any sequence of G -Lipschitz loss functions. Then for any sequence $\mathbf{u}_1, \dots, \mathbf{u}_T$ in $\mathcal{W}$, Algorithm 3.1 guarantees $\mathbb{E}[R_T(\mathbf{u}_1, \dots, \mathbf{u}_T)] \leq \min_{n=1, \dots, N} \mathbb{E}[R_T^{\mathcal{A}_n}(N\mathbf{u}_1, \dots, N\mathbf{u}_T)] + (N-1)G\epsilon$.*

Proof. For all $t \geq 0$, let $\mathcal{F}_t$ be the σ -algebra generated by $I_1, \dots, I_t$ and by any internal randomization used by each base algorithm $\mathcal{A}_i$ to generate $\mathbf{w}_s^{(i)}$ for $s = 1, \dots, t$. Let $\hat{\mathbf{g}}_t^{(n)} = \mathbb{I}\{I_t = n\} \mathbf{g}_t$, and note that $\mathbf{w}_t = \sum_i \mathbb{I}\{I_t = i\} \mathbf{w}_t^{(i)}$. Observe that for any $n \in [N]$ we have

$$\begin{aligned}
 \mathbb{E}[R_T(\mathbf{u}_1, \dots, \mathbf{u}_T)] &= \mathbb{E}\left[\sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{u}_t \rangle\right] \\
 &= \mathbb{E}\left[\sum_{t=1}^T \langle \mathbf{g}_t, \sum_{i=1}^N \mathbb{I}\{I_t = i\} \mathbf{w}_t^{(i)} \rangle\right] \\
 &\quad - \mathbb{E}\left[\sum_{t=1}^T \langle \mathbf{g}_t, \mathbf{u}_t \rangle \underbrace{\mathbb{E}[N \mathbb{I}\{I_t = n\} \mid \mathcal{F}_{t-1}]}_{=1}\right] \\
 &= \mathbb{E}\left[\sum_{i=1}^N \sum_{t=1}^T \langle \mathbb{I}\{I_t = i\} \mathbf{g}_t, \mathbf{w}_t^{(i)} \rangle\right] \\
 &\quad - \mathbb{E}\left[\sum_{t=1}^T \mathbb{E}[\langle \mathbb{I}\{I_t = n\} \mathbf{g}_t, N\mathbf{u}_t \rangle \mid \mathcal{F}_{t-1}]\right]
 \end{aligned} \tag{3.1}$$

$$\begin{aligned}
 &= \mathbb{E} \left[\sum_{i=1}^N \sum_{t=1}^T \langle \hat{\mathbf{g}}_t^{(i)}, \mathbf{w}_t^{(i)} \rangle \right] - \mathbb{E} \left[\sum_{t=1}^T \langle \hat{\mathbf{g}}_t^{(n)}, N\mathbf{u}_t \rangle \right] \\
 &= \mathbb{E} \left[\sum_{t=1}^T \langle \hat{\mathbf{g}}_t^{(n)}, \mathbf{w}_t^{(n)} - N\mathbf{u}_t \rangle \right. \\
 &\quad \left. + \sum_{i \neq n} \sum_{t=1}^T \langle \hat{\mathbf{g}}_t^{(i)}, \mathbf{w}_t^{(i)} \rangle \right] \\
 &\leq \mathbb{E} \left[R_T^{A_n}(N\mathbf{u}_1, \dots, N\mathbf{u}_T) + \sum_{i \neq n} R_T^{A_i}(\mathbf{0}) \right] \\
 &\leq \mathbb{E} \left[R_T^{A_n}(N\mathbf{u}_1, \dots, N\mathbf{u}_T) \right] + (N-1)G\epsilon
 \end{aligned}$$

concluding the proof. $\square$

This theorem illustrates how uniform sampling acts as a rudimentary coordination mechanism for comparator-adaptive algorithms. Because each learner individually adapts to an arbitrary comparator, our combined strategy, on average, can be compared against the performance of any single algorithm in the collection. As we will see in the following sections, this mechanism enables us to “tune” hyperparameters on-the-fly, allowing us to adapt to unknown problem parameters such as the distributional path-length P_T^Δ without any prior knowledge.

It is important to note that, unlike the iterate-adding approach of Cutkosky (2019), this method for combining guarantees ends up increasing the comparator norm by a factor of N . Therefore, in the context of OCO, the uniform sampling approach is a generally a worse option than the iterate-adding approach. Indeed, this trick is primarily of interest in settings where the iterate-adding approach cannot be applied.

3.2 Linear Bandits

Recall from the previous section that the key property that we need to apply the uniform sampling strategy (Algorithm 3.1) is that the base algorithms $\mathcal{A}_i$ guarantee a comparator-adaptive property, $R_T(\mathbf{0}) = \mathcal{O}(1)$. To obtain guarantees of this form in the bandit setting, we use a reduction introduced by van der Hoeven et al. (2020) which only works for bandits with linear losses $\ell_t = \langle \ell_t, \cdot \rangle$. The idea is to decompose the action $\mathbf{w}_t$ into a *scale* $v_t \in \mathbb{R}$ and a *direction* $\beta_t \in \mathcal{B}$, which will be learned by separate online learning algorithms. In particular, observe that for $M = \max_t \|\mathbf{u}_t\|$ we have

$$\begin{aligned}
 R_T(\mathbf{u}_1 \dots \mathbf{u}_T) &= \sum_{t=1}^T \langle \ell_t, \mathbf{w}_t - \mathbf{u}_t \rangle \\
 &= \sum_{t=1}^T \langle \ell_t, \beta_t \rangle v_t - \langle \ell_t, \mathbf{u}_t \rangle \\
 &= \underbrace{\sum_{t=1}^T \langle \ell_t, \beta_t \rangle (v_t - M)}_{=: R_T^{A_V}(M)} + M \underbrace{\sum_{t=1}^T \langle \ell_t, \beta_t - \mathbf{u}_t/M \rangle}_{=: R_T^{A_B}(\mathbf{u}_1/M, \dots, \mathbf{u}_T/M)}.
 \end{aligned}$$

Algorithm 3.2: Scale and Direction Decomposition (van der Hoeven et al., 2020)

Input: Scale learner $\mathcal{A}_V$, direction learner $\mathcal{A}_B$
for $t = 1, \dots, T$ **do**
 Get scale prediction v_t from $\mathcal{A}_V$ and direction prediction β_t from $\mathcal{A}_B$
 Play $\mathbf{w}_t = v_t \beta_t$
 Observe $\langle \mathbf{w}_t, \ell_t \rangle$ and compute $g_t = \langle \mathbf{w}_t, \ell_t \rangle / v_t$
 Send $\langle \ell_t, \beta_t \rangle = \langle \ell_t, \mathbf{w}_t \rangle / v_t$ to $\mathcal{A}_B$ as the feedback on round t
 Send $v \mapsto \langle \ell_t, \beta_t \rangle v$ to $\mathcal{A}_V$ as the t^{th} loss

In order to ensure $R_T(\mathbf{0}) = \mathcal{O}(1)$ it suffices to provide a scale learner algorithm $\mathcal{A}_V$ which guarantee a $R_T^{\mathcal{A}_V}(\mathbf{0}) = \mathcal{O}(1)$ against the losses $v \mapsto \langle \ell_t, \beta_t \rangle v$. And this is a 1-dimensional *static regret* Online Linear Optimization problem with full information feedback, so we can apply any of the existing comparator-adaptive algorithms to ensure this property (Orabona and Pál, 2016, Cutkosky and Orabona, 2018, Mhammedi and Koolen, 2020, Jacobsen and Cutkosky, 2022).

We can now show in the following proposition that this scale and direction decomposition extends to a collection of algorithms combined via the uniform sampling method we introduced before.

Proposition 3.2. *Pick N base algorithms as described in Algorithm 3.2, where the scale learner $\mathcal{A}_V$ is over $\mathbb{R}$ and the direction learner is over $\mathcal{B}_N$. Suppose that for any sequence of G -Lipschitz linear losses $g_1, \dots, g_T$ in $\mathbb{R}$ and for any $\epsilon > 0$, the regret of $\mathcal{A}_V$ satisfies $R_T^{\mathcal{A}_V}(\mathbf{0}) = \sum_{t=1}^T g_t(v_t^{(n)} - \mathbf{0}) \leq G\epsilon$. Then, for any $n \in [N]$ and any $\mathbf{u}_1, \dots, \mathbf{u}_T$ in $\mathbb{R}^d$, Algorithm 3.1 guarantees*

$$\mathbb{E}[R_T(\mathbf{u}_1, \dots, \mathbf{u}_T)] \leq \mathbb{E}\left[R_T^{\mathcal{A}_V^{(n)}}(MN)\right] + MN\mathbb{E}\left[R_T^{\mathcal{A}_B^{(n)}}\left(\frac{\mathbf{u}_1}{M}, \dots, \frac{\mathbf{u}_T}{M}\right)\right] + (N-1)G\epsilon$$

where $M = \max_t \|\mathbf{u}_t\|$.

Proof. The result is immediate by applying Proposition 3.1 followed by the scale / direction decomposition:

$$\begin{aligned} \mathbb{E}[R_T(\mathbf{u}_1, \dots, \mathbf{u}_T)] &= \mathbb{E}\left[\sum_{t=1}^T \langle \ell_t, \mathbf{w}_t - \mathbf{u}_t \rangle\right] \\ &= \mathbb{E}\left[\sum_{t=1}^T \langle \tilde{\ell}_t^{(n)}, \beta_t^{(n)} \rangle v_t^{(n)} - \langle \tilde{\ell}_t^{(n)}, N\mathbf{u}_t \rangle\right] \\ &= \mathbb{E}\left[\sum_{t=1}^T \langle \tilde{\ell}_t^{(n)}, \beta_t^{(n)} \rangle v_t^{(n)} \pm \langle \tilde{\ell}_t^{(n)}, \beta_t^{(n)} \rangle NM - \langle \tilde{\ell}_t^{(n)}, N\mathbf{u}_t \rangle\right] \\ &\leq \underbrace{\mathbb{E}\left[\sum_{t=1}^T \langle \tilde{\ell}_t^{(n)}, \beta_t^{(n)} \rangle (v_t - MN)\right]}_{R_T^{\mathcal{A}_V^{(n)}}(MN)} + MN \underbrace{\mathbb{E}\left[\sum_{t=1}^T \langle \tilde{\ell}_t^{(n)}, \beta_t^{(n)} - \frac{\mathbf{u}_t}{M} \rangle\right]}_{R_T^{\mathcal{A}_B^{(n)}}\left(\frac{\mathbf{u}_1}{M}, \dots, \frac{\mathbf{u}_T}{M}\right)} + (N-1)G\epsilon \\ &= \mathbb{E}\left[R_T^{\mathcal{A}_V^{(n)}}(MN)\right] + MN\mathbb{E}\left[R_T^{\mathcal{A}_B^{(n)}}\left(\frac{\mathbf{u}_1}{M}, \dots, \frac{\mathbf{u}_T}{M}\right)\right] + (N-1)G\epsilon \quad \square \end{aligned}$$

As observed above, the key insight of van der Hoeven et al. (2020) is that the feedback received by the scale learner is actually full-information feedback; indeed, a scale learner's loss function $v \mapsto \langle \ell_t, \beta_t \rangle v$ can be precisely recovered from $\langle \ell_t, \mathbf{w}_t \rangle = \langle \ell_t, \beta_t \rangle v_t$ by dividing it by the scale v_t , so

Algorithm 3.3: Continuous Exponential Weight for Linear Bandits

Input: Action set $\mathcal{W}$, initial exploration distr. π , learning rate $\eta > 0$, exploration param. $\beta > 0$

Initialize: $\tilde{p}_1 = \pi$

for $t = 1 \dots T$ **do**

 Sample $\mathbf{w}_t \sim \tilde{p}_t$, observe $\langle \ell_t, \mathbf{w}_t \rangle$

 Let $\mathbf{Q}_t = \mathbb{E}_{\mathbf{w} \sim \tilde{p}_t}[\mathbf{w}\mathbf{w}^\top]$ and estimate $\hat{\ell}_t = \mathbf{Q}_t^{-1} \langle \ell_t, \mathbf{w}_t \rangle \mathbf{w}_t$

 Update $p_{t+1} = \arg \min_{\mathbf{p} \in \mathcal{M}_1(\mathcal{W})} \mathbb{E}_{\mathbf{w} \sim \mathbf{p}}[\langle \ell_t, \mathbf{w} \rangle] + D_\psi(\mathbf{p} | \tilde{p}_t)$

 Update $\tilde{p}_{t+1} = (1 - \beta)p_{t+1} + \beta\pi$

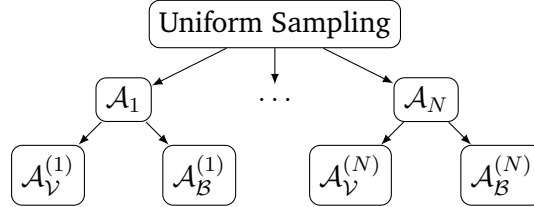


Figure 3.1: Illustration of how the Uniform Sampling interface interacts with each base algorithm $\mathcal{A}_i$. Each base algorithm internally applies the direction and scale decomposition, using its own hyperparameters.

no tricky loss estimation is required for the scale learner. Moreover, the scale learner faces a *static* regret problem, so overall to meet the condition of Proposition 3.2,

it will suffice to apply any comparator-adaptive OCO algorithm for static regret as the scale learner, leading to a guarantee of the form $R_T^{\mathcal{A}_V^{(n)}}(M) = \tilde{O}(M\sqrt{T})$.

All that remains is to design a bandit algorithm which can handle dynamic regret on a ball. To achieve this, we use the Continuous Exponential Weights algorithm (Algorithm 3.3) and derive the following dynamic regret guarantee, proven in Appendix A.1.

Lemma 3.1. *Let $(\|\cdot\|, \|\cdot\|_*)$ be a dual-norm pair and suppose that $\|\mathbf{w}\| \leq D$ for all $\mathbf{w} \in \mathcal{W}$. Then, for any sequence $\mathbf{u}_1, \dots, \mathbf{u}_T$ in $\mathcal{W}$ and for any sequence of distributions $\mathbf{p}_1^*, \dots, \mathbf{p}_T^*$ satisfying $\mathbb{E}_{\mathbf{w} \sim \mathbf{p}_t^*}[\mathbf{w}] = \mathbf{u}_t$, Algorithm 3.3 with regularizer $\psi(\mathbf{w}) = \frac{1}{\eta} \sum_{i=1}^d (w_i \log(w_i) - w_i)$ and exploration distribution uniform over $\mathcal{W}$ guarantees*

$$\mathbb{E}[R_T(\mathbf{u}_1, \dots, \mathbf{u}_T)] \leq \frac{1 + \log(d)(1 + P_T^\Delta)}{\eta} + \eta d D^2 \sum_{t=1}^T \|\ell_t\|_*^2.$$

Note that, in this specific setting, we obtain a guarantee on the distributional path length P_T^Δ . This results from the fact that Continuous Exponential Weights works with the distributions over actions rather than the actions themselves.

With this guarantee at hand, we can see that the optimal choice of learning rate would be $\eta^* \sim \sqrt{\frac{P_T^\Delta \log d}{dT}}$. However, choosing this step-size would require knowledge of P_T^Δ . Instead, we will use the guarantee-combining argument suggested above to consider a grid of exponentially spaced candidate step-sizes:

$$\mathcal{S} = \left\{ \frac{2^i}{GT} \wedge \frac{1}{G} : i = 0, 1, \dots \right\}. \quad (3.2)$$

By running a base algorithm for each $\eta_i \in \mathcal{S}$, we guarantee that at least one of them achieves regret within a constant factor of the optimal choice η^* (see Appendix A.2). Combining this with the guarantees of the scale and direction learners above and applying Proposition 3.2, we obtain the

following guarantee (proof in Appendix A.1.2).

Theorem 3.1. *Let $\mathcal{S}$ as defined in Equation (3.2). For all $i \in |\mathcal{S}|$, let $\mathcal{A}_i$ be an instance of Algorithm 3.2 such that the direction learner $\mathcal{A}_B^{(i)}$ is an instance of Algorithm 3.3 with step-size η_i on domain $\mathcal{B}_N$, and $\mathcal{A}_V^{(i)}$ is a comparator-adaptive OLO routine. Then for any sequence $\mathbf{u}_1, \dots, \mathbf{u}_T$ in $\mathcal{W}$ and sequence of distributions $\mathbf{p}_1^*, \dots, \mathbf{p}_T^*$ satisfying $\mathbb{E}_{\mathbf{w} \sim \mathbf{p}_t^*}[\mathbf{w}] = \mathbf{u}_t$, Algorithm 3.1 guarantees*

$$\mathbb{E}[R_T(\mathbf{u}_1, \dots, \mathbf{u}_T)] = \tilde{\mathcal{O}}\left(G\epsilon + dMG(1 + P_T^\Delta) + dM\sqrt{N(1 + P_T^\Delta)\sum_{t=1}^T\|\ell_t\|_*^2}\right),$$

where $M = \max_t \|\mathbf{u}_t\|$.

As noted above, P_T^Δ is a generalization of the number of switches S_T . Hence, this result implies the optimal switching regret guarantee $\mathcal{O}(dMG\sqrt{(S_T + 1)T})$ up to poly-logarithmic terms.

3.3 Conclusions and Future Work

The approach proposed in this paper is remarkably simple and allows hyperparameters to be tuned on-the-fly, enabling adaptivity to problem parameters that cannot be directly observed or estimated, such as the path-length. This leads to the first optimal parameter-free dynamic regret bound for linear bandits. This technique can easily be extended to other settings involving hard-to-estimate quantities, but it crucially relies on the scale and direction decomposition, which may not apply to arbitrary action sets. Extending this to more general or irregular action sets—e.g., combinatorial sets—is non-trivial, as these often lack a clear notion of direction or have an irregular geometry that complicates such decoupling.

A natural question is whether it is possible to recover guarantees that scale with the standard path length P_T in linear bandits. One possible approach is to directly optimize in the action space—e.g., via mirror descent—and then construct a distribution p_t such that $\mathbb{E}_{p_t}[\mathbf{w}] = \mathbf{u}_t$, playing $\mathbf{w} \sim p_t$. However, the feasibility and variance control of such sampling methods remain challenging.

Chapter 4

Bandits with Abstention under Expert Advice

In this chapter we introduce more structure in the bandits with expert advice framework introduced in Section 2.2.1. We consider a finite number of experts that suggest a probability distributions over an action space. The learner will therefore choose between playing an action **or** abstain.

In practical applications, errors often lead to severe consequences, and consistently making predictions is neither safe nor economically practical. For this reason, the abstention option has gained a lot of interest in the literature, both in the batch and online setting (Chow, 1957, 1970, Hendrickx, Perini, Van der Plas, Meert, and Davis, 2021, Cortes, DeSalvo, Gentile, Mohri, and Yang, 2018). Similarly to previous works, the setting studied in this chapter is based on the assumption that one of the actions always has *zero reward*: such an action is equivalent to an abstention of the learner from play. Besides the rewards being bounded, we make no additional assumptions regarding how the rewards or expert predictions are generated. In this chapter, we present an efficient algorithm CBA (Confidence-rated Bandits with Abstentions) which exploits the abstention action to get reward bounds that can be dramatically higher than those of EXP4 (Auer et al., 2002). In the worst case, our reward bound matches that of EXP4 so that CBA can be seen as a strict improvement, since the time-complexities of the two algorithms are, up to a factor logarithmic in the time horizon, identical in the general case.

Our problem can also be seen as that of aggregating *confidence-rated predictors* (Blum and Mansour, 2007, Gaillard, Stoltz, and Van Erven, 2014, Luo and Schapire, 2015) when the learner has the option of abstaining from taking actions. When the problem is phrased in this way, at the start of each trial, each predictor recommends a probability distribution over the actions (which now may not include an action with zero reward) but with a confidence rating. A low confidence rating can mean that either the predictor thinks that all actions are bad (so that the learner should abstain) or simply does not know which action is the best. Previous works on confidence-rated experts measure the performance of their algorithm in terms of the sum of *scaled* per-trial rewards. In contrast to previous algorithms, our approach allows for the derivation of bounds on the expected cumulative reward of CBA.

This formulation enables us to extend our work to the problem of adversarial contextual bandits with the abstention option, which has not been studied before. Previous work has considered the abstention option in the standard (context-free) adversarial bandit setting or in stochastic settings

(Cortes et al., 2018, Cortes, DeSalvo, Gentile, Mohri, and Zhang, 2020, Neu and Zhivotovskiy, 2020), but not in the contextual and adversarial case. Moreover, their results and methods cannot be applied to confidence-rated predictors. To get more intuition on this setup, we can think of any deterministic policy that maps contexts into actions. Any such policy can be viewed as a classifier, with *foreground* classes associated with each action and a *background* class associated with abstaining. Our learning bias is represented by a set of information we refer to as the *basis*, which we formally define later. It encodes contextual structural assumptions that hold exclusively for the foreground classes and are provided to the algorithm a priori. A particular type of basis is generated by a set of potential clusters that can overlap. Alternatively, a basis can also be created using balls generated by any kind of distance function, which groups contexts believed to be close together. For this latter family of basis, we can also achieve a significant speedup in the per-trial time complexity of CBA. This result is very different (and incomparable) to other results about adversarial bandits in metric spaces (Pasteris, Hicks, and Mavroudis, 2023b, Pasteris, Dwyer, Hicks, and Mavroudis, 2023a).

4.1 Additional related work

As discussed in the introduction, a key aspect of this chapter is the option to abstain from making any decision. In the batch setting (Chow, 1957, 1970), this option is usually referred to as “rejection”. These works study whether to use or reject a specific model prediction based on specific requests (see Hendrickx et al. (2021) for a survey). In online learning, “rejection” can be the possibility of abstention by the learner. These works usually rely on a cost associated with the abstention action. Neu and Zhivotovskiy (2020) studied the magnitude of the cost associated with abstention in an expert setting with bounded losses. They state that if the cost is lower than half of the amplitude of the interval of the loss, it is possible to derive bounds that are independent of the time. In Cortes et al. (2018), a non-contextual and partial information setting with the option of abstention is studied. The sequel model (Cortes et al., 2020) regards this model as a special case of their stochastic feedback graph model. Schreuder and Chzhen (2021) studied the fairness setting when using the option of abstaining as it may lead to discriminatory predictions.

One specific scenario where prior algorithms can establish cumulative reward bounds is as follows: on any given trial, the predictors are *specialists* (Freund, Schapire, Singer, and Warmuth, 1997), having either full confidence (a.k.a. *awake*) or no confidence (a.k.a. *asleep*). The SPECIALISTEXP algorithm by Herbster, Pasteris, Vitale, and Pontil (2021), a bandit version of the standard specialist algorithm, achieves regret bounds with respect to any subset of specialists where exactly one specialist is awake on each trial. We differ from this work as abstention is an algorithmic choice. Instead of sleeping in the rounds where the specialist is not active, the specialist will vote for abstention, which is a proper action of our algorithm. In Section 4.5.2, we present an illustrative problem involving learning balls in a space equipped with a metric. This example demonstrates our capability to significantly improve on SPECIALISTEXP. For this problem, we also present subroutines that significantly speed up CBA.

4.2 Problem formulation and notation

In this problem we consider $K + 1$ actions, E experts, and T trials. On each trial t :

1. *Question*: Each expert suggests, to the learner, a probability distribution over the $K + 1$ actions.
2. *Answer*: The learner selects an action a_t .
3. *Feedback*: The reward incurred by action a_t on trial t (which is in $[-1, 1]$) is revealed to the learner.

We note that the experts' suggestions and the rewards (associated with each action) are chosen a-priori and hence do not depend on the learner's actions. The aim of the learner is to maximize the cumulative reward obtained by its selected actions. As discussed in the introduction, we consider the case in which there is an action (the abstention action) that incurs zero reward on every trial.

We denote our action set by $[K] \cup \{\square\}$ where $\square$ is the abstention action. For each trial $t \in [T]$ we define the vector $\mathbf{r}_t \in [-1, 1]^K$ such that for all $a \in [K]$, $r_t(a)$ is the reward obtained by action a on trial t . Moreover, we define $r_t(\square) := 0$ which is the reward of the abstention action $\square$.

It will be useful for us to represent probability distributions over the actions by vectors in the set:

$$\mathcal{A} := \{\mathbf{s} \in [0, 1]^K \mid \|\mathbf{s}\|_1 \leq 1\}.$$

Any vector $\mathbf{s} \in \mathcal{A}$ represents the probability distribution over actions which assigns, for all $a \in [K]$, a probability of s_a to action a , and assigns a probability of $1 - \|\mathbf{s}\|_1$ to the abstention action $\square$, where $\|\mathbf{s}\|_1$ denotes 1-norm of $\mathbf{s}$. We write $a \sim \mathbf{s}$ to represent that action a is drawn from the probability distribution $\mathbf{s}$. We will refer to the elements of the set $\mathcal{A}$ as *stochastic actions*.

A *policy* is any element of $\mathcal{A}^T$ (noting that any such policy is a matrix in $[0, 1]^{T \times K}$). Any policy $\mathbf{e} \in \mathcal{A}^T$ defines a stochastic sequence of actions: on every trial $t \in [T]$ an action $a \in [K] \cup \{\square\}$ being drawn as $a \sim \mathbf{e}_t$. Note that if the learner plays according to a policy $\mathbf{e} \in \mathcal{A}^T$, then on each trial t it obtains an expected reward of $\mathbf{r}_t \cdot \mathbf{e}_t$, where the operator $\cdot$ denotes the dot product. Note that each expert is equivalent to a policy. Thus, for all $i \in [E]$ we denote the i -th expert by $\mathbf{e}^i \in \mathcal{A}^T$. Hence, at the start of each trial $t \in [T]$, the learner views the sequence $\langle \mathbf{e}_t^i \mid i \in [E] \rangle$.

We can also view the experts as *confidence-rated predictors* over the set $[K]$: for each $i \in [E]$ and $t \in [T]$, the vector $\mathbf{e}_t^i$ can be viewed as suggesting the probability distribution $\mathbf{e}_t^i / \|\mathbf{e}_t^i\|_1$ over $[K]$, but with confidence $\|\mathbf{e}_t^i\|_1$. We denote this confidence by $c_t(i) := \|\mathbf{e}_t^i\|_1$ and write $\mathbf{c}_t := (c_t(1), \dots, c_t(E))$.

In this chapter, we will refer to the *unnormalized relative entropy* defined by:

$$\Delta(\mathbf{u}, \mathbf{v}) := \sum_{i \in [E]} u_i \ln \left(\frac{u_i}{v_i} \right) - \|\mathbf{u}\|_1 + \|\mathbf{v}\|_1$$

for any $\mathbf{u}, \mathbf{v} \in \mathbb{R}_+^E$. We will also use the Iverson bracket notation $\mathbb{I}\{\text{Pred}\}$ as the indicator function, meaning that it is equal to 1 if Pred is true, and 0 otherwise. All the proofs are in the Appendix.

4.3 Main result

Our main result is represented by a bound on the cumulative reward of our algorithm CBA. We note that any *weight* vector $\mathbf{u} \in \mathbb{R}_+^E$ induces a matrix $\boldsymbol{\pi}(\mathbf{u}) \in \mathbb{R}_+^{T \times K}$ defined by

$$\boldsymbol{\pi}(\mathbf{u}) := \sum_{i \in [E]} u_i \mathbf{e}^i,$$

which is the linear combination of the experts with coefficients given by $\mathbf{u}$. However, only some of such linear combinations generate valid policies. Thus, we define

$$\mathcal{V} := \{\mathbf{u} \in \mathbb{R}_+^E \mid \boldsymbol{\pi}(\mathbf{u}) \in \mathcal{A}^T\}$$

as the set of all weight vectors that generate valid policies. Particularly, note that $\mathbf{u} \in \mathcal{V}$ if and only if, on every trial t , the weighted sum of the confidences $\mathbf{u} \cdot \mathbf{c}_t$ is no greater than one. Given some $\mathbf{u} \in \mathcal{V}$, we define

$$\rho(\mathbf{u}) := \sum_{t \in [T]} r_t \cdot \pi_t(\mathbf{u}),$$

which would be the expected cumulative reward of the learner if it was to follow the policy $\boldsymbol{\pi}(\mathbf{u})$. We point out that the learner does not know $\mathcal{V}$ or the function $\boldsymbol{\pi}$ a-priori.

The following theorem (proved in Appendix B.1) allows us to bound the regret of CBA with respect to any valid linear combination $\mathbf{u}$ of experts.

Theorem 4.1. *CBA takes parameters $\eta \in (0, 1)$ and $\mathbf{w}_1 \in \mathbb{R}_+^E$. For any $\mathbf{u} \in \mathcal{V}$ the expected cumulative reward of CBA is bounded below by:*

$$\sum_{t \in [T]} \mathbb{E}[r_t(a_t)] \geq \mathbb{E}[\rho(\mathbf{u})] - \frac{\Delta(\mathbf{u}, \mathbf{w}_1)}{\eta} - \eta(12K + 2)T,$$

where the expectations are with respect to the randomization of CBA's strategy. The per-trial time complexity of CBA is in $\mathcal{O}(KE)$.

We now compare our bound to those of previous algorithms. Firstly, EXP4 can only achieve bounds relative to a $\mathbf{u} \in \mathcal{V}$ with $\|\mathbf{u}\|_1 = 1$, in which case it essentially matches our bound but with $12K + 2$ replaced by $8K + 8$. Hence, for any $\mathbf{u} \in \mathcal{V}$ the EXP4 bound essentially replaces the term $\rho(\mathbf{u})$ in our bound by $\rho(\mathbf{u})/\|\mathbf{u}\|_1$. Note that $\|\mathbf{u}\|_1$ could be as high as the number of experts which implies we can dramatically outperform EXP4*.

Secondly, when viewing our experts as confidence-rated predictors, we note that previous algorithms for this setting only give bounds on a weighted sum of the per-trial rewards where the weight on each trial is $\mathbf{u} \cdot \mathbf{c}_t$ for some $\mathbf{u} \in \mathcal{V}$. This is only a cumulative reward bound when $\mathbf{u} \cdot \mathbf{c}_t = 1$ for all $t \in [T]$, and finding such a $\mathbf{u}$ is typically impossible. When there does exist $\mathbf{u}$ that satisfies this constraint, the reward relative to $\mathbf{u}$ is essentially the same as for us (Blum and Mansour, 2007). However, there will often be another value of $\mathbf{u}$ that will give us a much better bound, as we show in Section 4.5.2.

4.4 The CBA algorithm

The CBA algorithm is given in Algorithm 1. In this section, we describe its derivation via a modification of the classic *mirror descent* algorithm.

Our modification of mirror descent is based on the following mathematical objects. For all $t \in [T]$

*Precisely, if for each expert there exists a trial in which the confidence is 1, then we have $0 \leq \|\mathbf{u}\|_1 \leq E$. Otherwise can be high as $0 \leq \|\mathbf{u}\|_1 \leq E/c^*$, where $c^* = \max_{t \in [T]} c_t^i$.

Algorithm 4.1: CBA(w_1, η)

```

for  $t = 1, 2, \dots, T$  do
  For all  $i \in [E]$  receive  $e_t^i$ 
  For all  $i \in [E]$  set  $c_t(i) \leftarrow \|e_t^i\|_1$ 
  if  $\|c_t\|_1 \leq 1$  then
    Set  $\tilde{w}_t \leftarrow w_t$ 
  else
    By interval bisection find  $\lambda > 0$  such that:
      
$$\sum_{i \in [E]} c_t(i) w_{t,i} \exp(-\lambda c_t(i)) = 1$$

    For all  $i \in [E]$  set  $\tilde{w}_t(i) \leftarrow w_{t,i} \exp(-\lambda c_t(i))$ 
  Set:
      
$$s_t \leftarrow \sum_{i \in [E]} \tilde{w}_t(i) e_t^i$$

  Draw  $a_t \sim s_t$ 
  Receive  $r_t(a_t)$ 
  For all  $a \in [K]$  set:
      
$$\hat{r}_{t,a} \leftarrow 1 - \mathbb{I}\{a = a_t\} (1 - r_t(a_t)) / s_t(a_t)$$

  For all  $i \in [E]$  set  $w_{(t+1),i} \leftarrow \tilde{w}_t(i) \exp(\eta e_t^i \cdot \hat{r}_t)$ 

```

we first define:

$$\mathcal{V}_t := \{v \in \mathbb{R}_+^E \mid \|\pi_t(v)\|_1 \leq 1\},$$

which is the set of all weight vectors that give rise to linear combinations producing valid stochastic actions at trial t . Given some $t \in [T]$, we define our *objective function* $\rho_t : \mathcal{V}_t \rightarrow [-1, 1]$ as

$$\rho_t(v) := r_t \cdot \pi(v) \text{ for all } v \in \mathcal{V}_t.$$

Like mirror descent, CBA maintains, on each trial $t \in [T]$, a weight vector $w_t \in \mathbb{R}_+^E$. However, unlike mirror descent on the simplex, we do not keep w_t normalized, but we will instead project it into $\mathcal{V}_t$ at the start of trial t , producing a vector $\tilde{w}_t$. Also, unlike mirror descent, CBA does not use the actual gradient (which it does not know) of ρ_t at $\tilde{w}_t$, but (inspired by the Exp3 algorithm) uses an unbiased estimator instead. Specifically, on each trial $t \in [T]$, CBA does the following:

1. Set $\tilde{w}_t \leftarrow \operatorname{argmin}_{v \in \mathcal{V}_t} \Delta(v, w_t)$.
2. Randomly construct a vector $g_t \in \mathbb{R}^E$ such that $\mathbb{E}[g_t] = \nabla \rho_t(\tilde{w}_t)$.
3. Set $w_{t+1} \leftarrow \operatorname{argmin}_{v \in \mathbb{R}_+^E} (\eta g_t \cdot (\tilde{w}_t - v) + \Delta(v, \tilde{w}_t))$.

This naturally raises two questions: how is a_t selected and how is g_t constructed? On each trial $t \in [T]$ we define

$$s_t := \sum_{i \in [E]} \tilde{w}_t(i) e_t^i,$$

which is the stochastic action generated by the linear combination $\tilde{w}_t$, and select $a_t \sim s_t$. Note that:

$$\mathbb{E}[r_t(a_t)] = \rho_t(\tilde{w}_t), \tag{4.1}$$

which confirms that ρ_t is our objective function at trial t . Once $r_t(a_t)$ is revealed to us we can proceed to construct the gradient estimator $\mathbf{g}_t$. It is important that we construct this estimator in a specific way. Inspired by Exp4 we first define a reward estimator $\hat{\mathbf{r}}_t$ such that for all $a \in [K]$ we have:

$$\hat{r}_{t,a} := 1 - \mathbb{I}\{a = a_t\} (1 - r_t(a_t)) / s_t(a_t).$$

This reward estimate is unbiased as:

$$\mathbb{E}[\hat{r}_{t,a}] = 1 - \Pr[a = a_t] (1 - r_t(a_t)) / s_t(a_t) = r_t(a_t).$$

We then define, for all $i \in [E]$, the component:

$$g_{t,i} := \mathbf{e}_t^i \cdot \hat{\mathbf{r}}_t.$$

Note that for all $i \in [E]$ we have:

$$\mathbb{E}[g_{t,i}] = \mathbf{e}_t^i \cdot \mathbb{E}[\hat{\mathbf{r}}_t] = \mathbf{e}_t^i \cdot \mathbf{r}_t = \partial_i \rho_t(\tilde{\mathbf{w}}_t)$$

so that $\mathbb{E}[\mathbf{g}_t] = \nabla \rho_t(\tilde{\mathbf{w}}_t)$ as required.

Now that we defined the process by which CBA operates we must show how to compute $\tilde{\mathbf{w}}_t$ and $\mathbf{w}_{t+1}$. First we show how to compute $\tilde{\mathbf{w}}_t$ from $\mathbf{w}_t$. If $\|\mathbf{c}_t\|_1 \leq 1$ it holds that $\mathbf{w}_t \in \mathcal{V}_t$ so we immediately have $\tilde{\mathbf{w}}_t = \mathbf{w}_t$. Otherwise we must find $\tilde{\mathbf{w}}_t \in \mathbb{R}_+^E$ that minimizes $\Delta(\tilde{\mathbf{w}}_t, \mathbf{w}_t)$ subject to the constraint:

$$\sum_{i \in [E]} \tilde{w}_t(i) c_t(i) = 1,$$

which is equivalent to the constraint that $\|\boldsymbol{\pi}(\tilde{\mathbf{w}}_t)\|_1 = 1$. Hence, by Lagrange's theorem there exists λ such that:

$$\nabla_{\tilde{\mathbf{w}}_t} \left(\Delta(\tilde{\mathbf{w}}_t, \mathbf{w}_t) + \lambda \sum_{i \in [E]} \tilde{w}_t(i) c_t(i) \right) = 0$$

which is solved by setting, for all $i \in [E]$:

$$\tilde{w}_t(i) := w_{t,i} \exp(-\lambda c_t(i)).$$

The constraint is then satisfied if λ is such that:

$$\sum_{i \in [E]} c_t(i) w_{t,i} \exp(-\lambda c_t(i)) = 1.$$

Since this function is monotonic decreasing, λ can be found by interval bisection. For this computation step, we treat our numerical precision as a constant in our time complexity. In Appendix B.1.1, we show that, even if the numerical precision is unbounded, we incur a time complexity equal to that of Exp4, up to a factor logarithmic in T , adding only 1 to the regret.

Turning to the computation of $\mathbf{w}_{t+1}$, since it is unconstrained it is found by the equation:

$$\nabla_{\mathbf{w}_{t+1}} (\mathbf{g}_t \cdot \mathbf{w}_{t+1} + \eta^{-1} \Delta(\mathbf{w}_{t+1}, \tilde{\mathbf{w}}_t)) = 0.$$

which is solved by setting, for all $i \in [E]$:

$$w_{(t+1),i} := \tilde{w}_t(i) \exp(\eta g_{t,i}). \quad (4.2)$$

4.5 Adversarial contextual bandits with abstention

One main application of CBA is in the problem of adversarial contextual bandits with a finite context set. In this problem, we have a finite set of *contexts* $\mathcal{X}$. A-priori nature selects a sequence:

$$\langle (x_t, \mathbf{r}_t) \in \mathcal{X} \times [-1, 1]^K \mid t \in [T] \rangle,$$

but does not reveal it to the learner. For all $t \in [T]$ we define $r_t(\square) := 0$. On each trial $t \in [T]$ the following happens:

1. *Question*: The context x_t is revealed to the learner.
2. *Answer*: The learner selects an action $a_t \in [K] \cup \{\square\}$.
3. *Feedback*: The learner sees and incurs reward $r_t(a_t) \in [-1, 1]$.

We will assume that we are given, a-priori, a set $\mathcal{B} \subseteq 2^{\mathcal{X}}$ that we call the *basis*. We call each element of $\mathcal{B}$ a *basis element* (which is a set of contexts). We will later introduce various potential bases, determined by the nature of the context's structure: points within a metric space, nodes within a graph, and beyond. Importantly, our method is capable of accommodating any type of basis and, thus, any potential inductive bias that might be present in the data.

Given our basis we run our algorithm CBA with each expert corresponding to a pair $(B, k) \in \mathcal{B} \times [K]$. The expert corresponding to each pair (B, k) will deterministically choose action k when the current context x_t is in B , and abstain otherwise.

Corollary 4.1. *Given any basis $\mathcal{B}$ of cardinality N and any $M \in \mathbb{N}$ we can implement CBA such that for any sequence of disjoint basis elements $\langle B_j \mid j \in [M] \rangle$ with corresponding actions $\langle b_j \in [K] \mid j \in [M] \rangle$ we have:*

$$\sum_{t \in [T]} \mathbb{E}[r_t(a_t)] \geq \sum_{t \in [T]} \sum_{j \in [M]} \mathbb{I}\{x_t \in B_j\} r_t(b_j) - \sqrt{2M \ln(N)(6K+1)T}.$$

The per-trial time complexity of this implementation of CBA is in $\mathcal{O}(KN)$.

Proof. The choice of experts for CBA that leads to Corollary 4.1 is defined by the set of pairs so that $E = NK$ and for each $B \in \mathcal{B}$ and action $a \in [K]$ there exists a unique $i \in [E]$ such that for all $t \in [T]$ and $b \in [K]$ we have:

$$e_{t,b}^i := \mathbb{I}\{x_t \in B\} \mathbb{I}\{b = a\}.$$

By choosing $w_{1,i} := M/NK$ for all $i \in [E]$, and choosing

$$\eta := (M \ln(N)/(6K+1)T)^{-1/2},$$

Theorem 4.1 implies the reward bound in Corollary 4.1. The per-trial time complexity of a direct implementation of CBA for this set of experts would be $\mathcal{O}(KN)$. $\square$

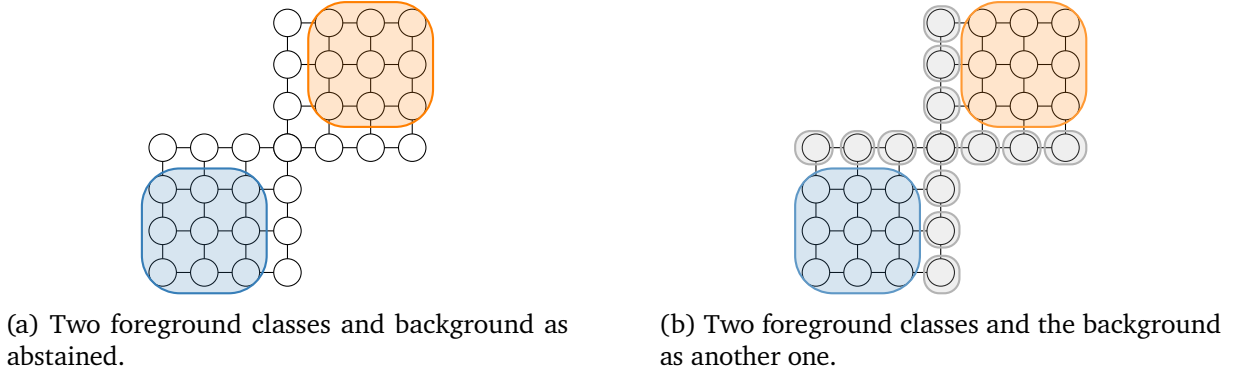


Figure 4.1: Illustrative example of abstention where we cover the foreground and background classes with metric balls. We consider two clusters (blue and orange) as the foreground and one background class (white), using the shortest path d_∞ metric. Using abstention, we can cover two clusters with one ball for each and abstain the background with no balls required (Fig. (a)). In contrast, if we treat the background class as another class, it would require significantly more balls to cover the background class, as seen by the 10 gray balls in Fig. (b). If the number of balls to cover significantly increases like in this case, the bound involving the number of balls also gets significantly worse.

We briefly comment on the term:

$$\sum_{j \in [M]} \mathbb{I}\{x_t \in B_j\} r_t(b_j),$$

that appears in the theorem statement. If x_t does not belong to any of the sets in $\langle B_j \mid j \in [M] \rangle$ then this term is equal to zero (which is the reward of abstaining). Otherwise, since the sets are disjoint, x_t belongs to exactly one of them and the term is equal to the reward induced by the action that corresponds to that set. In other words, the total cumulative reward is bounded relative to that of the policy that abstains whenever x_t is outside the union of the sets and otherwise selects the action corresponding to the set that x_t lies in.

Note the vast improvement of our reward bound over that of `SPECIALISTEXP` with abstention as one of the actions. Let's assume our context set is a metric space and our basis is the set of all balls. In order to get a reward bound for `SPECIALISTEXP`, the sets in which the specialists are awake must partition the set $\mathcal{X}$. This means that we must add to our M balls a disjoint covering (by balls) of the complement of the union of the original M balls. Note that the added balls correspond to the sets in which the specialists predicting the abstention action are awake. Typically this would require a huge number of balls so that the total number of specialists is huge (much larger than M); this huge number of specialists essentially replaces the term M in our reward bound (we illustrate an example in Figure 4.1).

Furthermore, in Appendix B.4, we show that the same implementation of CBA is capable of learning a weighted set of *overlapping* basis elements, as long as the sum of the weights of the basis elements covering any context is bounded above by one, which `SPECIALISTEXP` cannot do in general.

As we will see below, the practical bases we propose have a moderate size of $|\mathcal{B}| = \mathcal{O}(|\mathcal{X}|^2)$ leading to a per-step runtime of $\mathcal{O}(K|\mathcal{X}|^2)$ for CBA in this contextual bandit problem. In Section 4.5.2, we show how to significantly improve the runtime for a broad family of bases.

4.5.1 A lower bound

In this section, we show that CBA is, up to an $\mathcal{O}(\ln(|\mathcal{B}|))$ factor, essentially best possible on this contextual bandit problem:

Proposition 4.1. *Take any learning algorithm. Given any basis $\mathcal{B}$ and any $M \in \mathbb{N}$, for any sequence of disjoint basis elements $\langle B_j \mid j \in [M] \rangle$ there exists a sequence of corresponding actions $\langle b_j \in [K] \mid j \in [M] \rangle$ such that an adversary can force:*

$$\sum_{t \in [T]} \sum_{j \in [M]} \mathbb{I}\{x_t \in B_j\} r_t(b_j) - \sum_{t \in [T]} \mathbb{E}[r_t(a_t)] \in \Omega\left(\sqrt{MKT}\right).$$

4.5.2 Efficient learning with balls

In practice we can often quantify the similarity between any pair of contexts. That is, the contexts form a metric space, equipped with a *distance* function $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_+$ known to the learner a-priori. For example, contexts could have feature vectors in $\mathbb{R}^m$ (and the metric is the standard Euclidean distance or cosine similarity) or be nodes in a graph with the metric given by the shortest-path distance. A natural basis for this situation is the set of metric *balls*. Specifically, a ball is any set $B \subseteq \mathcal{X}$ in which there exists some $x \in \mathcal{X}$ and $\delta \in \mathbb{R}_+$ with:

$$B = \{z \in \mathcal{X} \mid d(x, z) \leq \delta\}.$$

For this broad family of bases[†] we can achieve the following speed-up, relying on a sophisticated data structure based on binary trees.

Theorem 4.2. *Let $N := |\mathcal{X}|$. Given any $M \in \mathbb{N}$ we can implement CBA such that for any sequence of disjoint balls $\langle B_j \mid j \in [M] \rangle$ with corresponding actions $\langle b_j \in [K] \mid j \in [M] \rangle$ we have:*

$$\sum_{t \in [T]} \mathbb{E}[r_t(a_t)] \geq \sum_{t \in [T]} \sum_{j \in [M]} \mathbb{I}\{x_t \in B_j\} r_t(b_j) - \sqrt{4M \ln(N)(6K+1)T}.$$

The per-trial time complexity of this implementation of CBA is in $\mathcal{O}(KN \ln(N))$.

As there are at most $\mathcal{O}(N^2)$ metric balls, this improves the runtime of the direct CBA implementation from $\mathcal{O}(KN^2)$ to $\mathcal{O}(KN \ln(N))$, that is almost linear per step. All the details are in Appendix B.2.

4.6 Experiments

This section conducts preliminary experiments, the code is available at GitHub[‡]. We evaluate our method to compare existing algorithms using graph data, since it is common to consider graph structures under the confidence-rated expert setting (Cesa-Bianchi, Gentile, and Zappella, 2013, Herbster et al., 2021). As mentioned above, the bases used in our algorithm can be constructed

[†]Actually we require a weaker condition. We only use the fact that for each context $z \in \mathcal{X}$ we have a set $\mathcal{B}_z = \{B_1^z, \dots, B_\ell^z\}$ of monotonically increasing basis elements, that is, $B_i^z \subseteq B_j^z$ for $i < j$, and the whole basis is formed by the union of these: $\mathcal{B} = \bigcup_{z \in \mathcal{X}} \mathcal{B}_z$.

[‡]<https://github.com/albertorumi/ContextualBanditsWithAbstention>

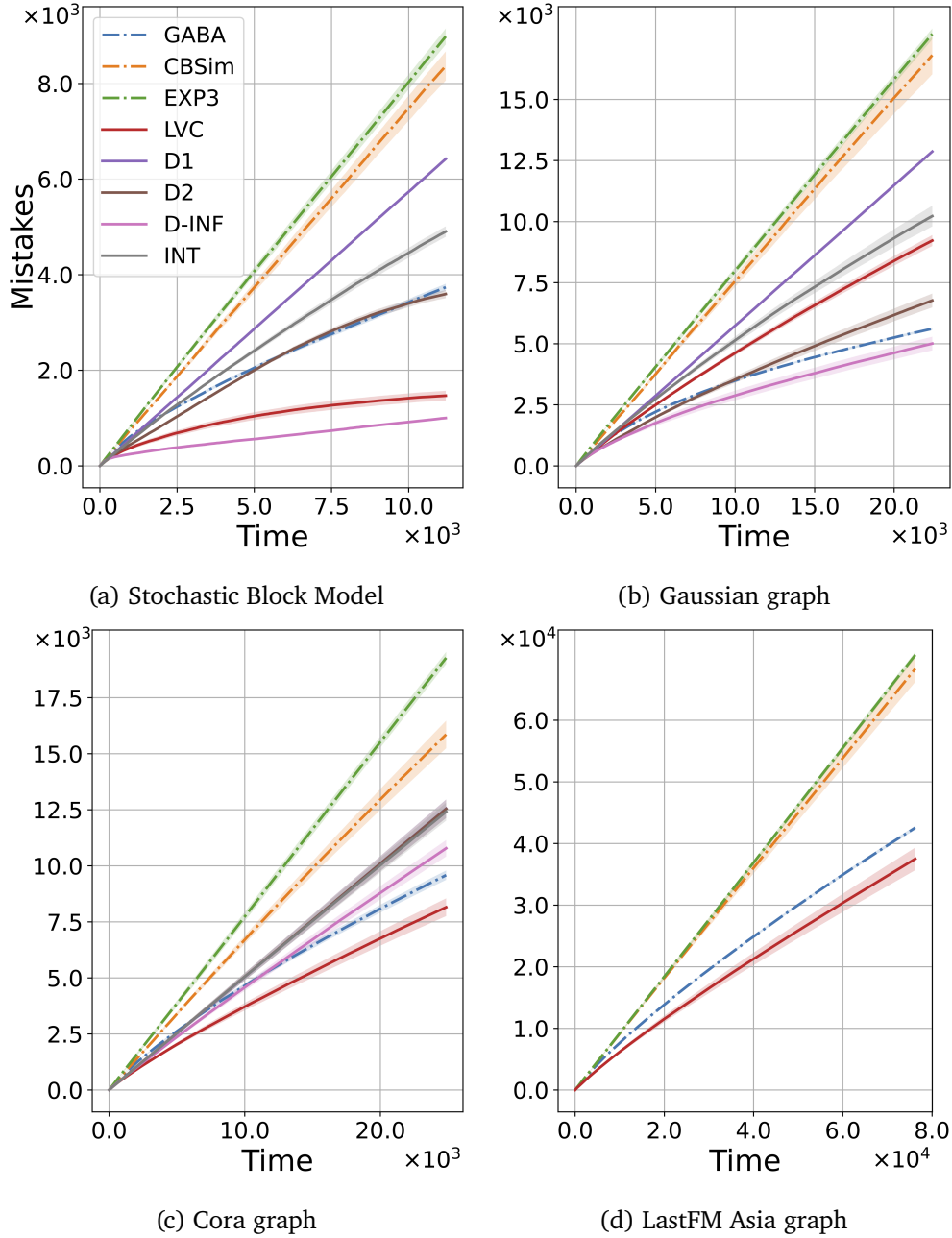


Figure 4.2: Results regarding the number of mistakes over time, the four main settings are presented from left to right: the Stochastic Block Model, Gaussian graph, Cora graph and LastFM Asia graph. In this context, D1, D2, and D-INF represent the p -norm bases, LVC represents the community detection basis, and INT represents the interval basis. The baselines, EXP3 for each context, Contextual Bandit with similarity, and GABA-II, are denoted as EXP3, CBSim, and GABA, respectively, and are represented with dashed lines. All the figures display the data with 95% confidence intervals over 20 runs, calculated using the standard error multiplied by the z -score 1.96.

arbitrarily, allowing to encompass different inductive biases based on applications. Thus, we consider some representative bases used on learning tasks on graphs before, each leading to different inductive priors on the contexts. We provide a short description of the bases here and refer to Appendix B.5 for more details.

Effective p -resistance basis d_p : Balls given by the metric

$$d_p(i, j) := \left(\min_{\substack{\mathbf{u} \in \mathbb{R}^N \\ u_i - u_j = 1}} \sum_{s, t \in V} |u_s - u_t|^p \right)^{-1/p}.$$

We use d_1 , d_2 , and d_∞ (Herbster and Lever, 2009).

Louvain method basis (LVC): Communities returned by the Louvain method (Blondel, Guillaume, Lambiotte, and Lefebvre, 2008), processed by the greedy peeling algorithm (Lanciano, Miyauchi, Fazzzone, and Bonchi, 2024).

Geodesic intervals basis (INT): All sets of the form $I(x, y) := \{z \in \mathcal{X} \mid z \text{ is on a shortest } x\text{-}y \text{ path}\}$ for all $x, y \in \mathcal{X}$ (Pelayo, 2013, Thiessen and Gärtner, 2021).

Let N be the cardinality of $|\mathcal{X}|$. For all three basis types, we immediately get an $\mathcal{O}(KN^2)$ runtime per step of CBA as there are $\mathcal{O}(N^2)$ basis elements. Moreover, for d_p balls and the LVC basis we can use the more efficient $\mathcal{O}(KN \ln N)$ implementation through Theorem 4.2. We empirically evaluate our approach in the context of online multi-class node classification on a given graph with bandit feedback. At each time step, the algorithm is presented with a node chosen uniformly at random and must either predict an action from the set of possible actions $[K]$ or abstain. The node can accept (resulting in a positive reward) or reject (resulting in a negative reward) the suggestion based on its preferred class with a certain probability. In a real-world application, this models a scenario where each user has a category preference (such as music genre or interest). When the item we decide to present matches their interest, there is a high probability of receiving a reward.

We compare our approach CBA using each of these bases on real-world and artificial graphs against the following baselines: an implementation of CONTEXTUALBANDIT from Slivkins (2011), the GABA-II algorithm proposed by Herbster et al. (2021), and an EXP3 instance for each data point. We use the following graphs for evaluation.

Stochastic block model. We use an established synthetic graph, *stochastic block model* (Holland, Laskey, and Leinhardt, 1983). This graph is generated by spawning an arbitrary number of disjoint cliques representing the foreground classes. Then an arbitrary number of background points are generated and connected to every possible point with a low probability. Figure 4.2a are displayed the results for the case of $F = 160$ nodes for each foreground class and $B = 480$ nodes for the background class. Connecting each node of the background class with a probability of $1/\sqrt{FB}$.

Gaussian graph. The points on this graph are generated in a two-dimensional space using five different Gaussian distributions with zero mean. Four of them are positioned at the corners of the unit square, representing the foreground classes and having a relatively low standard deviation. Meanwhile, the fifth distribution, representing the background class, is centered within the square and is characterized by a larger standard deviation. The points are linked in a k -nearest neighbors graph. In Figure 4.2b are displayed the results for 160 nodes for each foreground class and a standard deviation of 0.2, 480 nodes for the background class with a standard deviation of 1.75, along with a 7-nearest neighbors graph.

Real-world dataset. We tested our approach on the Cora dataset (Sen, Namata, Bilgic, Getoor, Galligher, and Eliassi-Rad, 2008) and the LastFM Asia dataset (Leskovec and Krevl, 2014). While both of these graphs contain both features and a graph, we exclusively utilized the largest connected component of each graph, resulting in 2485 nodes and 5069 edges for the Cora graph and 7624

nodes and 27806 edges for the LastFM Asia graph. Subsequently, we randomly chose a subset of three out of the original seven and eighteen classes, respectively, to serve as the background class. Additionally, we selected 15% of the nodes from the foreground classes randomly to represent noise points, and we averaged the results over multiple runs, varying the labels chosen for noise. Both in Figures 4.2c and 4.2d we averaged over 5 different label sets as noise. For the LastFM Asia graph, we exclusively tested the LVC bases, as it is the most efficient one to compute given the large size of the graph.

Results. The results from both synthetically generated tests (Figures 4.2a and 4.2b) demonstrate the superiority of our method when compared to the baselines. In particular, d_∞ -balls delivered exceptional results for both graphs, implying that d_∞ -balls effectively cover the foreground classes as expected. For the Cora dataset (Figure 4.2c), we observed that our method outperforms GABA-II only when employing the community detection basis. This similarity in performance is likely attributed to the dataset’s inherent lack of noise. Worth noting that the method we employed to inject noise into the dataset may not have been the optimal choice for this specific context. However, it is essential to highlight that our primary focus revolves around the abstention criteria, which plays a central role in ensuring the robustness of our model in the presence of noise. For the LastFM Asia dataset, our objective was to assess the practical feasibility of the model on a larger graph. We tested the LVC bases as they were the most promising and most efficient to compute. We outperform the baselines in our evaluation as shown in Figure 4.2d.

In summary, our first results confirm what we expected: our approach excels when we choose basis functions that closely match the context’s structure. However, it also encounters difficulties when the chosen basis functions are not a good fit for the context.

Chapter 5

Sum-Max Submodular Bandits

In many concrete settings of sequential decision-making, decisions are subsets of a finite set $[K]$ (possibly with cardinality constraints) and utilities, or rewards, are non-linear set functions over $[K]$. Although we may know that utility functions have some specific structure, e.g., they are submodular, the feedback may not reveal anything beyond the utility of the current decision. For example, consider an advertising campaign over $[K]$ digital channels (e.g., web, apps, and social media). Due to budget constraints, the campaign can show ads only on a subset of M channels for every user. If a user ends up buying the advertised product, we observe that a sale occurred, but we may not know which of the M channels triggered the purchase. The advertiser's goal is to choose the subset of channels for each new user in order to maximize the number of sales.

The same problem was studied (with a different motivation) by Simchowitz, Jamieson, and Recht (2016) under stochastic assumptions on the generation of the Bernoulli random variables each indicating whether displaying an ad on a certain channel triggers a purchase for the current user. In this chapter, we study the nonstochastic variant of this problem, where the binary variables associated with the channels are chosen, for each user, by an oblivious adversary. The main result we present in this chapter is an efficient algorithm minimizing regret in a much larger class of problems containing the multichannel advertising problem as a special case. In particular, our regret analysis applies to any sequential decision-making problem where reward functions belong to a subclass of all monotone submodular functions called *sum-max*.

A sum-max function is defined by a nonnegative matrix with K columns and an arbitrary number of rows. The value of the function evaluated at a subset $S \subset [K]$ of columns is the sum over the rows of the maximum row element over the subset S of columns. In the multichannel campaign example, the matrix is binary with a single row. The j -th entry indicates whether the current user would buy the product if advertised on channel j . If the matrix is square and symmetric, then we recover the non-metric facility location problem as a special case.

As we said earlier, our analysis of regret for sum-max functions assumes bandit feedback: at each time t we only observe the reward $r_t(\mathcal{A}_t)$ associated with our decision $\mathcal{A}_t$, where r_t is the sum-max function chosen by the adversary at time t . Hence, the reward $r_t(S)$ that we would have obtained by choosing any $S \neq \mathcal{A}_t$ remains unknown. We also consider cardinality constraints, in the form of a parameter M requiring that the decision $\mathcal{A}_t$ at each time t satisfy $|\mathcal{A}_t| \leq M$. Note that when $M = 1$ we recover the adversarial K -armed bandit problem.

The main result we present in this chapter is an efficient algorithm, MSE3, achieving a $\tilde{\mathcal{O}}(\sqrt{MKT})$

bound on the γ_M -regret for $\gamma_M = 1 - (1 - 1/M)^M$. For comparison, for the class of all monotone submodular functions, Niazadeh, Golrezaei, Wang, Susan, and Badanidiyuru (2021) obtain a $(1 - \frac{1}{e})$ -regret bound of $\mathcal{O}((\ln K)^{1/3} M(KT)^{2/3})$. As $\gamma_M > 1 - \frac{1}{e}$ for all $M > 1$, this bound is worse than ours in both approximation factor and regret.

When $M = 1$, algorithm MSE3 reduces to the standard Exp3 algorithm for K -armed bandits and our result specializes to the standard $\mathcal{O}(\sqrt{K(\ln K)T})$ regret bound of Exp3. This implies that the $\sqrt{KT}$ dependence in the regret bound is not improvable, even disregarding efficiency. Moreover, we show that improving on the approximation factor γ_M with an efficient algorithm would give an efficient randomized algorithm for solving set cover on $[K]$ with an approximation ratio of $(1 - \varepsilon) \ln K$, which is NP-hard for any $\varepsilon > 0$ (Dinur and Steurer, 2014).

In many real world problems, including an element i in the decision $\mathcal{A}_t$ at round t invokes a cost (i.e., a negative reward) $c_t(i) \geq 0$. When this is the case we would like to maximize the cumulative reward:

$$\sum_{t \in [T]} \left(r_t(\mathcal{A}_t) - \sum_{i \in \mathcal{A}_t} c_t(i) \right).$$

We show that MSE3 can handle this generalized problem if it receives, at the end of each round t , the values of $c_t(i)$ for all $i \in \mathcal{A}_t$. We note that the bandit MSE3 without costs is a special case of MSE3 with costs.

The inclusion of costs creates a tension between including arms in $\mathcal{A}_t$ to increase the reward and, simultaneously, avoid including too many arms to control the costs. We address this trade-off in Section 5.4 by introducing and analyzing a variant of MSE3 for regret minimization with costs and bandit feedback where the rewards are sum-max functions without cardinality constraints. We call this setting the bandit facility location problem because it is a bandit version of the online facility location problem studied by Pasteris, He, Vitale, Wang, and Herbster (2021).

For $M > 1$ and arbitrary costs, MSE3 selects $\mathcal{A}_t$ by performing M independent draws $a_t(1), \dots, a_t(M)$ from a distribution $\mathbf{p}_t = (p_t(1), \dots, p_t(K)) \in \Delta_K$. Then, a reward estimate for each $i \in [K]$ is computed using

$$g_t(i) = \frac{r_t(\mathcal{A}_t) - c_t(i)}{p_t(i)} \sum_{j \in [M]} \mathbb{I}\{a_t(j) = i\}, \quad (5.1)$$

where, for any statement S , the Iverson bracket notation $\mathbb{I}\{S\}$ is defined as $\mathbb{I}\{S\} = 1$ if S is true and $\mathbb{I}\{S\} = 0$ otherwise. Note that for $M = 1$ and $c_t(i) = 0$, the above reduces to the standard reward estimate of Exp3.

We now give an overview of how MSE3 works when we have no costs (i.e., $c_t(i) = 0$). For all set functions r , we construct a function $\Phi^r : \mathbb{R}_+^K \rightarrow \mathbb{R}$ such that for all $\mathbf{q} \in \Delta_K$ we have that $\Phi^r(\mathbf{q})$ is the expected value of $r(\mathcal{A})$ when $\mathcal{A}$ is constructed by drawing M arms i.i.d. with replacement from $\mathbf{q}$. Specifically, we first show that there exists a function $\tilde{r} : 2^{[K]} \rightarrow \mathbb{R}$ such that for all $\mathcal{Q} \subseteq [K]$ we have $r(\mathcal{Q}) = \sum_{\mathcal{S} \subseteq [K]} \mathbb{I}\{\mathcal{Q} \subseteq \mathcal{S}\} \tilde{r}(\mathcal{S})$. For all $\mathbf{q} \in \mathbb{R}_+^K$ we then define:

$$\Phi^r(\mathbf{q}) = \sum_{\mathcal{S} \subseteq [K]} \tilde{r}(\mathcal{S}) \left(\sum_{i \in [K]} \mathbb{I}\{i \in \mathcal{S}\} q(i) \right)^M.$$

We learn via online exponentiated gradient ascent using the unbiased estimates (5.1) of the

gradient of Φ^{r_t} . Clearly, for exponentiated gradient ascent to work we must have that, for all rounds t , our objective function Φ^{r_t} is concave over the simplex. We show that a sufficient condition for this to hold is that the function r_t is pseudo-concave, see Section 5.1 for a formal definition.

Next, we bound the regret with respect to any vector $\mathbf{p}^* \in \Delta_K$. Namely, we bound the expected reward of our algorithm relative to $\sum_{t \in [T]} \Phi^{r_t}(\mathbf{p}^*)$. By taking $\mathbf{p}^*$ such that $p^*(i) = \mathbb{I}\{i \in \mathcal{S}\} / |\mathcal{S}|$ for some set $\mathcal{S}$ we show that, because r_t is submodular, we have $\Phi^{r_t}(\mathcal{S}) \geq (1 - \alpha^M) r_t(\mathcal{S})$ where $\alpha = (|\mathcal{S}| - 1) / |\mathcal{S}|$. By bounding the variance of the gradient estimate we show that the regret term is $\tilde{\mathcal{O}}(\sqrt{MKT})$.

We have provided an overview of how, when we have no costs, MSE3 works and why we require r_t to be pseudo-concave and submodular. We now describe how costs are incorporated. This is done by using, instead of Φ^{r_t} , the function Ψ_t defined by:

$$\Psi_t(\mathbf{q}) = \Phi^{r_t}(\mathbf{q}) - M \sum_{i \in [K]} q(i) c_t(i),$$

so that $\Psi_t(\mathbf{p}_t)$ lower bounds the expected profit on trial t . Since Ψ_t differs from Φ^{r_t} by a linear function it is straightforward to extend the above methodology to this new objective function.

5.1 Sum-Max Functions

We now introduce sum-max functions and define the key property of this class that allows us to learn it with bandit feedback.

Definition 5.1. A set function $r : 2^{[K]} \rightarrow \mathbb{R}$ is sum-max if and only if there exists some $N \in \mathbb{N}$ and some matrix $\mathbf{V} \in \mathbb{R}^{N \times K}$ such that for all $\mathcal{S} \subseteq [K]$ with $\mathcal{S} \neq \emptyset$ we have:

$$r(\mathcal{S}) = \sum_{k \in [N]} \max_{i \in \mathcal{S}} V(k, i) \quad \text{and} \quad r(\emptyset) \leq \sum_{k \in [N]} \min_{i \in [K]} V(k, i)$$

For example, consider a marketplace with N buyers and K sellers. The value $V(k, i)$ is the combined utility of buyer k going to seller i . The value $r(\mathcal{S})$ is the social welfare when a subset $\mathcal{S}$ of sellers participate in the marketplace, and buyers match up with sellers to optimize their combined utilities. When there is only one buyer ($N = 1$), $\mathbf{V}$ is a vector $(V(1), \dots, V(K))$ and we view each $i \in [K]$ as an arm with reward V_i . Then $r(\mathcal{S}) = \max_{i \in \mathcal{S}} V(i)$, the maximum reward of an arm in the chosen set $\mathcal{S}$.

As sum-max functions are sums of monotone submodular functions, they are monotone submodular. We now list a number of sequential decision-making problems that can be expressed as regret minimization of specific sum-max functions under bandit feedback.

The multichannel campaign problem. This is our nonstochastic variant of the best-of- k bandit problem of Simchowitz et al. (2016). To view it as an instance of sum-max optimization, set $N = 1$ and let $V(i) \in \{0, 1\}$ indicate whether a user makes a purchase when the ad is displayed on channel i . Then $(V(1), \dots, V_K)$ can be viewed as the incidence vector of a subset $\mathcal{D} \subseteq [K]$ of channels, and the reward is defined by $r(\mathcal{S}) = \mathbb{I}\{\mathcal{S} \cap \mathcal{D} \neq \emptyset\}$. The feedback is bandit because we do not know what channel triggered the sale for that user.

Bandit hitting sets. This is a generalization of the previous example where $N \geq 1$ and $\mathbf{V}$ is

a boolean matrix. Each row of $\mathbf{V}$ denotes a subset $\mathcal{C}_k$ of $[K]$ and $V(k, i)$ indicates whether $i \in \mathcal{C}_k$. The value $r(\mathcal{S})$ then counts how many sets $\mathcal{C}_k$ have a non-empty intersection with $\mathcal{S}$. Bandit setting occurs when the sets remain unknown and each time we only observe the number of intersected sets.

Combinatorial bandits. Another important special case is when we receive the sum of the rewards r_i of the arms $i \in \mathcal{S}$. In this case $N = K$ and $V(k, i) = \mathbb{I}\{k = i\} r_i$. The problem is then equivalent to a combinatorial bandit (with full bandit feedback) over the class of M -sized subsets (Cesa-Bianchi and Lugosi, 2012).

Bandit k -medians. Given $\mathbf{x}_1, \dots, \mathbf{x}_N$ points in a metric space $(\mathcal{X}, d)$, consider the version of the k -medians problem (for $k = M$) where the M centroids have to be chosen in the given set of points. The value of the objective function at a candidate solution $\mathcal{S} \subset [K]$ with $|\mathcal{S}| \leq M$ can be written as

$$r(\mathcal{S}) = - \sum_{k \in [N]} \min_{i \in \mathcal{S}} d(\mathbf{x}_k, \mathbf{x}_i).$$

Clearly, this is a sum-max function for $\mathbf{V}$ with elements $V(k, i) := -d(\mathbf{x}_k, \mathbf{x}_i)$. The feedback is bandit when we do not know the metric, but we can observe the value of the objective function.

Some sum-max functions examples can be seen in Section 5.1

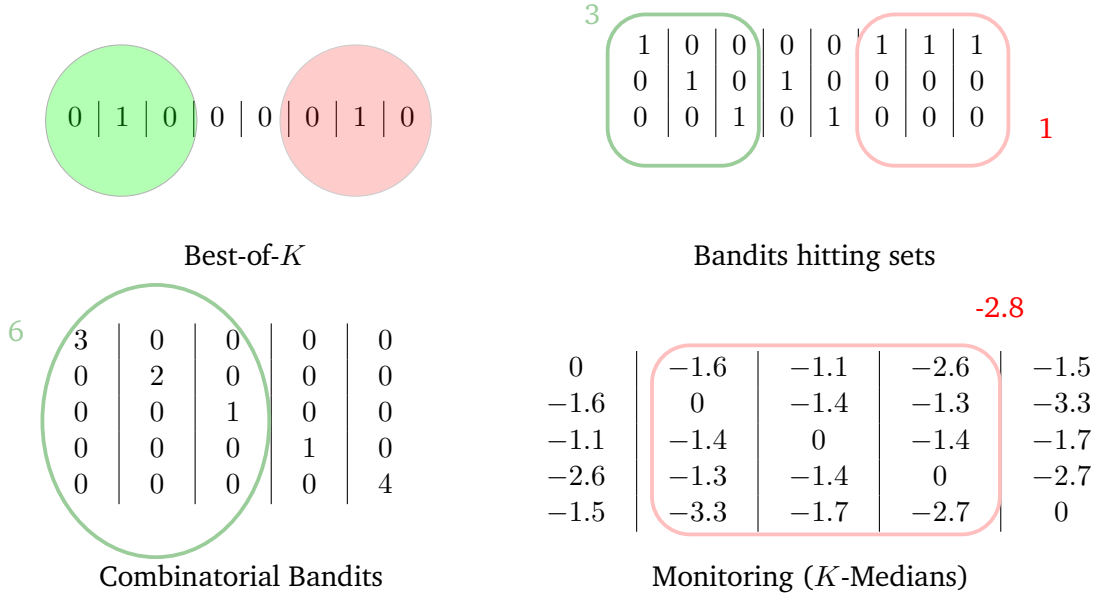


Table 5.1: Examples of Sum-Max Submodular functions

Next, we introduce an important property of sum-max functions.

Definition 5.2. Suppose we have a set function $r : 2^{[K]} \rightarrow \mathbb{R}$. For any $\mathcal{S} \subseteq [K]$ define the matrix $\mathbf{U}^{r, \mathcal{S}} \in \mathbb{R}^{K \times K}$ such that $U_{i,j}^{r, \mathcal{S}} = r(\mathcal{S} \cup \{i, j\})$ for all $i, j \in [K]$. We call the function r pseudo-concave if and only if $\mathbf{x}^\top \mathbf{U}^{r, \mathcal{S}} \mathbf{x} \leq 0$ for all $\mathcal{S} \subseteq [K]$ and all $\mathbf{x} \in \mathbb{R}^K$ with $\mathbf{x} \cdot \mathbf{1} = 0$.

In Appendix C.3, we show that there are monotone submodular functions that are not pseudo-concave. As a consequence, sum-max functions are indeed a proper subset of the class of monotone submodular functions. The following theorem confirms that all sum-max functions are pseudo-concave:

Theorem 5.1. Any sum-max set function is pseudo-concave.

Proof. Suppose we have some sum-max function $r : 2^{[K]} \rightarrow [0, 1]$. Let $\mathbf{V}$ be as in Definition 5.1. Without loss of generality, we will assume that all components of $\mathbf{V}$ are non-negative and $r(\emptyset) = 0$ (since any sum-max function can be transformed into this form by the addition of a constant). Define, for any $\mathcal{Q} \subseteq [K]$, the set function $r^{\mathcal{Q}} : 2^{[K]} \rightarrow [0, 1]$ such that for all $\mathcal{S} \subseteq [K]$ we have

$$r^{\mathcal{Q}}(\mathcal{S}) := \mathbb{I}\{\mathcal{S} \cap \mathcal{Q} \neq \emptyset\}.$$

We shall now show that for all such $\mathcal{Q}$ we have that $r^{\mathcal{Q}}$ is pseudo-concave. Choose any $\mathbf{x} \in \mathbb{R}^K$ with $\mathbf{x} \cdot \mathbf{1} = 0$ and any $\mathcal{S} \subseteq [K]$. We have two cases:

1. If $\mathcal{S} \cap \mathcal{Q} \neq \emptyset$, for all $i, j \in [K]$ we have $r^{\mathcal{Q}}(\mathcal{S} \cup \{i, j\}) = 1$, this implies $\mathbf{U}^{r^{\mathcal{Q}}, \mathcal{S}} = \mathbf{1}\mathbf{1}^\top$ and hence $\mathbf{x}^\top \mathbf{U}^{r^{\mathcal{Q}}, \mathcal{S}} \mathbf{x} = 0$.
2. If $\mathcal{S} \cap \mathcal{Q} = \emptyset$ then for all $i, j \in [K]$ we have

$$r^{\mathcal{Q}}(\mathcal{S} \cup \{i, j\}) = \mathbb{I}\{(i \in \mathcal{Q}) \vee (j \in \mathcal{Q})\}.$$

Let $\mathbf{z} \in \{0, 1\}^K$ be such that for all $k \in [K]$ we have $z_k := \mathbb{I}\{k \notin \mathcal{Q}\}$. Then for all $i, j \in [K]$ we have

$$\mathbb{I}\{(i \in \mathcal{Q}) \vee (j \in \mathcal{Q})\} = 1 - z_i z_j,$$

so that, by above, we have $\mathbf{U}^{r^{\mathcal{Q}}, \mathcal{S}} = \mathbf{1}\mathbf{1}^\top - \mathbf{z}\mathbf{z}^\top$. This implies that: $\mathbf{x}^\top \mathbf{U}^{r^{\mathcal{Q}}, \mathcal{S}} \mathbf{x} = -(\mathbf{x} \cdot \mathbf{z})^2 \leq 0$.

And therefore, $r^{\mathcal{Q}}$ is pseudo-concave.

Now suppose we have a vector $\mathbf{v} \in \mathbb{R}_+^K$ and define the set function $r^{\mathbf{v}} : 2^{[K]} \rightarrow \mathbb{R}_+$ such that for all $\mathcal{S} \subseteq [K]$ we have

$$r^{\mathbf{v}}(\mathcal{S}) := \max_{i \in \mathcal{S}} v(i),$$

where the maximum of the empty set is defined as equal to zero. We can order the set $[K]$ into the sequence $\langle j_i \mid i \in [K] \rangle = [K]$ where $v_{j_{i+1}} \leq v_{j_i}$ for all $i \in [K-1]$. For all $i \in [K]$ we can define $\mathcal{Q}_i := \{j_k \mid k \leq i\}$. Now note then that for all $\mathcal{S} \subseteq [K]$ the set function $r^{\mathbf{v}}(\mathcal{S})$ can be expressed as

$$\begin{aligned} & \sum_{i \in [K-1]} (v_{j_i} - v_{j_{i+1}}) \mathbb{I}\{\mathcal{S} \cap \mathcal{Q}_i \neq \emptyset\} + v_{j_K} \mathbb{I}\{\mathcal{S} \cap \mathcal{Q}_K \neq \emptyset\} \\ &= \sum_{i \in [K-1]} (v_{j_i} - v_{j_{i+1}}) r^{\mathcal{Q}_i}(\mathcal{S}) + v_{j_K} r^{\mathcal{Q}_K}(\mathcal{S}), \end{aligned}$$

so, by above, $r^{\mathbf{v}}$ is a positive sum of pseudo-concave functions and is hence itself pseudo-concave. Note also that $r^{\mathbf{v}}$ is clearly submodular. Noting that r is a positive sum of functions of the form $r^{\mathbf{v}}$ we have now shown that it is both pseudo-concave and submodular as required. $\square$

5.2 Additional Related Work

The work closest to ours is Pasteris et al. (2021), where they study online facility location with full information feedback. Our work improves on theirs in many respects: First, we solve the problem with bandit feedback, which requires designing an entirely different algorithm based on our discovery of an unbiased estimator for the gradient of our expected reward (we find it remarkable that such an estimator exists). As a consequence, our algorithm is also applicable to the full-information setting,

Algorithm 5.1: MSE3

 Set $\eta := \ln(K)/R$ and $p_{1,i} := 1/K$ for $i \in [K]$
for $t = 1, 2, \dots, T$ **do**:

 1. For all $j \in [M]$ draw $a_t(j) \in [K]$ from distribution $\mathbf{p}_t$

 2. Define $\mathcal{A}_t := \{a_t(j) \mid j \in [M]\}$

 3. Receive $r_t(\mathcal{A}_t)$ and $\{c_t(i) \mid i \in \mathcal{A}_t\}$

 4. For all $i \in [K]$ set

$$g_t(i) := \frac{r_t(\mathcal{A}_t) - c_t(i)}{p_t(i)} \sum_{j \in [M]} \mathbb{I}\{a_t(j) = i\}$$

 5. For all $i \in [K]$ define $\tilde{p}_t(i) := p_t(i) \exp(\eta g_t(i))$

 6. Define $\mathbf{p}_{t+1} := \tilde{\mathbf{p}}_t / \|\tilde{\mathbf{p}}_t\|_1$

where we obtain a per-trial running time of $\mathcal{O}(MK)$ when given an oracle for the reward function. When considering general sum-max functions, the methodology of Pasteris et al. (2021) would instead require a per-trial running time exponential in K^* . Second, our algorithm can efficiently learn classes that are even more general than sum-max functions. Third, we obtain tighter approximation ratios and show optimality for the multichannel campaign problem (and thus optimality in general).

Sum-max functions are a special case of linear submodular functions (Yue and Guestrin, 2011), which are of the form $r(\mathcal{S}) = \sum_{i \in [N]} w_i F_i(\mathcal{S})$ for $F_1, \dots, F_N$ monotone submodular functions and $w_1, \dots, w_N$ non-negative coefficients. However, linear submodular functions have been only studied in stochastic settings, assuming preliminary knowledge of $F_1, \dots, F_N$, and using a feedback model more informative than our bandit feedback.

Click-models (Lattimore and Szepesvári, 2020, Lattimore, Kveton, Li, and Szepesvari, 2018, Kveton, Szepesvari, Wen, and Ashkan, 2015) provide a different stochastic formalization of the best-of- k bandit problem. Here the user is presented with an ordered list of items, and the learner receives a positive reward if the user clicks on one of the presented items. The difference with our multichannel campaign problem is that the items are ordered, and the likelihood of clicking an item is also affected by the position of the item within the list.

5.3 Main Result

Our learning problem is formally defined as follows. The values $M, K \in \mathbb{N}$ and $C \in \mathbb{R}_+$ are all preliminarily known to the learner. Hidden from the learner, the adversary selects a sequence of set functions $\langle r_t \mid t \in [T] \rangle$, each with domain $2^{[K]}$ and a sequence of vectors $\langle \mathbf{c}_t \mid t \in [T] \rangle$ each in $[0, C]^K$. On each trial $t \in [T]$:

1. *Question*: The learner is asked to chose an action.
2. *Answer*: The learner chooses some $\mathcal{A}_t \subseteq [K]$ with $|\mathcal{A}_t| \leq M$.
3. *Feedback*: The value $r_t(\mathcal{A}_t)$ is revealed and for all $i \in \mathcal{A}_t$ the value $c_t(i)$ is also revealed.

The learner maintains a probability vector $\mathbf{p}_t \in \Delta_K$, and behaves as described in Algorithm 5.1.

To aid our theorem statement we add the following definitions. For all $t \in [T]$ and $\mathcal{Q} \subseteq [K]$ we define $\hat{r}_t(\mathcal{Q}) := r_t(\mathcal{Q}) - r_t(\emptyset)$, which is the difference between the learner's profit on trial t

*The work of Pasteris et al. (2021) only considered single-user cases, but it is straightforward to extend their methodology to general sum-max functions.

and that which it would have obtained by selecting the empty set, $\psi_t := \hat{r}_t(\mathcal{A}_t) - \gamma_t(\mathcal{A}_t)$, and $\gamma_t(\mathcal{Q}) := \sum_{i \in \mathcal{Q}} c_t(i)$. We note that by considering $\hat{r}_t$ instead of r_t our bounds do not change when r_t is shifted by an additive constant (which can be different for different trials t) as long as the range of r_t falls within the bounds described as follows.

We assume that the learner knows upper and lower bounds on the range r_t for all trials t . Hence, without loss of generality, assume that $r_t(\mathcal{Q}) \in [-1, 0]$ for all $t \in [T]$ and $\mathcal{Q} \subseteq [K]$ (otherwise scale and shift r_t and C). Let

$$R := (1 + C)\sqrt{2 \ln(K)M(K + M - 1)T}.$$

Our results hold for a relaxed notion of submodularity, which we call *pseudo-submodularity*.

Definition 5.3. A set function $r : 2^{[K]} \rightarrow \mathbb{R}$ is pseudo-submodular if and only if for every set $\mathcal{S} \subseteq [K]$ with $\mathcal{S} \neq \emptyset$ there exists some $i \in \mathcal{S}$ such that for all $\mathcal{Q} \subseteq \mathcal{S} \setminus \{i\}$ we have $r(\mathcal{Q} \cup \{i\}) - r(\mathcal{Q}) \geq r(\mathcal{S}) - r(\mathcal{S} \setminus \{i\})$.

Note that all pseudo-submodular set functions are also submodular. We now present our main result.

Theorem 5.2. Given r_t is pseudo-concave and pseudo-submodular for all $t \in [T]$, then for any set $\mathcal{S} \subseteq [K]$ with $\mathcal{S} \neq \emptyset$ we have

$$\sum_{t \in [T]} \mathbb{E}[\psi_t] \geq (1 - \alpha^M) \sum_{t \in [T]} \hat{r}_t(\mathcal{S}) - \frac{M}{|\mathcal{S}|} \sum_{t \in [T]} \gamma_t(\mathcal{S}) - R,$$

where

$$\alpha := 1 - \frac{1}{|\mathcal{S}|}.$$

Proof. See Section 5.5 □

We note that both the standard facility location and k -medians problems are often phrased as the minimization of a loss rather than a maximization of a profit. Our results easily capture this by considering the reward as a negative loss.

We now show that the approximation ratio $1 - \alpha^M$ is not improvable in general in the class of sum-max functions. In particular, we show that obtaining an efficient online learning algorithm for the multichannel advertising problem with a sublinear γ -regret with $\gamma < 1 - \alpha^M$ would give an efficient randomized algorithm for solving set cover on $[K]$ with an approximation better than $\ln K$. As shown in (Dinur and Steurer, 2014), obtaining an approximation of $(1 - \varepsilon) \ln K$ for set cover is NP-hard for any $\varepsilon > 0$.

Recall that an instance of the multichannel campaign problem over K ads is defined by a sequence $\langle r_t \mid t \in [T] \rangle$ of set functions over $[K]$ such that for all $t \in [T]$ there exists some $\mathcal{D}_t \subseteq [K]$ with $r_t(\mathcal{Q}) = \mathbb{I}\{\mathcal{Q} \cap \mathcal{D}_t \neq \emptyset\}$ for all $\mathcal{Q} \subseteq [K]$.

Theorem 5.3. Suppose that there exists some $d \in \mathbb{N}$, $s \in (0, 1)$, $\gamma > 1$, and a randomized polynomial time algorithm for the learner such that for all $K, M \in \mathbb{N}$ and for any instance of the multichannel advertising problem, it holds that $|\mathcal{A}_t| \leq M$ for all $t = 1, \dots, T$ and, for any subset $\mathcal{S} \subseteq [K]$,

$$\mathbb{E} \left[\sum_{t \in [T]} r_t(\mathcal{A}_t) \right] \geq (1 - \alpha^{\gamma M}) \sum_{t \in [T]} r_t(\mathcal{S}) - R',$$

Algorithm 5.2: FLE3

Run MSE3 with $L = 2K$ arms and $M := \frac{K}{2} \ln(T/K^2)$.

On each trial $t \in [T]$:

1. Let $\mathcal{A}'_t$ be the output of MSE3
 2. Output $\mathcal{A}_t := \mathcal{A}'_t \cap [K]$
 3. Receive $r_t(\mathcal{A}_t)$ and $\{c_t(i) \mid i \in [K]\}$
 4. For all $i \in [L] \setminus [K]$ set $c_t(i) := 0$
 5. Feed $r_t(\mathcal{A}_t)$ and $\{c_t(i) \mid i \in [L]\}$ back to MSE3
-

where $R' \in \mathcal{O}(K^d T^s)$ and $\alpha := 1 - \frac{1}{|S|}$. Then, for all $\varepsilon \in (0, 1 - 1/\gamma)$ and $B > 4^{1/((1-\varepsilon)\gamma-1)}$, there exists a randomized polynomial-time algorithm for the set cover problem on $[B]$ that, with probability at least $\frac{1}{2}$, achieves approximation ratio at least $(1 - \varepsilon) \ln(B)$.

The proof can be found in Appendix C.2.

5.4 Bandit Facility Location

We can view this setting as a generalization of the marketplace example where sellers pay a known cost to enter the market. At each round, the platform admits a subset $\mathcal{A}_t$ of sellers and only observes the resulting social welfare (bandit feedback).

In this application, there are no restrictions on the set of arms $\mathcal{A}_t$ that we choose. We seek to maximize $r(\mathcal{A}_t) - \gamma(\mathcal{A}_t)$ where r is the sum-max reward function and γ is the linear and positive cost function.

For the facility location problem we must choose M , noting that although a high value of M increases the approximation ratio on the reward, it also increases that on the costs. To decrease the potentially large approximation ratio on the costs, we borrow from Pasteris et al. (2021) the idea of *dummy arms* and the tuning of M . This leads to our algorithm FLE3 described in Algorithm 5.2. The bound on the total profit of FLE3 is given in the following theorem.

Theorem 5.4. *Given that $C = 1$ and $r_t : 2^K \rightarrow [-1, 0]$ is pseudo-concave and pseudo-submodular for all $t \in [T]$, we have that the algorithm FLE3 obtains the following bound for all $\mathcal{S} \subseteq [K]$ with $\mathcal{S} \neq \emptyset$:*

$$\sum_{t \in [T]} \mathbb{E}[\psi_t] \geq \sum_{t \in [T]} \hat{r}_t(\mathcal{S}) - \frac{1}{2} \ln \left(\frac{T}{K^2} \right) \sum_{t \in [T]} \gamma_t(\mathcal{S}) - R'',$$

where $R'' \in \tilde{\mathcal{O}}(K\sqrt{T})$.

Proof. For all $t \in [T]$ define the set function $r'_t : 2^L \rightarrow [0, 1]$ such that for all $\mathcal{Q} \subseteq [L]$, $r'_t(\mathcal{Q}) := r_t(\mathcal{Q} \cap [K])$ and, as consequence, $\hat{r}'_t(\mathcal{Q}) := r'_t(\mathcal{Q}) - r'_t(\emptyset)$. Now taking into consideration any possible comparator set $\mathcal{S} \subseteq [K]$, we define

$$\mathcal{S}' := \mathcal{S} \cup \{K + i \mid i \in [K - |\mathcal{S}|\]\},$$

noting that $|\mathcal{S}'| = K$. Note that r'_t is sum-max and hence, by Theorem 5.1, is pseudo-concave and

submodular for all $t \in [T]$. This allows us to apply Theorem 5.2, that gives us:

$$\begin{aligned} \sum_{t \in [T]} \mathbb{E}[\psi_t] &\geq (1 - \alpha^M) \sum_{t \in [T]} \hat{r}'_t(\mathcal{S}') - \frac{M}{|\mathcal{S}'|} \sum_{t \in [T]} \gamma_t(\mathcal{S}') - R \\ &= (1 - \alpha^M) \sum_{t \in [T]} \hat{r}_t(\mathcal{S}) - \frac{M}{|\mathcal{S}'|} \sum_{t \in [T]} \gamma_t(\mathcal{S}) - R \end{aligned} \quad (5.2)$$

$$= (1 - \alpha^M) \sum_{t \in [T]} \hat{r}_t(\mathcal{S}) - \frac{1}{2} \ln \frac{T}{K^2} \sum_{t \in [T]} \gamma_t(\mathcal{S}) - R, \quad (5.3)$$

where equation (5.2) comes from the contribution of the dummy arms and equation (5.3) from the definition of M . Given that

$$\alpha := \frac{|\mathcal{S}'| - 1}{|\mathcal{S}'|} = \frac{K - 1}{K} \leq \exp(-1/K),$$

we can therefore see that

$$\begin{aligned} \alpha^M \sum_{t \in [T]} \hat{r}_t(\mathcal{S}) &\leq \alpha^M T = \exp(-M/K) T \\ &= \frac{1}{\sqrt{T/K^2}} T = \sqrt{TK^2}, \end{aligned} \quad (5.4)$$

where we used the definition of M given in Algorithm 5.2. Putting together (5.3) and (5.4) gives us the result, where $R'' = R + \sqrt{TK^2}$. $\square$

5.5 Analysis

We now give an overview of the proof of Theorem 5.2.

We first consider the case that we have no costs (i.e. $c_t = \mathbf{0}$). MSE3 works by maintaining a probability distribution over the set of arms. Specifically, $\mathbf{p}_t \in \Delta_K$ is the vector whose components are the probabilities of drawing the actions on trial t . On trial t the algorithm constructs the set $\mathcal{A}_t$ by drawing a sequence $\langle a_t(j) \mid j \in [M] \rangle$ of arms i.i.d. with replacement from $\mathbf{p}_t$ and then setting $\mathcal{A}_t := \{a_t(j) \mid j \in [M]\}$.

This stochastic draw of a sequence and set from a probability vector will be represented by the following notation.

Definition 5.4. For all $\mathbf{q} \in \Delta_K$ let $\langle b_j(\mathbf{q}) \mid j \in [M] \rangle$ be a sequence of stochastic quantities drawn i.i.d. at random from (the probability distribution characterised by) $\mathbf{q}$. In addition, let $\mathcal{B}(\mathbf{q}) := \{b_j(\mathbf{q}) \mid j \in [M]\}$.

Note that our expected reward on trial t is $\mathbb{E}[r_t(\mathcal{B}(\mathbf{p}_t))]$ and hence, for all set functions r we shall construct a differentiable function $\Phi^r : \mathbb{R}^K \rightarrow \mathbb{R}$ such that for all $\mathbf{q} \in \Delta_K$ we have $\Phi^r(\mathbf{q}) = \mathbb{E}[r(\mathcal{B}(\mathbf{q}))]$. This construction is based on the following notion of a *subset decomposition*.

Definition 5.5. Given a function $r : 2^{[K]} \rightarrow \mathbb{R}$, we call a function $\tilde{r} : 2^{[K]} \rightarrow \mathbb{R}$ a subset decomposition of r if and only if for all $\mathcal{Q} \subseteq [K]$ we have

$$r(\mathcal{Q}) = \sum_{\mathcal{S} \subseteq [K]} \mathbb{I}\{\mathcal{Q} \subseteq \mathcal{S}\} \tilde{r}(\mathcal{S}).$$

The following lemma confirms that every set function has a unique subset decomposition.

Lemma 5.1. *Given a function $r : 2^{[K]} \rightarrow \mathbb{R}$ there exists a unique subset decomposition $\tilde{r}$ of r .*

Proof. See Appendix C.1.1. □

Now we can define our function Φ^r .

Definition 5.6. *For all $r : 2^K \rightarrow \mathbb{R}$ and all $\mathbf{q} \in \mathbb{R}^K$ define*

$$\Phi^r(\mathbf{q}) := \sum_{\mathcal{S} \subseteq [K]} \tilde{r}(\mathcal{S}) \left(\sum_{i \in [K]} \mathbb{I}\{i \in \mathcal{S}\} q(i) \right)^M,$$

where, by Lemma 5.1, $\tilde{r}$ is the unique subset decomposition of r .

The following lemma confirms that our function Φ^r indeed satisfies our condition.

Lemma 5.2. *For all $r : 2^K \rightarrow \mathbb{R}$ and all $\mathbf{q} \in \Delta_K$ we have $\Phi^r(\mathbf{q}) = \mathbb{E}[r(\mathcal{B}(\mathbf{q}))]$.*

Proof. See Appendix C.1.2. □

Drawing inspiration from Auer, Freund, and Schapire (2001) we will learn via online exponentiated gradient ascent with the functions Φ^{r_t} using unbiased gradient estimates. Of course, this means that we must be able to construct unbiased gradient estimates. Remarkably, we now show that we can use our sequence $\langle a_t(j) \mid j \in [M] \rangle$ and the observed reward $r_t(\mathcal{A}_t)$ to construct an unbiased gradient estimate $\mathbf{g}_t$ defined in Algorithm 5.1 of the function Φ^{r_t} at $\mathbf{p}_t$.

Lemma 5.3. *For all $r : 2^K \rightarrow \mathbb{R}$, all $\mathbf{q} \in \Delta_K$ and all $i \in [K]$ we have*

$$\partial_i \Phi^r(\mathbf{q}) = \mathbb{E} \left[\frac{r(\mathcal{B}(\mathbf{q}))}{q(i)} \sum_{j \in [M]} \mathbb{I}\{b_j(\mathbf{q}) = i\} \right].$$

Proof. See Appendix C.1.3. □

For exponentiated gradient ascent to work, we must have that, for all trials t , our objective function Φ^{r_t} is concave over the simplex. We now show that a sufficient condition for this to hold is that the function r_t is pseudo-concave.

Lemma 5.4. *For all pseudo-concave set functions $r : 2^K \rightarrow \mathbb{R}$ we have that Φ^r is concave over the simplex Δ_K .*

Proof. See Appendix C.1.4. □

Now that we have all the underpinnings for exponentiated gradient ascent to function properly, we can establish a bound on the regret relative to any vector $\mathbf{p}^* \in \Delta_K$ via the following classic result.

Lemma 5.5. *For any vector $\mathbf{p}^* \in \Delta_K$ we have*

$$\begin{aligned} \sum_{t \in [T]} (\mathbf{p}^* - \mathbf{p}_t) \cdot \mathbf{g}_t &\leq \frac{1}{\eta} \sum_{i \in [K]} p^*(i) \ln(K p^*(i)) \\ &\quad + \eta \sum_{t \in [T]} \sum_{i \in [K]} p_t(i) g_t(i)^2. \end{aligned}$$

Proof. A classic result from the analysis of HEDGE. $\square$

This lemma gives a bound on the regret since, because we have shown that $\mathbf{g}_t$ is an unbiased estimate of the gradient and the objective function is concave over the simplex, the term $(\mathbf{p}^* - \mathbf{p}_t) \cdot \mathbf{g}_t$ is bounded below by $\Phi^{r_t}(\mathbf{p}^*) - \Phi^{r_t}(\mathbf{p}_t)$. Note that we have shown above that $\Phi^{r_t}(\mathbf{p}_t)$ is equal to $\mathbb{E}[r_t(\mathcal{A}_t)]$.

We will later discuss the bounding of the regret itself, but first we shall show how to choose $\mathbf{p}^*$ such that we can bound $\Phi^{r_t}(\mathbf{p}^*)$ relative to $r_t(\mathcal{S})$ for some set $\mathcal{S} \subseteq [K]$. Specifically, we will choose $\mathbf{p}^*$ equal to $\mathbf{p}^{\mathcal{S}}$ in the following definition.

Definition 5.7. For all $\mathcal{S} \subseteq [K]$ with $\mathcal{S} \neq \emptyset$ define $\mathbf{p}^{\mathcal{S}} \in \Delta_K$ such that for all $i \in [K]$ we have

$$p^{\mathcal{S}}(i) := \frac{\mathbb{I}\{i \in \mathcal{S}\}}{|\mathcal{S}|}.$$

We use the following lemma will to bound $\Phi^{r_t}(\mathbf{p}^{\mathcal{S}})$, and it explains why we require r_t to be pseudo-submodular.

Lemma 5.6. Let $\mathcal{S} \subseteq [K]$ with $\mathcal{S} \neq \emptyset$, $r : 2^{[K]} \rightarrow \mathbb{R}$ be a pseudo-submodular function, and $\mathcal{Z} \subseteq [K]$ be a set formed by drawing M elements uniformly at random (with replacement) from $\mathcal{S}$. Then we have

$$\mathbb{E}[r(\mathcal{Z}) - r(\emptyset)] \geq \left(1 - \left(\frac{|\mathcal{S}| - 1}{|\mathcal{S}|}\right)^M\right) (r(\mathcal{S}) - r(\emptyset)).$$

Proof. See Appendix C.1.5 $\square$

With this lemma in hand, we can now bound $\Phi^{r_t}(\mathbf{p}^{\mathcal{S}})$.

Lemma 5.7. Given any $\mathcal{S} \subseteq [K]$ and any pseudo-submodular set function $r : 2^{[K]} \rightarrow \mathbb{R}$ we have

$$\Phi^r(\mathbf{p}^{\mathcal{S}}) \geq r(\emptyset) + \left(1 - \left(\frac{|\mathcal{S}| - 1}{|\mathcal{S}|}\right)^M\right) (r(\mathcal{S}) - r(\emptyset)).$$

Proof. See Appendix C.1.6. $\square$

Before we bound the regret term, we show how to incorporate the costs, so that $\mathbf{c}_t$ can be non-zero. This is done by choosing, instead of Φ^{r_t} , the objective function Ψ_t defined as follows.

Definition 5.8. For all trials $t \in [T]$ define $\Psi_t : \mathbb{R}^K \rightarrow \mathbb{R}$ such that for all $\mathbf{q} \in \mathbb{R}^K$ we have

$$\Psi_t(\mathbf{q}) := \Phi^{r_t}(\mathbf{q}) - M\mathbf{q} \cdot \mathbf{c}_t.$$

Note that by Lemma 5.2 we have that $\Psi_t(\mathbf{p}_t)$ is a lower bound on the expected profit and by Lemma 5.4 Ψ_t is concave over the simplex. It can hence serve as a surrogate concave objective function.

Lemma 5.3 leads to the following lemma, which confirms that $\mathbf{g}_t$ is an unbiased gradient estimate of Ψ_t at $\mathbf{p}_t$.

Lemma 5.8. For all trials $t \in [T]$ and Ψ_t as defined in Definition 5.8, we have

$$\nabla \Psi_t(\mathbf{p}_t) = \mathbb{E}[\mathbf{g}_t | \mathbf{p}_t],$$

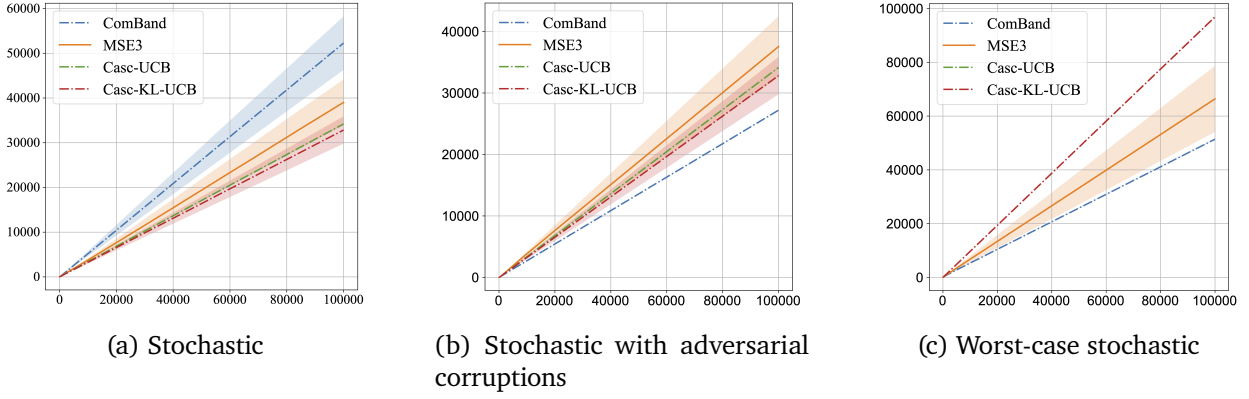


Figure 5.1: Cumulative reward over time in the three environment settings. The results also display the 95% confidence intervals over 35 runs with an Intel Xeon Gold 6312U, calculated using the standard error multiplied by the z -score of 1.96.

Proof. See Appendix C.1.7. □

Now we have shown that our results carry over to the case of non-zero costs, we can finally bound the regret via Lemma 5.5 and the following lemma.

Lemma 5.9. *For all trials $t \in [T]$ we have*

$$\mathbb{E} \left[\sum_{i \in [K]} p_t(i) g_t(i)^2 \right] \leq (1 + C)^2 M(K + M - 1).$$

Proof. See Appendix C.1.8. □

This completes the analysis. Although discussed here, Appendix C.1.9 formally shows how to piece the lemmas together in order to prove Theorem 5.2.

5.6 Experiments

We experimentally evaluated the performance of our method by comparing it against two baselines: CASCADEBANDIT from Kveton et al. (2015) (in both the UCB and KL settings) and COMBAND from Cesa-Bianchi and Lugosi (2012) for M -sized subsets, whose efficient implementation is described in Appendix C.4. We conducted our experiments in various synthetic settings. In each of these environments, a hidden vector $\theta \in \mathbb{R}^K$ is maintained. For each $k \in [K]$, the entry θ_k represents the probability of obtaining a unit reward. These values can be viewed as *attraction probabilities*: the probability that a user clicks on the specific item. After presenting a subset of M elements, the learner gets a unit reward if any of the selected items returns a 1, and 0 otherwise. It is worth emphasizing that our model does not necessitate binary rewards; it offers the flexibility to accommodate any sum-max reward function (as discussed in Section 5.1). The use of a binary reward model is specifically required for comparisons with click models such as CASCADEBANDIT.

Environments for the experiments. We experimentally evaluated our method in three different synthetic environments. We conducted experiments across a wide range of values for K , M , T , and

for the probabilities associated with both optimal and suboptimal arms. In Figure 5.1, we display the cumulative reward over time obtained with $T = 10^5$, $K = 20$, $M = 3$ when the environments are set as follows:

1. **Stochastic** (Figure 5.1a): we randomly select M *good actions* to which we assign a reward probability of 0.3. The reward probabilities of the remaining $k - M$ arms are set to 0.1.
2. **Stochastic with adversarial corruptions** (Figure 5.1b): the rewards are generated as in the stochastic setting. However, in the first $\sqrt{T}$ rounds all good actions have a deterministic reward of 0.
3. **Worst-case stochastic** (Figure 5.1c): this setting is inspired by the lower bound of Cohen, Hazan, and Koren (2017). Here the set $\mathcal{M} \subset [K]$ of M *good actions* is drawn uniformly at random. Then, for each $k \in [K]$, the probabilities are assigned as follows:

$$\theta_k = \begin{cases} X_k + \epsilon & \text{if } k \in \mathcal{M} \\ X_k & \text{otherwise,} \end{cases} \quad \text{where } X_k \sim N\left(\frac{1}{2}, \sigma^2\right),$$

$$\sigma^2 = \frac{1}{192 + 96 \log T} \quad \text{and} \quad \epsilon = \sigma \sqrt{\frac{KM}{8T}}.$$

In Appendix C.5 we present also results obtained varying the subset size M .

Results As expected, our most compelling results were achieved in the adversarial setting, where our approach demonstrated its superiority. In the two stochastic settings, we observed results that were on par with the established baseline methods, affirming the competitiveness of our proposed approach. These findings collectively underscore the effectiveness of our method, particularly in the challenging adversarial context, while also highlighting its versatility in stochastic scenarios. We emphasize that our method is the most efficient one, as each prediction only requires sampling M times from a probability distribution over the K available actions.

Chapter 6

Stochastic Shortest Path under Sparsity

We now transition to a setting with significantly more structure: the Stochastic Shortest Path problem. The underlying structure of a stochastic shortest path is governed by a Markov Decision Process, $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, s_0, g)$, along with a sequence of obliviously selected cost functions. This formalism directly instantiates the ‘question, answer, feedback’ loop used throughout this thesis as follows:

1. *Question*: the learner sees the current environment state $s_t \in \mathcal{S}$.
2. *Answer*: the learner picks an action $a_t \in \mathcal{A}$ from the action set, which causes a transition to a new state s_{t+1} according to the dynamics P .
3. *Feedback*: when the goal is reached, the overall cost function for the episode is revealed (full-information).

The Stochastic Shortest Path (SSP) problem is a fundamental model in reinforcement learning (Bertsekas and Tsitsiklis, 1991, Tarbouriech, Garcelon, Valko, Pirotta, and Lazaric, 2020), which describes tasks where an agent interacts with an environment over episodes and must reach a designated goal state within each episode while minimizing accumulated costs. This covers problems such as car navigation while trying to avoid traffic jams, or internet routing. Recently, this classical setting has been extended to the adversarial regime, where costs may vary arbitrarily between episodes (Rosenberg and Mansour, 2021, Chen, Luo, and Wei, 2021b, Chen and Luo, 2021, Zhao, Li, and Zhou, 2022) and the goal is to obtain theoretical guarantees robust to any cost-generation mechanism. Under *full-information* feedback where the full cost vector is observed after each episode and known transitions, current algorithms achieve regret bounds that scale as $\mathcal{O}(\sqrt{DKT_\star \log SAT_\star})$, where D is the diameter (the smallest expected hitting time of any policy from any state), $T_\star$ is the expected hitting time of the optimal policy, S is the number of states, A is the number of actions, and K is the number of episodes. These bounds are independent of any cost structure and are shown to be minimax optimal up to logarithmic factors in Chen et al. (2021b).

In SSP, the size of the state-action space SA – which we consider and refer to as the *dimension* of the problem – appears in the minimax regret as $\sqrt{\log SA}$. While in the worst-case this is unimprovable (we show this in Theorem 6.1), many real-world problems have costs with structural properties that may be leveraged for improved regret. A common property often considered in the statistics and machine learning literature Wainwright (2019) is sparsity, which can naturally arise for SSP problems. For instance, in the car navigation example, the number of traffic jams is usually much

smaller than the number of roads. Motivated by this, we consider *sparse* SSP problems where M , the maximum number of state-action pairs with non-zero cost in an episode, can be much smaller than SA .

In such scenarios, the regret bounds should capture some dependence on M , reflecting an improvement in performance on easier sparse problems as $M \rightarrow 1$ and recovering the standard bounds on worst-case problems as $M \rightarrow SA$. In fact, for the so-called experts setting* ($S = 1$), the minimax regret scales with $\sqrt{KMA^{-1} \log A}$ instead of $\sqrt{K \log A}$ in the worst-case Kwon and Perchet (2016), providing a polynomial improvement in the dimension A . Furthermore, this is achieved with Online Mirror Descent (OMD) with negative entropy regularization. However for SSP where the problem dimension also includes the size of the state space S , we show that existing approaches also based on OMD with the same negative entropy regularization Chen et al. (2021b) fail to exploit sparse costs. We construct a sparse SSP problem where this algorithm suffers $\sqrt{\log S}$ regret (Theorem 6.1), providing no improvement in terms of S compared to the minimax regret for the non-sparse worst-case $M = SA$ problem. The failure of existing SSP methods to exploit sparsity, let alone match the polynomial improvements from the experts setting, leads us to ask the following questions:

Does sparsity improve the minimax-regret in the full-information feedback SSP problem? How much?

We answer the first question positively for the known-transition case by designing a novel family of regularizers based on ℓ_r -norms for $r \in (1, 2)$ for which we show a $\mathcal{O}(\sqrt{DKT_* \log MT_*})$ regret bound that depends logarithmically on the sparsity level M , rather than on the size of the state-action space SA , without requiring the knowledge of M in advance. This family of regularizers interpolates between the negative entropy and the squared Euclidean norm (see Section 6.4), allowing flexibility for much weaker regularization on sparse points in sparse settings and recovering existing algorithms (and guarantees) in the non-sparse setting.

We show that the above $\sqrt{\log M}$ dependence on M is unimprovable by constructing a matching lower bound. Interestingly, this establishes that the benefit of sparsity in SSP is logarithmic in SA instead of polynomial, as in the simpler experts problem, thus answering the second question. It also highlights that M plays the role of effective dimension, replacing the general dimension SA in controlling the scaling of the minimax regret.

While the benefits of sparsity in SSP are only logarithmic, we emphasize that due to the often combinatorial nature of the state-action space, these improvements can be significant. For instance, in many problems the size of the state-action space grows exponentially in some parameters, while the assigned costs remain linear or even constant. This occurs in many real-world problems (e.g. Surynek, Kumar, and Koenig (2019), Shi, Steenkiste, and Veloso (2019)) in which settings exploiting sparsity can lead to polynomial improvements.

Finally, it is natural to ask whether sparsity may provide similar benefits in the unknown transition setting. However, in Theorem 6.5, we show a lower bound with polynomial dependence on SA in a sparse SSP instance with unknown transitions. This illustrates that in the unknown transition setting the sparsity level M does not play the same role of effective dimension, and that the general dimension SA is crucial in controlling the scaling of the minimax regret polynomially, motivating our focus on the known transitions setting. In particular, this result shows that sparse problems with

*The experts setting is the single-state full-information feedback A -action online learning problem Cesa-Bianchi, Freund, Haussler, Helmbold, Schapire, and Warmuth (1997), Freund and Schapire (1997a).

combinatorial state-action spaces will remain very challenging.

Our results provide a complete characterisation of the benefits of sparsity in removing dimension dependence (i.e. SA) for adversarial SSP problems under full-information feedback.

Contributions We highlight the main contributions of this chapter below:

- We design a novel family of ℓ_r -norm ($r \in (1, 2)$) regularizers for OMD that allows interpolation between the negative entropy and squared Euclidean norm, adjusting its geometry to the sparsity of the cost functions (see Section 6.4). The regularizer naturally plugs into the standard OMD analysis.
- We show OMD with the above regularizer achieves sparsity-adaptive regret bounds of order $\mathcal{O}(\sqrt{DKT_\star \log MT_\star})$ (Theorem 6.2). We also give a parameter-free version achieving the same bound (Theorem 6.3) that does not require prior knowledge of the sparsity level M nor the expected hitting time of the optimal policy $T_\star$ (the only unknown parameters).
- We establish a lower bound of order $\Omega(\sqrt{DKT_\star \log M})$ (Theorem 6.4), matching our regret guarantees up to a logarithmic factor of $T_\star$ (already present in prior work Chen et al. (2021b)) and improving over Chen et al. (2021b) in the $M = SA$ non-sparse setting by including the $\sqrt{\log SA}$ dependence.
- We show that OMD with the negative entropy used in prior work Rosenberg and Mansour (2021), Chen et al. (2021b) suffers regret at least $\Omega(\sqrt{K \log S})$ even when $M = 3$ (Theorem 6.1). This rules out the negative entropy as a viable regularizer in the sparse setting and provides justification for the use of our novel regularizer.
- We establish that results independent of SA are not achievable in the unknown transitions setting via a lower bound in the sparse ($M = 1$) setting of order $\Omega(D\sqrt{SAK})$ (Theorem 6.5).

Technical Contributions: Proving these results requires new technical ideas. For the general sparse lower-bound, we derive a result on the expectation of the maximum of asymmetric zero-mean random walks, generalizing the result for the symmetric case from Orabona and Pál (2015). The negative-entropy-specific lower-bound relies on the careful design of an MDP with skewed initial occupancy measures that highlights both the reasons for the failure of the negative entropy as well as the more general difficulty of the stochastic nature of SSP problems.

6.1 Related works

Regret minimisation for SSP problems under full-information feedback was initiated by a line of work studying stochastic costs Tarbouriech et al. (2020), Rosenberg, Cohen, Mansour, and Kaplan (2020), Tarbouriech, Zhou, Du, Pirodda, Valko, and Lazaric (2021), Cohen, Efroni, Mansour, and Rosenberg (2021), Chen, Jafarnia-Jahromi, Jain, and Luo (2021a), Chen, Jain, and Luo (2022a), Jafarnia-Jahromi, Chen, Jain, and Luo (2023). In the adversarial setting, it was first studied by Rosenberg and Mansour (2021) in the known transition case. Their bounds were later improved by Chen et al. (2021b). There have since been many extensions: Chen and Luo (2021) consider the

unknown transition setting, Zhao et al. (2022) establish dynamic regret bounds, Chen, Luo, and Rosenberg (2022b) consider a policy optimisation approach in the unknown transition setting.

Regret minimisation for SSP problems under bandit feedback where only the costs of the visited state-action pairs in an episode are revealed to the learner has also been studied both in the stochastic Chen et al. (2022b) and adversarial settings Chen et al. (2021b), Chen and Luo (2021), Chen et al. (2022b). In the adversarial known transition setting, the minimax regret is of the order $\sqrt{KDT_*SA}$ (ignoring log terms) Chen et al. (2021b). It is an interesting future direction to study the sparse SSP problem with bandit feedback and understand if the regret scales with $\sqrt{M}$ instead of $\sqrt{SA}$, in which case M would play the same role of effective dimension as in the setting we consider.

Regret minimisation with sparse costs was studied in the classical online learning setting Kwon and Perchet (2016) ($S = 1$). The minimax regret goes from $\mathcal{O}(\sqrt{K \log A})$ to $\mathcal{O}(\sqrt{KMA^{-1} \log A})$ under full-information feedback (experts problem). For rewards instead of costs, it goes from $\mathcal{O}(\sqrt{K \log A})$ to $\mathcal{O}(\sqrt{K \log M})$, which matches the benefits of sparsity we establish for the SSP problem with costs. Note that we restrict our focus to costs since it is unclear how to interpret rewards within the SSP framework. Under bandit feedback, the sparse minimax regret goes from $\tilde{\mathcal{O}}(\sqrt{KA})$ to $\tilde{\mathcal{O}}(\sqrt{KM})$ for both rewards and costs Kwon and Perchet (2016), Bubeck, Cohen, and Li (2018b). The above minimax regrets can also be achieved by sparse-agnostic methods Kwon and Perchet (2016), Tsuchiya, Ito, and Honda (2023). Finally, sparsity was also considered in the case of stochastic losses by Kwon, Perchet, and Vernade (2017).

6.2 Preliminaries

6.2.1 Problem setting

We consider the *Stochastic Shortest Path* (SSP) problem with adversarial costs. The environment is modeled as a Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, s_0, g)$ along with a sequence of cost functions $\{c_k\}_{k=1}^K$ chosen by an oblivious adversary over K episodes. $\mathcal{S}$ is the state space with cardinality $S = |\mathcal{S}|$, and $s_0 \in \mathcal{S}$ is the fixed starting state. The goal state g is a special absorbing state not included in $\mathcal{S}$. $\mathcal{A}$ is the action space with cardinality $A = |\mathcal{A}|$ and we assume for simplicity that it is the same in every state. Let $\Gamma = \mathcal{S} \times \mathcal{A}$ denote the set of all state-action pairs. The dynamics in the MDP are given by the *known* transition function P , where $P(s'|s, a)$ specifies the probability of moving to state $s' \in \mathcal{S} \cup \{g\}$ after taking action a in state s .

Each episode begins in state s_0 and proceeds with the learner selecting actions until the goal state g is reached. When the goal state is reached, the current episode ends and a new one begins. At the start of each episode k , the adversary selects a cost function $c_k : \Gamma \rightarrow [0, 1]$, assigning a cost to each state-action pair. We denote the sparsity level as $M = \max_k \sum_{(s,a) \in \Gamma} \mathbb{I}\{c_k(s, a) > 0\}$ the maximum number of non-zero costs in an episode. We work in the full-information setting where the entire function c_k is revealed to the learner at the end of the episode. The objective is to minimize the total cost over all episodes, which requires a balance of minimizing the accumulated costs while ensuring the goal state is reached efficiently.

We use super-scripts to denote the time-step within an episode and sub-scripts to denote the episode: e.g. (s_k^t, a_k^t) refers to the state-action pair at the t -th time-step of the k -th episode. We sometimes omit the sub-script when referring to an arbitrary episode. We now define some important

concepts:

- A **stationary policy** π is a mapping such that $\pi(\cdot|s)$ is a probability distribution over the choice of action $a \sim \pi(\cdot|s)$ in state s . A policy is called **proper** if it reaches the goal g in finite time from any initial state in $\mathcal{S}$ with probability one, and improper if not. Let Π_p be the set of all stationary proper policies. We assume the existence of at least a proper policy.
- The **expected hitting time** $T^\pi(s)$ is the expected number of steps required to reach g from state s under π . Letting $I_\pi(s)$ be the random number of time-steps used to reach the goal state when executing a policy π in an episode starting from state s , then $T^\pi(s) = \mathbb{E}[I_\pi(s)]$. For any proper policy π , $I_\pi(s)$ and $T^\pi(s)$ are finite for all $s \in \mathcal{S}$.
- The **fast policy** π_f is the deterministic policy that minimizes the worst-state expected hitting time, and the **diameter** D of the MDP is the corresponding expected hitting time:

$$\pi_f = \arg \min_{\pi \in \Pi} \max_{s \in \mathcal{S}} T^\pi(s), \quad D = \max_{s \in \mathcal{S}} T^{\pi_f}(s) = \min_{\pi \in \Pi} \max_{s \in \mathcal{S}} T^\pi(s).$$

Since the transition function P is known, both the fast policy π_f and the diameter D can be computed offline prior to the learning process.

- The **cost-to-go** function $J_c^\pi : \mathcal{S} \rightarrow [0, \infty)$ is the expected cost suffered during an episode executing policy π and starting from state s , given a cost function c and a proper policy π . It is defined as

$$J_c^\pi(s) = \mathbb{E} \left[\sum_{t=1}^{I_\pi(s)} c(s^t, a^t) \middle| P, \pi, s^1 = s \right],$$

where the expectation is with respect to the randomness in the action sampling and state transitions. We use J_k^π to denote the cost-to-go from the initial state s_0 using the cost function c_k in episode k .

- The **regret** R_K is the primary measure of performance by which the learner is evaluated. It is the difference between the total cost over all episodes of the policies $\pi_1, \dots, \pi_K$ chosen by the learner, and the total cost of the best proper deterministic policy in hindsight, $\pi^* \in \arg \min_{\pi \in \Pi_p} \sum_{k=1}^K J_k^\pi$:

$$R_K = \sum_{k=1}^K \sum_{t=1}^{I_{\pi_k}(s_0)} c_k(s_k^t, a_k^t) - \sum_{k=1}^K J_k^{\pi^*},$$

We denote by $T_\star = T^{\pi^*}(s_0)$ the expected hitting time of π^* .

- The **occupancy measure** $q_\pi \in \mathcal{R}_{\geq 0}^\Gamma$ of a proper policy π is the expected number of visits to state-action pairs in an episode executing policy π starting from s_0 :

$$q_\pi(s, a) = \mathbb{E} \left[\sum_{i=1}^{I_\pi(s_0)} \mathbb{I}\{s^i = s, a^i = a\} \middle| P, \pi, s^1 = s_0 \right].$$

The marginal $q_\pi(s) = \sum_{a \in \mathcal{A}} q_\pi(s, a)$ gives the expected number of visits to state s . Given a vector $q \in \mathcal{R}_{\geq 0}^\Gamma$, if it corresponds to a valid occupancy measure, the corresponding policy π_q can be recovered via normalization as $\pi_q(a|s) = q(s, a) / \sum_{a'} q(s, a')$ Zimin and Neu (2013), Rosenberg and Mansour (2021).

6.2.2 SSP as online linear optimisation and online mirror descent

Occupancy measures allow the cost-to-go to be expressed in a linear form:

$$J_k^\pi = \sum_{(s,a) \in \Gamma} q_\pi(s,a) c_k(s,a) = \langle q_\pi, c_k \rangle.$$

If the learner executes a stationary proper policy π_k in episode k , the expected regret can thus be reformulated as an online linear optimisation problem on the space of occupancy measures:

$$\mathbb{E}[R_K] = \mathbb{E}\left[\sum_{k=1}^K J_k^{\pi_k} - J_k^{\pi^*}\right] = \mathbb{E}\left[\sum_{k=1}^K \langle q_{\pi_k} - q_{\pi^*}, c_k \rangle\right].$$

Online linear optimisation is a well studied problem and can be solved using Online Mirror Descent (OMD) (see e.g. Orabona (2019)). In the SSP framework, OMD is applied on the space of occupancy measures corresponding to proper policies with expected hitting time bounded by some $T > 0$ defined as:

$$\Delta(T) = \left\{ q \in \mathbb{R}^\Gamma : \sum_{(s,a) \in \Gamma} q(s,a) \leq T, \right. \\ \left. \forall s \in \mathcal{S}, \sum_{a \in \mathcal{A}} q(s,a) - \sum_{(s',a') \in \Gamma} P(s|s',a') q(s',a') = \mathbb{I}\{s = s_0\} \right\}.$$

The first constraint ensures the expected hitting time is bounded by T , while the second is a flow constraint ensuring the vector corresponds to the occupancy measure of a policy. The regret bounds of OMD will hold against any fixed comparator policy as long as T is large enough such that $\Delta(T)$ contains the occupancy measure of the optimal policy, i.e. $q_{\pi^*} \in \Delta(T)$. OMD with a strictly convex differentiable regularizer ψ and step-size η selects occupancy measures computed through the update

$$q_1 = \arg \min_{q \in \Delta(T)} \psi(q), \quad q_{k+1} = \arg \min_{q \in \Delta(T)} \left\{ \eta \cdot \langle q, c_k \rangle + D_\psi(q, q_k) \right\}, \quad (6.1)$$

where $D_\psi(x, y) = \psi(x) - \psi(y) - \langle \nabla \psi(y), x - y \rangle$ is the Bregman divergence with respect to ψ . This update can be computed efficiently for all the regularizers we will discuss (see Appendix D.3). As discussed in the previous section, we can easily recover via normalization the corresponding policy π_{q_k} that will be executed by the learner.

If the regularizer satisfies for some $\alpha > 0$, any $q \in \mathbb{R}^\Gamma$ and all $k \geq 1$:

$$\nabla \psi(q) \in [\nabla \psi(q_k), \nabla \psi(q_k) - \eta c_k] \implies \nabla^2 \psi(q) \succeq \alpha \nabla^2 \psi(q_k), \quad (6.2)$$

(this is satisfied by many common regularizers), then a standard result (see e.g. Theorem 6 in Bubeck et al. (2018b), Theorem 5.5 in Bubeck and Cesa-Bianchi (2012)) gives the following general regret bound for OMD:

$$\sum_{k=1}^K \langle q_k - q_{\pi^*}, c_k \rangle \leq \underbrace{\frac{\psi(q_{\pi^*}) - \psi(q_1)}{\eta}}_{\text{Penalty}} + \underbrace{\frac{\eta}{2\alpha} \sum_{k=1}^K \|c_k\|_{\nabla^2 \psi(q_k)}^2}_{\text{Stability}} \quad (6.3)$$

where $\|q\|_A^2 = \sum_{s,a,s',a'} q(s,a)A((s,a),(s',a'))q(s',a')$ for a matrix $A \in \mathbb{R}^{\Gamma \times \Gamma}$. Various regret bounds can be obtained by instantiating the above with different regularizers. In particular, Chen et al. (2021b) use the negative-entropy to obtain a $\mathcal{O}(\sqrt{DKT_\star \log SAT_\star})$ regret bound.

6.3 Failure of negative entropy regularization

In the general non-sparse setting, Chen et al. (2021b) use the negative entropy to achieve a regret of $\mathcal{O}(\sqrt{DKT_\star \log SAT_\star})$, which in the non-sparse setting has optimal dependence on SA (as we show later in Theorem 6.3). Despite this success, the negative entropy fails to benefit from sparsity in its dependence on S , as shown by the result below. As we will see in Section 6.4, this establishes the negative entropy as a sub-optimal choice for sparse SSP problems.

Theorem 6.1. *For any $S \geq 6$, there exists an SSP instance with a fixed horizon of 3, sparsity level $M = 3$, an action space of size $A = 2$ and state space of size S such that the regret of OMD (6.1) with negative-entropy regularization and any step-size $\eta > 0$ after K episodes is $\mathbb{E}[R_K] = \Omega(\min\{\sqrt{K \log S}, K\})$.*

This result shows that despite the SSP instance being sparse ($M = 3$), the regret of OMD with negative entropy regularization nevertheless scales as $\sqrt{\log S}$, which is the same dependence on S as in the non-sparse setting. For sparse problems, the negative entropy provides no improvement on the regret with respect to S . This highlights that existing approaches and regularizers are inadequate to appropriately exploit sparse problems and motivates considering alternate regularizers specifically designed for the geometry of sparse problems, as we do in the next section.

To better understand the failure of negative entropy regularization in sparse settings, we highlight the main intuition behind the lower bound construction and defer the details of the proof to Appendix D.2.

Proof intuition: The key idea is to reduce SSP to an experts problem with 2 actions and a heavily skewed initial distribution over the actions. The initial occupancy measure played by OMD in (6.1) is $q_1 = \arg \min_{q \in \Delta(T)} \psi(q)$. For most regularizers, including the negative entropy, this encourages q_1 to be uniform across the state-action space while maintaining the constraints on the flow and expected hitting time. Since we consider a fixed-horizon MDP, only the flow constraint is relevant.

Consider the SSP problem shown in Figure 6.1 with $N = S - 2$. Since $s_1, \dots, s_N$ constitute a large majority of the states (especially for large N), $\psi(q)$ is mainly affected by the values of q in these states. In order to minimize $\psi(q)$, q_1 needs to ensure the expected number of visits to these N states is sufficiently high. However since for any q and any $i \geq 1$, $q(s_i) = \frac{1}{N}q(s_0, a_2)$, for $q_1(s_i)$ to be sufficiently large then $q_1(s_0, a_2)$ needs to be much larger (by a factor of N). This results in q_1 being heavily skewed towards a_2 in s_0 . For the negative entropy, this gives specifically $q_1(s_0, a_1) \approx \frac{1}{\sqrt{N}}$.

If the costs in all states but s_0 are set to 0, the problem is sparse ($M = 2$) and reduces to a experts problem with 2 actions where the initial probability for the first action, which in our case is $q_1(s_0, a_1)$, scales as $1/\sqrt{N}$. The regret for OMD with the negative entropy in this setting can be shown to scale for any step-size at least as $\Omega(\sqrt{K \log N}) = \Omega(\sqrt{K \log S})$, providing the dependence from the statement of the theorem. To prove this formally for the SSP reduction, we use the above

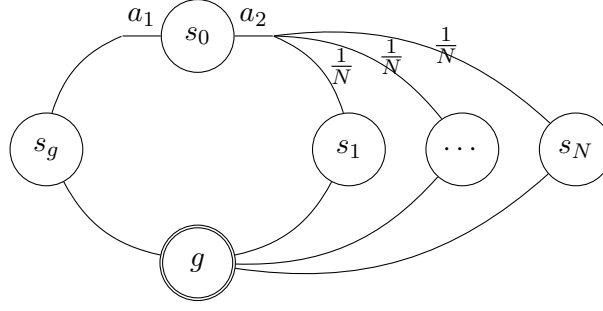


Figure 6.1: MDP for the reduction to a skewed experts problem with 2 actions: $\mathcal{S} = \{s_0, s_g, s_1, \dots, s_N\}$ ($N = S - 2$), $\mathcal{A} = \{a_1, a_2\}$. The transitions are given by $p(s_g|s_0, a_1) = 1, p(g|s_g, a) = 1$ for all $a \in \mathcal{A}$, for $i \geq 1$: $p(s_i|s_0, a_2) = 1/N, p(g|s_i, a) = 1$ for all $a \in \mathcal{A}$.

construction coupled with a non-skewed reduction and careful setting of the costs. We include the details in Appendix D.2.

Finally, we remark that this failure comes from the negative entropy inducing an OMD update at points near the boundary of the space that can only move to new points in very close proximity. This property does not generalize to all regularizers and in fact provides insights for designing a regularizer to appropriately handle sparsity. In particular, the regularizer we consider in the next section does not suffer from the same issue because it induces an OMD update that can relatively easily move away from points that are near or exactly on the boundary of the space. This is because its gradient does not diverge at the boundary (i.e. as $q(s, a) \rightarrow 0$) unlike the negative entropy.

6.4 The benefits of sparsity

In this section, we show that it is possible to achieve a regret bound of order $\mathcal{O}(\sqrt{DKT_\star \log(MT_\star)})$, where M is the maximum number of non-zero entries in the cost. This is our main result and together with the lower bound in Theorem 6.4 establishes that the sparsity level M acts as a measure of effective dimension instead of the state-action space size SA for SSP with full-information feedback.

In the previous section, we showed and discussed that the negative entropy, the regularizer used in OMD by existing methods, is inadequate to handle sparse SSP problems. Motivated by this failure, we consider alternate regularizers. However, identifying a suitable regularizer poses two key challenges. Firstly, it must work for SSP and the associated technical complexities compared to other simpler online learning problems. In particular, it needs to match the dependence in terms of the other non-sparsity-related quantities appearing in the regret of the negative entropy (i.e. $D, T_\star, K$). Second, it must explicitly leverage sparsity to improve performance. We propose the following novel family of regularizers parameterised by $p > 1$:

$$\psi_p(q) = p \cdot \left(-1 + \|q\|_{1+1/p}^{1+1/p} \right) = p \cdot \left(-1 + \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} |q(s, a)|^{1+1/p} \right). \quad (6.4)$$

As $p \rightarrow \infty$, the regularizer in Equation (6.4) converges to the negative entropy. On the other hand, as $p \rightarrow 1$, the regularizer converges to the squared Euclidean norm that enforces much weaker regularization on sparse points. Therefore, ψ_p allows smooth interpolation between dense and sparse regimes via the tunable parameter p (see Figure 6.2 for a comparison). In particular, ψ_p for small

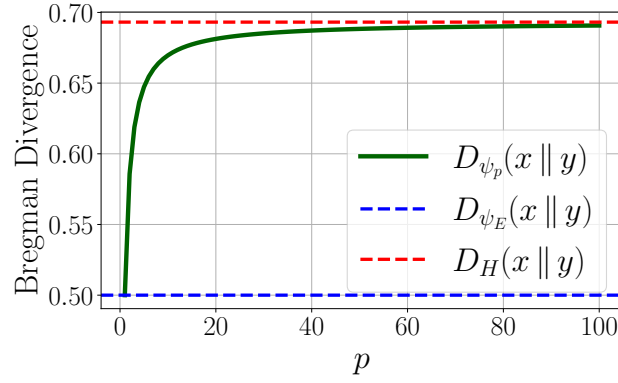


Figure 6.2: Bregman Divergence between a deterministic distribution $x = [0, 1]$ and the uniform distribution $y = [1/2, 1/2]$ for our regularizer ψ_p , squared Euclidean norm ψ_E and negative entropy H for increasing values of p .

p induces an OMD update that is able to easily move to and away from the boundary of the space, making it robust to the skewed initial occupancy measure on the SSP instance which caused the failure of the negative entropy in Section 6.3. The parameter p also controls a trade-off between the stability and penalty term in (6.3), which ultimately will enable the removal of the dependence on SA .

We note that OMD with the above regularizer can be implemented efficiently for any p : the projection step of OMD over $\Delta(T)$ can be written as a convex optimization problem as in Rosenberg and Mansour (2021), which can be solved efficiently (details in Appendix D.3).

We can now turn to our main result, which establishes regret bounds that scale with the sparsity level M for OMD with the regularizer in (6.4) when M is known.

Theorem 6.2. *Consider OMD with ψ_p as regularizer. If T is such that $q_{\pi^*} \in \Delta(T)$, $\eta = \sqrt{\frac{2pT^{1+1/p}}{KDM^{1/p}}}$, $p = 1 + \log(TM)$, then $\mathbb{E}[R_K] \leq \mathcal{O}\left(\sqrt{DKT \log(MT)}\right)$.*

We present below the outline of the proof and include the missing details in Appendix D.1.

Proof. It can be shown that ψ_p satisfies the condition (6.2) with $\alpha = 1$, allowing us to use the bound in (6.3) as a starting point. Using that $q_\star \in \Delta(T)$, we can bound the Penalty term:

$$\psi_p(q_\star) - \psi_p(q_1) = p \left(\|q_\star\|_{1+1/p}^{1+1/p} - \|q_1\|_{1+1/p}^{1+1/p} \right) \leq p \cdot \|q_\star\|_1^{1+1/p} \leq p \cdot T^{1+1/p} \quad (6.5)$$

It can also be shown that $\nabla^2 \psi_p(q)^{-1} = \text{diag}\left(\frac{p}{p+1} q^{1-1/p}\right)$. Using that $c_k(s, a)^2 \leq c_k(s, a)$, we get

$$\begin{aligned} \|c_k\|_{\nabla^2 \psi_p(q_k)^{-1}}^2 &\leq \frac{p}{p+1} \sum_{s,a} c_k(s, a) q_k(s, a)^{1-1/p} \leq \|c_k\|_1 \sum_{s,a} \frac{c_k(s, a)}{\|c_k\|_1} q_k(s, a)^{1-1/p} \\ &\leq \|c_k\|_1 \left(\sum_{s,a} \frac{c_k(s, a)}{\|c_k\|_1} q_k(s, a) \right)^{1-1/p} \\ &= \|c_k\|_1^{1/p} \langle c_k, q_k \rangle^{1-1/p} \leq M^{1/p} \max\{1, \langle c_k, q_k \rangle\} \leq M^{1/p} (1 + \langle c_k, q_k \rangle), \end{aligned} \quad (6.5)$$

where the key step (6.5) uses Jensen's inequality on the concave function $x^{1-1/p}$ ($p > 1$) and

probability distribution $c_k/\|c_k\|_1$. Plugging this into the Stability term and combining with (6.5):

$$\begin{aligned} \sum_{k=1}^K \langle q_k - q_{\pi^*}, c_k \rangle &\leq \frac{pT^{1+1/p}}{\eta} + \frac{\eta M^{1/p} K}{2} + \frac{\eta M^{1/p}}{2} \sum_{k=1}^K \langle q_k, c_k \rangle \\ \implies \sum_{k=1}^K \langle q_k - q_{\pi^*}, c_k \rangle &\leq \frac{1}{1 - \frac{\eta M^{1/p}}{2}} \left(\frac{pT^{1+1/p}}{\eta} + \frac{\eta M^{1/p} K}{2} + \frac{\eta M^{1/p}}{2} \sum_{k=1}^K \langle q_k, c_k \rangle \right) \\ \implies \sum_{k=1}^K \langle q_k - q_{\pi^*}, c_k \rangle &\leq \frac{2pT^{1+1/p}}{\eta} + \eta M^{1/p} DK, \end{aligned}$$

where the last step uses that $\sum_{k=1}^K \langle q_{\pi^*}, c_k \rangle \leq \sum_{k=1}^K \langle q_{\pi_f}, c_k \rangle \leq K \|q_{\pi_f}\|_1 \|c_k\|_\infty \leq DK$, $D \geq 1$ and $\eta \leq 4M^{-1/p} \iff 1 - \eta M^{1/p}/2 \geq 1/2$. Tuning $\eta = \sqrt{\frac{2pT^{1+1/p}}{KDM^{1/p}}}$ (so $\eta \leq 2M^{-1/p}$ for sufficiently large K) and $p = 1 + \log(TM)$ gives the result. $\square$

Provided we can suitably select $T \approx T_*$ (see Section 6.4.1), this result establishes that sparsity does lead to an improvement in the minimax regret. In Section 6.4.2, we show that the dependence on M and SA is optimal, ruling out polynomial improvements from sparsity such as in the experts setting Kwon and Perchet (2016). This highlights that M acts as the effective dimension of the problem instead of SA . In particular, if the sparsity level M is constant, then we obtain a dimension-independent regret of $\mathcal{O}(DKT \log T)$.

Remark 6.1. Although we express the bound in terms of the sparsity level M , it can be seen that the analysis above holds more generally if M is instead an upper bound on the ℓ_1 norm of the costs: $M = \max_k \|c_k\|_1$. This relaxation allows our result to cover "softly sparse" cost structures and aligns with the notion of first-order bounds commonly studied in the online learning literature (Neu, 2015, Wei and Luo, 2018, Wagenmaker, Chen, Simchowitz, Du, and Jamieson, 2022).

6.4.1 Sparse-agnostic parameter-free upper bound

The procedure in Theorem 6.3 assumes knowledge of the sparsity level M to tune the parameter p of our regularizer and uses knowledge of the expected hitting time of the optimal policy T_* to consider OMD over the space of suitable occupancy measures. We now adapt existing techniques to remove both of these assumptions and derive fully parameter-free guarantees.

For the unknown sparsity level, we use the same approach as in Kwon and Perchet (2016). We divide the K episodes into batches. Within each batch, we independently run OMD tuning the parameter p of our regularizer with the sparsity level observed up to the current batch, as described in Algorithm D.1 in Appendix D.1.

For the unknown expected hitting time of the optimal policy T_* , we can exploit the same meta-algorithm technique as in Chen et al. (2021b), using the sparse agnostic algorithms introduced above as base learners. We run $N \approx \log K$ instances of Algorithm D.1 where the j -th instance sets its parameter T as $b(j) \approx 2^j$. Therefore, there exists a good instance j_* such that $b(j_*)$ is close to the unknown T_* . The regret of a scale-invariant meta-algorithm, described for completeness in Algorithm D.2 in Appendix D.1, closely matches that of this good instance.

Together, these two techniques yield the following parameter-free regret bound (proof in Appendix D.1):

Theorem 6.3. *For any $K \geq MT$, Algorithm D.2 guarantees:*

$$\mathbb{E}[R_K] \leq \mathcal{O} \left(B\sqrt{DT_\star K \log(T_\star M)} + BM\sqrt{\frac{DT_\star \log(T_\star M)}{K}} + \sqrt{DT_\star K \log N \sqrt{T_\star}} \right),$$

where $B = \lceil \log_2 \log_2 M \rceil$ and $N = \mathcal{O}(\log K)$.

The leading term matches the regret bound from Theorem 6.2 up to a $\log \log$ term, while the second vanishes with large K and the third does not depend on M . Therefore, running a procedure that does not assume knowledge of M and $T_\star$ comes at no additional cost in terms of the regret bound. We note that we can also obtain a bound that holds with high-probability since the high-probability analysis given in Chen et al. (2021b) can easily be adapted to work with our regularizer.

6.4.2 Lower bound

In this section, we provide a general lower bound for sparse SSP problems.

Theorem 6.4. *For any $D, T_\star, K, S, A$ with $T^\star \geq D \geq 3 \log S$, $S(A-1) \geq 400$, $K \geq \frac{800T^\star}{D} \log M$ and $M \geq 101$, there exists an SSP instance with stochastic M -sparse costs, S states and A actions such that its diameter is D , the expected hitting time of the optimal policy is $T^\star$, and the expected regret with respect to the randomness of the losses for any learner after K episodes is $\mathbb{E}[R_K] \geq \Omega(\sqrt{KT^\star D \log M})$.*

For general M (> 100), the lower bound matches the upper bound established in Theorem 6.2 in its dependence on M , characterizing the minimax regret for general sparse problems (up to a $\log T_\star$ term). For $M = SA$, our result gives a $\Omega(\sqrt{KT^\star D \log SA})$ lower bound improving on the $\Omega(\sqrt{KT^\star D})$ lower bound of Chen et al. (2021b). In particular, this establishes the optimal dependence on the size of the state-action space SA in the minimax regret for the general non-sparse SSP problem.

Proof intuition: The proof is based on the combination of an SSP instance from Chen et al. (2021b) and a probabilistic costs construction, which then requires some non-trivial arguments to extend to the sparse SSP problem. We give an overview of the construction and defer the details to Appendix D.4.

The MDP construction is essentially a reduction to a non-sparse experts problem with $\mathcal{O}(M)$ actions. First, there is a reduction to an experts problem with $\mathcal{O}(SA)$ actions. Then within these, there are $\mathcal{O}(M)$ good actions, while the remaining are bad. The good actions suffer small costs in expectation and can lead directly to the goal-state. The bad actions are zero-cost but all lead to the same unique bad state, where only one action leads to the goal-state and suffers high cost. This allows a big proportion of the actions to be bad while still guaranteeing sparsity and forcing the learner to only consider the $\mathcal{O}(M)$ good actions, completing the reduction to the non-sparse experts problem with $\mathcal{O}(M)$ actions.

However, we cannot directly apply lower bounds for the experts problem because of subtleties in the reduction and the cost-generating mechanism. We use a similar approach to the experts lower bounds by sampling the costs i.i.d. from a Bernoulli distribution, however with a scaled parameter to ensure the reduction above holds. The regret in this stochastic environment can then be expressed as the maximum of asymmetric zero-mean i.i.d. random walks, capturing how much better the optimal policy can be by choosing the best action after the i.i.d. Bernoulli costs have been sampled for all episodes. The result then follows from a technical result on the expectation of this maximum that we derive in Appendix D.6. We note that the reduction and costs are constructed in such a way that the diameter of the MDP and expected hitting time of the optimal policy are indeed D and T_* .

6.5 Unknown transition setting

In this section, we consider the setting where the transitions are unknown and show through the following lower bound that the benefits of sparsity are limited.

Theorem 6.5. *For any D, K, S, A with $S \geq 2, A \geq 16, D \geq 2$ and $K \geq SA$, there exists an SSP instance with $M = 1$, S states and A actions such that its diameter is D and the expected regret for any learner without knowledge of the transitions after K episodes is $\mathbb{E}[R_K] \geq \Omega(D\sqrt{SAK})$.*

The above result establishes that the minimax regret for the unknown transition setting must scale polynomially with SA , regardless of the sparsity. In particular, this highlights the limited benefits of sparsity in removing the dependence on the state-action space size in the unknown transition setting, which is in stark contrast to the known transition setting.

The proof is based on an SSP instance used by Rosenberg et al. (2020) to prove an $\Omega(D\sqrt{SAK})$ lower bound in the unknown transition non-sparse setting. It turns out that this instance can be adapted such that the cost is only non-zero for a single state-action pair, while keeping the regret lower bound unchanged, giving the above result. We include the details in Appendix D.5.

6.6 Conclusion, limitations and future-work

In this chapter, we studied the SSP problem under sparse adversarial costs and full-information feedback. When the transitions are known, we have shown that existing methods fail to appropriately exploit sparsity. Instead, we designed a family of regularizers to use with Online Mirror Descent that allowed us to characterize the sparse minimax regret, establishing the extent of the benefits of sparsity in this setting. When the transitions are unknown, we showed that even the sparse minimax regret scales polynomially in the size of the state-action space, suggesting fundamental limits in such settings.

Our results open up many further directions of research. In particular, we established the benefits of sparsity under known transition as limited to logarithmic, however, there could be structural properties of an MDP that could break this logarithmic limit and achieve polynomial benefits. Moreover, we have limited our focus to the adversarial full-information feedback setting, but the study of sparse SSP problems in other settings, such as partial feedback, stochastic environments, or structured decision problems remains unexplored.

Bibliography

- Alekh Agarwal, Haipeng Luo, Behnam Neyshabur, and Robert E Schapire. Corraling a band of bandit algorithms. In *Conference on Learning Theory*, pages 12–38. PMLR, 2017. URL <http://proceedings.mlr.press/v65/agarwal17b/agarwal17b.pdf>.
- Morteza Alamgir and Ulrike von Luxburg. Phase transition in the family of p -resistances. In *Proc. NIPS*, pages 379–387, 2011.
- Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.
- Peter Auer, Yoav Freund, and Robert E. Schapire. The non-stochastic multi-armed bandit problem. 2001. URL <https://api.semanticscholar.org/CorpusID:7732525>.
- Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, 2002.
- Peter Auer, Pratik Gajane, and Ronald Ortner. Adaptively tracking the best arm with an unknown number of distribution changes. In *European Workshop on Reinforcement Learning*, volume 14, page 375, 2018.
- Amir Beck. *First-Order Methods in Optimization*. SIAM, 2017.
- Dimitri P Bertsekas and John N Tsitsiklis. An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, 16(3):580–595, 1991.
- Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.*, 2008(10):P10008, 2008.
- Avrim Blum and Yishay Mansour. From external to internal regret. *J. Mach. Learn. Res.*, 8(6), 2007.
- A. V. Boyd. Inequalities for mills’ ratio. *Reports of Statistical Application Research (Union of Japanese Scientists and Engineers)*, 6(1959):44–46, 1959.
- Marco Bressan, Nicolò Cesa-Bianchi, Silvio Lattanzi, and Andrea Paudice. Exact recovery of clusters in finite metric spaces using oracle queries. In *Proc. COLT*, 2021.
- Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.
- Sébastien Bubeck, Michael Cohen, and Yuanzhi Li. Sparsity, variance and curvature in multi-armed bandits. In *Algorithmic Learning Theory*, pages 111–127. PMLR, 2018a.

- Sébastien Bubeck, Michael Cohen, and Yuanzhi Li. Sparsity, variance and curvature in multi-armed bandits. *Proceedings of Algorithmic Learning Theory*, 83:111–127, 2018b.
- Nicolò Campolongo and Francesco Orabona. A closer look at temporal variability in dynamic online learning, 2021. URL <https://arxiv.org/abs/2102.07666>.
- Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.
- Nicolo Cesa-Bianchi and Gábor Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012.
- Nicolò Cesa-Bianchi, Yoav Freund, David P. Helmbold, David Haussler, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, pages 382–391, 1993.
- Nicolo Cesa-Bianchi, Yoav Freund, David Haussler, David P Helmbold, Robert E Schapire, and Manfred K Warmuth. How to use expert advice. *Journal of the ACM (JACM)*, 44(3):427–485, 1997.
- Nicolo Cesa-Bianchi, Claudio Gentile, and Giovanni Zappella. A gang of bandits. In *Proc. NIPS*, volume 26, 2013.
- Liyu Chen and Haipeng Luo. Finding the stochastic shortest path with low regret: the adversarial cost and unknown transition case. *Proceedings of the 38th International Conference on Machine Learning*, 139:1651–1660, 2021.
- Liyu Chen, Mehdi Jafarnia-Jahromi, Rahul Jain, and Haipeng Luo. Implicit finite-horizon approximation and efficient optimal algorithms for stochastic shortest path. *Advances in Neural Information Processing Systems*, 34:10849–10861, 2021a.
- Liyu Chen, Haipeng Luo, and Chen-Yu Wei. Minimax regret for stochastic shortest path with adversarial costs and known transition. *Proceedings of the 34th Conference on Learning Theory*, 134: 1180–1215, 2021b.
- Liyu Chen, Rahul Jain, and Haipeng Luo. Improved no-regret algorithms for stochastic shortest path with linear MDP. *Proceedings of the 39th International Conference on Machine Learning*, 162: 3204–3245, 2022a.
- Liyu Chen, Haipeng Luo, and Aviv Rosenberg. Policy optimization for stochastic shortest path. *Proceedings of the 35th Conference on Learning Theory*, 178:982–1046, 2022b.
- Wang Chi Cheung, David Simchi-Levi, and Ruihao Zhu. Learning to optimize under non-stationarity. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1079–1087. PMLR, 2019.
- C Chow. On optimum recognition error and reject tradeoff. *IEEE Trans. Inf. Theory*, 16(1):41–46, 1970.
- Chi-Keung Chow. An optimum character recognition system using decision functions. *IRE Trans. Electron. Comput.*, EC-6(4):247–254, 1957.

- Alon Cohen, Tamir Hazan, and Tomer Koren. Tight bounds for bandit combinatorial optimization. In *Conference on Learning Theory*, pages 629–642. PMLR, 2017.
- Alon Cohen, Yonathan Efroni, Yishay Mansour, and Aviv Rosenberg. Minimax regret for stochastic shortest path. *Advances in Neural Information Processing Systems*, 34:28350–28361, 2021.
- Corinna Cortes, Giulia DeSalvo, Claudio Gentile, Mehryar Mohri, and Scott Yang. Online learning with abstention. In *Proc. ICML*, pages 1059–1067, 2018.
- Corinna Cortes, Giulia DeSalvo, Claudio Gentile, Mehryar Mohri, and Ningshan Zhang. Online learning with dependent stochastic feedback graphs. In *Proc. ICML*, pages 2154–2163, 2020.
- Ashok Cutkosky. Combining online learning guarantees. In Alina Beygelzimer and Daniel Hsu, editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 895–913, Phoenix, USA, 2019. PMLR. URL <http://proceedings.mlr.press/v99/cutkosky19b.html>.
- Ashok Cutkosky. Parameter-free, dynamic, and strongly-adaptive online learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2250–2259, Virtual, 2020. PMLR. URL <http://proceedings.mlr.press/v119/cutkosky20a.html>.
- Ashok Cutkosky and Francesco Orabona. Black-box reductions for parameter-free online learning in banach spaces. In *Conference On Learning Theory*, pages 1493–1529. PMLR, 2018.
- Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 624–633, 2014.
- Benjamin Doerr. An elementary analysis of the probability that a binomial random variable exceeds its expectation. *Statistics & Probability Letters*, 139:67–74, 2018.
- Peter G Doyle and J Laurie Snell. *Random Walks and Electric Networks*, volume 22 of *The Carus Mathematical Monographs*. American Mathematical Society, 1984.
- Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. *arXiv preprint cs/0408007*, 2004.
- Robert W Floyd. Algorithm 97: shortest path. *Commun. ACM*, 5(6):345, 1962.
- Santo Fortunato. Community detection in graphs. *Phys. Rep.*, 486(3):75–174, 2010.
- Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997a.
- Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997b.
- Yoav Freund, Robert E Schapire, Yoram Singer, and Manfred K Warmuth. Using and combining predictors that specialize. In *Proc. STOC*, pages 334–343, 1997.

- Pierre Gaillard, Gilles Stoltz, and Tim Van Erven. A second-order bound with excess losses. In *Proc. COLT*, pages 176–196, 2014.
- Armengol Gasull and Frederic Utzet. Approximating mills ratio. *Journal of Mathematical Analysis and Applications*, 420(2):1832–1853, 2014.
- Ralph E Gomory and Tien Chung Hu. Multi-terminal network flows. *J. Soc. Ind. Appl. Math.*, 9(4): 551–570, 1961.
- David Harvey and Joris Van Der Hoeven. Integer multiplication in time $o(n \log n)$. *Annals of Mathematics*, 193(2):563–617, 2021.
- Kilian Hendrickx, Lorenzo Perini, Dries Van der Plas, Wannes Meert, and Jesse Davis. Machine learning with a reject option: A survey. *arXiv preprint arXiv:2107.11277*, 2021.
- Mark Herbster and Guy Lever. Predicting the labelling of a graph via minimum p -seminorm interpolation. In *Proc. COLT*, 2009.
- Mark Herbster and Manfred K Warmuth. Tracking the best regressor. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 24–31, 1998. URL <https://dl.acm.org/doi/pdf/10.1145/279943.279949>.
- Mark Herbster and Manfred K Warmuth. Tracking the best linear predictor. *Journal of Machine Learning Research*, 1(281-309):10–1162, 2001. URL <https://www.jmlr.org/papers/volume1/herbster01a/herbster01a.pdf>.
- Mark Herbster, Stephen Pasteris, Fabio Vitale, and Massimiliano Pontil. A gang of adversarial bandits. In *Proc. NeurIPS*, pages 2265–2279, 2021.
- Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Soc. Netw.*, 5(2):109–137, 1983.
- Andrew Jacobsen and Ashok Cutkosky. Parameter-free mirror descent. In *Conference on Learning Theory*, pages 4160–4211. PMLR, 2022.
- Andrew Jacobsen and Ashok Cutkosky. Unconstrained online learning with unbounded losses. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 14590–14630. PMLR, 2023. URL <https://proceedings.mlr.press/v202/jacobsen23a.html>.
- Andrew Jacobsen and Ashok Cutkosky. Online linear regression in dynamic environments via discounting. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 21083–21120. PMLR, 2024. URL <https://proceedings.mlr.press/v235/jacobsen24a.html>.
- Mehdi Jafarnia-Jahromi, Liyu Chen, Rahul Jain, and Haipeng Luo. Posterior sampling-based online learning for the stochastic shortest path model. *Proceedings of the 39th Conference on Uncertainty in Artificial Intelligence*, 216:922–931, 2023.

- Emmeran Johnson, Alberto Rumi, Ciara Pike-Burke, and Patrick Rebeschini. Stochastic shortest path with sparse adversarial costs. In *Eighteenth European Workshop on Reinforcement Learning*, 2025.
- Yuko Kuroki, Alberto Rumi, Taira Tsuchiya, Fabio Vitale, and Nicolò Cesa-Bianchi. Best-of-both-worlds algorithms for linear contextual bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 1216–1224. PMLR, 2024.
- Branislav Kveton, Csaba Szepesvari, Zheng Wen, and Azin Ashkan. Cascading bandits: Learning to rank in the cascade model. In *International conference on machine learning*, pages 767–776. PMLR, 2015.
- Joon Kwon and Vianney Perchet. Gains and losses are fundamentally different in regret minimization: The sparse case. *Journal of Machine Learning Research*, 17(227):1–32, 2016.
- Joon Kwon, Vianney Perchet, and Claire Vernade. Sparse stochastic bandits. In *Proceedings of the 30th Conference on Learning Theory*, volume 65, pages 1269–1270, 2017.
- Tommaso Lanciano, Atsushi Miyauchi, Adriano Fazzzone, and Francesco Bonchi. A survey on the densest subgraph problem and its variants. *ACM Comput. Surv.*, 56(8):1–40, 2024.
- Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.
- Tor Lattimore, Branislav Kveton, Shuai Li, and Csaba Szepesvari. Toprank: A practical algorithm for online stochastic ranking. *Advances in Neural Information Processing Systems*, 31, 2018.
- Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. <http://snap.stanford.edu/data>, June 2014.
- Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.
- Haipeng Luo and Robert E Schapire. Achieving all with no parameters: Adanormalhedge. In *Proc. COLT*, pages 1286–1304, 2015.
- Haipeng Luo, Mengxiao Zhang, Peng Zhao, and Zhi-Hua Zhou. Corraling a larger band of bandits: A case study on switching regret for linear bandits. In *Conference on Learning Theory*, pages 3635–3684. PMLR, 2022.
- Teodor V. Marinov and Julian Zimmert. The pareto frontier of model selection for general contextual bandits, 2021. URL <https://arxiv.org/abs/2110.13282>.
- Brendan D McKay. On littlewood’s estimate for the binomial distribution. *Advances in Applied Probability*, 21(2):475–478, 1989.
- Brendan McMahan and Jacob Abernethy. Minimax optimal algorithms for unconstrained linear optimization. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL <https://proceedings.neurips.cc/paper/2013/file/e00406144c1e7e35240afed70f34166a-Paper.pdf>.

- Brendan McMahan and Matthew Streeter. No-regret algorithms for unconstrained online convex optimization. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL <https://proceedings.neurips.cc/paper/2012/hash/38ca89564b2259401518960f7a06f94b-Abstract.html>.
- H. Brendan McMahan and Francesco Orabona. Unconstrained online linear learning in hilbert spaces: Minimax algorithms and normal approximations. In Maria Florina Balcan, Vitaly Feldman, and Csaba Szepesvári, editors, *Proceedings of The 27th Conference on Learning Theory*, volume 35 of *Proceedings of Machine Learning Research*, pages 1020–1039, Barcelona, Spain, 13–15 Jun 2014. PMLR. URL <http://proceedings.mlr.press/v35/mcmahan14.html>.
- Zakaria Mhammedi and Wouter M. Koolen. Lipschitz and comparator-norm adaptivity in online learning. In Jacob Abernethy and Shivani Agarwal, editors, *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 2858–2887. PMLR, 09–12 Jul 2020. URL <http://proceedings.mlr.press/v125/mhammedi20a.html>.
- Aryan Mokhtari, Shahin Shahrampour, Ali Jadbabaie, and Alejandro Ribeiro. Online optimization in dynamic environments: Improved regret rates for strongly convex problems. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 7195–7201. IEEE, 2016.
- Gergely Neu. First-order regret bounds for combinatorial semi-bandits. *Proceedings of the 28th Conference on Learning Theory*, 40:1360–1375, 2015.
- Gergely Neu and Nikita Zhivotovskiy. Fast rates for online prediction with abstention. In *Proc. COLT*, pages 3030–3048, 2020.
- Mark E.J. Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69(2):026113, 2004.
- Rad Niazadeh, Negin Golrezaei, Joshua R Wang, Fransisca Susan, and Ashwinkumar Badanidiyuru. Online learning via offline greedy algorithms: Applications in market design and optimization. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 737–738, 2021.
- Maillard Odalric and Rémi Munos. Adaptive bandits: Towards the best history-dependent strategy. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 570–578. JMLR Workshop and Conference Proceedings, 2011.
- Francesco Orabona. A modern introduction to online learning. *CoRR*, abs/1912.13213, 2019. URL <http://arxiv.org/abs/1912.13213>.
- Francesco Orabona and Dávid Pál. Optimal non-asymptotic lower bound on the minimax regret of learning with expert advice. *Preprint arXiv:1511.02176*, 2015.
- Francesco Orabona and Dávid Pál. Coin betting and parameter-free online learning. *Advances in Neural Information Processing Systems*, 29, 2016.
- Stephen Pasteris, Ting He, Fabio Vitale, Shiqiang Wang, and Mark Herbster. Online learning of facility locations. In *Algorithmic Learning Theory*, pages 1002–1050. PMLR, 2021.

- Stephen Pasteris, Madeleine Dwyer, Chris Hicks, and Vasilios Mavroudis. A hierarchical nearest neighbour approach to contextual bandits. *arXiv preprint arXiv:2312.09332*, 2023a.
- Stephen Pasteris, Chris Hicks, and Vasilios Mavroudis. Nearest neighbour with bandit feedback. In *Proc. NeurIPS*, 2023b.
- Stephen Pasteris, Alberto Rumi, Maximilian Thiessen, Shota Saito, Atsushi Miyauchi, Fabio Vitale, and Mark Herbster. Bandits with abstention under expert advice. *Advances in Neural Information Processing Systems*, 37:126059–126087, 2024a.
- Stephen Pasteris, Alberto Rumi, Fabio Vitale, and Nicolò Cesa-Bianchi. Sum-max submodular bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 1216–1224. PMLR, 2024b.
- Ignacio M. Pelayo. *Geodesic Convexity in Graphs*. Springer, 2013.
- Christos Pelekis and Jan Ramon. A lower bound on the probability that a binomial random variable is exceeding its mean. *Statistics & Probability Letters*, 119:305–309, 2016.
- Herbert Robbins. A remark on stirling’s formula. *The American mathematical monthly*, 62(1):26–29, 1955.
- Aviv Rosenberg and Yishay Mansour. Stochastic shortest path with adversarially changing costs. *Proceedings of the 30th International Joint Conference on Artificial Intelligence*, 2021.
- Aviv Rosenberg, Alon Cohen, Yishay Mansour, and Haim Kaplan. Near-optimal regret bounds for stochastic shortest path. *Proceedings of the 37th International Conference on Machine Learning*, 119: 8210–8219, 2020.
- Alberto Rumi, Andrew Jacobsen, Nicolò Cesa-Bianchi, and Fabio Vitale. Parameter-free dynamic regret for unconstrained linear bandits. In *Eighteenth European Workshop on Reinforcement Learning*, 2025.
- Shota Saito and Mark Herbster. Multi-class graph clustering via approximated effective p -resistance. In *Proc. ICML*, pages 29697–29733, 2023.
- Nicolas Schreuder and Evgenii Chzhen. Classification with abstention but without disparities. In *Proc. UAI*, pages 1227–1236, 2021.
- Yevgeny Seldin and Gábor Lugosi. A lower bound for multi-armed bandits with expert advice. In *Proc. EWRL*, volume 2, page 7, 2016.
- Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI Mag.*, 29(3):93–93, 2008.
- Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.
- Rongye Shi, Peter Steenkiste, and Manuela M. Veloso. Sc-m*: A multi-agent path planning algorithm with soft-collision constraint on allocation of common resources. *Applied Sciences*, 9(19), 2019.

- Max Simchowitz, Kevin Jamieson, and Benjamin Recht. Best-of-k-bandits. In *Conference on Learning Theory*, pages 1440–1489. PMLR, 2016.
- Aleksandrs Slivkins. Contextual bandits with similarity information. In *Proc. COLT*, pages 679–702, 2011.
- Pavel Surynek, T. K. Satish Kumar, and Sven Koenig. Multi-agent path finding with capacity constraints. In *Advances in Artificial Intelligence*, 2019.
- Jean Tarbouriech, Evrard Garcelon, Michal Valko, Matteo Pirodda, and Alessandro Lazaric. No-regret exploration in goal-oriented reinforcement learning. *Proceedings of the 37th International Conference on Machine Learning*, 119:9428–9437, 2020.
- Jean Tarbouriech, Runlong Zhou, Simon S Du, Matteo Pirodda, Michal Valko, and Alessandro Lazaric. Stochastic shortest path: Minimax, parameter-free and towards horizon-free regret. *Advances in Neural Information Processing Systems*, 34:6843–6855, 2021.
- Maximilian Thiessen and Thomas Gärtner. Active learning of convex halfspaces on graphs. In *Proc. NeurIPS*, 2021.
- Vincent A Traag. Faster unfolding of communities: Speeding up the louvain algorithm. *Phys. Rev. E*, 92(3):032801, 2015.
- Taira Tsuchiya, Shinji Ito, and Junya Honda. Stability-penalty-adaptive follow-the-regularized-leader: Sparsity, game-dependency, and best-of-both-worlds. *Advances in Neural Information Processing Systems*, 36:47406–47437, 2023.
- Marcel LJ van De Vel. *Theory of Convex Structures*. Elsevier, 1993.
- Dirk van der Hoeven, Ashok Cutkosky, and Haipeng Luo. Comparator-adaptive convex bandits. *CoRR*, abs/2007.08448, 2020. URL <https://arxiv.org/abs/2007.08448>.
- Andrew J Wagenmaker, Yifang Chen, Max Simchowitz, Simon Du, and Kevin Jamieson. First-order regret in reinforcement learning with linear function approximation: A robust estimation approach. *Proceedings of the 39th International Conference on Machine Learning*, 162:22384–22429, 2022.
- Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge University Press, 2019.
- Chen-Yu Wei and Haipeng Luo. More adaptive algorithms for adversarial bandits. *Proceedings of the 31st Conference On Learning Theory*, 75:1263–1291, 2018.
- Yu-Hu Yan, Peng Zhao, and Zhi-Hua Zhou. Online non-stochastic control with partial feedback. *Journal of Machine Learning Research*, 24(273):1–50, 2023.
- Tianbao Yang, Lijun Zhang, Rong Jin, and Jinfeng Yi. Tracking slowly moving clairvoyant: Optimal dynamic regret of online learning with true and noisy gradient. In *International Conference on Machine Learning*, pages 449–457. PMLR, 2016.
- Yisong Yue and Carlos Guestrin. Linear submodular bandits and their application to diversified retrieval. *Advances in Neural Information Processing Systems*, 24, 2011.

- Lijun Zhang, Shiyin Lu, and Zhi-Hua Zhou. Adaptive online learning in dynamic environments. *Advances in neural information processing systems*, 31, 2018.
- Zhiyu Zhang, Ashok Cutkosky, and Yannis Paschalidis. Unconstrained dynamic regret via sparse coding. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 74636–74670. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/ec2833cda146c277cdaa39066764f25c-Paper-Conference.pdf.
- Peng Zhao, Yu-Jie Zhang, Lijun Zhang, and Zhi-Hua Zhou. Dynamic regret of convex and smooth functions. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12510–12520. Curran Associates, Inc., 2020. URL <https://proceedings.neurips.cc/paper/2020/file/939314105ce8701e67489642ef4d49e8-Paper.pdf>.
- Peng Zhao, Guanghui Wang, Lijun Zhang, and Zhi-Hua Zhou. Bandit convex optimization in non-stationary environments. *Journal of Machine Learning Research*, 22(125):1–45, 2021.
- Peng Zhao, Long-Fei Li, and Zhi-Hua Zhou. Dynamic regret of online Markov decision processes. *Proceedings of the 39th International Conference on Machine Learning*, 162:26865–26894, 2022.
- Peng Zhao, Yu-Jie Zhang, Lijun Zhang, and Zhi-Hua Zhou. Adaptivity and non-stationarity: Problem-dependent dynamic regret for online convex optimization. *Journal of Machine Learning Research*, 25(98):1–52, 2024. URL <http://jmlr.org/papers/v25/21-0748.html>.
- Alexander Zimin and Gergely Neu. Online learning in episodic markovian decision processes by relative entropy policy search. *Advances in Neural Information Processing Systems*, 26, 2013.
- Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936, 2003.

Appendix A

Appendix - Parameter Free

A.1 Unconstrained Linear Bandits

In this section, we show how to derive the unconstrained linear bandit result stated in Theorem 3.1. We begin by establishing a base regret guarantee for the continuous exponential weights learner. To that end, we first prove the base guarantee.

Lemma 3.1. *Let $(\|\cdot\|, \|\cdot\|_*)$ be a dual-norm pair and suppose that $\|w\| \leq D$ for all $w \in \mathcal{W}$. Then, for any sequence $u_1, \dots, u_T$ in $\mathcal{W}$ and for any sequence of distributions $p_1^*, \dots, p_T^*$ satisfying $\mathbb{E}_{w \sim p_t^*}[w] = u_t$, Algorithm 3.3 with regularizer $\psi(w) = \frac{1}{\eta} \sum_{i=1}^d (w_i \log(w_i) - w_i)$ and exploration distribution uniform over $\mathcal{W}$ guarantees*

$$\mathbb{E}[R_T(u_1, \dots, u_T)] \leq \frac{1 + \log(d)(1 + P_T^\Delta)}{\eta} + \eta d D^2 \sum_{t=1}^T \|\ell_t\|_*^2.$$

Proof. First observe that we can relate the dynamic regret w.r.t. the chosen actions to the dynamic regret w.r.t. the action sampling distributions as follows

$$\begin{aligned} \mathbb{E}[R_T(u_1, \dots, u_T)] &= \mathbb{E}\left[\sum_{t=1}^T \langle \ell_t, w_t - u_t \rangle\right] = \mathbb{E}\left[\sum_{t=1}^T \langle \ell_t, \int w \tilde{p}_t(w) dw - u_t \rangle\right] \\ &= \mathbb{E}\left[\sum_{t=1}^T \int \langle \ell_t, w \rangle (\tilde{p}_t(w) - p_t^*(w)) dw\right] \\ &= \mathbb{E}\left[\sum_{t=1}^T \int \langle \hat{\ell}_t, w \rangle (\tilde{p}_t(w) - p_t^*(w)) dw\right] \end{aligned}$$

where $p_t^*(w)$ is the distribution supported on $\mathcal{W}$ such that $\mathbb{E}_{p_t^*}[w] = u_t$. Equivalently, we will slightly abuse notation by writing $\hat{g}_t(w) = \langle \hat{\ell}_t, w \rangle$ and $\langle \hat{g}_t, p \rangle = \int \hat{g}_t(w) p(w) dw$, we have

$$\mathbb{E}[R_T((u_1, \dots, u_T))] \leq \mathbb{E}\left[\sum_{t=1}^T \langle \hat{g}_t, \tilde{p}_t - p_t^* \rangle\right]. \quad (\text{A.1})$$

Now we apply the following regret guarantee, proven in Appendix A.1.1:

Proposition A.1. Let $\beta = 1/(T + 1)$, let $\eta > 0$ satisfy $\eta|\widehat{g}_t(\mathbf{w})| \leq 1$ for all $\mathbf{w} \in \mathcal{W}$, and let $\psi(p) = \mathbb{E}_{\mathbf{x} \sim p} \left[\frac{\log(p(\mathbf{x}))}{\eta} \right] = \int \frac{p(\mathbf{x}) \log(p(\mathbf{x}))}{\eta} d\mathbf{x}$. Set $\tilde{p}_1 = \text{Uniform}(\mathcal{W})$ and on each round suppose we play $\tilde{p}_t = (1 - \beta)\mathbf{p}_t + \beta\tilde{p}_1$ and update $\mathbf{p}_{t+1} = \arg \min_{\mathbf{p} \in \mathcal{M}_1(\mathcal{W})} \mathbb{E}_{\mathbf{w} \sim p} [\widehat{g}_t(\mathbf{w})] + D_\psi(\mathbf{p}|\tilde{p}_t)$. Then for any sequence of distributions $p_1^*, \dots, p_T^*$ in $\mathcal{M}_1(\mathcal{W})$, it holds that

$$\sum_{t=1}^T \langle \widehat{g}_t, \tilde{p}_t - \mathbf{p}_t^* \rangle \leq \frac{1 + \log(\text{Vol}(\mathcal{W})) (1 + P_T^\Delta)}{\eta} + \eta \sum_{t=1}^T \mathbb{E}_{\mathbf{w} \sim \tilde{p}_t} [\widehat{g}_t(\mathbf{w})^2],$$

where $P_T^\Delta = \sum_{t=2}^T \|\mathbf{p}_t^* - \mathbf{p}_{t-1}^*\|_1$.

Hence, plugging this back into Equation (A.1), the regret of Algorithm 3.3 is bounded as

$$\begin{aligned} \mathbb{E}[R_T(\mathbf{u}_1, \dots, \mathbf{u}_T)] &\leq \mathbb{E} \left[\frac{1 + \log(\text{Vol}(\mathcal{W})) (1 + P_T^\Delta)}{\eta} + \eta \sum_{t=1}^T \mathbb{E}_{\mathbf{w} \sim \tilde{p}_t} [\widehat{g}_t(\mathbf{w})^2] \right] \\ &= \mathbb{E} \left[\frac{1 + \log(\text{Vol}(\mathcal{W})) (1 + P_T^\Delta)}{\eta} + \eta \sum_{t=1}^T \mathbb{E}_{\mathbf{w} \sim \tilde{p}_t} [\langle \widehat{\ell}_t, \mathbf{w} \rangle^2] \right]. \end{aligned}$$

To bound the last term, define $\mathbf{Q}_t = \mathbb{E}_{\mathbf{w} \sim \tilde{p}_t} [\mathbf{w}\mathbf{w}^\top]$ and observe that

$$\begin{aligned} \mathbb{E}_{\mathbf{w} \sim \tilde{p}_t} [\langle \widehat{\ell}_t, \mathbf{w} \rangle^2] &= \int \langle \ell_t, \mathbf{w}_t \rangle^2 \mathbf{w}_t^\top \mathbf{Q}_t^{-1} \mathbf{w} \mathbf{w}^\top \mathbf{Q}_t^{-1} \mathbf{w}_t^\top \tilde{p}_t(\mathbf{w}) d\mathbf{w} \\ &= \langle \ell_t, \mathbf{w}_t \rangle^2 \mathbf{w}_t^\top \mathbf{Q}_t^{-1} \underbrace{\mathbb{E}_{\mathbf{w} \sim \tilde{p}_t} [\mathbf{w}\mathbf{w}^\top]}_{\mathbf{Q}_t} \mathbf{Q}_t^{-1} \mathbf{w}_t \\ &= \langle \ell_t, \mathbf{w}_t \rangle^2 \|\mathbf{w}_t\|_{\mathbf{Q}_t^{-1}}^2 \end{aligned}$$

Therefore, for $\mathbf{w}_t \sim \tilde{p}_t$, we have

$$\begin{aligned} \mathbb{E} \left[\sum_{t=1}^T \int \langle \widehat{\ell}_t, \mathbf{w} \rangle^2 \tilde{p}_t(\mathbf{w}) d\mathbf{w} \right] &= \mathbb{E} \left[\sum_{t=1}^T \langle \ell_t, \mathbf{w}_t \rangle^2 \|\mathbf{w}_t\|_{\mathbf{Q}_t^{-1}}^2 \right] \\ &\leq \mathbb{E} \left[\sum_{t=1}^T \|\ell_t\|_*^2 D^2 \mathbb{E} [\|\mathbf{w}_t\|_{\mathbf{Q}_t^{-1}}^2 | \tilde{p}_t] \right] \\ &= \mathbb{E} \left[D^2 \sum_{t=1}^T \|\ell_t\|_*^2 \text{Tr} \left(\int \mathbf{w} \mathbf{w}^\top \tilde{p}_t(\mathbf{w}) d\mathbf{w} \mathbf{Q}_t^{-1} \right) \right] \\ &= \mathbb{E} \left[D^2 d \sum_{t=1}^T \|\ell_t\|_*^2 \right], \end{aligned}$$

for $\|\mathbf{w}_t\| \leq D$. Plugging this back in above, we have

$$\mathbb{E}[R_T(\mathbf{u}_1, \dots, \mathbf{u}_T)] \leq \frac{1 + \log(\text{Vol}(\mathcal{W})) (1 + P_T^\Delta)}{\eta} + \eta d D^2 \sum_{t=1}^T \|\ell_t\|_*^2.$$

□

A.1.1 Switching Regret on the Simplex

For completeness, in this section we show how to derive the dynamic regret of the base OLO algorithm. The result follows a standard mirror descent argument.

Throughout this section, we will use the short-hand notation $\langle g, p \rangle = \int g(w)p(w)dw$, where g is a function with domain $\mathcal{W}$, p is the density of a distribution of $\mathcal{W}$. We let $\mathcal{M}_1(\mathcal{W})$ denote the set of distributions over $\mathcal{W}$.

Proposition A.1. *Let $\beta = 1/(T + 1)$, let $\eta > 0$ satisfy $\eta|\hat{g}_t(w)| \leq 1$ for all $w \in \mathcal{W}$, and let $\psi(p) = \mathbb{E}_{\mathbf{x} \sim p} \left[\frac{\log(p(\mathbf{x}))}{\eta} \right] = \int \frac{p(\mathbf{x}) \log(p(\mathbf{x}))}{\eta} d\mathbf{x}$. Set $\tilde{p}_1 = \text{Uniform}(\mathcal{W})$ and on each round suppose we play $\tilde{p}_t = (1 - \beta)p_t + \beta\tilde{p}_1$ and update $p_{t+1} = \arg \min_{p \in \mathcal{M}_1(\mathcal{W})} \mathbb{E}_{w \sim p}[\hat{g}_t(w)] + D_\psi(p|\tilde{p}_t)$. Then for any sequence of distributions $p_1^*, \dots, p_T^*$ in $\mathcal{M}_1(\mathcal{W})$, it holds that*

$$\sum_{t=1}^T \langle \hat{g}_t, \tilde{p}_t - p_t^* \rangle \leq \frac{1 + \log(\text{Vol}(\mathcal{W}))(1 + P_T^\Delta)}{\eta} + \eta \sum_{t=1}^T \mathbb{E}_{w \sim \tilde{p}_t} [\hat{g}_t(w)^2],$$

where $P_T^\Delta = \sum_{t=2}^T \|p_t^* - p_{t-1}^*\|_1$.

Proof. Using a standard dynamic regret decomposition for mirror descent updates (see, e.g., Jacobsen and Cutkosky (2023, Lemma A.1)), we have

$$\begin{aligned} \sum_{t=1}^T \langle \hat{g}_t, \tilde{p}_t - p_t^* \rangle &\leq \sum_{t=1}^T \underbrace{D_\psi(p_t^*|\tilde{p}_t) - D_\psi(p_t^*|\tilde{p}_{t+1})}_{=: \mathcal{P}_t} + \sum_{t=1}^T \underbrace{D_\psi(p_t^*|p_{t+1}) - D_\psi(p_t^*|\tilde{p}_{t+1})}_{=: \xi_t} \\ &\quad + \sum_{t=1}^T \underbrace{\langle \hat{g}_t, \tilde{p}_t - p_{t+1} \rangle - D_\psi(p_{t+1}|\tilde{p}_t)}_{=: \delta_t}. \end{aligned}$$

Observe that for any $u, p, q \in \mathcal{M}_1(\mathcal{W})$ we have

$$\begin{aligned} D_\psi(u | p) - D_\psi(u | q) &= \frac{1}{\eta} \int u(w) \log \left(\frac{u(w)}{p(w)} \right) dw - \frac{1}{\eta} \int u(w) \log \left(\frac{u(w)}{q(w)} \right) dw \\ &= \frac{1}{\eta} \int u(w) \log \left(\frac{q(w)}{p(w)} \right) dw \end{aligned} \tag{A.2}$$

Thus, the terms ξ_t can be bound as

$$\begin{aligned} \sum_{t=1}^T \xi_t &= \sum_{t=1}^T D_\psi(p_t^* | \tilde{p}_{t+1}) - D_\psi(p_t^* | p_{t+1}) \\ &= \frac{1}{\eta} \sum_{t=1}^T \int p_t^*(w) \log \left(\frac{p_{t+1}(w)}{\tilde{p}_{t+1}(w)} \right) dw \\ &= \frac{1}{\eta} \sum_{t=1}^T \int p_t^*(w) \log \left(\frac{p_{t+1}(w)}{(1 - \beta)p_{t+1}(w) + \beta\tilde{p}_1(w)} \right) dw \\ &\leq \frac{1}{\eta} \sum_{t=1}^T \int p_t^*(w) \log \left(\frac{1}{(1 - \beta)} \right) dw \\ &\leq \frac{T}{\eta} \log \left(\frac{1}{(1 - \beta)} \right). \end{aligned}$$

Now, focusing on the path terms $\mathcal{P}_t$, we have

$$\begin{aligned}
\sum_{t=1}^T \mathcal{P}_t &= D_\psi(p_1^*|\tilde{p}_1) - D_\psi(p_T^*|\tilde{p}_{T+1}) + \sum_{t=2}^T D_\psi(p_t^*|\tilde{p}_t) - D_\psi(p_{t-1}^*|\tilde{p}_t) \\
&= D_\psi(p_1^*|\tilde{p}_1) - D_\psi(p_T^*|\tilde{p}_{T+1}) \\
&\quad + D_\psi(p_T^*|\tilde{p}_1) - D_\psi(p_1^*|\tilde{p}_1) + \sum_{t=2}^T \frac{1}{\eta} \int (p_{t-1}^*(\mathbf{w}) - p_t^*(\mathbf{w})) (\log(\tilde{p}_t(\mathbf{w})) - \log(\tilde{p}_1(\mathbf{w}))) d\mathbf{w} \\
&= D_\psi(p_T^*|\tilde{p}_1) - D_\psi(p_T^*|\tilde{p}_{T+1}) + \sum_{t=2}^T \frac{1}{\eta} \int (p_{t-1}^*(\mathbf{w}) - p_t^*(\mathbf{w})) \log\left(\frac{\tilde{p}_t(\mathbf{w})}{\tilde{p}_1(\mathbf{w})}\right) d\mathbf{w} \\
&\leq D_\psi(p_T^*|\tilde{p}_1) - D_\psi(p_T^*|\tilde{p}_{T+1}) + \sum_{t=2}^T \frac{1}{\eta} \|p_t^* - p_{t-1}^*\|_1 \sup_{\mathbf{w} \in \mathcal{W}} \left| \log\left(\frac{\tilde{p}_t(\mathbf{w})}{\tilde{p}_1(\mathbf{w})}\right) \right| \\
&\leq D_\psi(p_T^*|\tilde{p}_1) - D_\psi(p_T^*|\tilde{p}_{T+1}) + \sum_{t=2}^T \frac{1}{\eta} \|p_t^* - p_{t-1}^*\|_1 |\log(\text{Vol}(\mathcal{W}))|,
\end{aligned}$$

where the first inequality applies Hölder inequality. Moreover, using Equation (A.2), the first two terms can be bound as

$$\begin{aligned}
D_\psi(p_T^*|\tilde{p}_1) - D_\psi(p_T^*|\tilde{p}_{T+1}) &= \frac{1}{\eta} \int p_T^*(\mathbf{w}) \log\left(\frac{\tilde{p}_{T+1}(\mathbf{w})}{\tilde{p}_1(\mathbf{w})}\right) d\mathbf{w} \\
&= \frac{1}{\eta} \int p_T^*(\mathbf{w}) \log\left(\frac{(1-\beta)p_{T+1}(\mathbf{w}) + \beta\tilde{p}_1(\mathbf{w})}{\tilde{p}_1(\mathbf{w})}\right) d\mathbf{w} \\
&\leq \frac{1}{\eta} \int p_T^*(\mathbf{w}) \log\left(\frac{(1-\beta)p_{T+1}(\mathbf{w})}{\tilde{p}_1(\mathbf{w})} + \beta\right) d\mathbf{w} \\
&\leq \frac{1}{\eta} \log(\text{Vol}(\mathcal{W}) (1-\beta) + \beta) \leq \frac{\log_+(\text{Vol}(\mathcal{W}))}{\eta}
\end{aligned}$$

where we've recalled $\tilde{p}_t = (1-\beta)p_t + \beta\tilde{p}_1$ for any t and used $(1-\beta)a + \beta b \leq \max\{a, b\}$.

Returning to our regret bound, we have

$$\begin{aligned}
\sum_{t=1}^T \langle \hat{\mathbf{g}}_t, \tilde{\mathbf{p}}_t - \mathbf{p}_t^* \rangle &\leq \frac{1}{\eta} |\log(\text{Vol}(\mathcal{W}))| + \frac{1}{\eta} |\log(\text{Vol}(\mathcal{W}))| \sum_{t=2}^T \|p_{t-1}^* - p_t^*\|_1 + \frac{T}{\eta} \log\left(\frac{1}{(1-\beta)}\right) + \sum_{t=1}^T \delta_t \\
&= \frac{|\log(\text{Vol}(\mathcal{W}))|(1 + P_T^\Delta)}{\eta} + \frac{T}{\eta} \log\left(\frac{1}{(1-\beta)}\right) + \sum_{t=1}^T \delta_t \tag{A.3}
\end{aligned}$$

We now have to take care of the terms $\sum_{t=1}^T \delta_t$. Let p_{t+1}^+ denote the unconstrained minimizer $p_{t+1}^+ = \arg \min_p \langle \hat{\mathbf{g}}_t, \mathbf{p} \rangle + D_\psi(p|\tilde{p}_t)$, we have

$$\begin{aligned}
\sum_{t=1}^T \delta_t &= \sum_{t=1}^T \langle \hat{\mathbf{g}}_t, \tilde{\mathbf{p}}_t - \mathbf{p}_{t+1} \rangle - D_\psi(\mathbf{p}_{t+1}|\tilde{p}_t) \leq \sum_{t=1}^T \langle \hat{\mathbf{g}}_t, \tilde{\mathbf{p}}_t - \mathbf{p}_{t+1}^+ \rangle - D_\psi(\mathbf{p}_{t+1}^+|\tilde{p}_t) \\
&= \sum_{t=1}^T \langle \nabla \psi(\tilde{\mathbf{p}}_t) - \nabla \psi(\mathbf{p}_{t+1}^+), \tilde{\mathbf{p}}_t - \mathbf{p}_{t+1}^+ \rangle - D_\psi(\mathbf{p}_{t+1}^+|\tilde{p}_t) \\
&= \sum_{t=1}^T D_\psi(\tilde{p}_t|\mathbf{p}_{t+1}^+) = \sum_{t=1}^T \int \frac{1}{\eta} \left(\tilde{p}_t(\mathbf{w}) \log(\tilde{p}_t(\mathbf{w})/p_{t+1}^+(\mathbf{w})) - \tilde{p}_t(\mathbf{w}) + p_{t+1}^+(\mathbf{w}) \right) d\mathbf{w}
\end{aligned}$$

$$\begin{aligned}
 &= \sum_{t=1}^T \int \frac{1}{\eta} \left(\tilde{p}_t(\mathbf{w}) \log \left(e^{\eta \hat{g}_t(\mathbf{w})} \right) - \tilde{p}_t(\mathbf{w}) + \tilde{p}_t(\mathbf{w}) e^{-\eta \hat{g}_t(\mathbf{w})} \right) d\mathbf{w} \\
 &= \sum_{t=1}^T \int \frac{\tilde{p}_t(\mathbf{w})}{\eta} \left(\eta \hat{g}_t(\mathbf{w}) - 1 + e^{-\eta \hat{g}_t(\mathbf{w})} \right) d\mathbf{w} \\
 &\leq \sum_{t=1}^T \int \eta \hat{g}_t(\mathbf{w})^2 \tilde{p}_t(\mathbf{w}) d\mathbf{w},
 \end{aligned}$$

where we've used $p_{t+1}^+(\mathbf{w}) = \tilde{p}_t(\mathbf{w}) e^{-\eta \hat{g}_t(\mathbf{w})}$ via the first-order optimality condition for p_{t+1}^+ and $e^{-x} - 1 + x \leq x^2$ for $x \geq -1$. Hence,

$$\begin{aligned}
 \sum_{t=1}^T \langle \hat{\mathbf{g}}_t, \tilde{\mathbf{p}}_t - \mathbf{p}_t^* \rangle &\leq \frac{|\log(\text{Vol}(\mathcal{W}))|}{\eta} + \frac{|\log(\text{Vol}(\mathcal{W}))| \sum_{t=2}^T \|\mathbf{p}_t^* - \mathbf{p}_{t-1}^*\|_1}{\eta} \\
 &\quad + \frac{T}{\eta} \log \left(\frac{1}{(1-\beta)} \right) + \eta \sum_{t=1}^T \int \hat{g}_t(\mathbf{w})^2 \tilde{p}_t(\mathbf{w}) d\mathbf{w}.
 \end{aligned}$$

Setting $\beta = 1/(T+1)$, and using the following simple inequalities:

$$\log \left(\frac{1}{1-\beta} \right) = \log \left(\frac{1}{1-1/(T+1)} \right) = \log \left(\frac{T+1}{T} \right) = \log \left(1 + \frac{1}{T} \right) \leq \frac{1}{T},$$

we have an overall regret bound of

$$\sum_{t=1}^T \langle \hat{\mathbf{g}}_t, \tilde{\mathbf{p}}_t - \mathbf{p}_t^* \rangle \leq \frac{1 + |\log(\text{Vol}(\mathcal{W}))|(1 + P_T^\Delta)}{\eta} + \eta \sum_{t=1}^T \int \hat{g}_t(\mathbf{w})^2 \tilde{p}_t(\mathbf{w}) d\mathbf{w}.$$

□

A.1.2 Optimal Switching Regret without Prior Knowledge

Now combining the results from Proposition 3.2 and Lemma 3.1, we are ready to prove Theorem 3.1.

Theorem 3.1. *Let $\mathcal{S}$ as defined in Equation (3.2). For all $i \in |\mathcal{S}|$, let $\mathcal{A}_i$ be an instance of Algorithm 3.2 such that the direction learner $\mathcal{A}_B^{(i)}$ is an instance of Algorithm 3.3 with step-size η_i on domain $\mathcal{B}_N$, and $\mathcal{A}_V^{(i)}$ is a comparator-adaptive OLO routine. Then for any sequence $\mathbf{u}_1, \dots, \mathbf{u}_T$ in $\mathcal{W}$ and sequence of distributions $\mathbf{p}_1^*, \dots, \mathbf{p}_T^*$ satisfying $\mathbb{E}_{\mathbf{w} \sim \mathbf{p}_t^*}[\mathbf{w}] = \mathbf{u}_t$, Algorithm 3.1 guarantees*

$$\begin{aligned}
 \mathbb{E}[R_T(\mathbf{u}_1, \dots, \mathbf{u}_T)] &= \tilde{\mathcal{O}} \left(G\epsilon + dMG(1 + P_T^\Delta) \right. \\
 &\quad \left. + dM \sqrt{N(1 + P_T^\Delta) \sum_{t=1}^T \|\ell_t\|_*^2} \right),
 \end{aligned}$$

where $M = \max_t \|\mathbf{u}_t\|$.

Proof. Observe that $\eta_{\max}/\eta_{\min} = T$, so $N = \lceil \log_2(\eta_{\max}/\eta_{\min}) \rceil = \lceil \log_2(T) \rceil$. By Proposition 3.2, for

any $\eta_i \in \mathcal{S}$ we have

$$\mathbb{E}[R_T(\mathbf{u}_1, \dots, \mathbf{u}_T)] \leq \mathbb{E}\left[R_T^{\mathcal{A}_V^{(i)}}(MN) + MN R_T\left(\frac{\mathbf{u}_1}{M}, \dots, \frac{\mathbf{u}_T}{M}\right) + (N-1)G\epsilon\right],$$

where $M = \max_t \|\mathbf{u}_t\|$. The regret guarantee of $\mathcal{A}_V^{(i)}$ ensures that

$$\begin{aligned} \mathbb{E}\left[R_T^{\mathcal{A}_V^{(i)}}(NM)\right] &\leq G\epsilon + MNE\left[\sqrt{\sum_t \langle \hat{\ell}_t^{(i)}, \beta_t^{(i_t)} \rangle^2 \log\left(\frac{MN\Lambda_T}{\epsilon G} + 1\right)} \vee G \log\left(\frac{NM\Lambda_T}{\epsilon G} + 1\right)\right] \\ &= \tilde{O}\left(\epsilon G + M\sqrt{N \sum_t \|\ell_t\|_*^2}\right), \end{aligned}$$

since $\|\beta_t^{(i_t)}\| \leq 1$ for all t and $\mathbb{E}\left[\sqrt{\sum_{t=1}^T \|\hat{\ell}_t^{(i)}\|_*^2}\right] \leq \sqrt{\sum_{t=1}^T \mathbb{E}[\|\ell_t\|_*^2 \mathbb{I}\{i_t = i\}]} = \sqrt{\sum_{t=1}^T \|\ell_t\|_*^2 / N}$ via Jensen's inequality. Moreover, letting $\tilde{\mathbf{u}}_t = \frac{\mathbf{u}_t}{M}$, each of the algorithms $\mathcal{A}_B^{(i)}$ guarantees

$$\begin{aligned} \mathbb{E}\left[R_T^{\mathcal{A}_B^{(i)}}(\tilde{\mathbf{u}}_1, \dots, \tilde{\mathbf{u}}_T)\right] &\leq \frac{1 + \log(\text{Vol}(\mathcal{W}))(1 + P_T^\Delta)}{\eta_i} + \eta_i d \sum_{t=1}^T \mathbb{E}\left[\|\hat{\ell}_t^{(i)}\|_*^2\right] \\ &= \frac{1 + \log(\text{Vol}(\mathcal{W}))(1 + P_T^\Delta)}{\eta_i} + \frac{\eta_i d}{N} \sum_{t=1}^T \mathbb{E}\left[\|\ell_t\|_*^2\right], \end{aligned}$$

where we've used the fact that $\mathbb{E}\left[\|\hat{\ell}_t^{(i)}\|_*^2\right] = \mathbb{E}\left[\|\ell_t\|_*^2 \mathbb{I}\{i_t = i\}\right] = \|\ell_t\|_*^2 / N$. Hence, by Lemma A.1, there is an $i \in [N]$ such that

$$\begin{aligned} NM\mathbb{E}\left[R_T^{\mathcal{A}_B^{(i)}}(\tilde{\mathbf{u}}_1, \dots, \tilde{\mathbf{u}}_T)\right] &\leq 3M\sqrt{dN(1 + \log(\text{Vol}(\mathcal{W}))(1 + P_T^\Delta)) \sum_{t=1}^T \|\ell_t\|_*^2} \\ &\quad + NM(1 + \log(d)(1 + P_T^\Delta))G + \frac{dM}{GT} \sum_{t=1}^T \|\ell_t\|_*^2 \\ &\leq 3\sqrt{dN(1 + \log(\text{Vol}(\mathcal{W}))(1 + P_T^\Delta)) \sum_{t=1}^T \|\ell_t\|_*^2} \\ &\quad + NM(1 + \log(d)(1 + P_T^\Delta))G + MGd \end{aligned}$$

Hence, combining with the previous display we have

$$\begin{aligned} \mathbb{E}[R_T(\mathbf{u}_1, \dots, \mathbf{u}_T)] &= \tilde{O}\left(NG(\epsilon + dM) + NM(1 + \log(\text{Vol}(\mathcal{W}))(1 + P_T^\Delta))G\right. \\ &\quad \left.+ M\sqrt{N \sum_{t=1}^T \|\ell_t\|_*^2} + 3M\sqrt{dN(1 + \log(\text{Vol}(\mathcal{W}))(1 + P_T^\Delta)) \sum_{t=1}^T \|\ell_t\|_*^2}\right) \\ &= \tilde{O}\left(G\epsilon + dMG(1 + P_T^\Delta) + dM\sqrt{N(1 + P_T^\Delta) \sum_{t=1}^T \|\ell_t\|_*^2}\right), \end{aligned}$$

where the last line hides polylog factors and bounds $\log(\text{Vol}(\mathcal{W})) \leq \tilde{\mathcal{O}}(d)$.

□

A.2 Supporting Lemmas

In this section, we present the supporting lemmas used to combine the guarantees of the base algorithms.

Lemma A.1. *Let $b > 1$, $0 < \eta_{\min} \leq \eta_{\max}$ and let $\mathcal{S} = \{\eta_i = \eta_{\min} b^i \wedge \eta_{\max} : i = 0, 1, \dots\}$. Then for any $P, V \in \mathbb{R}_{\geq 0}$, there is an $\eta \in \mathcal{S}$ such that*

$$R(\eta) := \frac{P}{\eta} + \eta V \leq (b+1)\sqrt{PV} + \frac{P}{\eta_{\max}} + \eta_{\min} V$$

Proof. Define

$$R(\eta) = \frac{P}{\eta} + \eta V.$$

Clearly the bound is optimized by $\eta^* = \arg \min_{\eta} \frac{P}{\eta} + \eta V = \sqrt{P/V}$. Suppose that $\eta^* \in [\eta_{\min}, \eta_{\max}]$. Then there is an $i \in 0, 1, \dots$ such that $\eta_i \leq \eta^* \leq \eta_{i+1} = b\eta_i$, hence setting $\eta = \eta_i$ yields

$$R(\eta) = \frac{P}{\eta_i} + \eta_i V \leq b \frac{P}{\eta^*} + \eta^* V = (b+1)\sqrt{PV}. \quad (\text{A.4})$$

Otherwise, $\eta^* \notin [\eta_{\min}, \eta_{\max}]$ and there are two cases to consider. First suppose that $\eta^* \geq \eta_{\max}$, then setting $\eta = \eta_{\max}$ we have

$$R(\eta) = \frac{P}{\eta_{\max}} + \eta_{\max} V \leq \frac{P}{\eta_{\max}} + \eta^* V = \frac{P}{\eta_{\max}} + \sqrt{PV}. \quad (\text{A.5})$$

Likewise, if $\eta^* \leq \eta_{\min}$, then setting $\eta = \eta_{\min}$ yields

$$R(\eta) = \frac{P}{\eta_{\min}} + \eta_{\min} V \leq \frac{P}{\eta^*} + \eta_{\min} V = \sqrt{PV} + \eta_{\min} V. \quad (\text{A.6})$$

Overall, combining Equations (A.4) to (A.6), we have that there is an $\eta \in \mathcal{S}$ such that

$$R(\eta) \leq (b+1)\sqrt{PV} + \frac{P}{\eta_{\max}} + \eta_{\min} V.$$

□

Lemma A.2. *Let $b > 1, c > 1, 0 < \eta_{\min} \leq \eta_{\max}, 0 < \delta_{\min} \leq \delta_{\max}$ and let $\mathcal{S} = \{\eta_i = \eta_{\min} b^i \wedge \eta_{\max} : i = 0, 1, \dots\}$ and $\{\delta_j = \delta_{\min} c^j \wedge \delta_{\max} : j = 0, 1, \dots\}$. Then for any $P, V_1, V_2 \in \mathbb{R}_{\geq 0}$, there is an $\eta \in \mathcal{S}$ such that*

$$R(\eta, \delta) := \frac{P}{\eta} + \eta \frac{V_1}{\delta^2} T + \delta V_2 T \leq \frac{P}{\eta_{\max}} + \eta_{\min} \frac{V_1}{\delta_{\max}^2} T + \delta_{\min} V_2 T + \frac{1+b+c^2}{\sqrt{2}} V_2^{1/2} (PV_1)^{1/4} T^{3/4}$$

Proof. Define:

$$R(\eta, \delta) = \frac{P}{\eta} + \eta \frac{V_1}{\delta^2} T + \delta V_2 T$$

To obtain the optimal η^* and δ^* :

$$\frac{\partial R}{\partial \eta} = -\frac{P}{\eta^2} + \frac{V_1 T}{\delta^2} = 0 \implies \eta = \sqrt{\frac{P\delta^2}{V_1 T}}.$$

Then,

$$\begin{aligned} \frac{\partial R}{\partial \delta} &= TV_2 - 2\frac{\eta V_1 T}{\delta^3} = 0 \implies V_2 T - 2\frac{\left(\sqrt{\frac{P\delta^2}{V_1 T}}\right) V_1 T}{\delta^3} = 0 \\ &\implies V_2 T - 2\frac{\sqrt{PV_1 T}}{\delta^2} = 0 \implies \delta = \left(\frac{2}{V_2} \sqrt{\frac{PV_1}{T}}\right)^{1/2} \end{aligned}$$

$$\text{So, } \delta^* = \left(\frac{2}{V_2} \sqrt{\frac{PV_1}{T}}\right)^{1/2} \text{ and } \eta^* = \sqrt{\frac{P \frac{2}{V_2} \sqrt{\frac{PV_1}{T}}}{V_1 T}} = \sqrt{\frac{2P^{3/2}}{V_2 V_1^{1/2} T^{3/2}}} \text{ yields}$$

$$R(\eta^*, \delta^*) \leq 2\sqrt{2}V_2^{1/2}(PV_1)^{1/4}T^{3/4}.$$

Now suppose that $\eta^* \in [\eta_{\min}, \eta_{\max}]$ and $\delta^* \in [\delta_{\min}, \delta_{\max}]$. Then there are $i, j \in 0, 1, \dots$ such that $\eta_i \leq \eta^* \leq \eta_{i+1} = b\eta_i$ and $\delta_j \leq \delta^* \leq \delta_{j+1} = c\delta_j$, hence setting $\eta = \eta_i$ and $\delta = \delta_j$ yields

$$R(\eta, \delta) \leq \frac{P}{\eta} + \eta \frac{V_1}{\delta^2} T + \delta V_2 T \leq b \frac{P}{\eta^*} + c^2 \eta^* \frac{V_1}{\delta^{*2}} T + \delta^* V_2 T \leq \frac{1+b+c^2}{\sqrt{2}} V_2^{1/2} (PV_1)^{1/4} T^{3/4}$$

Now we need to analyze the other cases, note that we can proceed as in the previous lemma for both the parameters, if $\eta^* \geq \eta_{\max}$ then the second term can be optimized with η^* , while in the other case the first term will be. The same reasoning is used for the δ parameter, yielding:

$$R(\eta, \delta) \leq \frac{P}{\eta_{\max}} + \eta_{\min} \frac{V_1}{\delta_{\max}^2} T + \delta_{\min} V_2 T + \frac{1+b+c^2}{\sqrt{2}} V_2^{1/2} (PV_1)^{1/4} T^{3/4}$$

□

Appendix B

Appendix - Bandits with Abstention under Expert Advice

B.1 CBA analysis

Here we prove Theorem 4.1 from the modification of mirror descent (and the specific construction of g_t) given in Section 4.4. Whenever we take expectations in this analysis they are over the draw of a_t from s_t for some $t \in [T]$. As for mirror descent, our analysis hinges on the following classic lemma:

Lemma B.1. *Given any convex set $\mathcal{C} \subseteq \mathbb{R}_+^E$, any convex function $\xi : \mathbb{R}_+^E \rightarrow \mathbb{R}$, any $\mathbf{q} \in \mathcal{C}$ and any $\mathbf{z} \in \mathbb{R}_+^E$ with:*

$$\mathbf{q} = \operatorname{argmin}_{\mathbf{v} \in \mathcal{C}} (\xi(\mathbf{v}) + \Delta(\mathbf{v}, \mathbf{z})),$$

then for all $\mathbf{u} \in \mathcal{C}$ we have:

$$\xi(\mathbf{u}) + \Delta(\mathbf{u}, \mathbf{z}) \geq \xi(\mathbf{q}) + \Delta(\mathbf{u}, \mathbf{q}).$$

Proof. Theorem 9.12 in Beck (2017) shows that the theorem holds if Δ is Bregman divergence. In our case Δ is indeed a Bregman divergence: that of the convex function $f : \mathbb{R}_+^E \rightarrow \mathbb{R}$ for all $\mathbf{v} \in \mathbb{R}_+^E$ defined by:

$$f(\mathbf{v}) := \sum_{i \in [E]} v(i) \ln(v(i)),$$

which concludes the proof. □

Proof of Theorem 4.1. Choose any $\mathbf{u} \in \mathcal{V}$ and $t \in [T]$. We immediately have $\mathcal{V} \subseteq \mathcal{V}_t$ by definition, and therefore $\mathbf{u} \in \mathcal{V}_t$. Hence, by setting ξ such that $\xi(\mathbf{v}) := 0$ for all $\mathbf{v} \in \mathbb{R}_+^E$, setting $\mathcal{C} \in \mathcal{V}_t$ and setting $\mathbf{z} = \mathbf{w}_t$ in Lemma B.1 we have $\mathbf{q} = \tilde{\mathbf{w}}_t$ so that:

$$\Delta(\mathbf{u}, \mathbf{w}_t) \geq \Delta(\mathbf{u}, \tilde{\mathbf{w}}_t). \tag{B.1}$$

Alternatively, by setting ξ such that $\xi(\mathbf{v}) := \eta g_t \cdot (\tilde{\mathbf{w}}_t - \mathbf{v})$ for all $\mathbf{v} \in \mathbb{R}_+^E$, setting $\mathcal{C} = \mathbb{R}_+^E$ and setting $\mathbf{z} = \tilde{\mathbf{w}}_t$ in Lemma B.1 we have $\mathbf{q} = \mathbf{w}_{t+1}$ so that:

$$\eta g_t \cdot (\tilde{\mathbf{w}}_t - \mathbf{u}) + \Delta(\mathbf{u}, \tilde{\mathbf{w}}_t) \geq \eta g_t \cdot (\tilde{\mathbf{w}}_t - \mathbf{w}_{t+1}) + \Delta(\mathbf{u}, \mathbf{w}_{t+1}). \tag{B.2}$$

Since $\mathbb{E}[\mathbf{g}_t] = \nabla \rho_t(\tilde{\mathbf{w}}_t)$ and ρ_t is linear we have:

$$\mathbb{E}[\mathbf{g}_t \cdot (\tilde{\mathbf{w}}_t - \mathbf{u})] = \rho_t(\tilde{\mathbf{w}}_t) - \rho_t(\mathbf{u}). \quad (\text{B.3})$$

In what follows we use the fact that for all $x \leq 1$ we have:

$$x(1 - \exp(x)) \geq -2x^2. \quad (\text{B.4})$$

For all $i \in [E]$, we have, by definition, that $g_{t,i} = e_t^i \cdot \hat{\mathbf{r}}_t$ so by Equation (4.2) we have:

$$\mathbf{g}_t \cdot (\tilde{\mathbf{w}}_t - \mathbf{w}_{t+1}) = \sum_{i \in [E]} \tilde{w}_t(i) e_t^i \cdot \hat{\mathbf{r}}_t (1 - \exp(\eta e_t^i \cdot \hat{\mathbf{r}}_t)).$$

Since, for all $a \in [K]$, we have $\hat{r}_{t,a} \leq 1$ and hence, as $\eta < 1$ and, for all $i \in [E]$ we have $\|e_t^i\|_1 \leq 1$, we can invoke Equation (B.4), which gives us:

$$\eta \mathbf{g}_t \cdot (\tilde{\mathbf{w}}_t - \mathbf{w}_{t+1}) \geq -2 \sum_{i \in [E]} \tilde{w}_t(i) (\eta e_t^i \cdot \hat{\mathbf{r}}_t)^2. \quad (\text{B.5})$$

By definition of $\hat{\mathbf{r}}_t$ we have, for all $i \in [E]$, that:

$$e_t^i \cdot \hat{\mathbf{r}}_t = \|e_t^i\|_1 + e_t^i(a_t)(1 - r_t(a_t))/s_t(a_t) \leq c_t(i) + 2e_t^i(a_t)/s_t(a_t)$$

so that since, for all $a \in [K]$, we have $\Pr[a_t = a] = s_t(a)$ we also have:

$$\mathbb{E}[(e_t^i \cdot \hat{\mathbf{r}}_t)^2] \leq c_t(i)^2 + \sum_{a \in [K]} (2e_t^i(a)c_t(i) + 4(e_t^i(a))^2/s_t(a)). \quad (\text{B.6})$$

Since, for all $i \in [E]$ and $a \in [K]$, we have $e_t^i(a) \leq 1$ and $c_t(i) \leq 1$ and hence also $c_t(i)^2 \leq c_t(i)$ we then have:

$$\mathbb{E}[(e_t^i \cdot \hat{\mathbf{r}}_t)^2] \leq (2K + 1)c_t(i) + 4 \sum_{a \in [K]} e_t^i(a)/s_t(a). \quad (\text{B.7})$$

Note that since $\tilde{\mathbf{w}}_t \in \mathcal{V}_t$ we have:

$$\sum_{i \in [E]} \tilde{w}_t(i) c_t(i) \leq 1. \quad (\text{B.8})$$

Also, by definition of $\mathbf{s}_t$ we have:

$$\sum_{i \in [E]} \tilde{w}_t(i) \sum_{a \in [K]} e_t^i(a)/s_t(a) = \sum_{a \in [K]} \frac{1}{s_t(a)} \sum_{i \in [E]} \tilde{w}_t(i) e_t^i(a) = \sum_{a \in [K]} \frac{1}{s_t(a)} s_t(a) = K. \quad (\text{B.9})$$

Multiplying Inequality (B.7) by $\tilde{w}_t(i)$, summing over all $i \in [E]$, and then substituting in Inequality (B.8) and Equation (B.9) gives us:

$$\sum_{i \in [E]} \tilde{w}_t(i) \mathbb{E}[(e_t^i \cdot \hat{\mathbf{r}}_t)^2] \leq (2K + 1) + 4K = 6K + 1. \quad (\text{B.10})$$

Taking expectations on Inequality (B.5) and substituting in Inequality (B.10) (after taking expecta-

tions) gives us:

$$\mathbb{E}[\eta \mathbf{g}_t \cdot (\tilde{\mathbf{w}}_t - \mathbf{w}_{t+1})] \geq -\eta^2(12K + 2). \quad (\text{B.11})$$

Taking expectations (over the draw $a_t \sim \mathbf{s}_t$) on Inequality (B.2), substituting in Inequalities (B.1), (B.3) and (B.11), and then rearranging gives us:

$$\Delta(\mathbf{u}, \mathbf{w}_t) - \mathbb{E}[\Delta(\mathbf{u}, \mathbf{w}_{t+1})] \geq \eta(\rho_t(\mathbf{u}) - \rho_t(\tilde{\mathbf{w}}_t)) - \eta^2(12K + 2).$$

Summing this inequality over all $t \in [T]$, taking expectations (over the entire sequence of action draws) and noting that $\Delta(\mathbf{u}, \mathbf{w}_{T+1}) > 0$ gives us:

$$\Delta(\mathbf{u}, \mathbf{w}_1) \geq \eta \sum_{t \in [T]} \mathbb{E}[\rho_t(\mathbf{u}) - \rho_t(\tilde{\mathbf{w}}_t)] - \eta^2(12K + 2)T.$$

Substituting in Equation (4.1) and rearranging then gives us, by definition of ρ and ρ_t , the required goal:

$$\sum_{t \in [T]} \mathbb{E}[r_t(a_t)] \geq \mathbb{E}[\rho(\mathbf{u})] - \Delta(\mathbf{u}, \mathbf{w}_1)/\eta - \eta(12K + 2)T.$$

□

B.1.1 Unbounded precision case

We will now show how to handle the case in which our numerical precision is unbounded, incurring a time complexity equal, up to a factor logarithmic in T , to that of Exp4 and adding only 1 to the regret. This additive factor, however, can be made arbitrarily small.

Let us restrict ourselves to compare against $\mathbf{u}$ with $\|\mathbf{u}\|_\infty \leq Z$ for some arbitrary Z . Note that this always has to be the case when each expert has a confidence of at least $1/Z$ on some trial. Our time complexity will be logarithmic in Z . At the beginning of trial t we will now project (via the unnormalised relative entropy) $\mathbf{w}_t$ into the set $\{\mathbf{v} \in \mathbb{R}^E \mid \|\mathbf{v}\|_\infty \leq Z\}$ which simply requires clipping its components. Since the set $\{\mathbf{v} \in \mathbb{R}^E \mid \|\mathbf{v}\|_\infty \leq Z\}$ is convex and contains our comparator $\mathbf{u}$ this will not affect our regret bound.

For any $q \in \mathbb{R}$ let $\mathcal{V}_t(q)$ be the set of all $\mathbf{v}$ with $\mathbf{v} \cdot \mathbf{c}_t \leq q$. We note that given, for all $t \in [T]$, a value $q_t \in [1 - 1/T, 1]$ we have that there exists $\hat{\mathbf{u}} \in \bigcap_t \mathcal{V}_t(q_t)$ such that the cumulative reward of $\pi(\hat{\mathbf{u}})$ is no less than that of $\pi(\mathbf{u})$ minus 1. This means that, on any trial t we can, instead of projecting into the set $\mathcal{V}_t$, project into the set $\mathcal{V}_t(q_t)$ for some $q_t \in [1 - 1/T, 1]$ and add no more than one to the regret (by considering $\hat{\mathbf{u}}$ as the comparator instead of $\mathbf{u}$).

So the problem (for the projection step at time t if necessary) is now to project into the set of all $\{\mathbf{v} \mid \mathbf{v} \cdot \mathbf{c} \leq q_t\}$ for some arbitrary $q_t \in [1 - 1/T, 1]$. Following our use of Lagrange multipliers, this means that we need to find $\lambda > 0$ with $\sum_i c_t(i) w_t(i) \exp(-\lambda c_t(i)) \in [1 - 1/T, 1]$. So consider the function f defined by $f(\lambda') := \sum_i c_t(i) w_t(i) \exp(-\lambda' c_t(i))$.

Consider $\lambda' := ZE \ln(ZE)$ and take any $i \in [E]$. Since $w_t(i) \leq Z$ we have that when $c_t(i) < 1/ZE$ then $c_t(i) w_t(i) \exp(-\lambda' c_t(i)) \leq c_t(i) w_t(i) < 1/E$ and that when $c_t(i) \geq 1/ZE$ then $c_t(i) w_t(i) \exp(-\lambda' c_t(i)) \leq Z \exp(-\lambda'/ZE) = 1/E$. This implies that $f(\lambda') \leq 1$ and hence (since f is monotonic decreasing) an acceptable λ lies in $[0, ZE \ln(ZE)]$.

For general λ' we note that $\nabla f(\lambda') = -\sum_i c_t(i)^2 w_t(i) \exp(-\lambda' c_t(i)) \geq -f(\lambda')$. This means that

$|\nabla f(\lambda)| \leq 1$. Since the length of the interval $[1 - 1/T, 1]$ is $1/T$ this means that the length of the interval containing acceptable values of λ is at least $1/T$.

So we have shown that either $\lambda = ZE \ln(ZE)$ is acceptable or the range of acceptable values of λ is of length $1/T$ and lies in $[0, ZE \ln(ZE)]$ (which has length $ZE \ln(ZE)$). The ratio of these lengths is $ZET \ln(ZE)$ so interval bisection will find an acceptable value of λ in $O(\ln(ZET \ln(ZE))) = O(\ln(EZT))$ steps.

So we have a time complexity $O(EK + E \ln(EZT))$ and we have only added 1 to the regret (although this additive factor can be made arbitrarily small).

B.2 Efficient implementation proof

We here prove the time complexity of Theorem 4.2. The per-trial time complexity of a direct implementation of CBA for this set of experts would be $\mathcal{O}(KN^2)$. We now show how to implement CBA in a per-trial time of only $\mathcal{O}(KN \ln(N))$. To do this first note that we can assume, without loss of generality, that for all $q, x, z \in \mathcal{X}$ with $x \neq z$ we have $d(q, x) \neq d(q, z)$ since ties can be broken arbitrarily and balls can be duplicated.

Given $x, z \in \mathcal{X}$, $a \in [K]$ and $t \in [T]$ we let $y_{t,a}(x, z) := w_t(i)$ and $\tilde{y}_{t,a}(x, z) := \tilde{w}_t(i)$ where i is the index of the expert corresponding to the ball-action pair with ball: $\{q \in \mathcal{X} \mid d(x, q) \leq d(x, z)\}$, and action a . Given $x, z \in \mathcal{X}$ let $\mathcal{E}(x, z) := \{q \in \mathcal{X} \mid d(x, q) \geq d(x, z)\}$. It is straightforward to derive the following equations for the quantities in CBA at trial $t \in [T]$. First we have:

$$\|c_t\|_1 = \sum_{a \in [K]} \sum_{x \in \mathcal{X}} \sum_{z \in \mathcal{E}(x, x_t)} y_{t,a}(x, z).$$

For all $x, z \in \mathcal{X}$ and $a \in [K]$ we have the following:

- If $\|c_t\|_1 \leq 1$ or $z \notin \mathcal{E}(x, x_t)$ then $\tilde{y}_{t,a}(x, z) = y_{t,a}(x, z)$.
- If $\|c_t\|_1 > 1$ and $z \in \mathcal{E}(x, x_t)$ then $\tilde{y}_{t,a}(x, z) = y_{t,a}(x, z) / \|c_t\|_1$.

For all $a \in [K]$ we have:

$$s_t(a) = \sum_{x \in \mathcal{X}} \sum_{z \in \mathcal{E}(x, x_t)} \tilde{y}_{t,a}(x, z).$$

Finally, for all $x, z \in \mathcal{X}$ and $a \in [K]$ we have the following:

$$y_{(t+1),a}(x, z) = \begin{cases} \tilde{y}_{t,a}(x, z) & \text{if } z \notin \mathcal{E}(x, x_t), \\ \tilde{y}_{t,a}(x, z) \exp(\eta e_t^i \cdot \hat{\mathbf{r}}_t) & \text{if } z \in \mathcal{E}(x, x_t). \end{cases}$$

Hence, to implement CBA we need, for each $x \in \mathcal{X}$ and $a \in [K]$, a data structure that implicitly maintains a function $h : \mathcal{X} \rightarrow \mathbb{R}^+$ and has the following two subroutines, that take parameters $q \in \mathcal{X}$ and $p \in \mathbb{R}_+$.

1. **QUERY**(q): Compute $\sum_{z \in \mathcal{E}(x, q)} h(z)$.
2. **UPDATE**(q, p): Set $h(z) \leftarrow ph(z)$ for all $z \in \mathcal{E}(x, q)$.

Now fix $x \in \mathcal{X}$ and $a \in [K]$. Let h be as above. On each trial $t \in [T]$ and for all $z \in \mathcal{X}$, $h(z)$ will start equal to $y_{t,a}(x, z)$ and change to $\tilde{y}_{t,a}(x, z)$ and then $y_{(t+1),a}(x, z)$ by applying the **UPDATE** subroutine.

Algorithm B.1: QUERY(q)

1. For all $i \in [n] \cup \{0\}$ let γ_i be the ancestor of q at depth i in $\mathcal{D}$
 2. Set $\sigma_n \leftarrow \psi(\gamma_n)\phi(\gamma_n)$
 3. Climb $\mathcal{D}$ from γ_{n-1} to γ_0 . When at γ_i do as follows:
 - (a) If $\gamma_{i+1} = \triangleleft(\gamma_i)$ then set $\sigma_i \leftarrow \phi(\gamma_i)(\sigma_{i+1} + \psi(\triangleright(\gamma_i))\phi(\triangleright(\gamma_i)))$
 - (b) If $\gamma_{i+1} = \triangleright(\gamma_i)$ then set $\sigma_i \leftarrow \phi(\gamma_i)\sigma_{i+1}$
 4. Return σ_0
-

Algorithm B.2: UPDATE(q, p)

1. For all $i \in [n] \cup \{0\}$ let γ_i be the ancestor of q at depth i in $\mathcal{D}$
 2. Descend $\mathcal{D}$ from γ_0 to γ_{n-1} . When at γ_i set:
 - (a) $\phi(\triangleleft(\gamma_i)) \leftarrow \phi(\gamma_i)\phi(\triangleleft(\gamma_i))$
 - (b) $\phi(\triangleright(\gamma_i)) \leftarrow \phi(\gamma_i)\phi(\triangleright(\gamma_i))$
 - (c) $\phi(\gamma_i) \leftarrow 1$
 3. For all $i \in [n-1] \cup \{0\}$, if $\gamma_{i+1} = \triangleleft(\gamma_i)$ then set $\phi(\triangleright(\gamma_i)) \leftarrow p\phi(\triangleright(\gamma_i))$
 4. Set $\phi(\gamma_n) \leftarrow p\phi(\gamma_n)$
 5. Climb $\mathcal{D}$ from γ_{n-1} to γ_0 . When at γ_i set:

$$\psi(\gamma_i) \leftarrow \psi(\triangleleft(\gamma_i))\phi(\triangleleft(\gamma_i)) + \psi(\triangleright(\gamma_i))\phi(\triangleright(\gamma_i))$$
-

We now show how to implement these subroutines implicitly in a time of $\mathcal{O}(\ln(N))$ as required. Without loss of generality, assume that $N = 2^n$ for some $n \in \mathbb{N}$. Our data structure is based on a balanced binary tree $\mathcal{D}$ whose leaves are the elements of $\mathcal{X}$ in order of increasing distance from x . This implies that for any $z \in \mathcal{X}$ we have that $\mathcal{E}(x, z)$ is the set of leaves that do not lie on the left of z . Given a node $v \in \mathcal{D}$ we let $\uparrow(v)$ be the set of ancestors of v and let $\downarrow(v)$ be the set of all $z \in \mathcal{X}$ which are descendants of v . For any internal node v let $\triangleleft(v)$ and $\triangleright(v)$ be the left and right children of v respectively.

We maintain functions $\phi, \psi : \mathcal{D} \rightarrow \mathbb{R}_+$ such that for all $v \in \mathcal{D}$ we have:

$$\psi(v) \prod_{v' \in \uparrow(v)} \phi(v') = \sum_{z \in \downarrow(v)} h(z). \quad (\text{B.12})$$

The pseudo-code for the subroutines QUERY and UPDATE are given in Algorithms B.1 and B.2 respectively. We now prove their correctness. We first consider the QUERY subroutine with parameter $q \in \mathcal{X}$. From Equation (B.12) we see that, by (reverse) induction on $i \in [n] \cup \{0\}$, we have:

$$\sigma_i \prod_{v' \in \uparrow(\gamma_i) \setminus \{\gamma_i\}} \phi(v') = \sum_{z \in \downarrow(\gamma_i) \cap \mathcal{E}(x, q)} h(z).$$

Since γ_0 is the root of $\mathcal{D}$, we have $\sigma_0 = \sum_{z \in \mathcal{E}(x, q)} h(z)$ as required. Now consider the UPDATE subroutine with parameters $q \in \mathcal{X}$ and $p \in \mathbb{R}_+$. Let h be the implicitly maintained function before

the subroutine is called. For Equation (B.12) to hold after the subroutine is called we need:

$$\psi(v) \prod_{v' \in \uparrow(v)} \phi(v') = \sum_{z \in \downarrow(v)} h'(z). \quad (\text{B.13})$$

where for all $z \in \mathcal{X}$ we have:

$$h'(z) := \mathbb{I}\{z \notin \mathcal{E}(x, q)\} h(z) + \mathbb{I}\{z \in \mathcal{E}(x, q)\} p h(z).$$

We shall now show that Equation (B.13) does indeed hold after the subroutine is called, which will complete the proof. To show this we consider each step of the subroutine in turn. After Step 2 we have (via induction) that:

- For all $v \in \uparrow(q)$ we have $\phi(v) = 1$.
- For all $v \in \mathcal{D} \setminus \uparrow(q)$ we have:

$$\psi(v) \prod_{v' \in \uparrow(v)} \phi(v') = \sum_{z \in \downarrow(v)} h(z).$$

So, since $\mathcal{E}(x, q)$ is the set of all $z \in \mathcal{X}$ that do not lie to the left of q in $\mathcal{D}$ we have that, after Step 4 of the algorithm, the following holds:

- For all $v \in \uparrow(q)$ we have $\phi(v) = 1$,
- For all $v \in \mathcal{D} \setminus \uparrow(q)$ we have:

$$\psi(v) \prod_{v' \in \uparrow(v)} \phi(v') = \sum_{z \in \downarrow(v)} h'(z).$$

Hence, by induction, we have that, after Step 5 of the algorithm, it is the case that for all $v \in \uparrow(q)$ we have: $\psi(v) = \sum_{z \in \downarrow(v)} h'(z)$. So since $\phi(v) = 1$ for all $v \in \uparrow(q)$ and Step 5 does not alter $\phi(v)$ or $\psi(v)$ for any $v \in \mathcal{D} \setminus \uparrow(q)$ we have Equation (B.13). ■

B.3 Lower bound proof

Proposition B.1. *Take any learning algorithm. Given any basis $\mathcal{B}$ and any $M \in \mathbb{N}$ then for any sequence of disjoint basis elements $\langle \mathcal{B}_j \mid j \in [M] \rangle$ there exists a sequence of corresponding actions $\langle b_j \in [K] \mid j \in [M] \rangle$ such that an adversary can force:*

$$\sum_{t \in [T]} \sum_{j \in [M]} \mathbb{I}\{x_t \in \mathcal{B}_j\} r_t(b_j) - \sum_{t \in [T]} \mathbb{E}[r_t(a_t)] \in \Omega(\sqrt{MKT})$$

Proof. In this scenario, at each time step, either a single expert (i.e., the basis element containing the current context x_t) is active, making predictions based on its label, or no expert is active, prompting the learner to abstain and thus incur zero reward or cost.

Therefore we define $T' = \{t \in [T] \mid \sum_{j \in [M]} \mathbb{I}\{x_t \in \mathcal{B}_j\} = 1\}$ as the set of timesteps in which the learner is going to play. Since the concept of abstention is that our algorithm is not going to pay anything for the timesteps in which we abstain, we can see that:

$$\sum_{t \in [T]} \sum_{j \in [M]} \mathbb{I}\{x_t \in \mathcal{B}_j\} r_t(b_j) - \sum_{t \in [T]} \mathbb{E}[r_t(a_t)] = \sum_{t \in T'} r_t(b_j) - \sum_{t \in T'} \mathbb{E}[r_t(a_t)],$$

For any ball $j \in [M]$, we define $T_j = \{t \in [T'] \mid \mathbb{I}\{x_t \in \mathcal{B}_j\}\}$. Following the ideas of Seldin and Lugosi (2016), for any of the sets T_j we can create a multi-armed bandit instance as the one described in the lower bound by Auer et al. (2002). Note that in the lower bound construction, the abstention arm would be a forehand known suboptimal arm, which results in a lower bound of the order $c\sqrt{(K-1)T}$, for the constant $c = \frac{\sqrt{2}-1}{\sqrt{32 \ln(4/3)}} > 0$. Since the presented context x_t is chosen adversarially at each time step, we can ensure that each basis element is activated for $|T'|/M$ time steps, obtaining:

$$\begin{aligned} \sum_{j \in [M]} \left(\sum_{s \in T'_j} r_s(b_j) - \sum_{s \in T'_j} \mathbb{E}[r_s(a_s)] \right) &\geq \sum_{j \in [M]} c\sqrt{(K-1)|T'_j|} \\ &= \sum_{j \in [M]} c\sqrt{(K-1)|T'|/M} \\ &= c\sqrt{M(K-1)|T'|} \end{aligned}$$

As we can choose $|T'|$ to be any fraction of T , we end up with the desired lower bound of the order $\Omega(\sqrt{MKT})$, which matches, up to logarithmic factors, the cumulative reward bound presented in Theorem 4.2. $\square$

B.4 Overlapping balls extension

In this section, we present the theorem that allows us to present the results of overlapping balls as expressed in Section 4.5.2. Note that Theorem 4.2 is the special case of Theorem B.1 when the balls are disjoint and $u_j = 1$ for all $j \in [M]$.

Theorem B.1. *Let $M \in \mathbb{N}$ and $\{(B_j, b_j, u_j) \mid j \in [M]\}$ be any sequence such that B_j is a ball, $b_j \in [K]$ is an action, and $u_j \in [0, 1]$ is such that for all $x \in \mathcal{X}$ we have:*

$$\sum_{j \in [M]} \mathbb{I}\{x \in B_j\} u_j \leq 1.$$

For all $t \in [T]$ define:

$$r_t^* := \sum_{j \in [M]} \mathbb{I}\{x_t \in B_j\} u_j r_t(b_j),$$

which represents the reward of the policy induced by $\{(B_j, b_j, u_j) \mid j \in [M]\}$ on trial t . The regret of CBA, with the set of experts given in Section 4.5.2 and with correctly tuned parameters, is then bounded by:

$$\sum_{t \in [T]} r_t^* - \sum_{t \in [T]} \mathbb{E}[r_t(a_t)] \in \mathcal{O} \left(\sqrt{\ln(KN)KT \sum_{j \in [M]} u_j} \right).$$

Its per-trial time complexity is:

$$\mathcal{O}(KN \ln(N)) .$$

Proof. Direct from Theorem 4.1 using the experts (with efficient implementation) given in Section 4.5.2 $\square$

B.5 The details of the graph bases

This section expands the definition and explanations for the bases we used in the Experiment. Remember that we refer to any set of experts that correspond to set-action pairs of the form $(B, k) \in 2^{\mathcal{X}} \times [K]$ as a *basis elements*, and a set of basis elements as *basis*.

B.5.1 p -seminorm balls on graphs

As we see in Sec. 4.5.2, the CBA seems to work only for vector data. However, in the following sections, we explore how our CBA algorithm can be applied to graph data by creating a ball structure over the graph.

We first introduce the notations of a graph. A graph is a pair of *nodes* $V := [N]$ and *edges* E . An edge connects two nodes, and we assume that our graph is *undirected* and *weighted*. For each edge $\{i, j\} \in E$, we denote its weight by c_{ij} . For convenience, for each pair of nodes i, j with $\{i, j\} \notin E$, we define $c_{ij} = 0$.

To form a ball over a graph, a family of metrics we are particularly interested in is given by p -norms on a given graph G . Let

$$d_p(i, j) := \left(\min_{\substack{\mathbf{u} \in \mathbb{R}^N \\ u_i - u_j = 1}} \sum_{s, t \in V} c_{st} |u_s - u_t|^p \right)^{-1/p} . \quad (\text{B.14})$$

which is a well-defined metric for $p \in [1, \infty)$ if the graph is connected and may be defined for $p = \infty$ by taking the appropriate limits. When $p = 2$ this is the square root of the *effective resistance* circuit between nodes i and j which comes from interpreting the graph as an electric circuit where the edges are unit resistors and the denominator of Equation (B.14) is the power required to maintain a unit voltage difference between u and v (Doyle and Snell, 1984). More generally, $d_p(i, j)^p$ is known as p -(effective) resistance (Herbster and Lever, 2009, Alamgir and von Luxburg, 2011, Saito and Herbster, 2023). When $p \in \{1, 2, \infty\}$ there are natural interpretation of the p -resistance. In the case of $p = 1$, we have that the effective is equal to one over the number of edge-disjoint paths between i and j which is equivalently one over the minimal cut that separates i from j . When $p = 2$ it is the effective resistance as discussed above. And finally when $p = \infty$ we have that d_∞ is the geodesic distance (shortest path) between i and j . Note that, interestingly, there are at most $2N$ distinct balls for d_1 ; as opposed to the general bound $\mathcal{O}(N^2)$ on the number of metric balls. This follows since d_1 is an *ultrametric*. A nice feature of metric balls is that they are ordinal, i.e., we can take an increasing function of the distance and the distinct are unchanged. The time complexity for each ball is as follows. For d_1 ball, we compute every pair of distance in $\mathcal{O}(N^3)$ using the Gomory-Hu tree (Gomory and Hu, 1961). For d_2 ball, it is actually enough to compute the pseudoinverse of

graph Laplacian once, which costs $\mathcal{O}(N^3)$ (Doyle and Snell, 1984). For d_∞ ball, we can compute every pair of distance in $\mathcal{O}(N^3)$ by Floyd–Warshall algorithm (Floyd, 1962).

B.5.2 Community detection bases

In this section, we consider only bases formed via a set of subsets (a.k.a clusters) $C \subseteq 2^{[N]}$. Each of these subsets induces K basis elements: one for each action $a \in [K]$. Specifically, the basis element $\beta : [N] \rightarrow [K]$ corresponding to the pair (C, a) is such that $\beta(x)$ is equal to a whenever $x \in C$ and equal to $\square$ otherwise. Hence, in this section, we equate a basis with a set of subsets of $[N]$.

We can compute a basis for a given graph $G = (V, E)$ using community detection algorithms. Community detection is one of the most well-studied operations for graphs, where the goal is to find a partition $\{C_1, \dots, C_q\}$ of V (i.e., $\bigcup_{i=1}^q C_i = V$ and $C_i \cap C_j = \emptyset$ for $i \neq j$) so that each C_i is densely connected internally but sparsely connected to the rest of the graph (Fortunato, 2010). There are many community detection algorithms, all of which can be used here, but the most popular algorithm is the Louvain method (Blondel et al., 2008). We briefly describe how this algorithm works. The algorithm starts with an initial partition $\{\{v\} \mid v \in V\}$ and aggregates the clusters iteratively: For each $v \in V$, compute the gain when moving v from its current cluster to its neighbors' clusters and indeed move it to a cluster with the maximum gain (if the gain is positive). Note that the gain is evaluated using *modularity*, i.e., the most popular quality function for community detection (Newman and Girvan, 2004). The algorithm repeats this process until no movement is possible. Then the algorithm aggregates each cluster to a single super node (with appropriate addition of self-loops and change of edge weights) and repeats the above process on the coarse graph as long as the coarse graph is updated. Finally, the algorithm outputs the partition of V in which each cluster corresponds to each super node in the latest coarse graph. Note that it is widely recognized that the Louvain method works in $\mathcal{O}(N \log N)$ in practice (Traag, 2015).

To obtain a finer-grained basis, we apply the so-called greedy peeling algorithm for each C_i in the output of the Louvain method. For $C_i \subseteq V$ and $v \in C_i$, we denote by $d_{C_i}(v)$ the degree of v in the induced subgraph $G[C_i]$. For $G[C_i]$, the greedy peeling iteratively removes a node with the smallest degree in the currently remaining graph and obtains a sequence of node subsets from C_i to a singleton. Specifically, it works as follows: Set $j \leftarrow |C_i|$ and $C_i^{(j)} \leftarrow C_i$. For each $j = |C_i|, \dots, 2$, compute $v_{\min} \in \arg \min \{d_{C_i^{(j)}}(v) \mid v \in C_i^{(j)}\}$ and $C_i^{(j-1)} \leftarrow C_i^{(j)} \setminus \{v_{\min}\}$. Using a sophisticated data structure, this algorithm runs in linear time (Lanciano et al., 2024).

In summary, our community detection basis is the collection of node subsets $\{C_i^{(j)} \mid i = 1, \dots, q, j = 1, \dots, |C_i|\}$ together with $\{\{v\} \mid v \in V\}$ for completeness.

B.5.3 Graph convexity bases

An alternative to metric balls and communities are, for example, (geodesically) convex sets in a graph. They correspond to the inductive bias that if two nodes prefer the same action, then also the nodes on a shortest path between the two tend to prefer the same action. Geodesically convex sets are well-studied (van De Vel, 1993, Pelayo, 2013) and have been recently used in various learning settings on graphs (Bressan, Cesa-Bianchi, Lattanzi, and Paudice, 2021, Thiessen and Gärtner, 2021). Similarly to convex sets in the Euclidean space, a set C of nodes is *convex* if the nodes of any shortest path with endpoints in C are in C , as well. More formally, the (geodesic)

interval $I(u, v) = \{x \in V : x \text{ is on a shortest path between } u \text{ and } v\}$ of two nodes u and v contains all the nodes on a shortest path between them. For a set of node A we define $I(A) = \cup_{a,b \in A} I(a, b)$ as a shorthand notation for the union of all pairwise intervals in A . A set A is (geodesically) convex iff $I(A) = A$ and the *convex hull* $\text{conv}(A)$ of a set A is the (unique) smallest convex set containing A . Note that for $u, v \in V$, $I(u, v)$ and $\text{conv}(\{u, v\})$ are typically different sets. Indeed, $I(u, v)$ is in general non-convex, as nodes on a shortest path between two nodes in $I(u, v)$ (except for u, v) are not necessarily contained in $I(u, v)$. As the total number of convex sets can be exponential in N , e.g., all subsets of a complete subgraph are convex, we consider the basis consisting of all intervals: $I(u, v)$ for $u, v \in [N]$. This involves $\mathcal{O}(N^2)$ basis elements, each of size $\mathcal{O}(N)$. With a simple modification of the Floyd Warshall (Floyd, 1962) algorithm, computing the interval basis takes $\mathcal{O}(N^3)$ time complexity.

Appendix C

Appendix - Sum-Max

C.1 Analysis Proofs (Proof Of Theorem 5.2)

C.1.1 Lemma 5.1

Lemma 5.1. *Given a function $r : 2^{[K]} \rightarrow \mathbb{R}$ there exists a unique subset decomposition $\tilde{r}$ of r .*

Proof. For all $k \in [K] \cup \{0\}$ define $\mathcal{V}_k := \{\mathcal{S} \in 2^{[K]} \mid k \leq |\mathcal{S}|\}$. We take the inductive hypothesis that for all $k \in [K] \cup \{0\}$ there exists a unique function $\tilde{r}_k : \mathcal{V}_k \rightarrow \mathbb{R}$ such that for all $\mathcal{Q} \in \mathcal{V}_k$ we have

$$r(\mathcal{Q}) = \sum_{\mathcal{S} \in \mathcal{V}_k} \mathbb{I}\{\mathcal{Q} \subseteq \mathcal{S}\} \tilde{r}_k(\mathcal{S}).$$

We will prove the inductive hypothesis via reverse induction on k (i.e. from $k = K$ to $k = 0$).

The inductive hypothesis holds for $k = K$ since the only element of $\mathcal{V}_K$ is $[K]$ so we must have $\tilde{r}_K([K]) := r([K])$

Now suppose, for some $i \in [K]$, the inductive hypothesis holds when $k = i$. Now consider the case that $k = i - 1$. Note that for all $\mathcal{Q} \in \mathcal{V}_i$ and $\mathcal{S} \in \mathcal{V}_{i-1} \setminus \mathcal{V}_i$ we must have that $\mathcal{Q} \not\subseteq \mathcal{S}$ and hence we must have that:

$$r(\mathcal{Q}) = \sum_{\mathcal{S} \in \mathcal{V}_i} \mathbb{I}\{\mathcal{Q} \subseteq \mathcal{S}\} \tilde{r}_{i-1}(\mathcal{S}),$$

so, by the inductive hypothesis, the restriction of $\tilde{r}_{i-1}$ to $\mathcal{V}_i$ is equal to $\tilde{r}_i$. Now choose some arbitrary $\mathcal{Q} \in \mathcal{V}_{i-1} \setminus \mathcal{V}_i$ and define:

$$v(\mathcal{Q}) := \sum_{\mathcal{S} \in \mathcal{V}_i} \mathbb{I}\{\mathcal{Q} \subseteq \mathcal{S}\} \tilde{r}_{i-1}(\mathcal{S})$$

which, by above, is uniquely defined. Note that for all $\mathcal{S} \in \mathcal{V}_{i-1} \setminus \mathcal{V}_i$ we have that $\mathcal{Q} \subseteq \mathcal{S}$ if and only if $\mathcal{S} = \mathcal{Q}$ and hence we must have that:

$$r(\mathcal{Q}) = \sum_{\mathcal{S} \in \mathcal{V}_i} \mathbb{I}\{\mathcal{Q} \subseteq \mathcal{S}\} \tilde{r}_{i-1}(\mathcal{S}) + \tilde{r}_{i-1}(\mathcal{Q}) = v(\mathcal{Q}) + \tilde{r}_{i-1}(\mathcal{Q}),$$

so that $\tilde{r}_{i-1}(\mathcal{Q}) = r(\mathcal{Q}) - v(\mathcal{Q})$ which is unique.

We have hence shown that the inductive hypothesis holds for $k = i - 1$ and hence holds always. Noting that $\mathcal{V}_0 = 2^{[K]}$ we then get the result by necessarily setting $\tilde{r} = \tilde{r}_0$. $\square$

C.1.2 Lemma 5.2

Lemma 5.2. For all $r : 2^K \rightarrow \mathbb{R}$ and all $\mathbf{q} \in \Delta_K$ we have $\Phi^r(\mathbf{q}) = \mathbb{E}[r(\mathcal{B}(\mathbf{q}))]$.

Proof. Let $\tilde{r}$ be a subset decomposition of r . We have

$$\begin{aligned} \mathbb{E}[r(\mathcal{B}(\mathbf{q}))] &= \sum_{\mathcal{S} \subseteq [K]} \tilde{r}(\mathcal{S}) \mathbb{P}[\mathcal{B}(\mathbf{q}) \subseteq \mathcal{S}] \\ &= \sum_{\mathcal{S} \subseteq [K]} \tilde{r}(\mathcal{S}) \prod_{j \in [M]} \mathbb{P}[b_j(\mathbf{q}) \in \mathcal{S}] \\ &= \sum_{\mathcal{S} \subseteq [K]} \tilde{r}(\mathcal{S}) \prod_{j \in [M]} \sum_{i \in [K]} \mathbb{I}\{i \in \mathcal{S}\} q(i) \\ &= \sum_{\mathcal{S} \subseteq [K]} \tilde{r}(\mathcal{S}) \left(\sum_{i \in [K]} \mathbb{I}\{i \in \mathcal{S}\} q(i) \right)^M \\ &= \Phi^r(\mathbf{q}) \end{aligned}$$

as required. □

C.1.3 Lemma 5.3

Lemma 5.3. For all $r : 2^K \rightarrow \mathbb{R}$, all $\mathbf{q} \in \Delta_K$ and all $i \in [K]$ we have

$$\partial_i \Phi^r(\mathbf{q}) = \mathbb{E} \left[\frac{r(\mathcal{B}(\mathbf{q}))}{q(i)} \sum_{j \in [M]} \mathbb{I}\{b_j(\mathbf{q}) = i\} \right].$$

Proof. Let $\tilde{r}$ be a subset decomposition of r . For all $\mathbf{q}' \in \mathbb{R}^K$ and $\mathcal{S} \subseteq [K]$ define

$$\Lambda^{\mathcal{S}}(\mathbf{q}') := \left(\sum_{k \in [K]} \mathbb{I}\{k \in \mathcal{S}\} q(k)' \right)^M.$$

Fix some $j \in [M]$. Note that

$$\begin{aligned} \partial_i \Lambda^{\mathcal{S}}(\mathbf{q}) &= M \mathbb{I}\{i \in \mathcal{S}\} \left(\sum_{k \in [K]} \mathbb{I}\{k \in \mathcal{S}\} q(k) \right)^{M-1} \\ &= M \mathbb{I}\{i \in \mathcal{S}\} \prod_{j' \in [M] \setminus \{j\}} \sum_{k \in [K]} \mathbb{I}\{k \in \mathcal{S}\} q(k) \\ &= M \mathbb{I}\{i \in \mathcal{S}\} \prod_{j' \in [M] \setminus \{j\}} \mathbb{P}[b_{j'}(\mathbf{q}) \in \mathcal{S}] \\ &= \frac{M}{q(i)} \mathbb{P}[b_j(\mathbf{q}) = i] \mathbb{I}\{i \in \mathcal{S}\} \prod_{j' \in [M] \setminus \{j\}} \mathbb{P}[b_{j'}(\mathbf{q}) \in \mathcal{S}] \\ &= \frac{M}{q(i)} \mathbb{P}[(b_j(\mathbf{q}) = i) \wedge (i \in \mathcal{S})] \prod_{j' \in [M] \setminus \{j\}} \mathbb{P}[b_{j'}(\mathbf{q}) \in \mathcal{S}] \\ &= \frac{M}{q(i)} \mathbb{P}[(b_j(\mathbf{q}) = i) \wedge (b_j(\mathbf{q}) \in \mathcal{S})] \prod_{j' \in [M] \setminus \{j\}} \mathbb{P}[b_{j'}(\mathbf{q}) \in \mathcal{S}] \end{aligned}$$

$$\begin{aligned}
 &= \frac{M}{q(i)} \mathbb{P}[(b_j(\mathbf{q}) = i) \wedge (\forall j' \in [M], b_{j'}(\mathbf{q}) \in \mathcal{S})] \\
 &= \frac{M}{q(i)} \mathbb{P}[(b_j(\mathbf{q}) = i) \wedge (\mathcal{B}(\mathbf{q}) \subseteq \mathcal{S})] \\
 &= \frac{M}{q(i)} \mathbb{E}[\mathbb{I}\{b_j(\mathbf{q}) = i\} \mathbb{I}\{\mathcal{B}(\mathbf{q}) \subseteq \mathcal{S}\}],
 \end{aligned}$$

so since:

$$\Phi^r(\mathbf{q}) = \sum_{\mathcal{S} \subseteq [K]} \tilde{r}(\mathcal{S}) \Lambda^{\mathcal{S}}(\mathbf{q}),$$

we have

$$\begin{aligned}
 \partial_i \Phi^r(\mathbf{q}) &= \sum_{\mathcal{S} \subseteq [K]} \tilde{r}(\mathcal{S}) \partial_i \Lambda^{\mathcal{S}}(\mathbf{q}) \\
 &= \frac{M}{q(i)} \sum_{\mathcal{S} \subseteq [K]} \tilde{r}(\mathcal{S}) \mathbb{E}[\mathbb{I}\{b_j(\mathbf{q}) = i\} \mathbb{I}\{\mathcal{B}(\mathbf{q}) \subseteq \mathcal{S}\}] \\
 &= \frac{M}{q(i)} \mathbb{E} \left[\mathbb{I}\{b_j(\mathbf{q}) = i\} \sum_{\mathcal{S} \subseteq [K]} \tilde{r}(\mathcal{S}) \mathbb{I}\{\mathcal{B}(\mathbf{q}) \subseteq \mathcal{S}\} \right] \\
 &= \frac{M}{q(i)} \mathbb{E}[\mathbb{I}\{b_j(\mathbf{q}) = i\} r(\mathcal{B}(\mathbf{q}))].
 \end{aligned}$$

Summing over all $j \in [M]$ and dividing by M then gives us

$$\partial_i \Phi^r(\mathbf{q}) = \mathbb{E} \left[\frac{r(\mathcal{B}(\mathbf{q}))}{q(i)} \sum_{j \in [M]} \mathbb{I}\{b_j(\mathbf{q}) = i\} \right]$$

as required. $\square$

C.1.4 Lemma 5.4

Lemma 5.4. *For all pseudo-concave set functions $r : 2^K \rightarrow \mathbb{R}$ we have that Φ^r is concave over the simplex Δ_K .*

Proof. Choose any $\mathbf{q} \in \Delta_K$. Define $\langle b'_j(\mathbf{q}) \mid j \in [M-2] \rangle$ to be a sequence of stochastic quantities drawn i.i.d. at random from (the probability distribution characterised by) $\mathbf{q}$. In addition, let:

$$\mathcal{B}'(\mathbf{q}) := \{b'_j(\mathbf{q}) \mid j \in [M-2]\}.$$

Direct from the definition of Φ^r we have, for all $i, i' \in [K]$, that

$$\begin{aligned}
 \partial_i \partial_{i'} \Phi^r &= \sum_{\mathcal{S} \subseteq [K]} \tilde{r}(\mathcal{S}) \mathbb{I}\{i \in \mathcal{S}\} \mathbb{I}\{i' \in \mathcal{S}\} \left(\sum_{k \in [K]} \mathbb{I}\{k \in \mathcal{S}\} q(k) \right)^{M-2} \\
 &= \sum_{\mathcal{S} \subseteq [K]} \tilde{r}(\mathcal{S}) \mathbb{I}\{i \in \mathcal{S}\} \mathbb{I}\{i' \in \mathcal{S}\} \prod_{j \in [M-2]} \sum_{k \in [K]} \mathbb{I}\{k \in \mathcal{S}\} q(k) \\
 &= \sum_{\mathcal{S} \subseteq [K]} \tilde{r}(\mathcal{S}) \mathbb{I}\{i \in \mathcal{S}\} \mathbb{I}\{i' \in \mathcal{S}\} \prod_{j \in [M-2]} \mathbb{P}[b'_j(\mathbf{q}) \in \mathcal{S}]
 \end{aligned}$$

$$\begin{aligned}
 &= \sum_{\mathcal{S} \subseteq [K]} \tilde{r}(\mathcal{S}) \mathbb{I}\{i \in \mathcal{S}\} \mathbb{I}\{i' \in \mathcal{S}\} \mathbb{P}[\mathcal{B}'(\mathbf{q}) \subseteq \mathcal{S}] \\
 &= \sum_{\mathcal{S} \subseteq [K]} \tilde{r}(\mathcal{S}) \mathbb{P}[\mathcal{B}'(\mathbf{q}) \cup \{i, i'\} \subseteq \mathcal{S}] \\
 &= \mathbb{E} \left[\sum_{\mathcal{S} \subseteq [K]} \tilde{r}(\mathcal{S}) \mathbb{I}\{\mathcal{B}'(\mathbf{q}) \cup \{i, i'\} \subseteq \mathcal{S}\} \right] \\
 &= \mathbb{E}[r(\mathcal{B}'(\mathbf{q}) \cup \{i, i'\})] \\
 &= \sum_{\mathcal{S} \subseteq [K]} \mathbb{P}[\mathcal{B}'(\mathbf{q}) = \mathcal{S}] r(\mathcal{S} \cup \{i, i'\}) \\
 &= \sum_{\mathcal{S} \subseteq [K]} \mathbb{P}[\mathcal{B}'(\mathbf{q}) = \mathcal{S}] U_{i, i'}^{r, \mathcal{S}}.
 \end{aligned}$$

So for all $\mathbf{x} \in \mathbb{R}^K$ with $\mathbf{x} \cdot \mathbf{1} = 0$ we have

$$\begin{aligned}
 \mathbf{x}^\top (\nabla^2 \Phi^r(\mathbf{q})) \mathbf{x} &= \sum_{i, i' \in [K]} x_i (\partial_i \partial_{i'} \Phi^r) x_{i'} \\
 &= \sum_{i, i' \in [K]} x_i x_{i'} \sum_{\mathcal{S} \subseteq [K]} \mathbb{P}[\mathcal{B}'(\mathbf{q}) = \mathcal{S}] U_{i, i'}^{r, \mathcal{S}} \\
 &= \sum_{\mathcal{S} \subseteq [K]} \mathbb{P}[\mathcal{B}'(\mathbf{q}) = \mathcal{S}] \sum_{i, i' \in [K]} x_i U_{i, i'}^{r, \mathcal{S}} x_{i'} \\
 &= \sum_{\mathcal{S} \subseteq [K]} \mathbb{P}[\mathcal{B}'(\mathbf{q}) = \mathcal{S}] (\mathbf{x}^\top \mathbf{U}^{r, \mathcal{S}} \mathbf{x}) \\
 &\leq 0,
 \end{aligned}$$

which means that Φ^r is concave on Δ_K as required. $\square$

C.1.5 Lemma 5.6

Lemma 5.6. Let $\mathcal{S} \subseteq [K]$ with $\mathcal{S} \neq \emptyset$, $r : 2^{[K]} \rightarrow \mathbb{R}$ be a pseudo-submodular function, and $\mathcal{Z} \subseteq [K]$ be a set formed by drawing M elements uniformly at random (with replacement) from $\mathcal{S}$. Then we have

$$\mathbb{E}[r(\mathcal{Z}) - r(\emptyset)] \geq \left(1 - \left(\frac{|\mathcal{S}| - 1}{|\mathcal{S}|} \right)^M \right) (r(\mathcal{S}) - r(\emptyset)).$$

Proof. Without loss of generality assume that $r(\emptyset) = 0$.

We prove by induction on m that the lemma holds whenever $M \leq m$. In the case that $m = 0$ we have $\mathbb{E}[r(\mathcal{Z})] = r(\emptyset) = 0$ and $M = 0$ so the result holds. Now assume that it holds for all $M \leq m$ and consider the case that $M = m + 1$.

Since r is pseudo-submodular choose $i \in \mathcal{S}$ such that

$$r(\mathcal{Q} \cup \{i\}) - r(\mathcal{Q}) \geq r(\mathcal{S}) - r(\mathcal{S} \setminus \{i\}) \tag{C.1}$$

for all $\mathcal{Q} \subseteq \mathcal{S} \setminus \{i\}$. Define $\sigma := |\mathcal{S}|$ and

$$\phi := r(\mathcal{S}) - r(\mathcal{S} \setminus \{i\}). \tag{C.2}$$

Let $\langle z_s \mid s \in [M] \rangle$ be a sequence of M elements drawn uniformly at random from S such that $\mathcal{Z} = \{z_s \mid s \in [M]\}$. Define

$$\mu := \sum_{s \in [M]} \mathbb{I}\{z_s \neq i\}.$$

For all $j \in [M] \cup \{0\}$ let $\mathcal{Z}_j$ be a set formed by sampling j actions independently and uniformly at random from $S \setminus \{i\}$.

Note that by the inductive hypothesis, we have

$$\begin{aligned} \mathbb{E}[\mathbb{I}\{i \notin \mathcal{Z}\} r(\mathcal{Z})] &= \mathbb{P}[i \notin \mathcal{Z}] \mathbb{E}[r(\mathcal{Z}) \mid i \notin \mathcal{Z}] \\ &= \mathbb{P}[i \notin \mathcal{Z}] \left(1 - \left(\frac{|\mathcal{S} \setminus \{i\}| - 1}{|\mathcal{S} \setminus \{i\}|} \right)^M \right) r(\mathcal{S} \setminus \{i\}) \\ &= \mathbb{P}[i \notin \mathcal{Z}] \left(1 - \left(\frac{\sigma - 2}{\sigma - 1} \right)^M \right) r(\mathcal{S} \setminus \{i\}) \\ &= \mathbb{P}[\mu = M] \left(1 - \left(\frac{\sigma - 2}{\sigma - 1} \right)^M \right) r(\mathcal{S} \setminus \{i\}). \end{aligned} \quad (\text{C.3})$$

Note also that

$$\mathbb{E}[\mathbb{I}\{i \in \mathcal{Z}\} r(\mathcal{Z})] = \sum_{j \in [m] \cup \{0\}} \mathbb{P}[\mu = j] \mathbb{E}[r(\mathcal{Z}_j \cup \{i\})]. \quad (\text{C.4})$$

By equations (C.1) and (C.2) and the inductive hypothesis we have, for all $j \in [m] \cup \{0\}$, that

$$\begin{aligned} \mathbb{E}[r(\mathcal{Z}_j \cup \{i\})] &\geq \mathbb{E}[\phi + r(\mathcal{Z}_j)] \\ &= \phi + \mathbb{E}[r(\mathcal{Z}_j)] \\ &\geq \phi + \left(1 - \left(\frac{|\mathcal{S} \setminus \{i\}| - 1}{|\mathcal{S} \setminus \{i\}|} \right)^j \right) r(\mathcal{S} \setminus \{i\}) \\ &= \phi + \left(1 - \left(\frac{\sigma - 2}{\sigma - 1} \right)^j \right) r(\mathcal{S} \setminus \{i\}). \end{aligned} \quad (\text{C.5})$$

We also have that

$$\sum_{j \in [m] \cup \{0\}} \mathbb{P}[\mu = j] = \mathbb{P}[i \in \mathcal{Z}]. \quad (\text{C.6})$$

Substituting equations (C.5) and (C.6) into Equation (C.4) gives us

$$\mathbb{E}[\mathbb{I}\{i \in \mathcal{Z}\} r(\mathcal{Z})] = \mathbb{P}[i \in \mathcal{Z}] \phi + \sum_{j \in [m] \cup \{0\}} \mathbb{P}[\mu = j] \left(1 - \left(\frac{\sigma - 2}{\sigma - 1} \right)^j \right) r(\mathcal{S} \setminus \{i\}).$$

Adding this equation to Equation (C.3) gives us

$$\mathbb{E}[r(\mathcal{Z})] = \mathbb{P}[i \in \mathcal{Z}] \phi + \sum_{j \in [M] \cup \{0\}} \mathbb{P}[\mu = j] \left(1 - \left(\frac{\sigma - 2}{\sigma - 1} \right)^j \right) r(\mathcal{S} \setminus \{i\}). \quad (\text{C.7})$$

Take any $k \in \mathcal{S} \setminus \{i\}$. Note that

$$1 - \left(\frac{\sigma - 2}{\sigma - 1} \right)^j = 1 - (1 - 1/(\sigma - 1))^j = 1 - \mathbb{P}[k \notin \mathcal{Z}_j] = \mathbb{P}[k \in \mathcal{Z}_j],$$

so that

$$\begin{aligned} \sum_{j \in [M] \cup \{0\}} \mathbb{P}[\mu = j] \left(1 - \left(\frac{\sigma - 2}{\sigma - 1} \right)^j \right) &= \sum_{j \in [M] \cup \{0\}} \mathbb{P}[\mu = j] \mathbb{P}[k \in \mathcal{Z}_j] \\ &= \sum_{j \in [M] \cup \{0\}} \mathbb{P}[\mu = j] \mathbb{P}[k \in \mathcal{Z} \setminus \{i\} \mid \mu = j] \\ &= \sum_{j \in [M] \cup \{0\}} \mathbb{P}[\mu = j] \mathbb{P}[k \in \mathcal{Z} \mid \mu = j] \\ &= \mathbb{P}[k \in \mathcal{Z}]. \end{aligned}$$

Substituting into Equation (C.7) gives us:

$$\begin{aligned} \mathbb{E}[r(\mathcal{Z})] &\geq \mathbb{P}[i \in \mathcal{Z}] \phi + \mathbb{P}[k \in \mathcal{Z}] r(\mathcal{S} \setminus \{i\}) \\ &= \mathbb{P}[i \in \mathcal{Z}] (\phi + r(\mathcal{S} \setminus \{i\})) \\ &= \mathbb{P}[i \in \mathcal{Z}] r(\mathcal{S}) \\ &= (1 - \mathbb{P}[i \notin \mathcal{Z}]) r(\mathcal{S}) \\ &= (1 - (1 - 1/\sigma)^M) r(\mathcal{S}) \\ &= \left(1 - \left(\frac{|\mathcal{S}| - 1}{|\mathcal{S}|} \right)^M \right) r(\mathcal{S}). \end{aligned}$$

So the inductive hypothesis holds for all $M \in [m + 1]$ and hence holds always. $\square$

C.1.6 Lemma 5.7

Lemma 5.7. *Given any $\mathcal{S} \subseteq [K]$ and any pseudo-submodular set function $r : 2^{[K]} \rightarrow \mathbb{R}$ we have*

$$\Phi^r(\mathbf{p}^{\mathcal{S}}) \geq r(\emptyset) + \left(1 - \left(\frac{|\mathcal{S}| - 1}{|\mathcal{S}|} \right)^M \right) (r(\mathcal{S}) - r(\emptyset)).$$

Proof. Let $\mathcal{Z}$ be a set formed by drawing M elements i.i.d. with replacement from $\mathcal{S}$. Let z be an element drawn i.i.d. from $\mathcal{S}$. Let $\tilde{r}$ be a subset-decomposition of r . Note that for all $i \in [K]$ we have

$$p^{\mathcal{S}}(i) = \mathbb{P}[z = i].$$

Hence, we have

$$\Phi^r(\mathbf{p}^{\mathcal{S}}) = \sum_{\mathcal{Q} \subseteq [K]} \tilde{r}(\mathcal{Q}) \left(\sum_{i \in [K]} \mathbb{I}\{i \in \mathcal{Q}\} p^{\mathcal{S}}(i) \right)^M$$

$$\begin{aligned}
 &= \sum_{\mathcal{Q} \subseteq [K]} \tilde{r}(\mathcal{Q}) \left(\sum_{i \in [K]} \mathbb{I}\{i \in \mathcal{Q}\} \mathbb{P}[z = i] \right)^M \\
 &= \sum_{\mathcal{Q} \subseteq [K]} \tilde{r}(\mathcal{Q}) \mathbb{P}[z \in \mathcal{Q}]^M \\
 &= \sum_{\mathcal{Q} \subseteq [K]} \tilde{r}(\mathcal{Q}) \mathbb{P}[\mathcal{Z} \subseteq \mathcal{Q}] \\
 &= \sum_{\mathcal{Q} \subseteq [K]} \tilde{r}(\mathcal{Q}) \mathbb{E}[\mathbb{I}\{\mathcal{Z} \subseteq \mathcal{Q}\}] \\
 &= \mathbb{E} \left[\sum_{\mathcal{Q} \subseteq [K]} \tilde{r}(\mathcal{Q}) \mathbb{I}\{\mathcal{Z} \subseteq \mathcal{Q}\} \right] \\
 &= \mathbb{E}[r(\mathcal{Z})].
 \end{aligned}$$

So

$$\Phi^r(\mathbf{p}^S) - r(\emptyset) = \mathbb{E}[r(\mathcal{Z}) - r(\emptyset)],$$

Lemma 5.6 then gives us the result. $\square$

C.1.7 Lemma 5.8

Lemma 5.8. *For all trials $t \in [T]$ and Ψ_t as defined in Definition 5.8, we have*

$$\nabla \Psi_t(\mathbf{p}_t) = \mathbb{E}[\mathbf{g}_t \mid \mathbf{p}_t],$$

Proof. Take any $i \in [K]$. For any $j \in [M]$ we have

$$\begin{aligned}
 c_{t,i} &= p_{t,i} c_{t,i} / p_{t,i} \\
 &= \mathbb{P}[a_{t,j} = i \mid \mathbf{p}_t] c_{t,i} / p_{t,i} \\
 &= \mathbb{E}[\mathbb{I}\{a_{t,j} = i\} c_{t,i} / p_{t,i} \mid \mathbf{p}_t].
 \end{aligned}$$

So:

$$\begin{aligned}
 M c_{t,i} &= \sum_{j \in [M]} \mathbb{E}[\mathbb{I}\{a_{t,j} = i\} c_{t,i} / p_{t,i} \mid \mathbf{p}_t] \\
 &= \mathbb{E} \left[\frac{c_{t,i}}{p_{t,i}} \sum_{j \in [M]} \mathbb{I}\{a_{t,j} = i\} \mid \mathbf{p}_t \right].
 \end{aligned}$$

Hence, by Lemma 5.3, we have

$$\begin{aligned}
 \partial_i \Psi_t(\mathbf{p}_t) &= \partial_i \Phi^{r_t}(\mathbf{p}_t) - M c_{t,i} \\
 &= \mathbb{E} \left[\frac{r_t(\mathcal{B}(\mathbf{p}_t))}{p_{t,i}} \sum_{j \in [M]} \mathbb{I}\{b_j(\mathbf{p}_t) = i\} \right] - M c_{t,i}
 \end{aligned}$$

$$\begin{aligned}
&= \mathbb{E} \left[\frac{r_t(\mathcal{A}_t)}{p_{t,i}} \sum_{j \in [M]} \mathbb{I}\{a_{t,j} = i\} \middle| \mathbf{p}_t \right] - M c_{t,i} \\
&= \mathbb{E}[g_t(i) | \mathbf{p}_t]
\end{aligned}$$

as required. $\square$

C.1.8 Lemma 5.9

Lemma 5.9. *For all trials $t \in [T]$ we have*

$$\mathbb{E} \left[\sum_{i \in [K]} p_{t,i} g_t(i)^2 \right] \leq (1 + C)^2 M(K + M - 1).$$

Proof. Given $i \in [K]$ we have that

$$\begin{aligned}
\frac{\mathbb{E}[g_t(i)^2]}{(1 + C)^2} &= \frac{1}{(1 + C)^2} \mathbb{E} \left[(r_t(\mathcal{A}_t) - c_{t,i})^2 \sum_{j, j' \in [M]} \frac{\mathbb{I}\{a_{t,j} = i\} \mathbb{I}\{a_{t,j'} = i\}}{p_{t,i}^2} \right] \\
&\leq \sum_{j, j' \in [M]} \mathbb{E} \left[\frac{\mathbb{I}\{a_{t,j} = i\} \mathbb{I}\{a_{t,j'} = i\}}{p_{t,i}^2} \right] \\
&= \sum_{j \in [M]} \mathbb{E} \left[\frac{\mathbb{I}\{a_{t,j} = i\}}{p_{t,i}^2} \right] + \sum_{j, j' \in [M]} \mathbb{I}\{j \neq j'\} \mathbb{E} \left[\frac{\mathbb{I}\{a_{t,j} = i\} \mathbb{I}\{a_{t,j'} = i\}}{p_{t,i}^2} \right] \\
&= \sum_{j \in [M]} \frac{\mathbb{P}[a_{t,j} = i]}{p_{t,i}^2} + \sum_{j, j' \in [M]} \mathbb{I}\{j \neq j'\} \frac{\mathbb{P}[a_{t,j} = i] \mathbb{P}[a_{t,j'} = i]}{p_{t,i}^2} \\
&= \sum_{j \in [M]} \frac{1}{p_{t,i}} + \sum_{j, j' \in [M]} \mathbb{I}\{j \neq j'\} \\
&= \frac{M}{p_{t,i}} + M(M - 1),
\end{aligned}$$

and hence

$$\mathbb{E} \left[\sum_{i \in [K]} p_{t,i} g_t(i)^2 \right] = \sum_{i \in [K]} p_{t,i} \mathbb{E}[g_t(i)^2] \leq (1 + C)^2 M(K + M - 1)$$

as required. $\square$

C.1.9 Theorem 5.2

Theorem 5.2. *Given r_t is pseudo-concave and pseudo-submodular for all $t \in [T]$, then for any set $S \subseteq [K]$ with $S \neq \emptyset$ we have*

$$\sum_{t \in [T]} \mathbb{E}[\psi_t] \geq (1 - \alpha^M) \sum_{t \in [T]} \hat{r}_t(S) - \frac{M}{|S|} \sum_{t \in [T]} \gamma_t(S) - R,$$

where

$$\alpha := 1 - \frac{1}{|S|}.$$

Proof. Consider some trial $t \in [T]$. By Lemma 5.4 and the definition of Ψ_t we have that Ψ_t is concave over Δ_K . Hence, by Lemma 5.8, we have

$$\begin{aligned}\mathbb{E}[(\mathbf{p}^S - \mathbf{p}_t) \cdot \mathbf{g}_t \mid \mathbf{p}_t] &= (\mathbf{p}^S - \mathbf{p}_t) \cdot \mathbb{E}[\mathbf{g}_t \mid \mathbf{p}_t] \\ &= (\mathbf{p}^S - \mathbf{p}_t) \cdot \nabla \Psi_t(\mathbf{p}_t) \\ &\geq \Psi_t(\mathbf{p}^S) - \Psi_t(\mathbf{p}_t).\end{aligned}\tag{C.8}$$

Lemma 5.7 gives us:

$$\begin{aligned}\Psi_t(\mathbf{p}^S) &= \Phi^{r_t}(\mathbf{p}^S) - M \sum_{i \in [K]} p^S(i) c_{t,i} \\ &\geq r(\emptyset) + \left(1 - \left(\frac{|\mathcal{S}| - 1}{|\mathcal{S}|}\right)^M\right) (r(\mathcal{S}) - r(\emptyset)) - \frac{M}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} c_{t,i}\end{aligned}\tag{C.9}$$

and Lemma 5.2 gives us:

$$\begin{aligned}\Psi_t(\mathbf{p}_t) &= \Phi^{r_t}(\mathbf{p}_t) - M \sum_{i \in [K]} p_{t,i} c_{t,i} \\ &= \mathbb{E}[r_t(\mathcal{B}(\mathbf{p}_t))] - M \sum_{i \in [K]} p_{t,i} c_{t,i} \\ &= \mathbb{E}[r_t(\mathcal{A}_t) \mid \mathbf{p}_t] - M \sum_{i \in [K]} p_{t,i} c_{t,i} \\ &= \mathbb{E}[r_t(\mathcal{A}_t) \mid \mathbf{p}_t] - \sum_{j \in [M]} \sum_{i \in [K]} \mathbb{P}[a_{t,j} = i \mid \mathbf{p}_t] c_{t,i} \\ &= \mathbb{E}[r_t(\mathcal{A}_t) \mid \mathbf{p}_t] - \sum_{j \in [M]} \mathbb{E}[c_{t,a_{t,j}} \mid \mathbf{p}_t] \\ &= \mathbb{E}[r_t(\mathcal{A}_t) \mid \mathbf{p}_t] - \mathbb{E}\left[\sum_{j \in [M]} c_{t,a_{t,j}} \mid \mathbf{p}_t\right] \\ &\leq \mathbb{E}[r_t(\mathcal{A}_t) \mid \mathbf{p}_t] - \mathbb{E}\left[\sum_{i \in \mathcal{A}_t} c_{t,a_{t,j}} \mid \mathbf{p}_t\right] \\ &= \mathbb{E}[\psi_t \mid \mathbf{p}_t] + r_t(\emptyset).\end{aligned}\tag{C.10}$$

Substituting equations (C.9) and (C.10) into Equation (C.8) gives us:

$$\mathbb{E}[(\mathbf{p}^S - \mathbf{p}_t) \cdot \mathbf{g}_t \mid \mathbf{p}_t] \geq -\mathbb{E}[\psi_t \mid \mathbf{p}_t] + \left(1 - \left(\frac{|\mathcal{S}| - 1}{|\mathcal{S}|}\right)^M\right) \hat{r}_t(\mathcal{S}) - \frac{M}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} c_{t,i}$$

and hence:

$$\mathbb{E}[(\mathbf{p}^S - \mathbf{p}_t) \cdot \mathbf{g}_t] \geq -\mathbb{E}[\psi_t] + \left(1 - \left(\frac{|\mathcal{S}| - 1}{|\mathcal{S}|}\right)^M\right) \hat{r}_t(\mathcal{S}) - \frac{M}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} c_{t,i}.\tag{C.11}$$

Lemma 5.9 gives us:

$$\mathbb{E} \left[\sum_{i \in [K]} p_{t,i} g_t(i)^2 \right] \leq \frac{R^2}{T}, \quad (\text{C.12})$$

Lemma 5.5 gives us:

$$\sum_{t \in [T]} \mathbb{E}[(\mathbf{p}^S - \mathbf{p}_t) \cdot \mathbf{g}_t] \leq \frac{\ln(K)}{\eta} + \eta \sum_{t \in [T]} \mathbb{E} \left[\sum_{i \in [K]} p_{t,i} g_t(i)^2 \right]. \quad (\text{C.13})$$

Substituting equations (C.11) and (C.12) into Equation (C.13) gives us:

$$- \sum_{t \in [T]} \mathbb{E}[\psi_t] + \left(1 - \left(\frac{|\mathcal{S}| - 1}{|\mathcal{S}|} \right)^M \right) \sum_{t \in [T]} \hat{r}_t(\mathcal{S}) - \frac{M}{|\mathcal{S}|} \sum_{t \in [T]} \sum_{i \in \mathcal{S}} c_{t,i} \leq \frac{\ln(K)}{\eta} + \eta R^2.$$

Since $\eta = \ln(K)/R$ this implies the result. $\square$

C.2 Proof Of Theorem 5.3

Theorem 5.3. Suppose that there exists some $d \in \mathbb{N}$, $s \in (0, 1)$, $\gamma > 1$, and a randomized polynomial time algorithm for the learner such that for all $K, M \in \mathbb{N}$ and for any instance of the multichannel advertising problem, it holds that $|\mathcal{A}_t| \leq M$ for all $t = 1, \dots, T$ and, for any subset $\mathcal{S} \subseteq [K]$,

$$\mathbb{E} \left[\sum_{t \in [T]} r_t(\mathcal{A}_t) \right] \geq (1 - \alpha^{\gamma M}) \sum_{t \in [T]} r_t(\mathcal{S}) - R',$$

where $R' \in \mathcal{O}(K^d T^s)$ and $\alpha := 1 - \frac{1}{|\mathcal{S}|}$. Then, for all $\varepsilon \in (0, 1 - 1/\gamma)$ and $B > 4^{1/((1-\varepsilon)\gamma-1)}$, there exists a randomized polynomial-time algorithm for the set cover problem on $[B]$ that, with probability at least $\frac{1}{2}$, achieves approximation ratio at least $(1 - \varepsilon) \ln(B)$.

Proof. Suppose we have such an algorithm. Let $c > 0$ and $\gamma > 1$ be such that

$$\mathbb{E} \left[\sum_{t \in [T]} r_t(\mathcal{A}_t) \right] \geq \left(1 - \left(\frac{|\mathcal{S}| - 1}{|\mathcal{S}|} \right)^{\gamma M} \right) \sum_{t \in [T]} r_t(\mathcal{S}) - cK^d T^s. \quad (\text{C.14})$$

Choose any $\rho \in (1/\gamma, 1)$ and then consider any $B \in \mathbb{N}$ such that $B > 4^{1/(\rho\gamma-1)}$. Consider also any collection of sets $\{\mathcal{C}_k \mid k \in [K]\} \subseteq 2^{[B]}$ such that

$$\bigcup_{k \in [K]} \mathcal{C}_k = [B].$$

Let $\mathcal{S}$ be a subset of $[K]$ of minimum cardinality such that

$$\bigcup_{k \in \mathcal{S}} \mathcal{C}_k = [B].$$

Now choose

$$T := \left\lceil (4cK^d B)^{1/(1-s)} \right\rceil.$$

and choose any $M \in \mathbb{N}$ such that $M \geq \rho \ln(B)|\mathcal{S}|$. For all $t \in [T]$ draw $\mathcal{D}_t$ randomly as follows. First draw β_t uniformly at random from $[B]$ and then define

$$\mathcal{D}_t := \{k \in [K] \mid \beta_t \in \mathcal{C}_k\}.$$

It is a classic result that

$$\left(\frac{|\mathcal{S}| - 1}{|\mathcal{S}|}\right)^{|\mathcal{S}|} \leq e^{-1}.$$

so by the conditions on B and M we have

$$\left(\frac{|\mathcal{S}| - 1}{|\mathcal{S}|}\right)^{\gamma M} \leq \exp(-\gamma M/|\mathcal{S}|) = B^{-\rho\gamma} = \frac{B^{1-\rho\gamma}}{B} < \frac{1}{4B}. \quad (\text{C.15})$$

By definition of $\mathcal{S}$ we have, for all $t \in [T]$, that there exists some $k \in \mathcal{S}$ such that $\beta_t \in \mathcal{C}_k$ so that $\mathcal{D}_t \cap \mathcal{S} \neq \emptyset$. This implies

$$\sum_{t \in [T]} r_t(\mathcal{S}) = T,$$

and hence, by (C.14) and (C.15), we have

$$\mathbb{E} \left[\sum_{t \in [T]} (1 - r_t(\mathcal{A}_t)) \right] \leq T - T + \frac{T}{4B} + cK^d T^s \leq \frac{T}{4B} + \frac{TcK^d}{T^{1-s}} \leq \frac{T}{2B}. \quad (\text{C.16})$$

Fix t and a realization of $\mathcal{A}_t$. If we have

$$\bigcup_{k \in \mathcal{A}_t} \mathcal{C}_k \neq [B],$$

then we must also have that

$$\begin{aligned} \mathbb{E}[1 - r_t(\mathcal{A}_t) \mid \mathcal{A}_t] &= \mathbb{P}[\mathcal{A}_t \cap \mathcal{D}_t = \emptyset \mid \mathcal{A}_t] \\ &= \mathbb{P}[\forall k \in \mathcal{A}_t, \beta_t \notin \mathcal{C}_k \mid \mathcal{A}_t] \\ &= \mathbb{P} \left[\beta_t \notin \bigcup_{k \in \mathcal{A}_t} \mathcal{C}_k \mid \mathcal{A}_t \right] \geq \frac{1}{B}. \end{aligned}$$

Hence, by taking the randomness of $\mathcal{A}_1, \dots, \mathcal{A}_T$ into account,

$$\begin{aligned} &\mathbb{P} \left[\sum_{t \in [T]} \mathbb{I} \left\{ \bigcup_{k \in \mathcal{A}_t} \mathcal{C}_k \neq [B] \right\} = T \right] \\ &\leq \mathbb{P} \left[\mathbb{E} \left[\sum_{t \in [T]} (1 - r_t(\mathcal{A}_t)) \mid \mathcal{A}_1, \dots, \mathcal{A}_T \right] \geq \frac{T}{n} \right] \leq \frac{1}{2} \end{aligned}$$

by (C.16). Since T is polynomial in KB and $|\mathcal{A}_t| \leq M$, we have a randomized polynomial-time

algorithm that, with probability at least $\frac{1}{2}$, solves the set cover problem on $[B]$ with approximation ratio $(1 - \varepsilon) \ln(B)$ for $\varepsilon = 1 - \rho \in (0, 1 - 1/\gamma)$. $\square$

C.3 Submodular Monotone Non-Pseudoconcave Functions

We provide a function counterexample to show that there are submodular monotone functions which are not pseudoconcave.

Let $K = 8$, $\mathcal{P} = 2^{[K]}$, $\mathcal{S} = \{K\}$, and $\alpha > 0$. We define $U^{r,\mathcal{S}}$ as follows:

$$U^{r,\mathcal{S}} := \begin{pmatrix} 1 & 2 & 2 & 2 & 1+\alpha & 1+\alpha & 1+\alpha & 1 \\ 2 & 1 & 2 & 2 & 1+\alpha & 1+\alpha & 1+\alpha & 1 \\ 2 & 2 & 1 & 2 & 1+\alpha & 1+\alpha & 1+\alpha & 1 \\ 2 & 2 & 2 & 1 & 1+\alpha & 1+\alpha & 1+\alpha & 1 \\ 1+\alpha & 1+\alpha & 1+\alpha & 1+\alpha & 1 & 2 & 2 & 1 \\ 1+\alpha & 1+\alpha & 1+\alpha & 1+\alpha & 2 & 1 & 2 & 1 \\ 1+\alpha & 1+\alpha & 1+\alpha & 1+\alpha & 2 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

Now, let $\mathbf{x} = (1, 1, 1, 1, -1, -1, -1, -1)^\top$. Note that we have $\langle \mathbf{x}, \mathbf{1} \rangle = 0$ as required by the pseudoconcavity definition, and $\mathbf{x}^\top U^{r,\mathcal{S}} \mathbf{x} = 17 - 24\alpha$, which is positive for $\alpha \in (0, \frac{17}{24})$, implying therefore the non-pseudoconcavity of r for such values of α .

We now show how to define r starting from $U^{r,\mathcal{S}}$ in such a way that it is both monotone and submodular while being therefore also non-pseudoconcave.

We have $|\mathcal{P}| = 2^K = 256$ possible subsets as the arguments of r , 29 of which are already defined by the above matrix $U^{r,\mathcal{S}}$:

- 1 subset ($\{K\}$) with cardinality 1,
- 7 subsets ($\{j, K\}_{j \in [K-1]}$) with cardinality 2,
- 21 subsets ($\{i, j, K\}_{1 \leq j < i \leq K-1}$) with cardinality 3.

For any $i \in [K]$, let δ_i and Δ_i be equal respectively to the minimum and the maximum difference (gain) over all values of r for subsets with cardinality i and all the ones for subsets with cardinality $i - 1$. As previously anticipated, we construct function r starting from the above matrix $U^{r,\mathcal{S}}$ in such a way that for all $i \in [K - 1]$, we have

$$\delta_i \geq \Delta_{i+1},$$

which is a sufficient condition for submodularity because, for all $i \in [K]$, each subset $S_i \in \mathcal{P}$ with cardinality i can be generated by adding one of its element *only* from a subset $S_{i-1} \subset S_i$ with cardinality $i - 1$.

We set $\alpha = \frac{2}{3} < \frac{17}{24}$, which guarantees the non-pseudoconcavity of r . To ensure monotonicity and submodularity, we define

- $r(S_1) := 0$ for all subsets $S_1 \in \mathcal{P}$ with $|S_1| = 1$ (consistently with $U_{K,K}^{r,\mathcal{S}}$);

- $r(S_2) := 1$ for all subsets $S_2 \in \mathcal{P}$ with $|S_2| = 2$ (consistently with $U_{K,j}^{r,\mathcal{S}}, U_{j,j}^{r,\mathcal{S}}, U_{j,1}^{r,\mathcal{S}}$ for all $j \in [K-1]$);
- $r(S_3) := 1 + \frac{2}{3} = \frac{5}{3}$ for all subsets $S_3 \in \mathcal{P}$ with $|S_3| = 3$ that are not already defined by $U^{r,\mathcal{S}}$;
- $r(S_4) := r(S_3) + \frac{1}{2} = \frac{5}{3} + \frac{1}{2} = 2 + \frac{1}{6} > \max_{i,j} U_{i,j}^{r,\mathcal{S}} = 2$ for all subsets $S_4 \in \mathcal{P}$ with $|S_4| = 4$;
- $r(S_5) := r(S_4) + \frac{1}{6} = 2 + \frac{2}{6}$,
 $r(S_6) := r(S_5) + \frac{1}{6} = 2 + \frac{3}{6}$,
 $r(S_7) := r(S_6) + \frac{1}{6} = 2 + \frac{4}{6}$,
 $r(S_8) := r(S_7) + \frac{1}{6} = 2 + \frac{5}{6}$
 for all subsets $S_5, S_6, S_7, S_8 \in \mathcal{P}$ such that $|S_5| = 5, |S_6| = 6, |S_7| = 7, |S_8| = 8$.

Finally, we also set $r(\emptyset) = -1$. Note that, to ensure that submodularity is not violated, for each subset S_3^U with $|S_3^U| = 3$ defined by $U^{r,\mathcal{S}}$, we have that the difference $r(S_3^U) - r(S_2)$ for any subset $S_2 \subset S_3^U$ with $|S_2| = 2$ is either equal to $\alpha = \frac{2}{3}$ or 1, that is not smaller than the maximum difference $r(S_4) - r(S_3)$ over all $S_3, S_4 \in \mathcal{P}$ with $|S_3| = 3$ and $|S_4| = 4$, which in turn is equal to $\frac{1}{2} < \frac{2}{3}$. Furthermore, $r(S_4) = 2 + \frac{1}{6}$ is never smaller than any values of $r(S_3^U)$ for all subsets $S_3^U \in \mathcal{P}$ with $|S_3| = 3$ that are already defined by $U^{r,\mathcal{S}}$, because we have $r(S_3^U) \leq 2$, thereby preserving monotonicity for all subsets in $\mathcal{P}$ with cardinality smaller or equal to 4.

Now, we recall that for any $i \in [K]$, δ_i and Δ_i are defined to be respectively equal to the minimum and the maximum difference (gain) over all values of r for subsets with cardinality i and all the ones for subsets with cardinality $i-1$. Since we have

- $\delta_1, \Delta_1, \delta_2, \Delta_2 = 1$ (which immediately implies $\Delta_2 \leq \delta_1$),
- $\delta_3 = \frac{2}{3}; \quad \Delta_3 = 1 \leq \delta_2$,
- $\delta_4 = \frac{1}{6}; \quad \Delta_4 = \frac{1}{2} \leq \delta_3$,
- $\delta_5, \Delta_5, \delta_6, \Delta_6, \delta_7, \Delta_7, \delta_8, \Delta_8 = \frac{1}{6} \leq \delta_4$,

then $\delta_i \geq \Delta_{i+1}$ for all $i \in [K-1]$ which guarantees the submodularity of r . Finally, it is immediate to verify that r is monotone also for all subsets in $\mathcal{P}$ with cardinality larger than 4. Hence, we conclude that r is monotone submodular and non-pseudoconcave. $\square$

C.4 Efficient Implementation of COMBAND

To implement the algorithm the COMBAND presented in *Cesa-Bianchi and Lugosi (2012)*, it is necessary to devise an efficient method for sampling from a set whose size can be exponential in K . In fact, at each trial, given a set $\mathcal{S}$ of positive real numbers, we need to select any of the subsets S with a given size m from $\mathcal{S}$ with a probability proportional to the product of the elements contained in S itself.

To be consistent with the notation used in *Cesa-Bianchi and Lugosi (2012)*, henceforth we use the symbol d in place of K .

Given a set $S = \{q_1, q_2, \dots, q_d\}$ of real positive numbers, we now show how to select a m -sized subset of indices $\{u_1, \dots, u_m\}$ from $[d]$ with a probability proportional to $\prod_{i=1}^m q_{u_i}$ by using dynamic programming. The running time of this sampling method is always linear* in $m \cdot d$.

For each sampling operation, consider the sequence of element indices $u_1, u_2, \dots, u_m$ ordered according to the elements in $[d]$, i.e., $u_i < u_{i+1}$ for all $i \in [m-1]$.

The main idea of this method is to sample first u_m , and then $u_{m-1}, \dots, u_1$ (i.e., in reverse order) having derived in a preliminary phase via dynamic programming all the probabilities that $u_m = j$ for all $m \leq j \leq d$, and the conditional probabilities that $u_{m'} = j$ given that $u_{m'+1} = j'$, for all $m' \in [m-1]$ and $m' \leq j < j' \leq d - m + m'$.

We denote the conditional probability that $u_{m'} = j$ given that $u_{m'+1} = j'$, where $m' \in [m-1]$ and $m' \leq j < j' \leq d - m + m'$ by

$$P_{m',j|j'} := \mathbb{P}(u_{m'} = j | u_{m'+1} = j'),$$

and, for the selection of u_m , we define for all $j \in [d]$

$$P_{m,j} := \mathbb{P}(u_m = j),$$

because there is no element $u_{j'} > u_m$ (with $j' > m$) in the sequence of selected indices from $[d]$. We clearly have $\sum_{j=m'}^{j'-1} P_{m',j|j'} = 1$ and $\sum_{j=m}^d P_{m,j} = 1$.

For each $m' \in [m]$ and $m' \leq j \leq d - m + m'$ let $z_{m',j}$, be the the sum of the products of numbers of S with indices $u_1, u_2, \dots, u_{m'}$ contained in each m' -sized subset of $[j]$ such that $u_{m'} = j$. We define $Z_{m',k} := \sum_{i=m'}^k z_{m',i}$ for any integer k such that $m' \leq k \leq d - m + m'$. Thus, for all $m' \in [m-1]$ and $m' \leq j < j' \leq d - m + m'$ we have

$$P_{m',j|j'} = \frac{z_{m',j}}{Z_{m',j'-1}}.$$

Analogously, for the selection of u_m , for all $m \leq j \leq d$ we can write

$$P_{m,j} = \frac{z_{m,j}}{Z_{m,d}}.$$

Hence, once we obtain $z_{m',j}$ and $Z_{m',j'-1}$ for all $m' \in [m-1]$ and $m' \leq j < j' \leq d - m + m'$, $z_{m,j}$ for all $m \leq j \leq d$, and $Z_{m,d}$, we can immediately compute the desired probabilities to sample $u_m, u_{m-1}, \dots, u_1$ in this (reverse) order.

We now show how to calculate these values. To this goal, since $Z_{m',k} := \sum_{i=m'}^k z_{m',i}$, we only need to show how to compute the values appearing at the numerator in the above probability formulas.

The possibility to efficiently the above probabilities is given by the following observation:

$$z_{m',j} = Z_{m'-1,j-1} \cdot q_j.$$

Note that $Z_{m'-1,j-1}$ can be in turn defined in terms of $z_{m'-1,m'-1}, z_{m'-1,m'}, z_{m'-1,m'+1}, \dots, z_{m'-1,j-2}, z_{m'-1,j-1}$. This recurrence relation allows us to compute all these values once we know $z_{1,1}, z_{1,2}, \dots, z_{1,d}$. Since we clearly have $z_{1,j} = q_j$ for all $j \in [d]$, we can therefore compute all these values and the above

*We assume that multiplying two numbers requires a constant time. Removing this assumption, since it is known that it is possible to multiply two numbers represented by at most m bits in time equal to $\tilde{O}(m)$ when $m \gg 1$ Harvey and Van Der Hoeven (2021), the total sampling time would be $\tilde{O}(m^2 d)$ instead of $\mathcal{O}(md)$.

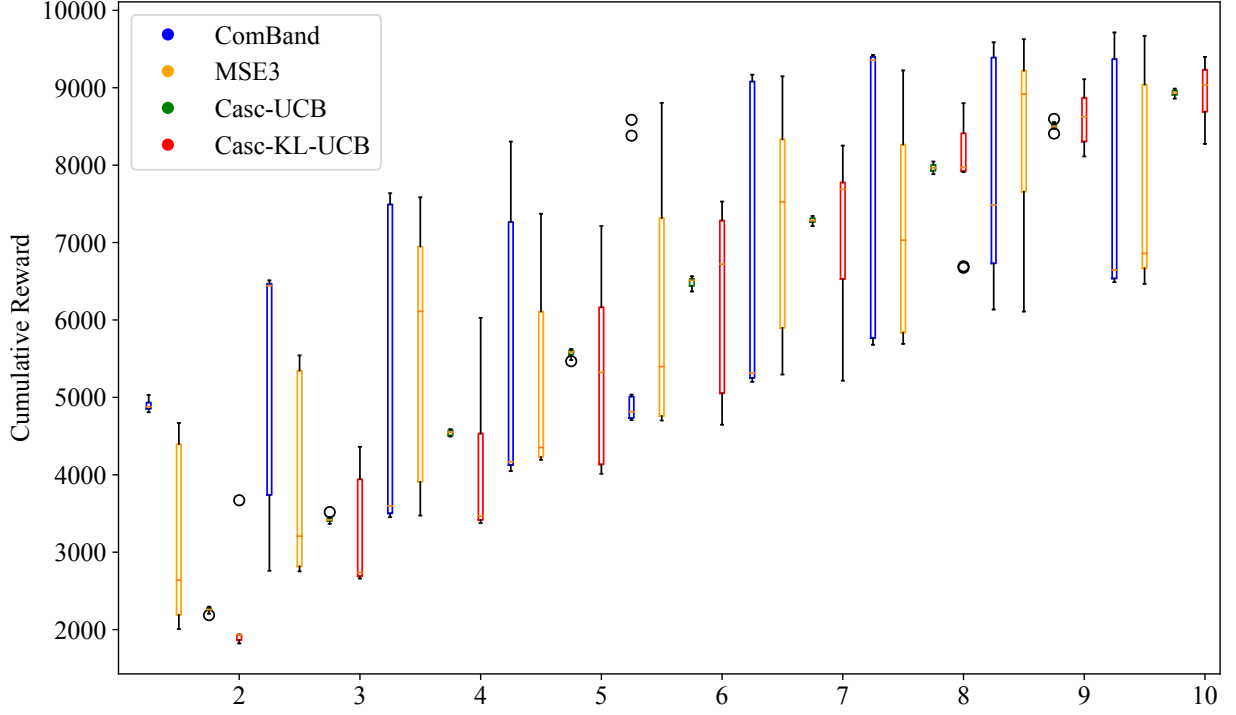


Figure C.1: Stochastic environment, cumulative reward with respect to the M parameter

probabilities to efficiently accomplish this sampling operation by finding the indices $u_m, u_{m-1}, \dots, u_1$ in this order. It is immediate to verify that both the number of sum and multiplication operations are equal to $\Theta(md)$.

C.5 Additional Experimental Results

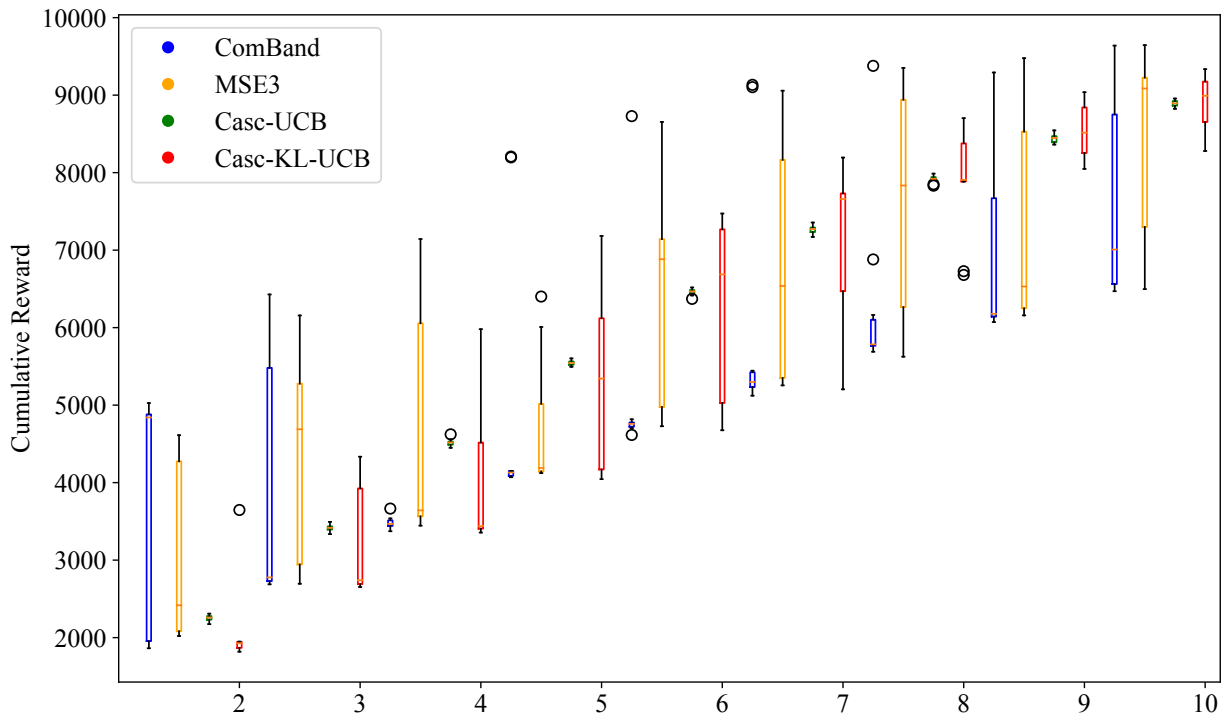


Figure C.2: Stochastic with adversarial corruptions environment, cumulative reward with respect to the M parameter

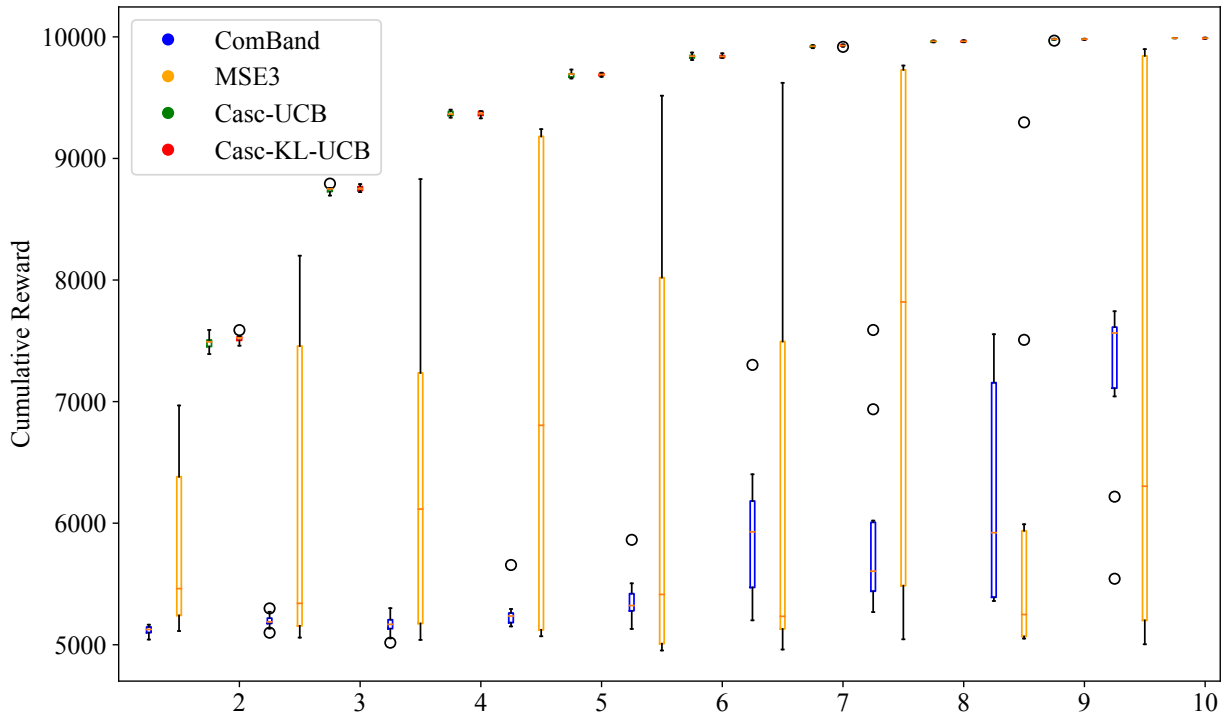


Figure C.3: Worst case stochastic environment, cumulative reward with respect to the M parameter

Appendix D

Appendix - Stochastic Shortest Path

D.1 The benefits of sparsity - upper bounds

We restate the main theorem proved in Section 6.4.

Theorem 6.2. *Consider OMD with ψ_p as regularizer. If T is such that $q_{\pi^*} \in \Delta(T)$, $\eta = \sqrt{\frac{2pT^{1+1/p}}{KDM^{1/p}}}$, $p = 1 + \log(TM)$, then $\mathbb{E}[R_K] \leq \mathcal{O}\left(\sqrt{DKT \log(MT)}\right)$.*

We include the missing details from the proof given in Section 6.4. First, recall from (6.4) that

$$\begin{aligned} \psi_p(q) &= p \cdot \left(-1 + \|q\|_{1+1/p}^{1+1/p}\right) = p \cdot \left(-1 + \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} |q(s, a)|^{1+1/p}\right) \\ \implies \frac{\partial \psi_p(q)}{\partial q(s, a)} &= (p+1)q(s, a)^{1/p} \\ \implies \frac{\partial^2 \psi_p(q)}{\partial q(s, a)^2} &= \left(1 + \frac{1}{p}\right)q(s, a)^{1/p-1} \\ \implies \nabla \psi_p(q) &= (p+1)q^{1/p}, \quad \nabla^2 \psi_p(q) = \text{diag}\left(\frac{p+1}{p}q^{1/p-1}\right) \end{aligned}$$

We implicitly assumed here that ψ_p is defined on $\mathcal{R}_{\geq 0}^\Gamma$. The missing details are:

- ψ_p satisfies the condition (6.2) with $\alpha = 1$:

$$\begin{aligned} \nabla \psi_p(q) \in \left[\nabla \psi_p(q_k), \nabla \psi_p(q_k) - \eta c_k\right] &\implies q^{1/p}(s, a) \leq q_k^{1/p}(s, a) \\ &\implies \frac{1}{q(s, a)} \geq \frac{1}{q_k(s, a)} \\ &\implies \frac{1}{q^{1-1/p}(s, a)} \geq \frac{1}{q_k^{1-1/p}(s, a)} \\ &\implies \nabla^2 \psi_p(q) \succeq \nabla^2 \psi_p(q_k). \end{aligned}$$

- $\nabla^2 \psi_p(q)^{-1} = \text{diag}\left(\frac{p}{p+1}q^{1-1/p}\right)$: follows directly from the expression for $\nabla^2 \psi_p(q)$ above.

We now turn to the description of the parameter-free algorithm and the proof of its corresponding regret bound (Theorem 6.3).

For the unknown sparsity level, we use the same approach as in Kwon and Perchet (2016), which divides the episode horizon into segments, where each segment will run OMD from scratch with an

Algorithm D.1: Sparse-Agnostic Mirror Descent

Input: T, K, D
Initialize: $p \leftarrow \log 2^2 T$, $\eta \leftarrow \sqrt{pT^{1+1/p}/(pDK2^{2/p})}$, $m \leftarrow 1$, $q_1 \leftarrow \arg \min_{q \in \Delta(T)} \psi_p(q)$
for $k = 1, \dots, K$ **do**
 Play q_k and Observe c_k
 if $\|c_k\|_0 \leq m$ **then**
 $q_{k+1} = \arg \min_{q \in \Delta_T} \langle q, c_k \rangle + D_{\psi_p}(q, c_k)$
 else
 $m \leftarrow \lceil \log_2 \log_2 \|c_k\|_0 \rceil$
 $p \leftarrow \log 2^{2^m} T$
 $\eta \leftarrow \sqrt{pT^{1+1/p}/(pDK2^{2/p})}$
 $q_{k+1} \leftarrow \arg \min_{q \in \Delta(T)} \psi_p(q)$

Algorithm D.2: Fully Parameter-Free Online Mirror Descent for Sparse SSPs

Define $j_0 = \lceil \log_2 T^{\pi_f} \rceil (s_0) - 1$, $b(j) = 2^{j_0+j}$, $N = \lceil \log_2 K \rceil - j_0$, $\eta_j = \sqrt{\frac{\log N \sqrt{b(j)}}{b(j)DK}}$
Define $\psi_p(p) = \sum_{j=1}^N \frac{1}{\eta_j} p(j) \log p(j)$
Initialize: $p_1 \in \Delta_N$, such that $p_1(j) = \frac{\eta_j}{\eta_1 N}$, $\forall j \neq 1$
Initialize: N instances of Algorithm D.1, with j -th instance $T = b(j)$
for $k = 1, \dots, K$ **do**
 Obtain occupancy measures q_k^j for $j \in [N]$
 Sample $j_k \sim p_k$, execute policy induced by $q_k^{j_k}$
 Receive c_k and send it to all instances.
 Compute $\ell_k(j) = \langle q_k^j, c_k \rangle$, $a_k(j) = 4\eta_j \ell_k^2(j)$
 Update $p_{k+1} = \arg \min_{p \in \Delta_N} \langle p, \ell_k + a_k \rangle + D_{\psi_p}(p, p_k)$

increasing sparsity level guess. By definition $M = \max_{k \in [K]} \|c_k\|_0$ is the higher sparsity level of the cost vectors. Now let $B = \lceil \log_2 \log_2 M \rceil$, $\tau(0) = 0$ then for $1 \leq b \leq B$:

$$\tau(b) = \min \left\{ 1 \leq k \leq K \mid \|c_k\|_0 > 2^{2^b} \right\} \quad \text{and} \quad \tau(B) = K.$$

This essentially says that $\tau(b)$ is the first episode in which the sparsity level of the loss vector exceeds 2^b . Then we denote $b_k = \min \{b \leq 1 \mid \tau(b) \leq k\}$, which is the index of the only episode k belongs to. Therefore we partition the horizon $[K]$ as intervals $(I(b))_{b \in [B]}$:

$$I(b) = \begin{cases} [\tau(b) + 1, \tau(b)] & \text{if } \tau(b-1) < \tau(b) \\ \emptyset & \text{if } \tau(b-1) = \tau(b) \end{cases}$$

Now we define:

$$p(b) = \log 2^{2^b} T$$

$$\eta(b) = \sqrt{\frac{p(b)T^{1+1/p(b)}}{DK2^{2^b/p(b)}}}.$$

We consider mirror descent updates with our regularizer:

$$\psi_{p(b)}(q) = p(b) \left(-1 + \|q\|_{1+1/p(b)}^{1+1/p(b)} \right),$$

then each mirror descent instance is defined by:

$$q_t = \nabla \psi_{p(b_t)}^* \left(\eta(b_t) \sum_{k' < k, k' \in I(b_t)} c_k \right), \quad k = 1, \dots, K.$$

Lemma D.1. *Running Algorithm Algorithm D.1 guarantees:*

$$R_K \leq \mathcal{O} \left(B \sqrt{DTK \log(TM)} + BM \sqrt{\frac{DT \log(TM)}{K}} \right)$$

Proof. On the time interval $I(b)$, we run OMD with regularizer $\psi_{p(b)}$, learning rate $\eta(b)$ and we consider the interval regret $R(b)$. Therefore, by the previous result, we see that the Penalty term is:

$$\psi_{p(b)}(q_\star) - \psi_{p(b)}(q_1) \leq p(b) \|q_\star\|^{1+1/p(b)} \leq p(b) T^{1+1/p(b)}$$

While for the Stability term:

$$\begin{aligned} \|c_k\|_{\nabla^2 \psi_p(q_k)}^2 &= \eta(b) \sum_{k \in I(b)} \sum_{s,a} c_k(s,a)^{1-1/p(b)} q_k(s,a)^{1-1/p(b)} \\ &= \eta(b) \sum_{\substack{k \in I(b) \\ k < \tau(b)}} \sum_{s,a} c_k(s,a)^{1-1/p(b)} q_k(s,a)^{1-1/p(b)} + \eta(b) \sum_{s,a} c_{\tau(b)}(s,a)^{1-1/p(b)} q_{\tau(b)}(s,a)^{1-1/p(b)} \\ &\leq \eta(b) \sum_{\substack{k \in I(b) \\ k < \tau(b)}} 2^{2^b} \left(\frac{1}{2^{2^b}} \sum_{s,a} c_k(s,a) q_k(s,a) \right)^{1-1/p(b)} \\ &\quad + \eta(b) \tau(b) \left(\frac{1}{\tau(b)} \sum_{s,a} c_{\tau(b)}(s,a) q_{\tau(b)}(s,a) \right)^{1-1/p(b)} \\ &\leq \eta(b) \sum_{\substack{k \in I(b) \\ k < \tau(b)}} 2^{2^b/p(b)} \langle c_k, q_k \rangle^{1-1/p(b)} + \eta(b) M^{1/p(b)} \langle c_{\tau(b)}, q_{\tau(b)} \rangle^{1-1/p(b)} \\ &\leq \eta(b) \sum_{\substack{k \in I(b) \\ k < \tau(b)}} 2^{2^b/p(b)} \max \{ \langle c_k, q_k \rangle, 1 \} + \eta(b) M^{1/p(b)} \max \{ \langle c_{\tau(b)}, q_{\tau(b)} \rangle, 1 \} \\ &\leq \eta(b) \sum_{\substack{k \in I(b) \\ k < \tau(b)}} 2^{2^b/p(b)} (\langle c_k, q_k \rangle + 1) + \eta(b) M^{1/p(b)} (\langle c_{\tau(b)}, q_{\tau(b)} \rangle + 1) \\ &= \eta(b) \sum_{\substack{k \in I(b) \\ k < \tau(b)}} 2^{2^b/p(b)} \langle c_k, q_k \rangle + \eta(b) (I_{\tau(b)} - 1) 2^{2^b/p(b)} + \eta(b) M^{1/p(b)} + \eta(b) M^{1/p(b)} \langle c_{\tau(b)}, q_{\tau(b)} \rangle \end{aligned}$$

Therefore

$$\begin{aligned}
& \sum_{k \in I(b)} \langle q_k - q_{\pi^*}, c_k \rangle \leq p(b) \frac{T^{1+1/p(b)}}{\eta(b)} + \eta(b) \sum_{\substack{k \in I(b) \\ k < \tau(b)}} 2^{2^b/p(b)} \langle c_k, q_k \rangle + \eta(b)(I_{\tau(b)} - 1)2^{2^b/p(b)} \\
& \quad + \eta(b)M^{1/p(b)} + \eta(b)M^{1/p(b)} \langle c_{\tau(b)}, q_{\tau(b)} \rangle \\
\Rightarrow & \sum_{\substack{k \in I(b) \\ k < \tau(b)}} \langle q_k - q_{\pi^*}, c_k \rangle + \langle c_{\tau(b)}, q_{\tau(b)} \rangle \leq p(b) \frac{T^{1+1/p(b)}}{\eta(b)} + \eta(b)(I_{\tau(b)} - 1)2^{2^b/p(b)} + \eta(b)M^{1/p(b)} \\
& \quad + \eta(b)2^{2^b/p(b)} \sum_{\substack{k \in I(b) \\ k < \tau(b)}} \langle c_k, q_k \rangle + \eta(b)M^{1/p(b)} \langle c_{\tau(b)}, q_{\tau(b)} \rangle \\
\Rightarrow & R_T(b) \leq 4p(b) \frac{T^{1+1/p(b)}}{\eta(b)} + 4\eta(b)(I_{\tau(b)} - 1)2^{2^b/p(b)} + 4\eta(b)M^{1/p(b)} \\
& \quad + 4\eta(b)2^{2^b/p(b)} \sum_{\substack{k \in I(b) \\ k < \tau(b)}} \langle c_k, q_{\pi^*} \rangle + 4\eta(b)M^{1/p(b)} \langle c_{\tau(b)}, q_{\pi^*} \rangle \\
\Rightarrow & R_T(b) \leq 4p(b) \frac{T^{1+1/p(b)}}{\eta(b)} + 4\eta(b)I_{\tau(b)}2^{2^b/p(b)} + 4\eta(b)M^{1/p(b)} \\
& \quad + 4\eta(b)2^{2^b/p(b)}I_{\tau(b)}D + 4\eta(b)M^{1/p(b)}D
\end{aligned}$$

Tuning $\eta(b) = \sqrt{\frac{p(b)T^{1+1/p(b)}}{DK2^{2^b/p(b)}}}$:

$$\begin{aligned}
R_T(b) & \leq 4\sqrt{KDT^{1+1/p(b)}p(b)2^{2^b/p(b)}} + 4I_{\tau(b)}\sqrt{\frac{DT^{1+1/p(b)}2^{2^b/p(b)}p(b)}{K}} + 4M^{1/p(b)}\sqrt{\frac{Dp(b)T^{1+1/p(b)}}{K2^{2^b/p(b)}}} \\
& \leq 4\sqrt{KDT \log(T2^{2^b})} + 4I_{\tau(b)}\sqrt{\frac{DT \log(T2^{2^b})}{K}} + 4M\sqrt{\frac{DT \log(T2^{2^b})}{K}}
\end{aligned}$$

Note that:

$$\begin{aligned}
\log(2^{2^b}) & \leq 2^{\log_2 \log_2 M + 1} \log 2 \\
& = \log 2 \exp(\log(2^{\log_2 \log_2 M + 1})) \\
& = \log 2 \exp((\log_2 \log_2 M + 1) \log 2) \\
& = 2 \log 2 \exp((\log_2 \log_2 M)) \\
& = 2 \log 2 \log M \leq 2 \log M
\end{aligned}$$

And therefore $\log(T2^{2^b}) \leq 2 \log(TM)$.

$$R_K = \sum_{b=1}^B R(b) \leq \sum_{b=1}^B 4\sqrt{KDT \log(T2^{2^b})} + \sum_{b=1}^B 4I_{\tau(b)}\sqrt{\frac{DT \log(T2^{2^b})}{K}} + \sum_{b=1}^B 4M\sqrt{\frac{DT \log(T2^{2^b})}{K}}$$

$$\leq 4\sqrt{KDT} \sum_{b=1}^B \sqrt{\log(T2^{2^b})} + 4\sqrt{\frac{DT}{K}} \sum_{b=1}^B I_{\tau(b)} \sqrt{\log(T2^{2^b})} + 4M\sqrt{\frac{DT}{K}} \sum_{b=1}^B \sqrt{\log(T2^{2^b})}$$

Now ignoring knowing that $\sum_{b=1}^B I_{\tau(b)} = K$:

$$\begin{aligned} R_K &\leq 4B\sqrt{KDT \log(TM)} + 4\sqrt{DTK \log(T2^{2^b})} + 4M\sqrt{\frac{DT}{K}} B\sqrt{\log(TM)} \\ &\leq \mathcal{O} \left(B\sqrt{DKT \log(TM)} + BM\sqrt{\frac{DT \log(TM)}{K}} \right) \end{aligned}$$

□

We now turn our attention to the unknown hitting time of the optimal policy T_* , where we can exploit the same technique presented in Chen et al. (2021b).

We will therefore run $N \approx \log K$ instances of Algorithm D.1 where the j -th instance will set its parameter T as $b(j)$ which is roughly 2^j , so that there always exists an instance j_* such that $b(j_*)$ is very close to the unknown T_* .

We will therefore run a scale invariant meta algorithm with a correction term as in Chen et al. (2021b) to obtain the desired bound (details in Algorithm D.2).

Theorem 6.3. *For any $K \geq MT$, Algorithm D.2 guarantees:*

$$\mathbb{E}[R_K] \leq \mathcal{O} \left(B\sqrt{DT_*K \log(T_*M)} + BM\sqrt{\frac{DT_* \log(T_*M)}{K}} + \sqrt{DT_*K \log N \sqrt{T_*}} \right),$$

where $B = \lceil \log_2 \log_2 M \rceil$ and $N = \mathcal{O}(\log K)$.

Proof. We start with a regret decomposition into the regret of the meta algorithm *w.r.t.* finding j_* and the regret of the j_* instance *w.r.t.* the best policy:

$$\begin{aligned} \mathbb{E}[R_K] &= \mathbb{E} \left[\sum_{k=1}^K \sum_{j=1}^N p_k(j) \langle q_k^j, c_k \rangle - \sum_{k=1}^K \langle q_k^*, c_k \rangle \right] \\ &= \mathbb{E} \left[\sum_{k=1}^K \sum_{j=1}^N p_k(j) \langle q_k^j, c_k \rangle - \langle q_k^{j_*}, c_k \rangle \right] + \mathbb{E} \left[\sum_{k=1}^K \langle q_k^* - q_k^{j_*}, c_k \rangle \right] \\ &= \underbrace{\mathbb{E} \left[\sum_{k=1}^K \langle p_k(j) - e_{j_*}, c_k \rangle \right]}_{\text{Meta-Regret}} + \underbrace{\mathbb{E} \left[\sum_{k=1}^K \langle q_k^* - q_k^{j_*}, c_k \rangle \right]}_{j_*\text{-Regret}} \end{aligned}$$

where e_{j_*} is the basis vector with the j_* coordinate equal to 1. By Lemma D.1 the j_* -Regret is bounded by a $\mathcal{O} \left(B\sqrt{DT_*K \log(b(j_*)M)} + BM\sqrt{\frac{Db(j_*) \log(b(j_*)M)}{K}} \right)$, where $b(j_*) = \Theta(T_*)$.

This also allows to say that:

$$\begin{aligned}
 \mathbb{E} \left[\sum_{k=1}^K \langle q_k^{j_\star}, c_k \rangle \right] &\leq \mathbb{E} \left[\sum_{k=1}^K \langle q_{\pi^\star}, c_k \rangle \right] + \mathcal{O} \left(B\sqrt{DTK \log(b(j_\star)M)} + BM\sqrt{\frac{Db(j_\star) \log(b(j_\star)M)}{K}} \right) \\
 &\leq \mathbb{E} \left[\sum_{k=1}^K \langle q_{\pi_f}, c_k \rangle \right] + \mathcal{O} \left(B\sqrt{DTK \log(b(j_\star)M)} + BM\sqrt{\frac{Db(j_\star) \log(b(j_\star)M)}{K}} \right) \\
 &\leq DK + \mathcal{O} \left(B\sqrt{DTK \log(b(j_\star)M)} + BM\sqrt{\frac{Db(j_\star) \log(b(j_\star)M)}{K}} \right).
 \end{aligned}$$

For the meta algorithm regret, we can use Lemma 12 of Chen et al. (2021b) which guarantees that:

$$\sum_{k=1}^K \langle p_k - e_{j_\star}, c_k \rangle \leq \frac{2 + \log \left(N\sqrt{\frac{b(j_\star)}{b(1)}} \right)}{\eta_{j_\star}} + 4\eta_{j_\star} b(j_\star) \sum_{k=1}^K \langle q_k^{j_\star}, c_k \rangle.$$

Therefore, when $K \geq MT$:

$$\begin{aligned}
 \langle p_k - e_{j_\star}, c_k \rangle &\leq \mathcal{O} \left(\frac{\log N\sqrt{T_\star}}{\eta_{j_\star}} + \eta_{j_\star} T_\star \left(DK + B\sqrt{DTK \log T_\star M} + BM\sqrt{\frac{DT_\star \log T_\star M}{K}} \right) \right) \\
 &= \mathcal{O} \left(\frac{\log N\sqrt{T_\star}}{\eta_{j_\star}} + \eta_{j_\star} T_\star DK \right) = \mathcal{O} \left(\sqrt{DT_\star K \log N\sqrt{T_\star}} \right)
 \end{aligned}$$

with $\eta_j = \sqrt{\frac{\log N\sqrt{b(j)}}{b(j)DK}}$, giving the desired result. $\square$

D.2 Failure of the negative entropy

In this appendix, we prove our lower bound result for OMD with the negative entropy from Section 6.3. We first restate the result.

Theorem 6.1. *For any $S \geq 6$, there exists an SSP instance with a fixed horizon of 3, sparsity level $M = 3$, an action space of size $A = 2$ and state space of size S such that the regret of OMD (6.1) with negative-entropy regularization and any step-size $\eta > 0$ after K episodes is $\mathbb{E}[R_K] = \Omega(\min\{\sqrt{K \log S}, K\})$.*

Proof. Fix K even, $S \geq 6$, $A = 2$ and $N = S - 5$. We first describe the SSP instance. Consider the following MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, s_0, g)$, where $\mathcal{S} = \{s_0, s_0^L, s_0^R, s_1^R, \dots, s_N^R, s_1\}$ and $\mathcal{A} = \{a_1, a_2\}$. The transitions and costs (in each episode k) are defined as:

- s_0 : $p(s_0^L | s_0, a) = p(s_0^R | s_0, a) = 1/2$ and $c_k(s_0, a) = 0$ for all $a \in \mathcal{A}$.
- s_0^L : $p(s_g^L | s_0^L, a) = 1$ for all $a \in \mathcal{A}$ and $c_k(s_0^L, a_1) = \frac{1+(-1)^k}{2}$, $c_k(s_0^L, a_2) = 1/2$.
- s_0^R : $p(s_g^R | s_0^R, a_1) = 1$, $p(s_i^R | s_0^R, a_2) = 1/N$ and $c_k(s_0^R, a_1) = 0$, $c_k(s_0^R, a_2) = 1$.
- s_i^R : $p(g | s_i^R, a) = 1$ and $c_k(s_i^R, a) = 0$ for all $a \in \mathcal{A}$.

- $B_k^v(s_0^R, a_1) = v(s_0^R) - v(s_g^R)$
- $B_k^v(s_0^R, a_2) = v(s_0^R) - \eta c_k(s_0^R, a_2) - \frac{1}{N} \sum_{i=1}^N v(s_i^R) = v(s_0^R) - \eta c_k(s_0^R, a_2) - v(s_1^R)$ since by symmetry $v(s_i^R) = v(s_1^R)$ for all $i \geq 1$ and any v solving the convex optimization problem specified in the OMD update.
- $B_k^v(s_i^R, a) = v(s_i^R) = v(s_1^R)$ for all $a \in \mathcal{A}$
- $B_k^v(s_g^R, a) = v(s_g^R)$ for all $a \in \mathcal{A}$

Plugging these into the optimization problem, we obtain (recall the notation $q(s) = \sum_{a \in \mathcal{A}} q(s, a)$):

$$\begin{aligned}
 v_{k+1} &= \arg \min_v \mathcal{D}_k(v) = \arg \min_v \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} q_k(s, a) e^{B_k^v(s, a)} - v(s_0) \\
 &= \arg \min_v q_k(s_0, a_1) e^{v(s_0) - 0.5v(s_0^L) - 0.5v(s_0^R)} + q_k(s_0, a_2) e^{v(s_0) - 0.5v(s_0^L) - 0.5v(s_0^R)} \\
 &\quad + q_k(s_0^L, a_1) e^{v(s_0^L) - \eta c_k(s_0^L, a_1) - v(s_g^L)} + q_k(s_0^L, a_2) e^{v(s_0^L) - \eta c_k(s_0^L, a_2) - v(s_g^L)} \\
 &\quad + q_k(s_g^L, a_1) e^{v(s_g^L)} + q_k(s_g^L, a_2) e^{v(s_g^L)} \\
 &\quad + q_k(s_0^R, a_1) e^{v(s_0^R) - v(s_g^R)} + q_k(s_0^R, a_2) e^{v(s_0^R) - \eta c_k(s_0^R, a_2) - v(s_1^R)} \\
 &\quad + \sum_{i=1}^N \left\{ q_k(s_i^R, a_1) e^{v(s_1^R)} + q_k(s_i^R, a_2) e^{v(s_1^R)} \right\} \\
 &\quad + q_k(s_g^R, a_1) e^{v(s_g^R)} + q_k(s_g^R, a_2) e^{v(s_g^R)} \\
 &\quad - v(s_0) \\
 &= \arg \min_v q_k(s_0) e^{v(s_0) - 0.5v(s_0^L) - 0.5v(s_0^R)} \\
 &\quad + q_k(s_0^L, a_1) e^{v(s_0^L) - \eta c_k(s_0^L, a_1) - v(s_g^L)} + q_k(s_0^L, a_2) e^{v(s_0^L) - \eta c_k(s_0^L, a_2) - v(s_g^L)} \\
 &\quad + q_k(s_g^L, a_1) e^{v(s_g^L)} + q_k(s_g^L, a_2) e^{v(s_g^L)} \\
 &\quad + q_k(s_0^R, a_1) e^{v(s_0^R) - v(s_g^R)} + q_k(s_0^R, a_2) e^{v(s_0^R) - \eta c_k(s_0^R, a_2) - v(s_1^R)} \\
 &\quad + \sum_{i=1}^N \left\{ q_k(s_i^R, a_1) e^{v(s_1^R)} \right\} \\
 &\quad + q_k(s_g^R, a_1) e^{v(s_g^R)} + q_k(s_g^R, a_2) e^{v(s_g^R)} \\
 &\quad - v(s_0).
 \end{aligned}$$

This being a convex optimization problem, it can be solved by differentiating and setting to 0:

$$\begin{aligned}
 \frac{\partial \mathcal{D}_k(v)}{\partial v(s_0)} &= q_k(s_0) e^{v(s_0) - 0.5v(s_0^L) - 0.5v(s_0^R)} - 1 = 0 \\
 \frac{\partial \mathcal{D}_k(v)}{\partial v(s_0^L)} &= -0.5 q_k(s_0) e^{v(s_0) - 0.5v(s_0^L) - 0.5v(s_0^R)} + q_k(s_0^L, a_1) e^{v(s_0^L) - \eta c_k(s_0^L, a_1) - v(s_g^L)} + q_k(s_0^L, a_2) e^{v(s_0^L) - \eta c_k(s_0^L, a_2) - v(s_g^L)} \\
 &= -0.5 + q_k(s_0^L, a_1) e^{v(s_0^L) - \eta c_k(s_0^L, a_1) - v(s_g^L)} + q_k(s_0^L, a_2) e^{v(s_0^L) - \eta c_k(s_0^L, a_2) - v(s_g^L)} = 0 \\
 \frac{\partial \mathcal{D}_k(v)}{\partial v(s_g^L)} &= -q_k(s_0^L, a_1) e^{v(s_0^L) - \eta c_k(s_0^L, a_1) - v(s_g^L)} - q_k(s_0^L, a_2) e^{v(s_0^L) - \eta c_k(s_0^L, a_2) - v(s_g^L)} + q_k(s_g^L, a_1) e^{v(s_g^L)} + q_k(s_g^L, a_2) e^{v(s_g^L)} \\
 &= q_k(s_g^L, a_1) e^{v(s_g^L)} + q_k(s_g^L, a_2) e^{v(s_g^L)} - 0.5 = 0 \\
 \frac{\partial \mathcal{D}_k(v)}{\partial v(s_0^R)} &= -0.5 q_k(s_0) e^{v(s_0) - 0.5v(s_0^L) - 0.5v(s_0^R)} + q_k(s_0^R, a_1) e^{v(s_0^R) - v(s_g^R)} + q_k(s_0^R, a_2) e^{v(s_0^R) - \eta c_k(s_0^R, a_2) - v(s_1^R)}
 \end{aligned}$$

$$\begin{aligned}
 &= -0.5 + q_k(s_0^R, a_1)e^{v(s_0^R)-v(s_g^R)} + q_k(s_0^R, a_2)e^{v(s_0^R)-\eta c_k(s_0^R, a_2)-v(s_1^R)} = 0 \\
 \frac{\partial \mathcal{D}_k(v)}{\partial v(s_1^R)} &= -q_k(s_0^R, a_2)e^{v(s_0^R)-\eta c_k(s_0^R, a_2)-v(s_1^R)} + e^{v(s_1^R)} \sum_{i=1}^N q_k(s_i^R) = 0 \\
 \frac{\partial \mathcal{D}_k(v)}{\partial v(s_g^R)} &= -q_k(s_0^R, a_1)e^{v(s_0^R)-v(s_g^R)} + q_k(s_g^R)e^{v(s_g^R)} = 0.
 \end{aligned}$$

Let's look specifically at the case $k = 0$ ($q_0(s, a) = 1, q_0(s) = 2, c_0(s, a) = 0$ for all $s \in \mathcal{S}, a \in \mathcal{A}$). For the left part of the MDP we have:

$$\begin{aligned}
 \frac{\partial \mathcal{D}_k(v)}{\partial v(s_0^L)} = 0 &\implies 2e^{v(s_0^L)-v(s_g^L)} = 0.5 \implies e^{v(s_0^L)} = 0.25e^{v(s_g^L)} \\
 \frac{\partial \mathcal{D}_k(v)}{\partial v(s_g^L)} = 0 &\implies e^{v(s_g^L)} = 0.25 \implies e^{v(s_0^L)} = 0.25^2 \\
 &\implies q_1(s_0^L, a) = e^{B_0^v(s_0^L, a)} = \frac{0.25^2}{0.25} = 0.25, \text{ for all } a \in \mathcal{A}.
 \end{aligned}$$

For the right part of the MDP, we have:

$$\begin{aligned}
 \frac{\partial \mathcal{D}_k(v)}{\partial v(s_0^R)} = 0 &\implies e^{v(s_0^R)-v(s_g^R)} + e^{v(s_0^R)-v(s_1^R)} = 0.5 \implies e^{v(s_0^R)} = \frac{0.5}{e^{-v(s_g^R)} + e^{-v(s_1^R)}} \\
 \frac{\partial \mathcal{D}_k(v)}{\partial v(s_1^R)} = 0 &\implies e^{v(s_0^R)-v(s_1^R)} = 2Ne^{v(s_1^R)} \implies e^{v(s_1^R)} = \frac{1}{\sqrt{2N}}e^{0.5v(s_0^R)} \\
 \frac{\partial \mathcal{D}_k(v)}{\partial v(s_g^R)} = 0 &\implies e^{v(s_0^R)-v(s_g^R)} = 2e^{v(s_g^R)} \implies e^{v(s_g^R)} = \frac{1}{\sqrt{2}}e^{0.5v(s_0^R)} \\
 &\implies e^{v(s_0^R)} = \frac{0.5}{\sqrt{2}e^{-0.5v(s_0^R)} + \sqrt{2N}e^{-0.5v(s_0^R)}} \\
 &\implies e^{0.5v(s_0^R)} = \frac{0.5}{\sqrt{2} + \sqrt{2N}} \\
 &\implies e^{v(s_0^R)} = \frac{0.25}{(\sqrt{2} + \sqrt{2N})^2} \\
 &\implies q_1(s_0^R, a_1) = e^{B_0^v(s_0^R, a_1)} = e^{v(s_0^R)-v(s_g^R)} = \sqrt{2}e^{0.5v(s_0^R)} = \frac{0.5}{1 + \sqrt{N}} \\
 q_1(s_0^R, a_2) &= e^{B_0^v(s_0^R, a_2)} = e^{v(s_0^R)-v(s_1^R)} = \sqrt{2N}e^{0.5v(s_0^R)} = \frac{0.5\sqrt{N}}{1 + \sqrt{N}} \\
 q_1(s_1^R, a) &= e^{B_0^v(s_1^R, a)} = e^{v(s_1^R)} = \frac{1}{\sqrt{2N}}e^{0.5v(s_0^R)} = \frac{0.25}{\sqrt{N}(1 + \sqrt{N})} \\
 q_1(s_g^R, a) &= e^{B_0^v(s_g^R, a)} = e^{v(s_g^R)} = \frac{1}{\sqrt{2}}e^{0.5v(s_0^R)} = \frac{0.25}{1 + \sqrt{N}}.
 \end{aligned}$$

Let's now look at general $k \geq 1$: Since q_k is an occupancy measure, it satisfies the properties of the dynamics of the MDP (see the definition of $\Delta(T)$ in Section 6.2) and we have that for any $s \in \mathcal{S}$: $\sum_{a \in \mathcal{A}} q_k(s, a) = \sum_{s' \in \mathcal{S}} \sum_{a' \in \mathcal{A}} p(s|s', a')q_k(s', a') + \mathbb{I}\{s = s_0\}$. In particular, this gives

- $s = s_0$: $q_k(s_0) = 1$
- $s = s_0^L$: $q_k(s_0^L) = 0.5q_k(s_0) = 0.5$
- $s = s_g^L$: $q_k(s_g^L) = q_k(s_0^L) = 0.5$

- $s = s_0^R$: $q_k(s_0^R) = 0.5q_k(s_0) = 0.5$
- $s = s_g^R$: $q_k(s_g^R) = q_k(s_0^R, a_1)$
- $s = s_1^R$: $q_k(s_1^R) = \frac{1}{N}q_k(s_0^R, a_2)$

This leads to the following simplifications in the derivatives of $\mathcal{D}_k(v)$:

$$\begin{aligned}\frac{\partial \mathcal{D}_k(v)}{\partial v(s_0)} &= e^{v(s_0)-0.5v(s_0^L)-0.5v(s_0^R)} - 1 = 0 \\ \frac{\partial \mathcal{D}_k(v)}{\partial v(s_0^L)} &= -0.5 + q_k(s_0^L, a_1)e^{v(s_0^L)-\eta c_k(s_0^L, a_1)-v(s_g^L)} + (1/2 - q_k(s_0^L, a_1))e^{v(s_0^L)-\eta c_k(s_0^L, a_2)-v(s_g^L)} = 0 \\ \frac{\partial \mathcal{D}_k(v)}{\partial v(s_g^L)} &= 0.5e^{v(s_g^L)} - 0.5 = 0 \implies e^{v(s_g^L)} = 1 \\ \frac{\partial \mathcal{D}_k(v)}{\partial v(s_0^R)} &= -0.5 + q_k(s_0^R, a_1)e^{v(s_0^R)-v(s_g^R)} + (1/2 - q_k(s_0^R, a_1))e^{v(s_0^R)-\eta-v(s_1^R)} = 0 \\ \frac{\partial \mathcal{D}_k(v)}{\partial v(s_1^R)} &= -q_k(s_0^R, a_2)e^{v(s_0^R)-\eta-v(s_1^R)} + q_k(s_0^R, a_2)e^{v(s_1^R)} = 0 \implies e^{v(s_0^R)} = e^{\eta+2v(s_1^R)} \\ \frac{\partial \mathcal{D}_k(v)}{\partial v(s_g^R)} &= -q_k(s_0^R, a_1)e^{v(s_0^R)-v(s_g^R)} + q_k(s_0^R, a_1)e^{v(s_g^R)} = 0 \implies e^{v(s_0^R)} = e^{2v(s_g^R)}.\end{aligned}$$

Left part of the MDP:

$$\begin{aligned}\frac{\partial \mathcal{D}_k(v)}{\partial v(s_0^L)} = 0 &\implies q_k(s_0^L, a_1)e^{v(s_0^L)-\eta\frac{1+(-1)^k}{2}} + (0.5 - q_k(s_0^L, a_1))e^{v(s_0^L)-0.5\eta} = 0.5 \\ &\implies e^{v(s_0^L)} = \frac{0.5}{q_k(s_0^L, a_1)e^{-\eta\frac{1+(-1)^k}{2}} + (0.5 - q_k(s_0^L, a_1))e^{-0.5\eta}} \\ k+1=2 &\implies e^{v(s_0^L)} = \frac{0.5}{0.25 + 0.25e^{-0.5\eta}} \\ &\implies q_2(s_0^L, a_1) = q_1(s_0^L, a_1)e^{B_1^v(s_0^L, a_1)} = 0.25e^{v(s_0^L)-v(s_g^L)} = \frac{0.5}{1 + e^{-0.5\eta}} \\ k+1=3 &\implies e^{v(s_0^L)} = \frac{0.5}{q_1(s_0^L, a_1)e^{-\eta} + (0.5 - q_1(s_0^L, a_1))e^{-0.5\eta}} \\ &\implies e^{v(s_0^L)} = \frac{0.5}{\frac{0.5}{1+e^{-0.5\eta}}e^{-\eta} + \frac{0.5e^{-0.5\eta}}{1+e^{-0.5\eta}}e^{-0.5\eta}} = 0.25\frac{e^\eta}{q_2(s_0^L, a_1)} \\ &\implies q_3(s_0^L, a_1) = q_2(s_0^L, a_1)e^{v(s_0^L)-\eta-v(s_g^L)} = 0.25 \\ k \text{ even} &\implies q_k(s_0^L, a_1) = \frac{0.5}{1 + e^{-0.5\eta}} \\ k \text{ odd} &\implies q_k(s_0^L, a_1) = 0.25,\end{aligned}$$

where the last two lines follow by a straightforward induction. Hence the losses suffered by OMD on the left part of the MDP are:

$$\begin{aligned}\sum_{k=1}^K \left\{ q_k(s_0^L, a_1)c_k(s_0^L, a_1) + q_k(s_0^L, a_2)c_k(s_0^L, a_2) \right\} &= \sum_{k=1}^K \left\{ q_k(s_0^L, a_1) \cdot \frac{1+(-1)^k}{2} + 0.5 \cdot (0.5 - q_k(s_0^L, a_1)) \right\} \\ &= \sum_{k=1}^K \left\{ q_k(s_0^L, a_1) \cdot \frac{(-1)^k}{2} + 0.25 \right\}\end{aligned}$$

$$\begin{aligned}
 &= 0.25K + 0.5 \sum_{t=1}^{K/2} \left\{ q_{2t}(s_0^L, a_1) - q_{2t-1}(s_0^L, a_1) \right\} \\
 &= 0.25K + 0.5 \sum_{t=1}^{K/2} \left\{ \frac{0.5}{1 + e^{-0.5\eta}} - 0.25 \right\} \\
 &= 0.25K + 0.5 \frac{K}{2} \frac{0.5 - 0.25 - 0.25e^{-0.5\eta}}{1 + e^{-0.5\eta}} \\
 &= 0.25K + \frac{K}{16} \cdot \frac{1 - e^{-0.5\eta}}{1 + e^{-0.5\eta}} \\
 &= 0.25K + \frac{K}{16} \cdot \min\left\{ \frac{\eta}{5}, \frac{1}{2} \right\}. \tag{D.1}
 \end{aligned}$$

Right part of the MDP:

$$\begin{aligned}
 \frac{\partial \mathcal{D}_k(v)}{\partial v(s_0^R)} = 0 &\implies e^{v(s_0^R)} = \frac{0.5}{q_k(s_0^R, a_1)e^{-v(s_g^R)} + (1/2 - q_k(s_0^R, a_1))e^{-\eta - v(s_1^R)}} \\
 \frac{\partial \mathcal{D}_k(v)}{\partial v(s_1^R)} = 0 &\implies v(s_0^R) = \eta + 2v(s_1^R) \\
 \frac{\partial \mathcal{D}_k(v)}{\partial v(s_g^R)} = 0 &\implies v(s_0^R) = v(s_g^R) \\
 \implies e^{v(s_0^R)} &= \frac{0.5}{q_k(s_0^R, a_1)e^{-0.5v(s_0^R)} + (1/2 - q_k(s_0^R, a_1))e^{-\eta - 0.5v(s_0^R) + 0.5\eta}} \\
 \implies e^{0.5v(s_0^R)} &= \frac{0.5}{q_k(s_0^R, a_1) + (1/2 - q_k(s_0^R, a_1))e^{-0.5\eta}} = \frac{0.5}{q_k(s_0^R, a_1) + q_k(s_0^R, a_2)e^{-0.5\eta}} \\
 \implies q_{k+1}(s_0^R, a_1) &= q_k(s_0^R, a_1)e^{B_k^v(s_0^R, a_1)} = q_k(s_0^R, a_1)e^{v(s_0^R) - v(s_g^R)} = q_k(s_0^R, a_1)e^{0.5v(s_0^R)} \\
 \implies q_{k+1}(s_0^R, a_1) &= 0.5 \frac{q_k(s_0^R, a_1)}{q_k(s_0^R, a_1) + (1/2 - q_k(s_0^R, a_1))e^{-0.5\eta}} \\
 \implies \frac{q_{k+1}(s_0^R, a_1)}{q_{k+1}(s_0^R, a_2)} &= \frac{q_{k+1}(s_0^R, a_1)}{(1/2 - q_{k+1}(s_0^R, a_1))} = \frac{q_k(s_0^R, a_1)}{(1/2 - q_k(s_0^R, a_1))e^{-0.5\eta}} = e^{0.5\eta} \frac{q_k(s_0^R, a_1)}{q_k(s_0^R, a_2)} \\
 \implies \frac{q_{k+1}(s_0^R, a_1)}{q_{k+1}(s_0^R, a_2)} &= e^{0.5k\eta} \frac{q_1(s_0^R, a_1)}{q_1(s_0^R, a_2)} = e^{0.5k\eta} \frac{0.5}{0.5\sqrt{N}} = \frac{1}{\sqrt{N}} e^{0.5k\eta} \\
 \implies q_{k+1}(s_0^R, a_2) &= \frac{0.5}{1 + \frac{1}{\sqrt{N}}e^{0.5k\eta}} = \frac{0.5\sqrt{N}}{\sqrt{N} + e^{0.5\eta k}}.
 \end{aligned}$$

This also holds for $k = 0$ (as shown above). Hence, the losses suffered by OMD on the right part of the MDP are

$$\begin{aligned}
 \sum_{k=1}^K q_k(s_0^R, a_2)c_k(s_0^R, a_2) &= \sum_{k=1}^K \frac{0.5\sqrt{N}}{\sqrt{N} + e^{0.5k\eta}} \\
 &\geq \int_1^{K+1} \frac{0.5\sqrt{N}}{\sqrt{N} + e^{0.5\eta x}} dx = 0.5K - \int_1^{K+1} \frac{0.5e^{0.5\eta x}}{\sqrt{N} + e^{0.5\eta x}} dx \\
 &= 0.5K - \left[\frac{1}{\eta} \log(\sqrt{N} + e^{0.5\eta x}) \right]_1^{K+1} \\
 &= 0.5K - \frac{1}{\eta} \log(\sqrt{N} + e^{0.5\eta(K+1)}) + \frac{1}{\eta} \log(\sqrt{N} + e^{0.5\eta}) \\
 &\geq 0.5K - \frac{1}{\eta} \log(2e^{0.5\eta(K+1)}) + \frac{1}{\eta} \log \sqrt{N} \quad \text{assuming } \sqrt{N} \leq e^{0.5\eta(K+1)}
 \end{aligned}$$

$$\begin{aligned}
 &= 0.5K - 0.5(K+1) - \frac{1}{\eta} \log 2 + \frac{1}{2\eta} \log N \\
 &= -0.5 + \frac{1}{2\eta} \log \frac{N}{4}.
 \end{aligned}$$

If $\sqrt{N} > e^{0.5\eta(K+1)}$, then we have

$$\begin{aligned}
 0.5K - \frac{1}{\eta} \log(\sqrt{N} + e^{0.5\eta(K+1)}) + \frac{1}{\eta} \log(\sqrt{N} + e^{0.5\eta}) &= 0.5K + \frac{1}{\eta} \log\left(\frac{\sqrt{N} + e^{0.5\eta}}{\sqrt{N} + e^{0.5\eta(K+1)}}\right) \\
 &\geq 0.5K + \frac{1}{\eta} \log\left(\frac{e^{0.5\eta(K+1)} + e^{0.5\eta}}{2e^{0.5\eta(K+1)}}\right) \\
 &\geq 0.5K + \frac{1}{\eta} \log\left(\frac{1 + e^{-0.5\eta K}}{2}\right) \\
 &\geq 0.25K,
 \end{aligned}$$

using that $\frac{1+e^{-0.5Kx}}{2} \geq e^{-0.25Kx}$ since $\cosh(x) \geq 1$. So we have

$$\sum_{k=1}^K q_k(s_0^R, a_2) c_k(s_0^R, a_2) \geq \min\left\{-0.5 + \frac{1}{2\eta} \log \frac{N}{4}, 0.25K\right\}. \quad (\text{D.2})$$

Combining the losses from the left part in (D.1) and from the right part in (D.2), we have:

$$\begin{aligned}
 \sum_{k=1}^K \langle q_k, c_k \rangle &= \sum_{k=1}^K \left\{ q_k(s_0^L, a_1) c_k(s_0^L, a_1) + q_k(s_0^L, a_2) c_k(s_0^L, a_2) \right\} + \sum_{k=1}^K \left\{ q_k(s_0^R, a_2) c_k(s_0^R, a_2) \right\} \\
 &\geq 0.25K + \frac{K}{16} \cdot \min\left\{\frac{\eta}{5}, \frac{1}{2}\right\} + \min\left\{-0.5 + \frac{1}{2\eta} \log \frac{N}{4}, 0.25K\right\}.
 \end{aligned}$$

Regret lower-bound: consider $q_\star$ defined as follows:

- $q_\star(s_0, a) = 1/2$ for all $a \in \mathcal{A}$
- $q_\star(s_0^L, a_2) = 1/2$, $q_\star(s_0^L, a_1) = 0$
- $q_\star(s_g^L, a) = 1/4$ for all $a \in \mathcal{A}$
- $q_\star(s_0^R, a_1) = 1/2$, $q_\star(s_0^R, a_2) = 0$, $q_\star(s_i^R, a) = 0$
- $q_\star(s_g^R, a) = 1/4$ for all $a \in \mathcal{A}$

It is straightforward to check that $q_\star$ satisfies the flow constraints and is an occupancy measure. We obtain

$$\begin{aligned}
 \sum_{k=1}^K \langle q_\star, c_k \rangle &= \sum_{k=1}^K \left\{ q_\star(s_0^L, a_2) \cdot 0.5 \right\} = 0.25K \\
 \implies R_K &\geq \sum_{k=1}^K \langle q_k - q_\star, c_k \rangle \geq \frac{K}{16} \cdot \min\left\{\frac{\eta}{5}, \frac{1}{2}\right\} + \min\left\{-0.5 + \frac{1}{2\eta} \log \frac{N}{4}, 0.25K\right\} \\
 &\geq \min\left\{\frac{1}{2} \sqrt{\frac{1}{10} K \log \frac{N}{4}}, \frac{K}{32}\right\} - 0.5.
 \end{aligned}$$

Recalling that $N = S - 5$, we have $R_K = \Omega(\min\{\sqrt{K \log S}, K\})$ for an MDP where the sparsity level is $M = 3$, concluding the proof. $\square$

D.3 Efficient implementation of OMD using our regularizer

In this section, we describe how the OMD update with our regularizer from Section 6.4 defined in (6.4) can be computed efficiently. This closely follows Appendix B.1 of Rosenberg and Mansour (2021), who provide a similar description for the negative entropy.

Recall the regularizer $\psi_p(q) = p \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} q(s, a)^{1+1/p} - p$ for $q \in \mathcal{R}_{\geq 0}^\Gamma$. We have

$$\nabla \psi_p(q) = (p+1) \cdot q(s, a)^{1/p}.$$

The Bregman divergence is defined as:

$$\begin{aligned} D_{\psi_p}(q, q') &= \psi_p(q) - \psi_p(q') - \langle \nabla \psi_p(q'), q - q' \rangle \\ &= \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \left\{ p \cdot q(s, a)^{1+1/p} - p \cdot q'(s, a)^{1+1/p} \right\} \\ &\quad - (p+1) \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \left\{ q'(s, a)^{1/p} q(s, a) - q'(s, a)^{1+1/p} \right\} \\ &= \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \left\{ q'(s, a)^{1+1/p} + q(s, a) \cdot [p \cdot q(s, a)^{1/p} - (p+1) \cdot q'(s, a)^{1/p}] \right\}. \end{aligned}$$

Recall that OMD with the above regularizer computes the occupancy measures as follows - see (6.1):

$$q_1 = \arg \min_{q \in \Delta(T)} \psi_p(q), \quad q_{k+1} = \arg \min_{q \in \Delta(T)} \left\{ \eta \cdot \langle q, c_k \rangle + D_{\psi_p}(q, q_k) \right\}.$$

As shown in Orabona (2019) (Theorem 6.15), each of these steps can be split into an unconstrained minimization step, and a projection step. Thus, q_1 can be computed as follows

$$\begin{aligned} q'_1 &= \arg \min_{q \in \mathcal{R}_{\geq 0}^\Gamma} \psi_p(q) \\ q_1 &= \arg \min_{q \in \Delta(T)} D_{\psi_p}(q, q'_1), \end{aligned}$$

where q'_1 has a closed-form solution $q'_1(s, a) = 1$ for every $s \in \mathcal{S}$ and $a \in \mathcal{A}$. Similarly, q_{k+1} is computed as follows for every $k = 1, \dots, K-1$:

$$\begin{aligned} q'_{k+1} &= \arg \min_{q \in \mathcal{R}_{\geq 0}^\Gamma} \left\{ \eta \cdot \langle q, c_k \rangle + D_{\psi_p}(q, q_k) \right\} \\ q_{k+1} &= \arg \min_{q \in \Delta(T)} D_{\psi_p}(q, q'_{k+1}), \end{aligned}$$

where again q'_{k+1} has a closed-form solution $q'_{k+1}(s, a) = \left[q_k(s, a)^{1/p} - \frac{\eta}{p+1} c_k(s, a) \right]_+^p$ for every $s \in \mathcal{S}$ and $a \in \mathcal{A}$ (follows from straightforwardly differentiating above objective and setting to 0 and accounting for the non-negativity of occupancy measures) - we use notation $a_+ = \max\{0, a\}$.

For the projection step, we start by formulating it as a constrained convex optimization problem:

$$\begin{aligned} \min_{q \in \mathcal{R}^\Gamma} D_{\psi_p}(q, q'_{k+1}) \quad \text{s.t.} \quad & \sum_{a \in \mathcal{A}} q(s, a) - \sum_{s' \in \mathcal{S}} \sum_{a' \in \mathcal{A}} P(s'|s, a) q(s', a') = \mathbb{I}\{s = s_0\} \quad \forall s \in \mathcal{S} \\ & \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} q(s, a) \leq T \\ & q(s, a) \geq 0 \quad \forall (s, a) \in \mathcal{S} \times \mathcal{A}. \end{aligned}$$

The problem can be solved by considering the Lagrangian with Lagrange multipliers λ and $\{v(s)\}_{s \in \mathcal{S}}$:

$$\begin{aligned} \mathcal{L}(q, \lambda, v) &= D_{\psi_p}(q, q'_{k+1}) + \lambda \left(\sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} q(s, a) - T \right) + \sum_s v(s) \left(\sum_{s', a'} P(s'|s, a') q(s', a') + \mathbb{I}\{s = s_0\} - \sum_a q(s, a) \right) \\ &= D_{\psi_p}(q, q'_{k+1}) + \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} q(s, a) \left(\lambda + \sum_{s' \in \mathcal{S}} P(s'|s, a) v(s') - v(s) \right) + v(s_0) - \lambda T, \end{aligned}$$

Differentiating the Lagrangian with respect to any $q(s, a)$ and setting to 0, we get

$$\begin{aligned} \frac{\partial \mathcal{L}(q, \lambda, v)}{\partial q(s, a)} &= \nabla \psi_p(q)(s, a) - \nabla \psi_p(q'_{k+1})(s, a) + \lambda + \sum_{s' \in \mathcal{S}} P(s'|s, a) v(s') - v(s) \\ &= (p+1)q(s, a)^{1/p} - (p+1)q'_{k+1}(s, a)^{1/p} + \lambda + \sum_{s' \in \mathcal{S}} P(s'|s, a) v(s') - v(s) = 0. \\ \implies q_{k+1}(s, a) &= \left[q'_{k+1}(s, a)^{1/p} - \frac{\lambda + \sum_{s' \in \mathcal{S}} P(s'|s, a) v(s') - v(s)}{p+1} \right]_+^p. \end{aligned}$$

This formula is also valid for $k = 0$ by setting $c_0(s, a) = 0$ and $q_0(s, a) = 1$ for every $s \in \mathcal{S}$ and $a \in \mathcal{A}$.

To compute the value of λ and v at the optimum, we write the dual problem $\mathcal{D}(\lambda, v) = \min_q \mathcal{L}(q, \lambda, v)$ by substituting q_{k+1} back into $\mathcal{L}$:

$$\mathcal{D}(\lambda, v) = D_{\psi_p}(q_{k+1}, q'_{k+1}) + \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} q_{k+1}(s, a) \left(\lambda + \sum_{s' \in \mathcal{S}} P(s'|s, a) v(s') - v(s) \right) + v(s_0) - \lambda T.$$

Recall that $q'_{k+1}(s, a) = \left[q_k(s, a)^{1/p} - \frac{\eta}{p+1} c_k(s, a) \right]_+^p$, so (ignoring terms independent of λ, v , e.g. $q'_{k+1}(s, a)$):

$$\begin{aligned} D_{\psi_p}(q_{k+1}, q'_{k+1}) &= \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \left\{ q'_{k+1}(s, a)^{1+1/p} + q_{k+1}(s, a) \cdot [p q_{k+1}(s, a)^{1/p} - (p+1) q'_{k+1}(s, a)^{1/p}] \right\} \\ &\propto \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \left\{ q_{k+1}(s, a) \cdot \left[p \left(q'_{k+1}(s, a)^{1/p} - \frac{\lambda + \sum_{s' \in \mathcal{S}} P(s'|s, a) v(s') - v(s)}{p+1} \right)_+ \right. \right. \\ &\quad \left. \left. - (p+1) q'_{k+1}(s, a)^{1/p} \right] \right\} \\ &= \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \left\{ q_{k+1}(s, a) \cdot \left[p \left(q'_{k+1}(s, a)^{1/p} - \frac{\lambda + \sum_{s' \in \mathcal{S}} P(s'|s, a) v(s') - v(s)}{p+1} \right) \right. \right. \\ &\quad \left. \left. - (p+1) q'_{k+1}(s, a)^{1/p} \right] \right\} \quad \text{since if } q_{k+1}(s, a) = 0, \text{ then the whole term is 0} \\ &= \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \left\{ q_{k+1}(s, a) \cdot \left[-q'_{k+1}(s, a)^{1/p} - \frac{p}{p+1} (\lambda + \sum_{s' \in \mathcal{S}} P(s'|s, a) v(s') - v(s)) \right] \right\} \end{aligned}$$

$$\begin{aligned}
 \Rightarrow \mathcal{D}(\lambda, v) &\propto \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \left\{ q_{k+1}(s, a) \cdot \left[-q'_{k+1}(s, a)^{1/p} - \frac{p}{p+1} \left(\lambda + \sum_{s' \in \mathcal{S}} P(s'|s, a) v(s') - v(s) \right) \right. \right. \\
 &\quad \left. \left. + \left(\lambda + \sum_{s' \in \mathcal{S}} P(s'|s, a) v(s') - v(s) \right) \right] \right\} + v(s_0) - \lambda T \\
 &= \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \left\{ q_{k+1}(s, a) \cdot \left[-q'_{k+1}(s, a)^{1/p} + \frac{\lambda + \sum_{s' \in \mathcal{S}} P(s'|s, a) v(s') - v(s)}{p+1} \right] \right\} + v(s_0) - \lambda T \\
 &= - \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} q_{k+1}(s, a)^{1+1/p} + v(s_0) - \lambda T \\
 &= - \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \left[q'_{k+1}(s, a)^{1/p} - \frac{\lambda + \sum_{s' \in \mathcal{S}} P(s'|s, a) v(s') - v(s)}{p+1} \right]_+^{1+p} + v(s_0) - \lambda T.
 \end{aligned}$$

Maximizing the dual gives λ and v or equivalently, we can minimize the negation of the dual:

$$\lambda_{k+1}, v_{k+1} = \arg \min_{\lambda \geq 0, v} \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \left[q'_{k+1}(s, a)^{1/p} - \frac{\lambda + \sum_{s' \in \mathcal{S}} P(s'|s, a) v(s') - v(s)}{p+1} \right]_+^{1+p} - v(s_0) + \lambda T.$$

This is a convex optimization problem subject only to non-negativity constraints, and can be efficiently solved using iterative methods (e.g. gradient descent).

D.4 The benefits of sparsity - lower bound under sparsity

In this appendix, we prove our sparse lower bound result from Section 6.4.2. We first restate the result.

Theorem 6.4. *For any D, T_*, K, S, A with $T^* \geq D \geq 3 \log S$, $S(A-1) \geq 400$, $K \geq \frac{800T^*}{D} \log M$ and $M \geq 101$, there exists an SSP instance with stochastic M -sparse costs, S states and A actions such that its diameter is D , the expected hitting time of the optimal policy is T^* , and the expected regret with respect to the randomness of the losses for any learner after K episodes is $\mathbb{E}[R_K] \geq \Omega(\sqrt{KT^*D \log M})$.*

Proof. Fix $B = \lceil \frac{\log S/2}{\log 2} \rceil - 2$. Fix $N = 2^{B+1} \geq 2^{\frac{\log S/2}{\log 2} - 1} = \frac{S}{4}$. Fix $L = \min\{M-1, N \cdot (A-1)\} \geq \frac{M}{8}$. Fix $D' = D - B - 2$, $T' = T_* - B - 1$, with $T_* \geq D$.

We first describe the SSP instance with stochastic costs. Consider the following MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, s_0, g)$ illustrated in Figure D.2 and that we formally define below.

The first part of the states are represented by a binary tree of depth $B+2$ and allow us to formerly consider the N states at the bottom of the tree that matter, while avoiding an assumption on the existence of a state with $A \approx S$ actions as was done in prior work Chen et al. (2021b). Each non-leaf node corresponds to a state with two actions transitioning (deterministically) to the left or right child respectively. The total number of nodes in the tree is

$$\sum_{i=0}^{B+1} 2^i = 2^{B+2} - 1 \leq 2^{\frac{\log S/2}{\log 2} + 1} - 1 = 2 \frac{S}{2} - 1 = S - 1.$$

The total number of leaf nodes is $N = 2^{B+1} \geq \frac{S}{4}$. Denote the set of states corresponding to the leaf nodes by $\mathcal{S}_\ell = \{s_1, \dots, s_N\}$. The root node is s_0 . There is also one additional state denoted by f (recall that the number of states in the tree is $\leq S-1$).

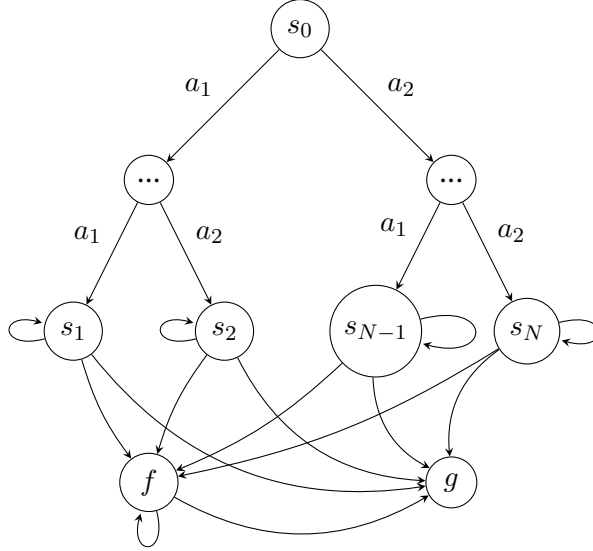


Figure D.2: Diagram illustrating MDP construction for the proof of Theorem 6.4. Details are given below.

We consider the same action set across each state: $\mathcal{A} = \{a_1, \dots, a_{A-1}, a_f\}$. In the states of the binary tree where we have only described two actions, we can consider the other actions to remain in the same state deterministically with 0 cost.

The transitions and costs are defined as follows:

- For all states and actions in the tree that are not leaves, the transitions are specified above. The costs are all 0.
- For $s_i \in \mathcal{S}_\ell$, and $a_j \in \mathcal{A}$
 - if $j = f$, then $p(f|s_i, a_f) = 1$ and $c_k(s_i, a_f) = 0$.
 - if $j \in \{1, \dots, A-1\}$ and $j + (A-1) \cdot (i-1) \leq L$, then $p(g|s_i, a_j) = \frac{1}{T'}$, $p(f|s_i, a_j) = 1 - \frac{1}{T'}$ and the cost is an independent sample from a Bernoulli distribution at each episode k : $c_k(s_i, a_j) \sim \text{Ber}\left(\frac{D'}{2T'}\right)$.
 - if $j \in \{1, \dots, A-1\}$ and $j + (A-1) \cdot (i-1) > L$, then a_j is the same as a_f , i.e. $p(f|s_i, a_j) = 1$ and $c_k(s_i, a_j) = 0$.
- For f ,
 - $p(g|f, a_f) = \frac{1}{D'}$, $p(f|f, a_f) = 1 - \frac{1}{D'}$ and $c_k(f, a_f) = 1$.
 - for all $a_j \in \mathcal{A} \setminus \{a_f\}$, $p(f|f, a_j) = 1$ and $c_k(f, a_j) = 0$.

Denote the above distribution for c_k by $\mathcal{D}$. In each episode there are at most $L + 1 \leq M$ non-zero costs, ensuring the condition on sparsity is respected.

For $i \in \{1, \dots, N\}$, let $\mathcal{A}_i$ correspond to the actions in state $s_i \in \mathcal{S}_L$ which can transition directly to g and $\mathcal{A} \setminus \mathcal{A}_i$ corresponds to the actions which deterministically transition to f (e.g. if $(A-1) \cdot i \leq L$, then $\mathcal{A}_i = \{a_1, \dots, a_{A-1}\}$). For any proper policy π independent of the stochastically generated costs

in episode k , we have

$$\begin{aligned}
 \mathbb{E}_{c_k \sim \mathcal{D}}[J_k^\pi(s_i)] &= \mathbb{E}_{c_k \sim \mathcal{D}}\left[\sum_{a \in \mathcal{A}_i} \pi(a|s_i) \left(c_k(s_i, a) + \left(1 - \frac{1}{T'}\right) J_k^\pi(s_i)\right) + J_k^\pi(f) \sum_{a \notin \mathcal{A}_i} \pi(a|s_i)\right] \\
 &= \sum_{a \in \mathcal{A}_i} \pi(a|s_i) \left(\mathbb{E}_{c_k \sim \mathcal{D}}[c_k(s_i, a)] + \left(1 - \frac{1}{T'}\right) \mathbb{E}_{c_k \sim \mathcal{D}}[J_k^\pi(s_i)]\right) + D' \cdot \sum_{a \notin \mathcal{A}_i} \pi(a|s_i) \\
 &= \sum_{a \in \mathcal{A}_i} \pi(a|s_i) \left(\frac{D'}{2T'} + \left(1 - \frac{1}{T'}\right) \mathbb{E}_{c_k \sim \mathcal{D}}[J_k^\pi(s_i)]\right) + D' \cdot \left(1 - \sum_{a \in \mathcal{A}_i} \pi(a|s_i)\right) \\
 \implies \mathbb{E}_{c_k \sim \mathcal{D}}[J_k^\pi(s_i)] \left(1 - \left(1 - \frac{1}{T'}\right) \sum_{a \in \mathcal{A}_i} \pi(a|s_i)\right) &= \frac{D'}{2T'} \sum_{a \in \mathcal{A}_i} \pi(a|s_i) + D' \cdot \left(1 - \sum_{a \in \mathcal{A}_i} \pi(a|s_i)\right) \\
 \implies \mathbb{E}_{c_k \sim \mathcal{D}}[J_k^\pi(s_i)] &= \frac{\frac{D'}{2T'} \sum_{a \in \mathcal{A}_i} \pi(a|s_i) + D' \cdot \left(1 - \sum_{a \in \mathcal{A}_i} \pi(a|s_i)\right)}{1 - \left(1 - \frac{1}{T'}\right) \sum_{a \in \mathcal{A}_i} \pi(a|s_i)} \\
 &= \frac{D'}{2} \cdot \frac{\frac{1}{T'} \sum_{a \in \mathcal{A}_i} \pi(a|s_i) + 2 \left(1 - \sum_{a \in \mathcal{A}_i} \pi(a|s_i)\right)}{\frac{1}{T'} \sum_{a \in \mathcal{A}_i} \pi(a|s_i) + \left(1 - \sum_{a \in \mathcal{A}_i} \pi(a|s_i)\right)} \\
 &\geq \frac{D'}{2}.
 \end{aligned}$$

The optimal policy π^* is the policy that takes actions in the binary tree to reach state s_{i^*} and then $\pi^*(a_{j^*}|s_{i^*}) = 1$ for $i^*, j^* = \arg \min_{i,j:j+(A-1) \cdot i \leq L} \sum_{k=1}^K c_k(s_i, a_j)$. We have $J_k^{\pi^*}(s_0) = J_k^{\pi^*}(s_{i^*})$ and for any $k \geq 1$

$$\begin{aligned}
 J_k^{\pi^*}(s_{i^*}) &= c_k(s_{i^*}, a_{j^*}) + \left(1 - \frac{1}{T'}\right) J_k^{\pi^*}(s_{i^*}) \\
 \implies J_k^{\pi^*}(s_0) &= T' c_k(s_{i^*}, a_{j^*}) \\
 \implies \sum_{k=1}^K J_k^{\pi^*}(s_0) &= T' \sum_{k=1}^K c_k(s_{i^*}, a_{j^*}) = T' \min_{i,j:j+(A-1) \cdot i \leq L} \sum_{k=1}^K c_k(s_i, a_j).
 \end{aligned}$$

Hence,

$$\begin{aligned}
 \mathbb{E}_{c_1, \dots, c_K \sim \mathcal{D}}[R_K] &\geq \frac{D'}{2} \cdot K - T' \cdot \mathbb{E}_{c_1, \dots, c_K \sim \mathcal{D}}\left[\min_{i,j:j+(A-1) \cdot i \leq L} \sum_{k=1}^K c_k(s_i, a_j)\right] \\
 &= T' \cdot \left(\frac{D'}{2T'} \cdot K - \mathbb{E}_{c_1, \dots, c_K \sim \mathcal{D}}\left[\min_{i,j:j+(A-1) \cdot i \leq L} \sum_{k=1}^K c_k(s_i, a_j)\right]\right) \\
 &= T' \cdot \mathbb{E}_{c_1, \dots, c_K \sim \mathcal{D}}\left[\max_{i,j:j+(A-1) \cdot i \leq L} \sum_{k=1}^K \left(\frac{D'}{2T'} - c_k(s_i, a_j)\right)\right]
 \end{aligned}$$

We now apply Theorem D.1 with $p = 1 - \frac{D'}{2T'} \geq \frac{1}{2}$, $d = L \geq 100$ (since $S(A-1) \geq 400$ and $M \geq 101$) and $n = K \geq \frac{800T^*}{D} \log M \geq \frac{400T'}{D'} \log M \geq 200 \frac{p}{1-p} \log d$. We obtain:

$$\begin{aligned}
 \sup_{c_1, \dots, c_K} \mathbb{E}[R_K] &\geq \mathbb{E}_{c_1, \dots, c_K \sim \mathcal{D}}[R_K] \geq 0.02T' \sqrt{K \left(1 - \frac{D'}{2T'}\right) \cdot \frac{D'}{2T'} \cdot \log L} - 1.5T' \\
 &= \Omega\left(\sqrt{KT^*D \log M}\right),
 \end{aligned}$$

since $L \geq M/8$. Note that since $T_\star \geq D$, the hitting-time of the fast-policy is $D' + B + 2 = D$ and the hitting time of the optimal is $T' + B + 1 = T_\star$, as required. This concludes the proof. $\square$

D.5 Lower bound under unknown transitions

Theorem 6.5. *For any D, K, S, A with $S \geq 2, A \geq 16, D \geq 2$ and $K \geq SA$, there exists an SSP instance with $M = 1$, S states and A actions such that its diameter is D and the expected regret for any learner without knowledge of the transitions after K episodes is $\mathbb{E}[R_K] \geq \Omega(D\sqrt{SAK})$.*

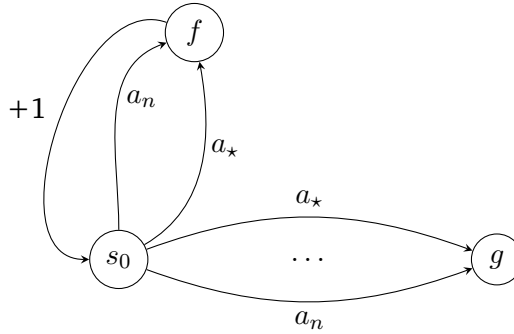


Figure D.3: base case

Proof. The idea is to inject sparsity into the lower bound construction of Rosenberg et al. (2020) and to see if sparsity helps. Imagine the simple SSP in Figure D.3, where at state s_0 there are A available actions, all with zero cost, while in the state f there is only one deterministic action with unit cost going back to s_0 . Among them, there exists an action $a_\star$ such that the transition probabilities are given by: $P(g \mid s_0, a) = \frac{1}{D} - \epsilon \mathbb{I}(a \neq a_\star)$, and consequently, $P(f \mid s_0, a) = 1 - \frac{1}{D} + \epsilon \mathbb{I}(a \neq a_\star)$. The cost is therefore only suffered when the selected action transitions to the f state. This will therefore not increase the hitting time of any proper deterministic policy while still inducing the desired sparsity.

Clearly, the optimal policy plays $a_\star$ at every time step to reach the goal as fast as possible and therefore $J^{\pi_\star}(s_0) = D - 1$.

Now, denote with N_k the number of steps that the learner spends in s_0 in episode k and $N_k^\star$ the number of steps that the learner picks action $a_\star$ in episode k . Note that N_k is also the total cost that the learner suffers during episode k minus one (since the last transition will not be paid). Thanks to our construction we can still prove Lemma C.1 in Rosenberg et al. (2020) as follows:

Lemma D.2.

$$\mathbb{E}[N_k] - 1 - J^{\pi_\star} = \epsilon \mathbb{E}[N_k - N_k^\star]$$

Proof.

$$\begin{aligned} \mathbb{E}[N_k] &= \sum_{t=1}^{\infty} P[s_t = s_0] \\ &= 1 + \sum_{t=2}^{\infty} P[s_t = s_0] \end{aligned}$$

$$\begin{aligned}
 &= 1 + \sum_{t=2}^{\infty} P[s_t = s_0 \mid s_{t+1} = s_0, a_{t-1} = a_{\star}] P[s_{t+1} = s_0, a_{t-1} = a_{\star}] \\
 &+ \sum_{t=2}^{\infty} P[s_t = s_0 \mid s_{t+1} = s_0, a_{t-1} \neq a_{\star}] P[s_{t+1} = s_0, a_{t-1} \neq a_{\star}] \\
 &= 1 + \left(1 - \frac{1}{D}\right) \sum_{t=2}^{\infty} P[s_{t+1} = s_0, a_{t-1} = a_{\star}] + \left(1 - \frac{1-\epsilon}{D}\right) \sum_{t=2}^{\infty} P[s_{t+1} = s_0, a_{t-1} \neq a_{\star}]
 \end{aligned}$$

Rearranging gives:

$$\mathbb{E}[N_k] - D = \epsilon \mathbb{E}[N_k - N_k^*]$$

Adding and subtracting 1 gives the desired result. $\square$

Hence:

$$\mathbb{E}[R_K] = \sum_{k=1}^K \sum_{i=1}^{I_k} c_k(s_k^i, a_k^i) - \sum_{k=1}^K J^{\pi^*}(s_0) = \mathbb{E}[N_k] - 1 - J^{\pi^*} = \epsilon[N - N^*]$$

where $N = \sum_{k=1}^K N_k$ and $N^* = \sum_{k=1}^K N_k^*$. Since we recovered Lemma C.1 in Rosenberg et al. (2020) as the starting block of the proof, following the derivation we can lower bound N in expectation and upper bound the expected value of N^* to retrieve

Lemma D.3 (Theorem C.4 in Rosenberg et al. (2020)). *Suppose that $D \geq 2$, $\epsilon \in (0, 1/8)$ and $A > 16$, for the problem described above we have:*

$$\mathbb{E}[R_K] \geq \epsilon K D \left(\frac{1}{8} - 2\epsilon \sqrt{\frac{2K}{A}} \right)$$

Now consider the following MDP. Let $\mathcal{S}$ be the set of states disregarding g and s_0 . The initial state s_0 has only one action which leads uniformly at random into one of the states $s \in \mathcal{S}$, where each one has its own optimal action a_s^* . Then the transition distributions are defined $P(g \mid a_s^*, s) = 1/D$, $P(s \mid a_s^*, s) = 1 - 1/D$, and $P(g \mid a, s) = (1 - \epsilon)/D$, $P(s \mid a, s) = 1 - (1 - \epsilon)/D$ for any other action $a \in \mathcal{A} \setminus \{a_s^*\}$. Note that for each state, the learner is faced with a simple problem as the one described above. Therefore, we can apply Lemma D.3 for each state separately and lower bound the learner's expected regret the sum of the regrets suffered at each state, which would depend on the number of times each state $s \in \mathcal{S}$ is visited from the initial state. Since reaching each state has uniform probability, there are many states (constant fraction) that are chosen $\Theta(K/S)$ times. Summing the regret bounds and choosing ϵ , gives the desired bound.

Denote by K_s the number of episodes the state $s \in \mathcal{S}$ is visited:

$$\mathbb{E}[R_K] \geq \sum_{s \in \mathcal{S}} \mathbb{E} \left[\epsilon K_s D \left(\frac{1}{8} - 2\epsilon \sqrt{\frac{2K_s}{A}} \right) \right] = \frac{\epsilon K D}{8} - 2\epsilon^2 D \sqrt{\frac{2}{A}} \sum_{s \in \mathcal{S}} \mathbb{E} \left[K_s^{3/2} \right]$$

Then:

$$\sum_{s \in \mathcal{S}} \mathbb{E} \left[K_s^{3/2} \right] \leq \sum_{s \in \mathcal{S}} \sqrt{\mathbb{E}[K_s]} \sqrt{\mathbb{E}[K_s^2]} = \sum_{s \in \mathcal{S}} \sqrt{\mathbb{E}[K_s]} \sqrt{\mathbb{E}[K_s^2] + \mathbb{V}[K_s]}$$

$$= \sum_{s \in S} \sqrt{\frac{K}{S}} \sqrt{\frac{K^2}{S^2} + \frac{K(S-1)}{S^2}} \leq K \sqrt{\frac{2K}{S}}$$

Leading to:

$$\mathbb{E}[R_K] \geq \frac{\epsilon K D}{8} - 2\epsilon^2 D K \sqrt{\frac{2K}{SA}} \geq \frac{1}{1024} D \sqrt{SAK}$$

for $\epsilon = 1/64 \sqrt{SA/K}$ $K \geq SA$, concluding the proof. $\square$

D.6 Lower bound on the maximum of asymmetric zero-mean random walks

We extend the lower bound of Orabona and Pál (2015) to asymmetric zero-mean random-walks. We consider $p \geq 1/2$ because it simplifies the proof below (lower-bounding ψ by 1 and upper-bounding C in proof below) and is what we need in the proof of Section 6.4 in Appendix D.4 (we use $p = 1 - D/2T^*$).

Theorem D.1. Fix $p \in [\frac{1}{2}, 1 - \frac{1}{n}]$. Consider random walks $Z_i^{(n)} = \sum_{t=1}^n X_t^i$, where

$$X_t^i = \begin{cases} -p, & \text{w.p. } 1-p \\ 1-p, & \text{w.p. } p. \end{cases}$$

If $n \geq 200 \frac{p}{1-p} \log d$ (also ensures that $p \leq 1 - \frac{1}{n}$) and $d \geq 100$. Then,

$$\mathbb{E}[\max_{1 \leq i \leq d} Z_i^n] \geq 0.02 \sqrt{np(1-p) \log d} - 1.5.$$

Proof. We follow the same lines as Orabona and Pál (2015) who show a special case of the result for $p = 1/2$. We generalize it to $p > 1/2$.

Consider $Z^{(n)} = \sum_{t=1}^n X_t$, a random-walk of length n , then $B_n = Z^{(n)} + pn \sim B(n, p)$, Binomial distribution with parameters n and p .

D.6.1 1st part of the proof:

The 1st part of the proof is all about providing a lower bound on $\mathbb{P}(B_n \geq pn + t - 1)$ in (D.4) for any $t \in [1, np + 1]$.

Lemma D.4 (Generalized version of Lemma 4 of Orabona and Pál (2015), Theorem 2 of McKay (1989)). Let n, k be integers satisfying $n \geq 1$ and $pn \leq k \leq n$. Define $x = \frac{k-pn}{\sqrt{p(1-p)n}}$. Then $B_n \sim B(n, p)$ satisfies

$$\mathbb{P}(B_n \geq k) \geq \sqrt{n} \binom{n-1}{k-1} p^{k-1/2} (1-p)^{n-k+1/2} \cdot \frac{1-\Phi(x)}{\phi(x)},$$

where $\phi(x)$ is the PDF of a standard Normal and $\Phi(x)$ is the CDF. The proof can be found in McKay (1989).

Denote $D(p, q) = p \log \frac{p}{q} + (1-p) \log \frac{1-p}{1-q}$ as the KL-divergence between two Bernoullis.

Lemma D.5 (Generalized version of Theorem 5 of Orabona and Pál (2015)). *Let n, k be integers satisfying $n \geq 1$, and $np \leq k \leq n$. Define $x = \frac{k-pn}{\sqrt{p(1-p)n}}$. Then $B_n \sim B(n, p)$ satisfies*

$$\mathbb{P}(B_n \geq k) \geq \frac{\exp\left(-nD\left(\frac{k}{n}, p\right)\right)}{e^{1/6}\sqrt{2\pi}} \cdot \frac{1 - \Phi(x)}{\phi(x)}.$$

Proof. For $k = n$, we verify the statement of the theorem directly. The left hand side is $\mathbb{P}(B_n \geq n) = p^n$. The right hand side is smaller because $\exp\left\{-nD\left(1, p\right)\right\} = p^n$ and for $x = \sqrt{n\frac{1-p}{p}} > 0$, we have $\frac{1-\Phi(x)}{\phi(x)} \leq \sqrt{2}$ (see e.g. Section 3.3 in Gasull and Utzet (2014)).

For $np \leq k < n$, we first bound the binomial coefficient $\binom{n}{k}$. Stirling's formula for the factorial Robbins (1955) gives for any $n \geq 1$,

$$\sqrt{2\pi n} \left(\frac{n}{e}\right)^n < n! < e^{1/12} \sqrt{2\pi n} \left(\frac{n}{e}\right)^n.$$

Since $0 < np \leq k \leq n-1$, we can use this approximation for k, n and $n-k$ and obtain

$$\begin{aligned} \binom{n}{k} &= \frac{n!}{k!(n-k)!} \\ &> \frac{n^n e^{-n} \sqrt{2\pi n}}{(e^{1/12} k^k e^{-k} \sqrt{2\pi k}) \cdot (e^{1/12} (n-k)^{n-k} e^{-(n-k)} \sqrt{2\pi(n-k)})} \\ &= \frac{1}{e^{1/6} \sqrt{2\pi}} \binom{n}{k}^k \left(\frac{n}{n-k}\right)^{n-k} \sqrt{\frac{n}{k(n-k)}} \\ &= \frac{1}{e^{1/6} \sqrt{2\pi}} \frac{1}{p^k (1-p)^{n-k}} \exp\left\{-nD\left(\frac{k}{n}, p\right)\right\} \sqrt{\frac{n}{k(n-k)}}, \end{aligned}$$

since

$$\begin{aligned} D\left(\frac{k}{n}, p\right) &= \frac{k}{n} \log\left(\frac{k}{np}\right) + \left(1 - \frac{k}{n}\right) \log\left(\frac{1-k/n}{1-p}\right) = -\frac{k}{n} \log\left(\frac{np}{k}\right) - \frac{n-k}{n} \log\left(\frac{n(1-p)}{n-k}\right) \\ \implies \exp\left\{-nD\left(\frac{k}{n}, p\right)\right\} &= \left(\frac{np}{k}\right)^k \cdot \left(\frac{n(1-p)}{n-k}\right)^{n-k} = p^k (1-p)^{n-k} \left(\frac{n}{k}\right)^k \left(\frac{n}{n-k}\right)^{n-k}. \end{aligned}$$

Since $k \geq 1$, we can write the binomial coefficient as $\binom{n-1}{k-1} = \frac{k}{n} \binom{n}{k}$. By Lemma D.4, we have

$$\begin{aligned} \mathbb{P}(B_n \geq k) &\geq \sqrt{n} \binom{n-1}{k-1} p^{k-1/2} (1-p)^{n-k+1/2} \cdot \frac{1 - \Phi(x)}{\phi(x)} \\ &= \sqrt{n} \frac{k}{n} \binom{n}{k} p^{k-1/2} (1-p)^{n-k+1/2} \cdot \frac{1 - \Phi(x)}{\phi(x)} \\ &\geq \frac{1}{e^{1/6} \sqrt{2\pi}} \frac{k}{\sqrt{n}} \sqrt{\frac{n}{k(n-k)}} \frac{p^{k-1/2} (1-p)^{n-k+1/2}}{p^k (1-p)^{n-k}} \exp\left\{-nD\left(\frac{k}{n}, p\right)\right\} \cdot \frac{1 - \Phi(x)}{\phi(x)} \\ &= \frac{1}{e^{1/6} \sqrt{2\pi}} \sqrt{\frac{k}{n-k}} \cdot \sqrt{\frac{1-p}{p}} \exp\left\{-nD\left(\frac{k}{n}, p\right)\right\} \cdot \frac{1 - \Phi(x)}{\phi(x)}. \end{aligned}$$

The result follows from $\sqrt{\frac{k}{n-k}} \geq \sqrt{\frac{np}{n-np}} = \sqrt{\frac{p}{1-p}}$ for $np \leq k \leq n-1$. $\square$

For $k = pn + xn$, the 2nd-order Taylor approximation of $u(x) = D\left(\frac{k}{n}, p\right) = D(p+x, p)$ around 0

is $\frac{x^2}{2p(1-p)}$. We define $\psi : [-p, 1-p] \rightarrow \mathcal{R}$ as the ratio of the divergence and the approximation:

$$\psi(x) = D(p+x, p) \cdot \frac{2p(1-p)}{x^2}.$$

In particular, we have that $1 \leq \psi(x) \leq \frac{p(1-p)}{(x+p)(1-p-x)}$ for $x \in [0, 1-p]$. This can be shown using Taylor's theorem on $u(x)$: for some $z \in [0, x]$,

$$\begin{aligned} D(p+x, p) &= \frac{u^{(2)}(z)}{2} x^2 = \left(\frac{1}{z+p} + \frac{1}{1-p-z} \right) \frac{x^2}{2} = \frac{x^2}{2(z+p)(1-p-z)} \\ \Rightarrow \frac{x^2}{2p(1-p)} &\leq D(p+x, x) \leq \frac{x^2}{2(x+p)(1-p-x)}, \end{aligned} \quad (\text{D.3})$$

since $\frac{1}{(x+p)(1-p-x)}$ is increasing on $[0, 1-p]$.

Let $t \in [1, np+1]$ be a real number. By Lemma D.5 and Lemma 1 in Orabona and Pál (2015) (also Mill's ratio for standard Gaussian Boyd (1959)), we have

$$\begin{aligned} \mathbb{P}(B_n \geq pn + t - 1) &= \mathbb{P}(B_n \geq \lceil pn + t - 1 \rceil) \\ &\geq \frac{\exp\left(-nD\left(\frac{\lceil pn+t-1 \rceil}{n}, p\right)\right)}{e^{1/6}\sqrt{2\pi}} \cdot \frac{\pi}{\pi \frac{\lceil pn+t-1 \rceil - np}{\sqrt{p(1-p)n}} + \sqrt{2\pi}} \\ &\geq \frac{\exp\left(-nD\left(\frac{pn+t}{n}, p\right)\right)}{e^{1/6}\sqrt{2\pi}} \cdot \frac{\pi}{\pi \frac{pn+t-np}{\sqrt{p(1-p)n}} + \sqrt{2\pi}} \\ &= \frac{\exp\left(-nD\left(p + \frac{t}{n}, p\right)\right)}{e^{1/6}\sqrt{2\pi}} \cdot \frac{\pi}{\pi \frac{t}{\sqrt{p(1-p)n}} + \sqrt{2\pi}} \\ &= e^{-1/6} \exp\left(-\frac{1}{2p(1-p)} \psi\left(\frac{t}{n}\right) \cdot \frac{t^2}{n}\right) \cdot \frac{1}{\frac{\sqrt{2\pi}t}{\sqrt{np(1-p)}} + 2}. \end{aligned} \quad (\text{D.4})$$

D.6.2 2nd part of the proof:

We can now turn to the actual proof of the result. Define the event A equal to the case that at least one of the $Z_i^{(n)}$ is greater or equal to $C\sqrt{np(1-p)\log d} - 1$. We will show this event / threshold controls the expectation of the maximum. First, we define C and provide some upper and lower bounds for it. Denote by $f(d) = \sqrt{2 - \frac{2\log\log d}{\log d}}$, then

$$C = C(d, n) = \frac{1}{\sqrt{\psi\left(\sqrt{\frac{2p(1-p)\log d}{n}}\right)}} \sqrt{2 - \frac{2\log\log d}{\log d}} = \frac{1}{\sqrt{\psi\left(\sqrt{\frac{2p(1-p)\log d}{n}}\right)}} f(d). \quad (\text{D.5})$$

We bound the two factors separately:

- $z = \sqrt{\frac{2p(1-p)\log d}{n}} \in [0, \frac{1}{10}(1-p)]$ for $n \geq 200\frac{p}{1-p}\log d$ and so

$$1 \leq \psi(z) \leq \frac{p(1-p)}{(z+p)(1-p-z)} \leq \frac{1-p}{(1-p-\frac{1-p}{10})} = \frac{10}{9}. \quad (\text{D.6})$$

- The function $f(d)$ is as in Orabona and Pál (2015): decreasing on $(1, e^e]$, increasing on $[e^e, +\infty)$, and $\lim_{d \rightarrow \infty} f(d) = \sqrt{2}$. Therefore for all $d \in [5, \infty)$,

$$1.12 \leq f(e^e) \leq f(d) \leq \max\{f(5), \sqrt{2}\} = \sqrt{2}$$

This gives for $n \geq 200 \frac{p}{1-p} \log d$,

$$1 \leq \frac{1.12}{\sqrt{10/9}} \leq C(d, n) \leq \sqrt{2} \quad (\text{D.7})$$

Since $p \geq 1/2$, if $n \geq 200 \frac{p}{1-p} \log d$, then $n > \frac{200}{p(1-p) \log d}$ (if $d \geq 8$) and $n \geq 200 \frac{1-p}{p} \log d$. The above implies:

$$1 < C \sqrt{np(1-p) \log d} \leq np \leq np + 1. \quad (\text{D.8})$$

Finally, we bound the quantity of interest:

$$\begin{aligned} \mathbb{E} \left[\max_{1 \leq i \leq d} Z_i^n \right] &= \mathbb{E} \left[\max_{1 \leq i \leq d} Z_i^n | A \right] \cdot \mathbb{P}(A) + \mathbb{E} \left[\max_{1 \leq i \leq d} Z_i^n | A^C \right] \cdot (1 - \mathbb{P}(A)) \\ &\geq \mathbb{E} \left[\max_{1 \leq i \leq d} Z_i^n | A \right] \cdot \mathbb{P}(A) + \mathbb{E} \left[Z_1^{(n)} | A^C \right] \cdot (1 - \mathbb{P}(A)) \\ &= \mathbb{E} \left[\max_{1 \leq i \leq d} Z_i^n | A \right] \cdot \mathbb{P}(A) + \mathbb{E} \left[Z_1^{(n)} | Z_1^{(n)} < C \sqrt{np(1-p) \log d} - 1 \right] \cdot (1 - \mathbb{P}(A)) \\ &\geq \mathbb{E} \left[\max_{1 \leq i \leq d} Z_i^n | A \right] \cdot \mathbb{P}(A) + \mathbb{E} \left[Z_1^{(n)} | Z_1^{(n)} \leq 0 \right] \cdot (1 - \mathbb{P}(A)) \quad \text{by (D.8)} \\ &\geq (C \sqrt{np(1-p) \log d} - 1) \cdot \mathbb{P}(A) + \mathbb{E} \left[Z_1^{(n)} | Z_1^{(n)} \leq 0 \right] \cdot (1 - \mathbb{P}(A)). \end{aligned} \quad (\text{D.9})$$

First, we lower bound $\mathbb{E} \left[Z_1^{(n)} | Z_1^{(n)} \leq 0 \right]$. Let $\beta = \frac{1}{1 - \sqrt{\frac{2n}{\pi \lfloor n(1-p) \rfloor (n - \lfloor n(1-p) \rfloor)}}}$. For $n \geq \frac{200}{p(1-p)} \log d$, we have $n \geq \frac{205}{\pi p(1-p)} \geq \frac{200 + \pi p}{\pi(1-p)p}$ and $\beta \leq \frac{10}{9}$.

Then Lemma 2.2 in Pelekis and Ramon (2016) combined with Lemma 8 in Doerr (2018) give that for $Y_n \sim B(n, 1-p)$:

$$\mathbb{E} [Y_n | Y_n \geq n(1-p)] < n(1-p) + \beta \sqrt{np(1-p)} < n(1-p) + \frac{10}{9} \sqrt{np(1-p)}.$$

Since $B_n = Z^{(n)} + pn \sim B(n, p)$ can be written as $n - Y_n$, we have:

$$\begin{aligned} \mathbb{E} \left[Z_1^{(n)} | Z_1^{(n)} \leq 0 \right] &= \mathbb{E} \left[B_n | B_n \leq np \right] - np \\ &= \mathbb{E} \left[n - Y_n | n - Y_n \leq np \right] - np \\ &= n - \mathbb{E} \left[Y_n | Y_n \geq n(1-p) \right] - np \\ &\geq n - n(1-p) - \frac{10}{9} \sqrt{np(1-p)} - np \\ &= -\frac{10}{9} \sqrt{np(1-p)}. \end{aligned} \quad (\text{D.10})$$

Next, we lower-bound $\mathbb{P}(A)$:

$$\begin{aligned}
 \mathbb{P}(A) &= 1 - \mathbb{P}(A^C) \\
 &= 1 - \left(\mathbb{P}\left[Z_1^{(n)} < C\sqrt{np(1-p)\log d} - 1\right] \right)^d \\
 &= 1 - \left(\mathbb{P}\left[B_n < np + C\sqrt{np(1-p)\log d} - 1\right] \right)^d \\
 &= 1 - \left(1 - \mathbb{P}\left[B_n \geq np + C\sqrt{np(1-p)\log d} - 1\right] \right)^d \\
 &\geq 1 - \exp\left(-d \cdot \mathbb{P}\left[B_n \geq np + C\sqrt{np(1-p)\log d} - 1\right]\right) \quad \text{since } 1 - x \leq e^{-x} \\
 &\geq 1 - \exp\left(-d \cdot \frac{e^{-1/6} \exp\left(-\frac{1}{2p(1-p)}\psi\left(\frac{C\sqrt{np(1-p)\log d}}{n}\right) \cdot \frac{C^2 np(1-p)\log d}{n}\right)}{\frac{\sqrt{2\pi}C\sqrt{np(1-p)\log d}}{\sqrt{np(1-p)}} + 2}\right) \quad \text{using (D.4) and (D.8)} \\
 &= 1 - \exp\left(-d \cdot \frac{e^{-1/6} \exp\left(-\frac{1}{2}\psi\left(C\sqrt{p(1-p)\log d/n}\right) \cdot C^2 \log d\right)}{\sqrt{2\pi}C\sqrt{\log d} + 2}\right) \\
 &= 1 - \exp\left(-\frac{e^{-1/6} d^{1-\frac{C^2}{2}} \psi\left(C\sqrt{p(1-p)\log d/n}\right)}{\sqrt{2\pi}C\sqrt{\log d} + 2}\right) \\
 &\geq 1 - \exp\left(-\frac{e^{-1/6} d^{1-\frac{C^2}{2}} \psi\left(\sqrt{\frac{2p(1-p)\log d}{n}}\right)}{2\sqrt{\pi}\log d + 2}\right) \quad \text{by (D.7).}
 \end{aligned}$$

We now use that $d^{1-\frac{C^2}{2}} \psi\left(\sqrt{2p(1-p)\log d/n}\right) = \log d$ by the definition of C in (D.5). Hence, we obtain:

$$\mathbb{P}(A) \geq 1 - \exp\left(-\frac{e^{-1/6} \log d}{2\sqrt{\pi}\log d + 2}\right) = 1 - g(d), \quad \text{for } g(d) = \exp\left(-\frac{e^{-1/6} \log d}{2\sqrt{\pi}\log d + 2}\right). \quad (\text{D.11})$$

Putting everything together: we plug (D.10) and (D.11) into (D.9) to get

$$\begin{aligned}
 \mathbb{E}\left[\max_{1 \leq i \leq d} Z_i^n\right] &\geq (C\sqrt{np(1-p)\log d} - 1) \cdot (1 - g(d)) - \frac{10}{9}g(d)\sqrt{np(1-p)} \\
 &= \frac{f(d) \cdot (1 - g(d))}{\sqrt{\psi\left(\sqrt{\frac{2p(1-p)\log d}{n}}\right)}} (\sqrt{np(1-p)\log d} - 2) - \frac{10}{9}g(d)\sqrt{np(1-p)} \quad \text{using (D.5)} \\
 &\geq \frac{f(d)(1 - g(d))}{\sqrt{10/9}} (\sqrt{np(1-p)\log d} - 2) - \frac{10}{9}g(d)\sqrt{np(1-p)} \quad \text{using (D.6)} \\
 &= \sqrt{np(1-p)\log d} \cdot \left(\frac{f(d)(1 - g(d))}{\sqrt{10/9}} - \frac{10}{9} \cdot \frac{g(d)}{\sqrt{\log(d)}} \right) - 2 \frac{f(d)(1 - g(d))}{\sqrt{10/9}} \\
 &\geq 0.02\sqrt{np(1-p)\log d} - 1.5,
 \end{aligned}$$

for $d \geq 100$ (we also used that $\sqrt{np(1-p)\log d} > 2$ in the 2nd inequality). This gives the result. $\square$

SEE YOU SPACE COWBOY...