

Multi-armed Bandits in Action

Présentée le 21 février 2020

à la Faculté informatique et communications
Laboratoire de la Dynamique de l'Information et des Réseaux 2
Programme doctoral en informatique et communications

pour l'obtention du grade de Docteur ès Sciences

par

Farnood SALEHI

Acceptée sur proposition du jury

Prof. O. N. A. Svensson, président du jury
Prof. P. Thiran, Prof. L. E. Celis, directeurs de thèse
Prof. S. Mandt, rapporteur
Dr S. Lattanzi, rapporteur
Prof. . M. Jaggi, rapporteur

Acknowledgments

It would not be possible for me to finish this thesis without the help and encouragement of many people, and I am glad that I have the opportunity to express my gratitude to them. First and foremost, I would like to express my warmest gratitude to my supervisors, Patrick Thiran and Elisa Celis for their mentorship. Their patience with me, sense of responsibility, passion for science, openness for new ideas, and constant and dedicated support give me the courage to explore new research directions, for which I am extremely grateful.

Next, I would like to thank the members of my jury committee: Ola Svensson, Martin Jaggi, Stephan Mandt, and Silvio Lattanzi. I thank them for the time they made in their busy schedules and for accepting to be my thesis reviewers.

I also would like to thank our wonderful staff at the lab: Holly Cogliati-Bauereis, Patricia Hjelt, and Angela Devenoge. Holly Cogliati-Bauereis proofread all my manuscripts and provided invaluable help to improve them. Patricia Hjelt and Angela Devenoge were always there to help me with administrative tasks.

During my Ph.D., I was fortunate to collaborate with a group of clever researchers who are also coauthors in some of my papers. Stephan and Robert, many thanks for giving me the opportunity to do an internship at Disney in which I learned a lot. William, working with you was one of the most fruitful collaborations that I had, thanks. Nicolas, it was a great experience working with you.

One of the best parts of my Ph.D. was working along with nice colleagues throughout my time at EPFL. Brunella and Lucas, thank you for your support and feedback for my manuscripts. William, Victor and Arnout, I sincerely appreciate your friendship, you are always helpful. Thank you Farid, Vincent, Mohamed, Julien, Runwei, Christina, Sébastien, Daniyar, Mahsa, Mladen, Greg and Aswin.

Many thanks to my Iranian friends at EPFL, whom I cannot name all. But I would like to thank Ehsan and Pedram. You were more than just a friend to me and your supportive advice played a big role during the toughest moments of my Ph.D. life.

Finally, I would like to thank my parents and my brother, who have always supported me in all moments of my life. I cannot find any words that can describe my gratitude to

Acknowledgments

them, so I would like to dedicate this thesis to them. Last but not least, I want to thank Fatemeh for bringing energy and love into my life!

Lausanne, November 25, 2019

Farnood Salehi.

Abstract

Making decisions is part and parcel of being human. Among a set of *actions*, we want to choose the one that has the highest *reward*. But the *uncertainty* of the outcome prevents us from always making the right decision. Making decisions under uncertainty can be studied in a principled way by the *exploitation-exploration* framework. The multi-armed bandit (MAB) framework is perhaps one of the simplest, and yet one of the most powerful settings to optimize a sequence of choices within an exploitation-exploration framework. In this thesis, I study several MAB related problems, from three different perspectives: I study (1) how machine-learning (ML) can benefit from MAB algorithms, (2) how MAB algorithms can affect humans, and (3) how human interactions can be studied in the MAB framework.

(1) Optimization lies at the heart of almost all ML algorithms. Stochastic-gradient descent (SGD) and stochastic-coordinate descent (CD) are perhaps two of the most well known and widely used optimization algorithms. In Chapters 2 and 3, I revisit the datapoint and coordinate-selection procedure of SGD and CD from an MAB point of view. The goal is to reduce the training time of ML models. SGD works by estimating, on the fly, the gradient of the cost function by sampling a *datapoint* uniformly at random from a training set, and CD works by updating only a single decision variable (*coordinate*) sampled at random. Updating the model's parameters based on, respectively, different datapoints or different coordinates, yields various improvements. However, a priori, it is not clear which datapoint or coordinate improves the model the most. I address this challenge by studying these problems in an MAB setting, and I develop algorithms to learn the optimal datapoint or coordinate selection strategies. Our methods often significantly reduce the training time of several machine-learning models.

(2) Although some MAB algorithms are designed to improve ML algorithms, they can affect humans' opinions about the outside world. In the personalized recommender systems, the goal is to predict the preference of a user and to suggest the best *content* to them. However, recent studies suggest that a personalized algorithm can learn and propagate systematic *biases and polarize opinions* [Pariser, 2011]. In Chapter 4, to combat bias, I propose to use *constraints* on the distribution from which a content is selected.

Abstract

The constraints can be designed to ameliorate polarization and biases. I combine the classic MAB setting with these constraints and show how an adaptation of an MAB algorithm can lead to a scalable algorithm with provable guarantees for the constrained setting.

(3) Interacting with others is one of the main sources of information for us. In Chapter 5, I study how this natural setting in the human world can be studied in the bandit framework. I extend the classic single decision-maker setting of MAB to multiple decision-makers, where a decision-maker observes her neighbors' decisions and rewards. Presumably, the additional information of the neighbors should improve the decisions. I show how to model such a decision-making process that appeals to the classic MAB framework. I study the new setting, in both stochastic and adversarial MAB frameworks, and I develop algorithms that incorporate the additional knowledge of the peers. Furthermore, I show that our algorithms often significantly outperform the existing algorithms that we could apply to this setting.

Keywords multi-armed bandit, recommender systems, uncertainty, decision making, machine learning, stochastic optimization, stochastic gradient descent, stochastic coordinate descent, polarization

Résumé

Prendre des décisions forme l'essence même des comportements humains. Parmi un ensemble d'actions, nous cherchons à choisir celle qui procurera la plus haute récompense. Mais l'incertitude du résultat nous empêche de toujours prendre la meilleure décision. La prise de décision en présence d'incertitude peut être analysée de manière rigoureuse et raisonnée par l'étude du compromis entre exploration (des différentes options) et exploitation (de la meilleure option). Les bandits manchots (*multi-armed bandits* en anglais, abrégé MAB) offrent peut-être l'une des approches les plus simples, et pourtant l'une des plus puissantes, afin d'optimiser une séquence de choix dans le cadre exploration-exploitation. Dans cette thèse, j'étudie plusieurs problèmes liés aux MAB sous trois angles différents : j'étudie (1) comment l'apprentissage automatique (*machine learning* en anglais) peut bénéficier des algorithmes de MAB, (2) comment les algorithmes de MAB peuvent s'appliquer aux êtres humains et (3) comment les interactions humaines peuvent être étudiées sous l'œil des MAB.

L'optimisation est au cœur de presque tous les algorithmes d'apprentissage automatique. L'algorithme du gradient stochastique (*stochastic gradient descent* en anglais, abrégé SGD) et la descente à coordonnées stochastiques (*stochastic coordinate descent* en anglais, abrégé CD) sont peut-être deux des algorithmes d'optimisation les plus connus et les plus largement utilisés. Dans les Chapitres 2 et 3, je revisite la procédure de sélection des données de SGD et des coordonnées de CD du point de vue des MAB. L'objectif est de réduire le temps d'apprentissage des modèles d'apprentissage automatique. L'algorithme SGD fonctionne en estimant à la volée le gradient de la fonction de coût en échantillonnant un point de donnée de manière uniforme et aléatoire au sein d'un ensemble de données d'apprentissage. L'algorithme CD, lui, fonctionne en mettant à jour une seule variable de décision (coordonnée) échantillonnée au hasard. La mise à jour des paramètres du modèle basée respectivement sur différents points de donnée ou différentes coordonnées fournit diverses améliorations. Cependant, il est difficile *a priori* de savoir quel point de donnée ou quelle coordonnée améliore le plus le modèle. J'aborde ce problème dans un contexte MAB et je développe des algorithmes pour apprendre les stratégies optimales de sélection des points de donnée ou des coordonnées. Nos méthodes réduisent souvent

de manière significative le temps d'apprentissage de plusieurs modèles d'apprentissage automatique.

Bien que certains algorithmes de MAB soient conçus pour améliorer les algorithmes d'apprentissage automatique, ils peuvent également affecter la perception qu'ont les êtres humains de leur environnement. Dans un système de recommandations personnalisées, l'objectif est de prédire les préférences des utilisateurs et de leur recommander le meilleur contenu. Cependant, des études récentes suggèrent qu'un algorithme de recommandations personnalisées peut également apprendre et propager un biais systématique et polariser les opinions [Pariser, 2011]. Dans le Chapitre 4, je propose de combattre ce biais en utilisant des contraintes sur la distribution à partir de laquelle le contenu est sélectionné. Ces contraintes peuvent être conçues spécifiquement pour atténuer les phénomènes de polarisation et de biais. Je combine l'approche classique des MAB avec ces contraintes et je montre comment l'adaptation d'un algorithme de MAB sous contraintes peut conduire à un algorithme flexible avec des garanties théoriques démontrables.

Interagir avec ses pairs est une de nos sources principales d'information. Dans le Chapitre 5, j'étudie comment ce trait inhérent aux êtres humains peut être étudié sous l'œil des MAB. J'étends l'analyse classique des MAB de un à plusieurs agents, où chaque agent peut observer les décisions et les récompenses de ses voisins. L'information supplémentaire venant des voisins permet alors d'améliorer la prise de décision. J'étudie ce nouveau problème dans le cadre des MAB stochastiques, ainsi que des MAB "adversaires". De plus, je développe des algorithmes qui incorporent l'information additionnelle des voisins et je montre que nos algorithmes peuvent fréquemment dépasser de façon significative les performances des algorithmes existants applicables dans ce cas.

Mots-clés bandits manchots, systèmes de recommandations, incertitude, prise de décision, apprentissage automatique, optimisation stochastique, algorithme du gradient stochastique, descente à coordonnées stochastiques, polarisation

Contents

Acknowledgments	iii
Abstract / Résumé	v
Mathematical Notation	xiii
1 Introduction	1
1.1 Motivation	1
1.2 Multi-armed Bandits	4
1.2.1 Framework	5
1.2.2 Adversarial Multi-armed Bandit	6
1.2.3 Stochastic Multi-armed Bandit	9
1.3 Outline and Contributions	12
2 Stochastic Gradient Descent with Bandit Sampling	17
2.1 Introduction	17
2.2 Preliminaries	21
2.3 Related Work	22
2.4 Technical Contributions	25
2.4.1 Multi-armed Bandit Sampling	28
2.5 Combining MABS with Stochastic Optimization Algorithms	38
2.5.1 SGD	39
2.5.2 First-order Algorithms	41
2.6 Empirical Evaluation	42
2.6.1 Experimental Setup	43
2.6.2 Empirical Results for Different Smoothness Ratios τ	45
2.6.3 Empirical Results on Real-World Data	47
2.6.4 Stability	47
2.6.5 Training Time	48

Contents

2.7	Summary	50
Appendix		51
2.A	Proofs	51
2.A.1	Omitted Proofs	51
2.A.2	MABS with IS	52
2.A.3	Omitted Proofs of Section 2.5	53
2.B	PSGD	57
2.C	Definitions	59
3	Coordinate Descent with Bandit Sampling	61
3.1	Introduction	61
3.2	Preliminaries	64
3.3	Related Work	65
3.4	Technical Contributions	66
3.4.1	Marginal Decreases	66
3.4.2	Greedy Algorithms (Full Information Setting)	68
3.4.3	Bandit Algorithms (Partial Information Setting)	76
3.5	Empirical Evaluation	78
3.5.1	Experimental Setup	79
3.5.2	Empirical Results	82
3.6	Summary	83
Appendix		85
3.A	Basic Definitions	85
3.A.1	Basic Definitions	85
3.B	Proofs	85
4	Controlling Polarization in Personalization	91
4.1	Introduction	91
4.1.1	Groups and Polarization	92
4.2	Preliminaries	94
4.2.1	Polarization in Existing Models	94
4.2.2	Constraint setting	95
4.3	Related Work	97
4.4	Technical Contributions	98
4.4.1	Overview of Algorithm 4.1: CONSTRAINED- ϵ -GREEDY	99
4.4.2	Alternate Approaches and Special Cases	102
4.5	Empirical Evaluation	104
4.5.1	Experimental Setup	105
4.5.2	Empirical Results on Effect of Reducing Polarization on the Reward	108
4.5.3	Empirical Results on Polarization Over Time	109
4.6	Summary	109

Appendix	111
4.A Constrained- L_1 -OFUL	111
4.B Laminar Constraints	115
4.B.1 Budget Type Constraints	116
5 Learn from Thy Neighbor	119
5.1 Introduction	119
5.2 Preliminaries	122
5.3 Related Work	123
5.4 Technical Contributions for the Stochastic Setting	125
5.5 Technical Contributions for the Adversarial Setting	126
5.5.1 The EXPN Algorithm	127
5.5.2 Comparison to Alternate Approaches	130
5.5.3 A Centralized Solution for the Network	133
5.6 Empirical Evaluation	134
5.6.1 Adversarial Setting: Experimental setup	134
5.6.2 Adversarial Setting: Empirical Results	134
5.6.3 Stochastic Setting: Empirical Results	136
5.7 Summary	138
Appendix	141
5.A Adversarial Bandits	141
5.B Stochastic Bandits	142
5.B.1 UCBN on Complete Graphs	146
5.B.2 Lower Bound	147
6 Conclusion	149
Bibliography	153
Curriculum Vitae	165

Mathematical Notation

Symbol	Description
x	Plain lowercase letters denote scalar values.
$\mathbf{x} = [x_i]$	Boldface lowercase letters denote column vectors.
$\mathbf{X} = [x_{ij}]$	Boldface uppercase letters denote matrices.
\mathcal{X}	Calligraphic uppercase letters denote sets.
$\mathbb{R}, \mathbb{R}_{>0}, \mathbb{N}$	Number types: real, positive real and natural numbers, respectively.
$[N]$	Set of consecutive natural numbers $\{1, \dots, N\}$.
$\mathbf{P}[\mathcal{A}]$	Probability of the event \mathcal{A} .
$\mathbb{1}_{\{\mathcal{A}\}}$	Indicator variable of the event \mathcal{A} .
$\mathbb{E}[x]$	Expectation of the random variable x .
$\mathbb{V}[x]$	Variance of the random variable x .
$O(f(x))$	$g(x) = O(f(x)) \iff \limsup_{x \rightarrow \infty} g(x) /f(x) < \infty$.
$o(f(x))$	$g(x) = o(f(x)) \iff \lim_{x \rightarrow \infty} g(x)/f(x) = 0$.
$\Omega(f(x))$	$g(x) = \Omega(f(x)) \iff f(x) = O(g(x))$.
$\omega(f(x))$	$g(x) = \omega(f(x)) \iff f(x) = o(g(x))$.
$\ \cdot\ $	Euclidean norm $\ \cdot\ _2$.
$B_\infty(\mathbf{q}, \eta)$	$\{\mathbf{p} : \ \mathbf{p} - \mathbf{q}\ _\infty \leq \eta\}$.

Mathematical Notation

Distribution Domain Density function $f(x)$

$$N(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad \mathbb{R}^D \quad \frac{1}{\sqrt{2\pi|\boldsymbol{\Sigma}|}} \exp \left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}) \right]$$

$$U(a, b) \quad [a, b] \quad \frac{1}{b - a}$$

1 Introduction

1.1 Motivation

Our daily life consists of making choices with uncertain outcomes and consequences. We choose the food we eat, the house where we live, the movies we watch, and the destination of our next trip. In the process of making decisions, sometimes, we can be certain of the outcome, hence we can choose the best decision for our objective. For example, an environmental activist advocates using public transport, as she is assured of its advantage over other transport means in terms of reducing pollution. But, other times we lack the necessary knowledge and we are not aware of the consequences of certain choices. We do not know if we are going to like the new food we ordered, the movie we are going to watch, or the trip we are going to take. The risk of these actions is low and can be appraised. For example, we can test a new food to check if we like the taste or not. But appraising all actions is not without risk. For example, regarding a new medicine, assessing its efficacy on a human might be dangerous.

In making a choice for an action, we are confronted with either *exploiting* an action whose reward is known to us, or *exploring* a new action in the hope of finding something more rewarding. This choice depends on whether we have sufficient information regarding the consequences of our actions, or whether on the contrary we face inadequate information regarding these consequences. Recent studies show that humans do uncertainty-driven explorations (see e.g., [Frank et al., 2009, Payzan-LeNestour and Bossaerts, 2012]), where the dilemma is solved between exploration and exploitation.

Exploration-exploitation is not unique to the human world, it arises in the digital world as well. For example,

- Recommender systems seek to predict the *preference* of a user for an item [Cremonesi et al., 2010]. A recommender algorithm either recommends an item to a user based on

the previous ratings (exploitation), or it recommends an item to gather information about the user's preference (exploration).

- In clinical trials, the effectiveness of different treatments on animals is assessed during the full range of the stages of the disease [Durand et al., 2018]. The effectiveness of a treatment might not be deterministic. For a specific animal, a treatment is selected either based on the previous responses to the selected treatment (exploitation), or to learn about its effectiveness (exploration).
- In the sequential portfolio selection, the objective is to maximize the cumulative reward by optimizing the allocation of wealth across a set of assets [Shen et al., 2015, Huo and Fu, 2017]. When allocating the resources across the set of assets, two strategies are employed. Either the resources are allocated based on the previously observed rewards of such an allocation (exploitation), or the resources are allocated to learn about their potential reward across a particular set of assets (exploration).
- Online retailers want to dynamically price their products to maximize their revenue [Misra et al., 2019]. The price of an item directly affects the number of purchases, hence the final profit. A retailer might have some data about how pricing affects the revenue. She could directly use this data (exploitation), or she could set a new price and see how much profit she obtains (exploration).
- A telecommunication system has to choose the best wireless link to maximize the quality perceived by the final user [Boldrini et al., 2018]. The quality of a link is a function of network congestion, current load, throughput, etc. The system can choose the link based on the previous quality that the users obtained (exploitation), or it can choose a new link for gathering information about its quality (exploration).

We need a general framework that can model all of the above problems. The unified framework should be able to take into account the uncertainty in the outcome of an action/choice. Now, imagine a gambling machine with K possible choices (or arms), where pulling each arm would result in a random reward from a probability distribution specific to that arm. Both the rewards and the distribution of the rewards are a priori unknown. A gambler wants to try her chance, and her budget allows her to play the game T times. She wants to maximize the sum of the rewards she receives. After each pull, she gathers information about the reward of each arm, and she can essentially use this information to refine her selection strategy. The described gambling game is one instance of a class of sequential decision-making problems called *multi-armed bandits*. It might be one of the simplest examples of sequential decision-making problems under uncertainty, yet it is powerful enough to model many applications in which exploration-exploitation trade-offs arise [Bouneffouf and Rish, 2019] (including those described above).

The purpose of this thesis is to exploit the power of multi-armed bandit settings to address four problems from three different perspectives: (1) how machine-learning (ML)

can benefit from multi-armed bandit algorithms (Chapters 2 and 3), (2) how multi-armed bandit algorithms can affect humans (Chapter 4), and (3) how human interactions can be studied in the MAB framework (Chapter 5). We focus in particular on designing algorithms that have a notion of *versatility* in practice and come with *guarantees* in theory. The theoretical guarantees ensure that the developed algorithms should work in practice, without leaving out any corner case. The versatility of the developed algorithms ensures that the developed algorithms are indeed capable of coping with the needs of the new large data era. Below, we present two important applications that we study with the help of the multi-armed bandit framework and two important extensions of the multi-armed bandit framework that were not studied before. They form the core of the 4 following chapters of this thesis.

1) Stochastic-Gradient Descent A machine-learning algorithm usually extracts patterns from the data in two steps: (1) Data is assumed to follow a certain generative model parameterized by $\theta \in \mathbb{R}^d$. (2) The parameters θ are found such that the model fits the data the best. Gradient descent and its variants form classic and often very effective methods for finding the parameters θ of the model. But, training a model on a large dataset with gradient descent is impractical, because gradient descent uses all of the data at once to update the parameters θ . Stochastic-gradient descent (SGD) reduces the computational complexity of an iteration by sampling a single data point and updating the parameters θ based on only that data point. At first glance, the connection between SGD and multi-armed bandits might not seem clear. But updating the model based on different data points yields various improvements in the model's capability. We study the iterative process of finding the data points that most improve the model in a multi-armed bandit setting.

2) Stochastic-Coordinate Descent Stochastic-coordinate descent (CD) is another algorithm that is developed to address the computational intractability of the gradient descent algorithm. CD selects a single parameter (a.k.a. coordinate) θ_i uniformly at random and updates it. CD has lower computational complexity because it does not require computing the full gradient, with respect to all of the parameters θ . We note that updating different coordinates does not yield the same improvement in the model. For example, if the data is independent (or little dependent) on a parameter θ_i , updating θ_i does not bring any (respectively, little) improvement. Ideally, we should update the coordinate that yields the most improvement in the model's capability, however, this coordinate is a priori unknown. We mold the CD method into a multi-armed bandit setting and choose the coordinates in an informed manner rather than purely at random.

3) Dealing with Polarization in Recommender Systems The purpose of a recommender system is to exploit the knowledge about the previously chosen items of a

user in order to suggest to the user an item that is potentially interesting to her. Because there is always uncertainty about a user’s preference over different items, a recommender system should carefully explore different options that the user might like. Even though research on content selection algorithms has produced a number of well-established methods, they share an imperative limitation: As the content-selection algorithm learns more about a user, the corresponding probability distribution begins to concentrate the mass on a small subset of items; this results in polarization where the feed is primarily composed of a single type of content (see e.g., [Li et al., 2010]). The situation might even escalate, as polarization can create biases that eventually influence decisions and opinions (see e.g., [Pariser, 2011, Epstein and Robertson, 2015]). A premise of this thesis is that these polarizations are unavoidable, but they can be dealt within a principled way using constraints.

4) Beyond a Single Decision Maker Humans’ personal knowledge about the environment and outcome of actions are not limited to only ones’ own experience. One of the main sources of knowledge for a person is the knowledge and experience of the neighbors/peers with whom she interacts. For example,

- Yoo [2012] found that farmers’ decisions are based on (1) their own experience, from previous years, of how different varieties performed, and (2) their peers’ experiences attained either directly (explicitly via conversations with social contacts) or indirectly (implicitly by observing the farming practices of peers).
- Sanditov [2006] studied the spread of knowledge in a social network. In particular, they consider a setting where neighboring nodes in a social network influence each other.
- Zhang et al. [2007] showed how users with high expertise can provoke the adaption of new technologies in industry.

In spite of this widespread knowledge and applications, classic bandit algorithms are still limited to a single-player setting. This hinders a bandit algorithm in combining the extra knowledge of the peers (or other decision makers) into their algorithm. This limitation calls for new bandit-algorithms that can adjust themselves with the new information coming from the neighbors.

1.2 Multi-armed Bandits

In this section, we introduce the multi-armed Bandit (*MAB*) framework that we use throughout this thesis. First, in Section 1.2.1, we define the general setting of a MAB problem. Then in Sections 1.2.2 and 1.2.3, we focus on two settings (adversarial and

stochastic) that are related to the problems solved in this thesis. We present existing algorithms for adversarial MAB and stochastic MAB with their theoretical guarantees. This section is a simple summary of MAB, but it contains pointers for a more thorough description of MAB and its applications.

1.2.1 Framework

MAB is the problem of making a sequence of decisions by a forecaster/player. The goal of the player is to maximize (minimize) the gained *cumulative rewards* (losses). In an MAB problem, there are K arms that a player can choose from. Selecting arm¹ $i \in [K]$ at time t results in a reward (or loss) r_i^t , that may vary among arms and time. After each round t of selection, the player observes the reward only of the selected arm i_t , hence has only access to *partial information*; that it, in turn, uses to refine its arm-selection strategy for the next round.

The nature of the reward processes considered here falls into two categories: (1) *stochastic*, and (2) *adversarial*. We formally define these two settings below.

Definition (Stochastic Multi-armed Bandit). In a stochastic multi-armed bandit, at each time step t , the rewards $\mathbf{r}^t = [r_1^t, \dots, r_K^t]$ are drawn from an unknown distribution \mathcal{D} , i.e., $[r_1^t, \dots, r_K^t] \sim \mathcal{D}$ for all $t \in [T]$. The distribution of the rewards \mathcal{D} is fixed but unknown. Let the expected reward of the arms be

$$[\mu_1, \dots, \mu_K] = \mathbb{E}_{\mathbf{r} \sim \mathcal{D}}[\mathbf{r}]. \quad (1.1)$$

Definition (Adversarial Multi-armed Bandit). In an adversarial multi-armed bandit, at each time step t , an *adversary* chooses the rewards \mathbf{r}^t . An adversary can be *oblivious* or *non-oblivious*. An oblivious adversary sets the sequence of rewards \mathbf{r}^t for $t \in [T]$ independently of the player's strategy. A non-oblivious adversary, at each iteration t , sets the reward \mathbf{r}^t based on the strategy of the player. Consequently, playing against a non-oblivious adversary is harder.

The player's goal is to maximize the cumulative reward

$$\arg \max_{i_1, i_2, \dots, i_T \in [K]} \mathbb{E} \left[\sum_{t=1}^T r_{i_t}^t \right],$$

where the expectation is taken over the randomness of the rewards.

In the MAB problem, the optimal cumulative reward that a player could obtain is not known, because the player does not access all the rewards at each iteration t . Hence, it is common to compare the performance of an algorithm to an ideal strategy. For both

¹ $[K] = [1, 2, \dots, K]$

stochastic and adversarial multi-armed bandits, the ideal strategy is to select the arm that has the maximum expected cumulative reward over the T rounds.

- In a stochastic multi-armed bandit, the distribution of rewards \mathcal{D} is fixed, hence the expected reward of the best arm $\arg \max_{i \in [K]} \mathbb{E} [r_i^t]$ does not change with time t . Therefore, in expectation, the ideal strategy is to select the arm $\arg \max_{i \in [K]} \mathbb{E} [r_i^t]$ with the highest expected reward.
- In an adversarial multi-armed bandit, the rewards \mathbf{r} have no pattern, and it is *impossible* to choose the best arm $\arg \max_{i \in [K]} \mathbb{E} [r_i^t]$ at each iteration t . Therefore, the ideal strategy of selecting an arm i^* that results in the highest cumulative expected reward makes sense. The definition of the ideal strategy in the adversarial setting might be a bit ambiguous because, although the efficacy of an algorithm is measured with respect to the defined strategy, implementing this strategy against a non-oblivious adversary would result in a small reward. Note that a non-oblivious adversary knows the player's strategy. If the player decides to deterministically choose any arm $i \in [K]$ throughout the game, then the adversary can exploit this deterministic strategy and set a small reward for the selected arm i , in all T rounds of the decision-making process.

For both settings, information-theoretic lower bounds show that there is no better strategy than selecting the arm that has the maximum expected cumulative reward [Bubeck and Cesa-Bianchi, 20120].

The efficacy of an algorithm is measured with respect to how well it minimizes *regret* – the difference between the algorithm's reward and the reward obtained from the (unknown) optimal strategy.

Definition (Regret). The regret is defined as

$$\bar{R}^T := \max_{j \in [K]} \mathbb{E} \left[\sum_{t=1}^T r_j^t - \sum_{t=1}^T r_{i_t}^t \right],$$

where the expectation is over the randomness of the rewards and the possible randomness of the algorithm.²

Next, we present some of the well-known algorithms with their theoretical guarantees on regret for both the adversarial and stochastic settings.

1.2.2 Adversarial Multi-armed Bandit

In an adversarial multi-armed bandit, an adversary chooses the sequence of rewards \mathbf{r}^t in all rounds $t \in [T]$. The implication of an adversary choosing the rewards \mathbf{r}^t is that

²Sometimes, \bar{R}^T is referred as pseudo-regret in the literature.

no deterministic algorithm would work in this setting, because the adversary can adapt itself to the algorithm and then it enlarges the regret. The probabilistic algorithms for the adversarial bandit setting can be seen as an extension of the algorithms for the simpler *full information* adversarial setting, where the player observes all rewards at each iteration t .

Many algorithms for the adversarial setting belong to the *multiplicative weight-update* methods. The multiplicative weight-update method has been discovered many times in many fields over the past century (see [Arora et al., 2012] for an overview). It is a simple yet surprisingly powerful way to *conservatively* update beliefs about the benefit of a given arm, it is extremely effective for adversarial settings, and it is asymptotically optimal up to log factors (see, e.g., [Auer et al., 2002b, Flaxman et al., 2005, Alon et al., 2013]).

Full-information Setting

Definition (Full information). In a full-information setting, at each iteration t , the player chooses arm i_t and receives the reward $r_{i_t}^t$. The player observes the rewards of all other arms as well. The problem of making decisions under a full-information setting is also known as *online learning*.

First, let us explain the Hedge algorithm [Freund and Schapire, 1997b] that belongs to the family of multiplicative weight-update methods and was developed for the full-information setting. The algorithms for the full-information setting (where all the rewards are observed at each time step) maintain a vector of weights w_i for each arm i , and (multiplicatively) update it at each time step, by the rule

$$w_i^{t+1} = w_i^t \exp(\delta r_i^t),$$

where δ is the *update parameter*. The probability of choosing arm i at time t is proportional to the weight w_i^t , namely,

$$p_i^t = \frac{w_i^t}{W^t},$$

where $w_i^0 = 1$, and $W^t = \sum_{i=1}^K w_i^t$.

Theorem 1.1 (Theorem 1.5 in [Hazan et al., 2016]). *Let the rewards $r_i^t \in [0, 1]$ for all $i \in [K]$ and $t \in [T]$. Running Hedge (Algorithm 1.1) for T rounds results in a sequence of chosen arms with rewards $r_{i_1}^1, r_{i_2}^2, \dots, r_{i_T}^T$ whose regret is bounded as follows*

$$\bar{R}^T \leq 2\delta T + \frac{\log K}{\delta} \tag{1.2}$$

for any choice of $\delta \in [0, 1]$.

Chapter 1. Introduction

Algorithm 1.1 Hedge Algorithm for full-information setting

- 1: **Input:** δ and T
 - 2: **Initialize:** $w_i^0 = 1$ \triangleright for all $i \in [K]$
 - 3: **for** $t = 1 : T$ **do**
 - 4: Play arm i with probability $p_i^t = \frac{w_i^{t-1}}{\sum_{j=1}^K w_j^{t-1}}$.
 - 5: Observe all rewards $\mathbf{r}^t = [r_1^t, \dots, r_K^t]$.
 - 6: Update $w_i^t = w_i^{t-1} \cdot \exp(\delta r_i^t)$. \triangleright for all $i \in [K]$
 - 7: **end for**
-

This algorithm, for an optimal choice of δ has regret $O(\sqrt{T \ln K})$. The regret bound $\bar{R}^T = O(\sqrt{T \ln K})$ is optimal as it matches the lower bound $\Omega(\sqrt{T \ln K})$ in [Freund and Schapire, 1999]. The full proof of Theorem 1.1 can be found in [Hazan et al., 2016]. A general template for the proofs of multiplicative weight-update method is by upper and lower bounding W^T . Then, putting the lower and upper bounds on W^T together leads to a tight bound on the regret.

Bandit Setting

Given that, in the bandit setting, we no longer observe all of the rewards, the problem becomes harder. Yet, surprisingly, with two simple but important tricks we can extend the Hedge algorithm to the bandit setting.

The first trick is to update the weights w_i^t by using an *unbiased estimator* \hat{r}_i^t for r_i^t , since only r_i^t is known at time t , but not r_j^t with $j \neq i$ (see, e.g., [Auer et al., 2002b, Flaxman et al., 2005]),

$$\hat{r}_i^t = \begin{cases} \frac{r_i^t}{p_i^t} & \text{if arm } i \text{ is chosen at time } t \\ 0 & \text{otherwise.} \end{cases} \quad (1.3)$$

The second trick ensures that some exploration is performed to estimate the rewards of the arms, which are no longer directly observed as before. This is achieved by setting a lower bound $\eta \in [0, 1]$ (the *exploration parameter*) on the probability of selecting arms:

$$p_i^t = (1 - \eta) \frac{w_i^t}{W^t} + \frac{\eta}{K}.$$

The resulting algorithm is called EXP3 (which stands for “Exponential-Weight Algorithm for Exploration and Exploitation”) [Auer et al., 2002b].

Theorem 1.2 (Lemma 6.3 in [Hazan et al., 2016]). *Let the rewards $r_i^t \in [0, 1]$ for all $i \in [K]$ and $t \in [T]$. Running EXP3 (Algorithm 1.2) with $\eta \in [0, 1]$ and $\delta = \eta/K$ for T rounds results in a sequence of chosen arms with rewards $r_{i_1}^1, r_{i_2}^2, \dots, r_{i_T}^T$ whose regret is*

Algorithm 1.2 EXP3 Algorithm for Bandit Setting

- 1: **Input:** η and T
 - 2: **Set:** $\delta = \eta/K$
 - 3: **Initialize:** $w_i^0 = 1$ ▷ for all $i \in [K]$
 - 4: **for** $t = 1 : T$ **do**
 - 5: Play arm i with probability $p_i^t = (1 - \eta) \frac{w_i^t}{W^t} + \frac{\eta}{K}$.
 - 6: Observe the reward $r_{i_t}^t$ of the selected arm i_t .
 - 7: Update $w_i^t = w_i^{t-1} \cdot \exp(\delta \hat{r}_i^t)$. ▷ for all $i \in [K]$
 - 8: **end for**
-

bounded as follows

$$\bar{R}^T \leq 2\eta T + K \frac{\log K}{\eta}. \tag{1.4}$$

EXP3 for an optimal choice of η has regret $O(\sqrt{TK \ln K})$, that is optimal up to log factors.

The regret bound of EXP3 for bandit setting is \sqrt{K} times worse, compared to the regret bound for Hedge. This difference is due to the lack of information that EXP3 has access to, compared to Hedge. The proof follows the same template as the proof of a multiplicative weight-update method, i.e., $W^T = \sum_{i=1}^K w_i^T$ is lower and upper bounded. Then, by putting the lower and upper bounds on W^T together the regret bound is derived. See [Hazan et al., 2016] for further details.

1.2.3 Stochastic Multi-armed Bandit

In a stochastic-bandit setting, the rewards \mathbf{r} are assumed to be drawn from an unknown *fixed* distribution \mathcal{D} . Therefore, the problem has more structure, and we could hope for a smaller regret bound. The assumption that the rewards \mathbf{r} are drawn from a fixed distribution \mathcal{D} enables us to use more tools such as concentration inequalities in our algorithms. To the best of our knowledge, all stochastic-bandit algorithms work by computing and using two metrics: (1) an empirical mean $\bar{\boldsymbol{\mu}}$ of the rewards, and (2) an estimation of the uncertainty around the empirical mean $\bar{\boldsymbol{\mu}}$ by using a concentration inequality. The estimation of the uncertainty is used either directly by the algorithm, or indirectly to set its parameters. In the stochastic setting, if the arms are selected many times, then the uncertainty is small and, with high probability, the arm with the highest empirical mean is the arm with the highest expected reward as well. Note that this is not the case in an adversarial setting, as the adversary can decide to change the pattern of rewards at any time t .

Algorithm 1.3 UCB

```

1: Input:  $\alpha$  and  $T$ 
2: Initialize:  $\bar{\mu}_i^t = 0$  and  $n_i^t = 0$  ▷ for all  $i \in [K]$ 
3: for  $t = 1 : T$  do
4:   Set  $U_i = \bar{\mu}_i^t + \sqrt{\frac{\alpha \ln(t)}{2n_i^t}}$  ▷ If  $n_i^t = 0$  replace  $n_i^t$  with 1
5:   Play arm  $i_t = \arg \max_i U_i$ .
6:   Observe the reward  $r_{i_t}^t$  of the selected arm  $i_t$ .
7:   Update the empirical mean  $\bar{\mu}_{i_t}^t$ .
8:   Update the counts  $n_{i_t}^{t+1} = n_{i_t}^t + 1$  and  $n_i^{t+1} = n_i^t$  for  $i \in [K] \setminus i_t$ .
9: end for

```

We describe below two of the most well-known algorithms used in the stochastic-bandit setting. The first algorithm, UCB, is a deterministic algorithm that uses the estimated uncertainty directly. The second algorithm, ε -GREEDY, is a probabilistic algorithm that uses the estimated uncertainty indirectly.

Upper Confidence Bound (UCB)

The UCB algorithm is an asymptotically optimal algorithm, which was first introduced by Auer et al. [2002a] and has been widely extended and studied (see, e.g., [Bubeck, 2010, Maillard et al., 2011, Garivier and Cappé, 2011]). The main idea behind the algorithm is the principle of *optimism in the face of uncertainty*; the algorithm maintains an optimistic *upper bound* on the mean reward of each arm, and selects the arm with the maximal upper bound. The expected reward of arm j (μ_j) at time t has an upper bound

$$\mu_j \leq U_j(t) \stackrel{\text{def}}{=} \bar{\mu}_j^t + \sqrt{\frac{\alpha \ln(t)}{2n_j^t}} \tag{1.5}$$

that holds with probability at least $1 - t^{-\alpha}$, where $\alpha > 2$ is a constant that depends on the variance of the rewards distribution \mathcal{D} , n_j^t is the number of samples we have for arm j by time t , and $\bar{\mu}_j^t$ is the sample mean of arm j over n_j^t samples.

At time t , UCB selects the arm $i_t = \arg \max_j \{U_j(t)\}$. After an initial period in which UCB collects information about the rewards, UCB selects the arm with the highest expected reward. The main idea behind UCB is that even if at a certain time UCB selects a non-optimal arm, it would correct itself later and eventually distinguish the non-optimal arms. Let us consider the situation in which UCB mistakenly considers a non-optimal arm j as the optimal arm. This happens when $U_j(t) > U_{i^*}(t) \geq \mu_{i^*}$, where $i^* = \arg \max_i \{\mu_i\}$ is the index of the arm with the highest expected reward. But as UCB selects the arm j increasingly often, the upper bound $U_j(t)$ is refined and decreases with the rate $\sqrt{1/n_j^t}$ because of the term $\sqrt{\alpha \ln(t)/2n_j^t}$. The upper bound $U_j(t)$ gets therefore closer to the true

mean μ_j in (1.1). Therefore, eventually at some iteration t , $U_j(t) \leq U_{i^*}(t)$ and UCB ranks the arm j below the optimal arm i^* .

Theorem 1.3 (Theorem 2.1 in [Bubeck and Cesa-Bianchi, 20120]). *Let the rewards $\mathbf{r}^t \in [0, 1]$ be random with a fixed unknown distribution \mathcal{D} . Running UCB (Algorithm 1.3) with $\alpha > 2$ for T rounds results in a sequence of chosen arms with rewards $r_{i_1}^1, r_{i_2}^2, \dots, r_{i_T}^T$ whose regret is bounded as follows*

$$\bar{R}^T \leq \frac{2\alpha K \ln T}{\Delta} + \frac{\alpha}{\alpha - 2}, \tag{1.6}$$

where $\Delta = \mu_{i^*} - \max_{j \neq i^*} \mu_j$ is the difference between the highest and the second highest expected reward.

The regret of UCB grows with $O(\log T)$, whereas the regret of EXP3 grows with $O(\sqrt{T})$. This means UCB learns at a much faster pace in a stochastic setting, compared to how EXP3 learns in an adversarial setting. Note that running UCB in an adversarial setting would result in a linear regret $\Omega(T)$, but the convergence guarantee of EXP3 ($\bar{R}^T = O(\sqrt{TK \log K})$) holds for any setting, including a stochastic one.

The convergence guarantee of UCB is proven by showing that when a sub-optimal arm i is selected $O(\log T)$ times, then with high probability the upper bound of the optimal arm i^* is larger than the ones of the sub-optimal arms i (i.e., $U_{i^*} \geq U_i$). See [Bubeck and Cesa-Bianchi, 20120] for further details.

ε -Greedy

As the name suggests ε -GREEDY, is an (almost) greedy algorithm, and it is perhaps one of the simplest algorithms for trading off exploration and exploitation (see [Sutton et al., 1998]). At each time step t , with probability $(1 - \varepsilon)$, ε -GREEDY selects the arm with the highest empirical mean $\bar{\mu}_i^t$ (exploitation); and in order to avoid converging to a sub-optimal strategy with probability ε , ε -GREEDY selects an arm uniformly at random (exploration).

The exploration parameter $\varepsilon \in [0, 1]$ controls the level of exploration and needs to be determined carefully. For example, for a constant exploration parameter ε , ε -GREEDY would have a linear regret $\bar{R} = O(T)$; also, a small exploration parameter ε can misguide the algorithm to choosing a sub-optimal strategy. Auer et al. [2002a] found that by setting $\varepsilon_t \sim 1/t$, ε -GREEDY has regret $O(K \log T / \Delta^2)$.

Theorem 1.4 (Theorem 3 in [Auer et al., 2002a]). *Let the rewards \mathbf{r}^t be random with a fixed unknown distribution \mathcal{D} . Running ε -GREEDY (Algorithm 1.4) for T rounds results in a sequence of chosen arms with rewards $r_{i_1}^1, r_{i_2}^2, \dots, r_{i_T}^T$ whose regret is bounded as*

Algorithm 1.4 ε -GREEDY

1: **Input:** $\gamma < \Delta$ and T
2: **Initialize:** $\bar{\mu}_i^t = 0$ \triangleright for all $i \in [K]$
3: **for** $t = 1 : T$ **do**
4: Set $\varepsilon_t = \min \left\{ 1, \frac{K}{\gamma^2 t} \right\}$.
5: Let $j = \arg \max_i \bar{\mu}_i^t$.
6: Play arm j with probability $1 - \varepsilon_t$ and with probability ε_t play a random arm.
7: Observe the reward $r_{i_t}^t$ of the selected arm i_t .
8: Update the empirical mean $\bar{\mu}_{i_t}^t$.
9: **end for**

follows

$$\bar{R}^T = O\left(\frac{K \ln T}{\Delta^2}\right), \tag{1.7}$$

where $\Delta = \mu_{i^*} - \max_{j \neq i^*} \mu_j$ is the difference between the highest and the second highest expected reward.

Although UCB has a better regret bound than ε -GREEDY, in practice ε -GREEDY usually outperforms UCB (see e.g., [Kuleshov and Precup, 2014]). Note that ε -GREEDY does not use the uncertainty estimate directly in the algorithm as UCB does, but instead sets the exploration rate ε based on the estimation of uncertainty. The exploration rate ε in ε -GREEDY is set such that all arms are at least $O(\log t)$ times selected up to time t . The convergence guarantee of ε -GREEDY is proven in [Auer et al., 2002a], by showing that when the arms are chosen $O(\log t)$ times, the arm with the highest empirical mean is the optimal arm with high probability.

1.3 Outline and Contributions

The MAB framework provides many strong tools for optimizing a sequence of decisions, when there is uncertainty about their outcomes. In this thesis, we use the MAB framework to handle several challenges that arise in modern optimization algorithms and their applications. The problems studied here share a common feature: they all make a sequence of choices, on the basis of limited information.

We will pay a particular attention to the scalability of the algorithms and their theoretical guarantees of convergence. Indeed, as datasets grow, it is more important to develop *computationally* efficient algorithms that achieve their best performance at a faster rate. Moreover, the theoretical guarantees of the algorithms developed in this thesis assure their performance under different conditions and different datasets.

In Chapter 2, we focus on accelerating the *stochastic-gradient descent* (SGD) method by using the MAB framework. SGD is one of the most well-known optimizers of an empirical risk function, and improving its performance is of great interest. The SGD method addresses the computational complexity obstacle of the gradient descent method by computing and using only the gradient of one of the data points. We begin by computing the variance of the estimator of the gradient and by noticing that sampling the data points from a non-uniform distribution can significantly reduce the variance of the estimator. The optimal sampling distribution depends on the gradient of each data point. As we do not want to compute the full gradient of the cost function, we cannot compute this distribution. Instead, we cast the variance-minimization problem as an MAB problem. As the norm of the gradients change over time and as these changes do not follow any specific pattern, we study the variance-minimization MAB problem in an adversarial setting. We develop a bandit algorithm to learn this distribution while optimizing the cost function. The proposed algorithm is called *Multi-armed Bandit Sampling* (MABS). MABS has a sublinear computational complexity that makes it applicable for large models. We prove that MABS can approximate the optimal distribution with regret $O(\sqrt{T})$ (Theorem 3.6 and Corollary 2.9 in Section 2.4.1). Next, we provide convergence guarantees of SGD and projected SGD (PSGD) when they are combined with MABS (Theorems 2.1 and 2.14 in Section 2.5) and we show MABS’s effectiveness in improving the convergence rate of SGD and PSGD. We also extensively evaluate the performance of MABS in conjunction with SGD, SVRG, and SAGA on both synthetic and real-world data (Section 2.6), and we verify its effectiveness in practice.

In Chapter 3, we shift our attention to the *stochastic-coordinate descent* (CD) method; it is another optimizer for an empirical risk function. CD addresses the computational complexity obstacle of the gradient descent method by optimizing the cost function along one of the decision variables (coordinates) at a time, rather than optimizing all of them at once. The CD method usually chooses a coordinate for the update, uniformly at random. However, different coordinates contribute differently to the prediction variable or output. Ideally, we would update the decision variable that contributes the most to the output. But finding this decision variable requires checking all of them, which effectively negates the improvement in computational tractability that CD is intended to afford. In this chapter, we develop an effective coordinate-selection method. First, we show the great enhancement in the convergence rate of CD with an optimal non-uniform coordinate-selection method (Theorems 3.5 and 3.6 in Section 3.4.2). The optimal non-uniform coordinate-selection method is found by a greedy procedure and is not computationally plausible, we call the greedy algorithm `max_r`. We then exploit the bandit setting in order to design a lightweight coordinate-selection mechanism, that maintains the computational tractability of CD. While in Chapter 2, the gradients of different data points vary over time and depend on each other, we notice that updating one coordinate has little impact on the improvement that CD gets when updating other coordinates, therefore, we study the coordinate-selection problem in a stochastic MAB setting and, inspired by ϵ -GREEDY,

we propose a bandit algorithm called B_max_r . We prove the effectiveness of B_max_r and we show that B_max_r can perform almost as well as max_r , yet it decreases the number of calculations required significantly (Proposition 3.8 in Section 3.4.3). Finally, we test B_max_r and max_r in practice and show their advantage over the state-of-the-art CD methods (Section 3.5).

In Chapter 4, we focus on the problem of polarization and biases in online personalized platforms such as online ad-services. Personalization is a must for online platforms, but it is also the reason for polarization. Polarization has been observed on many social media platforms (see, e.g., [Hong and Kim, 2016, Conover et al., 2011, Weber et al., 2013]), and new studies have shown that, over the past eight years, polarization has increased constantly [Garimella and Weber, 2017]. A feature of bandit algorithms is that the entire probability mass ends up on a single arm (hence in a single group) – causing polarization (see e.g., [Li et al., 2010]). For example, consider a news recommendation system. The bandit algorithms used for recommendation keep a probability distribution over a set of news, as the algorithm learns more about a user, the corresponding probability distribution begins to concentrate the mass on a small subset of topics; this results in polarization where the feed is primarily composed of a single type of content. To address polarization, we introduce constraints on the probability distribution that the bandit algorithm keeps. These constraints limit the total expected weights that can be allocated to a group of items. Despite the simplicity of the constraints, they are versatile enough to control polarization, with respect to a variety of metrics that can measure the extent of polarization in a given algorithm. Inspired by ϵ -GREEDY, we propose a new algorithm called $CONS\text{-}\epsilon\text{-GREEDY}$: it respects the constraints at each iteration. We show that $CONS\text{-}\epsilon\text{-GREEDY}$ learns the optimal constrained mechanism with the sub-linear regret $O(\log T)$ (Theorem 4.1 in Section 4.4). We evaluate $CONS\text{-}\epsilon\text{-GREEDY}$ on a curated dataset of online news articles, demonstrate that it can control polarization, and examine the trade-off between decreasing polarization and the resulting loss in revenue (Section 4.5).

Lastly, in Chapter 5, we study the bandit setting extended to a network. In this setting, multiple decision makers are connected in a network. The work of [Yoo, 2012] gives an example of network setting with which it is shown that farmers use the information from their neighbors in order to make the best decisions. Similar social learning phenomena appear in many other areas; see [Sanditov, 2006, Zhang et al., 2007, Accinelli and Sánchez-Carrera, 2012]. Although this setting is natural in the human world, it was not yet studied from an algorithmic point of view. We study the problem in both stochastic and adversarial settings. In the stochastic setting, the problem is easier: simply incorporating side information to a stochastic-bandit algorithm, such as UCB, is enough to have a near-optimal algorithm with the regret $O(\log T)$ (Theorem 5.1 in Section 5.4). In the adversarial setting, the problem is more challenging, and a simple modification of the existing algorithms does not work. Here, we use a new unbiased estimator of the rewards of the arms, and using the amount of the exploration of the neighbors we set the learning parameters of the algorithm. We show that the proposed algorithm (called

1.3. Outline and Contributions

EXPN) can exploit the side information efficiently: when the neighbors explore enough, EXPN can perform similarly to the Hedge algorithm in the full information setting, where the regret is $O(\sqrt{T})$ (Theorem 5.2 in Section 5.5). Finally, we experiment the developed algorithms on different network topologies and confirm their effectiveness, compared to several baselines (Section 5.6).

2 Stochastic Gradient Descent with Bandit Sampling

In this chapter¹, we study the problem of accelerating stochastic optimization algorithms with an adaptive sampling method. Many stochastic optimization algorithms work by estimating, on the fly, the gradient of the cost function by sampling datapoints uniformly at random from a training set. However, the estimator might have a large variance, which could slow down the convergence rate of the algorithm. One way to reduce this variance is to sample the datapoints from a carefully selected non-uniform distribution.

We study the datapoint-selection of stochastic optimization algorithms as a decision-making problem in the adversarial bandit setting (see Section 1.2.2). We develop bandit algorithms for datapoint-selection and we show that our algorithm asymptotically approximates the minimal variance within a constant factor. We propose several such approaches with different performance and running time. Empirically, we show that using this datapoint-selection technique results in a significant reduction of the convergence time and of the variance of several stochastic optimization algorithms such as SGD and SAGA. This approach for sampling datapoints is general and can be used in conjunction with *any* algorithm that uses an unbiased gradient estimation – we expect it to have a broad applicability beyond the specific examples explored in this chapter.

2.1 Introduction

Consider the following optimization problem, known as empirical risk minimization, which is ubiquitous in machine learning:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} F(\boldsymbol{\theta}) := \frac{1}{n} \sum_{i=1}^n \phi_i(\boldsymbol{\theta}), \quad (2.1)$$

¹This chapter is based on [Salehi et al., 2017a].

Chapter 2. Stochastic Gradient Descent with Bandit Sampling

where the coordinates $\boldsymbol{\theta} \in \mathbb{R}^d$ are the learning parameters. The empirical risk function $F(\boldsymbol{\theta})$ is the mean of n convex functions $\phi_i(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}$, which we call *sub-cost functions*. The i^{th} sub-cost function $\phi_i(\cdot)$ is parameterized by the i^{th} *datapoint* (\mathbf{x}_i, y_i) , where $\mathbf{x}_i \in \mathbb{R}^d$ denotes its feature vector and $y_i \in \mathbb{R}$ its label. Examples of common sub-cost functions include

- Logistic regression: $\phi_i(\boldsymbol{\theta}) = \log(1 + \exp(-y_i \langle \mathbf{x}_i, \boldsymbol{\theta} \rangle))$,
- SVM: $\phi_i(\boldsymbol{\theta}) = ([1 - y_i \langle \mathbf{x}_i, \boldsymbol{\theta} \rangle]_+)^2$ (where $[\cdot]_+ = \max\{0, \cdot\}$ is the hinge loss), and
- Linear regression: $\phi_i(\boldsymbol{\theta}) = \frac{1}{2} (\langle \mathbf{x}_i, \boldsymbol{\theta} \rangle - y_i)^2$.

Gradient descent and its variants form classic and often very effective methods for solving (2.1). However, when $F(\boldsymbol{\theta})$ is minimized using gradient descent, the value $\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^t)$ is computed at each iteration t for all $i \in [n]$ (there are n gradient calculations) which, for large n , can be prohibitively expensive (see, e.g., [Bottou, 2010]). Stochastic gradient descent (SGD) reduces the computational complexity of an iteration by sampling a datapoint $i_t \in [n]$ uniformly at random at each iteration t and by computing the gradient only at this datapoint; $\nabla_{\boldsymbol{\theta}} \phi_{i_t}(\boldsymbol{\theta}^t)$ is then an unbiased estimator for $\nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta}^t)$. However, this estimator might have a large variance, which negatively affects the convergence rate of the underlying optimization algorithm and requires an increased number of iterations. In stochastic optimization algorithms such as SGD and proximal SGD (PSGD), reducing this variance improves the speed of convergence to the optimal coordinate $\boldsymbol{\theta}^*$ (see, e.g., [Xiao and Zhang, 2014] and also Section 2.5).

This has motivated the development of several techniques to reduce this variance. One such technique, closely related to this work, is to sample a datapoint i_t from a non-uniform distribution $\mathbf{p} = \{p_1, \dots, p_n\}$ (see, e.g., [Needell et al., 2014, Zhao and Zhang, 2015a]), where the sampling distribution \mathbf{p} is chosen in order to minimize an upper bound of the variance of the estimator for $\nabla_{\boldsymbol{\theta}} F$ in (2.1). The upper bound is set a priori, independently from t and from $\boldsymbol{\theta}$. Therefore, the sampling distribution \mathbf{p} is also not a function of t , nor of the coordinates $\boldsymbol{\theta}$. The problem with this approach is that the gap between the minimum variance and the upper bound of this variance attained under these assumptions is, in general, unknown. Instead, if the non-uniform distribution $\mathbf{p}^{t*} = \{p_1^{t*}, \dots, p_n^{t*}\}$ that minimizes the variance (and is a function of t and $\boldsymbol{\theta}$) is available, the stochastic optimization algorithm converges much faster. In SGD, the ideal probability p_i^{t*} of sampling the datapoint i at time t is proportional to $\|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^t)\|$. If the $\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^t)$ s have similar magnitudes for all $i \in [n]$, then the optimal distribution is close to the uniform distribution. However, if the magnitude of $\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^t)$ at some datapoint i is comparatively very large, then the optimal distribution is far from uniform; in this case, the variance can be made roughly n times smaller than the variance when the uniform distribution is used. The challenge lies in finding the appropriate non-uniform sampling distribution with a lightweight mechanism that preserves the computational tractability of SGD.

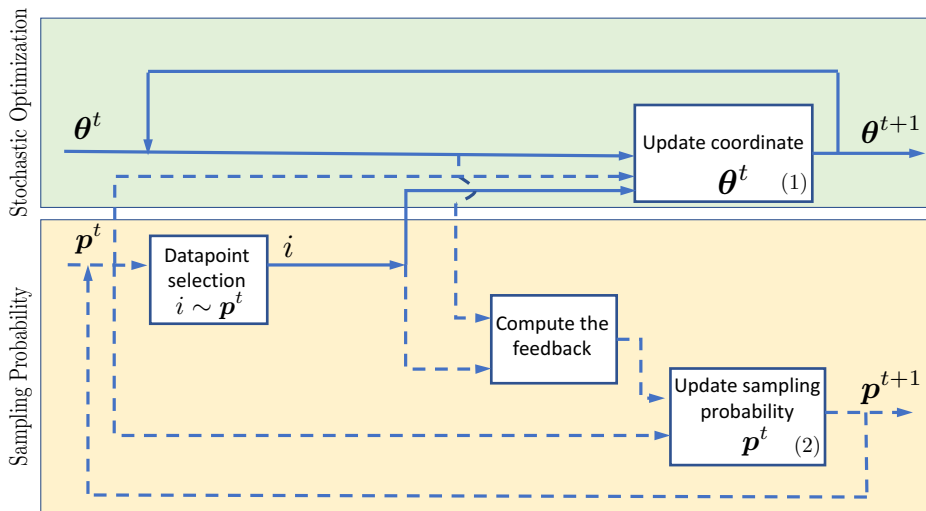


Figure 2.1 – Our approach to stochastic optimization with bandit sampling. The green (top) part of the mechanism updates the coordinates θ^t (using the selected datapoint i and the distribution \mathbf{p}^t). One can use SGD, SVRG, SAGA, or any other stochastic optimization method to do so in box (1) as long as they use an unbiased estimator for the gradient of $f(\theta)$. The yellow (bottom) part of the mechanism handles the selection of $i \in [n]$ according to a sampling distribution \mathbf{p}^t which is updated via bandit optimization in box (2) from feedback. For example, in SGD, the coordinates are updated as $\theta^{t+1} = \theta^t - \gamma \frac{\nabla_{\theta} \phi_i(\theta^t)}{np_i^t}$ and feedback is simply the norm of the gradient of the selected datapoint i , i.e., $\|\nabla_{\theta} \phi_i(\theta^t)\|$.

Our Contributions

In this work, inspired by active learning methods, we use an adaptive approach to define a probability distribution $\mathbf{p}^t = \{p_1^t, \dots, p_n^t\}$ over the datapoints, to sample a datapoint $i \in [n]$, instead of fixing it in advance (see the *Sampling Probability* part in Figure 2.1). If the set of datapoints selected during the first ℓ iterations is $\{i_t\}_{1 \leq t \leq \ell}$, then we refer to the corresponding gradients $\{\nabla_{\theta} \phi_{i_t}(\theta^t)\}_{1 \leq t \leq \ell}$ as *feedback*, which we use to refine $\mathbf{p}^{\ell+1}$. The problem of how to best define the distribution $\mathbf{p}^{\ell+1}$, given feedback, falls under the framework of multi-armed bandit problems. We call our approach *multi-armed bandit sampling* (MABS) and we show that it finds a distribution that is asymptotically close to the optimal. MABS can be used in conjunction with any algorithm that uses an unbiased gradient estimator to reduce the variance of the estimator for the gradient, not just SGD (see *Stochastic Optimization* part in Figure 2.1). This includes SAGA [Defazio et al., 2014], SVRG [Xiao and Zhang, 2014], Prox_SGD [Zhao and Zhang, 2015a], and S2GD [Konečný and Richtárik, 2017]. We present the empirical performance of some of these methods in Section 2.6, see Figures 2.2, 2.3 and 2.4.

In summary, our main contributions are as follows.

Multi-armed Bandit Sampling First, we show that the convergence guarantee of SGD linearly depends on the cumulative sum of variances over T rounds. Hence, minimizing this variance results in a faster convergence rate. We then compute the variance of the unbiased estimator for $\nabla_{\theta} F(\theta^t)$ as a function of the sampling distribution \mathbf{p}^t and of the magnitude of gradient $\nabla_{\theta} \phi_i(\theta^t)$. We recast the problem of minimizing the variance of stochastic optimization over \mathbf{p}^t as a multi-armed bandit problem (Section 2.2). We provide fast (sub-linear computational complexity) sampling algorithms (MABS) to minimize the variance. MABS is inspired by EXP3 [Auer et al., 2002b] and Hedge [Freund and Schapire, 1997a]. We show that MABS approximates the sum of variances over T rounds attained by the optimal distribution within a constant factor (see Theorem 2.6 and Corollary 2.9 in Section 2.4.1). As we explain next, approximating this optimal distribution is both necessary and sufficient (in our setting) when using SGD.

SGD in Conjunction with MABS The optimal sampling distribution depends on the trajectories of the coordinates $\theta^1, \theta^2, \dots, \theta^t$, while the sequence $\theta^1, \theta^2, \dots, \theta^t$ itself depends on the sampling distribution \mathbf{p} , therefore an optimal distribution for a sequence of coordinates $\theta^1, \theta^2, \dots, \theta^t$ might not reflect the minimum achievable cumulative variance. To address this problem, we use a biased sampling distribution as in [Needell et al., 2014]. The biased distribution has a non-zero weight for sampling each datapoint and ensures the convergence of SGD to the optimal coordinates θ^* , which is the minimizer of (2.1). The result of such a restriction is that no sampling algorithm can reach a sub-linear regret (i.e., approximate the optimal distribution with factor 1). Yet, we show that this is not an issue, as we prove that when θ^t converges to θ^* , the approximation factor only appears as a constant there. Interestingly, the term that dominates the minimum cumulative variance is the variance of the unbiased gradient estimator at the optimal coordinates θ^* . We provide the convergence guarantee of SGD when it is combined with MABS in Theorem 2.10, in Section 2.5, which we informally state below.

Theorem (Informal Statement of Theorem 2.10). *Assume that $F(\theta)$ is strongly convex and that each ϕ_i is convex and smooth. Then, with an appropriate learning rate, the convergence rate of SGD with MABS satisfies*

$$\mathbb{E} [\|\theta^{T+1} - \theta^*\|^2] = O \left(\frac{1}{Tn^2} \left(\sum_{i=1}^n \|\nabla_{\theta} \phi_i(\theta^*)\| \right)^2 \right).$$

Evaluation We extensively evaluate the performance of MABS in conjunction with SGD, SVRG, and SAGA on both synthetic and real-world data. More precisely, using a synthetic dataset, we vary the ratio τ between the average smoothness and the maximum smoothness of ϕ_i s, because it has been observed in many works (see, e.g., [Zhao and Zhang, 2015b]) that non-uniform sampling method helps more if τ is large. We observe that across different values of τ , SGD with MABS converges faster than other sampling

methods, and that the improvement is more significant for large τ (Section 2.6.2). Using real datasets, we test MABS with three different stochastic optimization algorithms and also observe significant improvements in practice, compared to other sampling methods (Section 2.6.3). We empirically show that using MABS makes a stochastic optimization algorithm more robust and that the stochastic optimization algorithm can use larger step sizes without diverging (Section 2.6.4). Finally, we evaluate the convergence rate of SGD, SVRG, and SAGA with different sampling methods as a function of wall-clock time, and we observe again that MABS improves the convergence rate (Section 2.6.5).

2.2 Preliminaries

First order stochastic optimization algorithms (such as SGD, SVRG, and SAGA) require an unbiased estimator for the gradient $\nabla_{\theta} f(\theta)$. The end goal of MABS is to find the sampling distribution \mathbf{p}^{t*} that keeps the estimator unbiased and minimizes its variance. The first step is therefore to find this variance as a function of the sampling distribution \mathbf{p}^t at timestep t , conditionally on θ^t .

Let us start with the example of SGD. In SGD with non-uniform sampling, the update rule is

$$\theta^{t+1} = \theta^t - \gamma_t \hat{g}(\theta^t, \mathbf{p}^t), \quad (2.2)$$

where γ_t is the step size and $\hat{g}(\theta^t, \mathbf{p}^t)$ is the unbiased estimator for $\nabla_{\theta} F(\theta^t)$ defined by

$$\hat{g}(\theta^t, \mathbf{p}^t) = \frac{\nabla_{\theta} \phi_{i_t}(\theta^t)}{np_{i_t}^t}, \quad (2.3)$$

where i_t is the sampled datapoint at timestep t . Taking expectations over the sampling distribution \mathbf{p}^t , conditionally² on θ^t , the *variance*³ of $\hat{g}(\theta^t, \mathbf{p}^t)$ can be written as

$$\mathbb{V}(\theta^t, \mathbf{p}^t) \triangleq \mathbb{E}_{\mathbf{p}^t} \left[\left\| \hat{g}(\theta^t, \mathbf{p}^t) - \nabla_{\theta} F(\theta^t) \right\|^2 \middle| \theta^t \right] = \mathbb{V}_e(\theta^t, \mathbf{p}^t) - \mathbb{V}_c(\theta^t), \quad (2.4)$$

where

$$\mathbb{V}_e(\theta^t, \mathbf{p}^t) = \mathbb{E}_{\mathbf{p}^t} \left[\left\| \hat{g}(\theta^t, \mathbf{p}^t) \right\|^2 \middle| \theta^t \right] = \frac{1}{n^2} \sum_{i=1}^n \frac{1}{p_i^t} \left\| \nabla_{\theta} \phi_i(\theta^t) \right\|^2 \quad (2.5)$$

²More precisely, conditionally on the filtration \mathcal{F}^{t-1} of all datapoints selected up to timestep $t-1$.

³Note that $\hat{g}(\theta^t, \mathbf{p}^t)$ is a d -dimensional random vector, with $d > 1$ in general, hence strictly speaking (2.4) is the sum of the variances of its d entries.

is referred to as the effective variance, and depends explicitly on \mathbf{p}^t contrary to

$$\mathbb{V}_c(\boldsymbol{\theta}^t) = \frac{1}{n^2} \left\| \sum_{i=1}^n \nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^t) \right\|^2.$$

As the only term under our control is \mathbf{p}^t , to minimize the variance (2.4), it suffices to minimize the effective variance $\mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p}^t)$. For SGD, this minimum is attained when

$$\mathbf{p}_i^{t*} = \frac{\|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^t)\|}{\sum_{j=1}^n \|\nabla_{\boldsymbol{\theta}} \phi_j(\boldsymbol{\theta}^t)\|}. \quad (2.6)$$

To show that minimizing the effective variance $\mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p}^t)$ improves the convergence rate of SGD, we state a simple extension of a theorem by Lacoste-Julien et al. [2012], that shows that the convergence rate of SGD is proportional to the expected value of the cumulative effective variance $\sum_{t=1}^T \mathbb{E}_{\boldsymbol{\theta}^t} [\mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p}^t)]$.

Theorem 2.1. *Assume that $F(\boldsymbol{\theta})$ is μ -strongly convex. Then, if $\gamma_t = 2/\mu(1+t)$ in (2.2), the following convergence guarantee holds for $T > 1$ in SGD:*

$$\mathbb{E} \left[\|\boldsymbol{\theta}^{T+1} - \boldsymbol{\theta}^*\|^2 \right] = O \left(\frac{1}{\mu^2 T^2} \sum_{t=1}^T \mathbb{E}_{\boldsymbol{\theta}^t} [\mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p}^t)] \right), \quad (2.7)$$

where the expectation of the left-hand side of (2.7) is taken over all $\boldsymbol{\theta}^1, \dots, \boldsymbol{\theta}^T$.

The convergence guarantee in (2.7) shows that to improve the convergence rate of SGD we can minimize the sum of the effective variances $\sum_{t=1}^T \mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p}^t)$ over \mathbf{p}^t . Theorem 2.1 follows directly from [Lacoste-Julien et al., 2012], but for the sake of completeness, we present the proof of Theorem 2.1 in Appendix 2.A.1.

2.3 Related Work

Non-uniform datapoint selection has been proposed first for constant (non-adaptive) probability distribution $\mathbf{p} = [p_1, \dots, p_n]$ (see, e.g., [Zhao and Zhang, 2015a, Needell et al., 2014, Kern and György, 2016, Zhao and Zhang, 2015b, 2014, Zhang et al., 2017, Schmidt et al., 2015a, Csiba and Richtárik, 2016]). To compute these probability distributions, a precomputation is required before starting the update of coordinates $\boldsymbol{\theta}$, for example in [Zhao and Zhang, 2015a,b] the smoothness of each ϕ_i is computed. Then, using these precomputed parameters a fixed distribution is found. For example, in [Zhao and Zhang, 2015a,b], an upper bound on the effective variance $\mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p}^t)$ is derived by upper bounding $\|\nabla_{\boldsymbol{\theta}} \phi_{i_t}(\boldsymbol{\theta}^t)\|^2$ in (2.5) as:

$$\|\nabla_{\boldsymbol{\theta}} \phi_{i_t}(\boldsymbol{\theta}^t)\|^2 \leq \sup\{\|\nabla_{\boldsymbol{\theta}} \phi_{i_t}(\boldsymbol{\theta}^t)\|^2\} = a_i.$$

Then, the effective variance is simply upper bounded as:

$$\mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p}^t) = \frac{1}{n^2} \sum_{i=1}^n \frac{1}{p_i^t} \|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^t)\|^2 \leq \frac{1}{n^2} \sum_{i=1}^n \frac{a_i}{p_i^t}. \quad (2.8)$$

If a_i s are known and fixed for all $i \in [n]$ and $1 \leq t \leq T$, the optimal sampling distribution \mathbf{p}^t at which the upper-bound in (2.8) attains its minimum is time invariant and known. This method is known as *importance sampling* (IS). A drawback of this method is that the upper-bound on (2.8) might be loose, when $\|\nabla_{\boldsymbol{\theta}} \phi_{i_t}(\boldsymbol{\theta}^t)\|^2$ is much smaller than a_i , hence the sampling distribution found by minimizing the upper-bound in (2.8) is far from the optimal sampling distribution that minimizes the variance. Unlike our approach, there is no theoretical guarantee on the distance between the importance sampling distribution and the optimal sampling distribution. Shen et al. [2016] and Papa et al. [2015] developed adaptive sampling methods that directly compute $\|\nabla_{\boldsymbol{\theta}} \phi_{i_t}(\boldsymbol{\theta}^t)\|^2$. For example, in [Shen et al., 2016] the whole gradient is computed every few epochs. If the number of datapoints is not large, then the method works fine but if the number of datapoints is large then the algorithm needs to wait considerably until the next update, which could affect its performance. Schmidt et al. [2015b] also developed an adaptive method for the stochastic average gradient algorithm (SAG) that uses a biased gradient. In [Schmidt et al., 2015b], the smoothness ratio is estimated in an online manner.

In contrast, we use a simple and efficient learning procedure (that needs only $O(\log n)$ computations per iteration) to learn the probability distribution \mathbf{p}^t that fits the data best. We transform the problem of finding the optimal probability distribution as a multi-armed bandit problem, where the observed gradient $\nabla_{\boldsymbol{\theta}} \phi_{i_t}(\boldsymbol{\theta}^t)$ is used as feedback to update the probability distribution \mathbf{p}^t . Therefore, instead of minimizing the upper bound on the effective variance $\mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p}^t)$, the actual effective variance is directly minimized by an online approach. Moreover, our sampling approach does not depend on any specific type of optimization algorithm and can be used in conjunction with any stochastic optimization algorithm, as long as they use an unbiased estimator for the gradient.

Several other techniques were also developed to reduce the variance of the estimator for the gradient used in SGD: They use previous information to refine the estimation for the gradient; e.g., by occasionally calculating and using the full gradient to refine the estimation [Xiao and Zhang, 2014, Allen-Zhu and Yuan, 2016], or by using the previous calculations of $\nabla_{\boldsymbol{\theta}} \phi_i$ (at the most recent selection of each datapoint i) [Defazio et al., 2014]. For example in SAGA (see [Defazio et al., 2014]), the following estimation for the full gradient is maintained

$$\tilde{g} = \frac{1}{n} \sum_{i=1}^n \nabla_{\boldsymbol{\theta}} \phi_i(\hat{\boldsymbol{\theta}}_i), \quad (2.9)$$

where $\hat{\theta}_i$ is the coordinate at the most recent time that datapoint i was chosen. The estimation \tilde{g} has shown to have good correlation with the true gradient g . At time t , the estimation \tilde{g} and the gradient of selected sub-cost function ϕ_{i_t} are used to build an unbiased estimator for the gradient, which is

$$\hat{g}(\theta^t) = \left(\nabla_{\theta} \phi_{i_t}(\theta^t) - \nabla_{\theta} \phi_{i_t}(\tilde{\theta}_{i_t}) \right) + \tilde{g}.$$

As the coordinate θ^t approaches the optimal coordinate θ^* , the variance of the estimator $\hat{g}(\theta^t)$ decreases.

Independently of the work presented in this chapter, [Namkoong et al., 2017] developed a similar approach by formulating the variance reduction for the problems of SGD and coordinate descent as a bandit problem. The difference with our work is that theoretical guarantees in our work are given with respect to the best achievable cumulative variance, whereas the theoretical guarantee in [Namkoong et al., 2017] is limited to the best distribution in hindsight for a random sequence of coordinates $\theta^1, \dots, \theta^T$ in a neighborhood of the uniform distribution. The proof of the regret guarantee in [Namkoong et al., 2017] is based on Theorem 5.3 of [Bubeck and Cesa-Bianchi, 20120], which is a regret analysis for online mirror descent. In contrast, the proof of the regret guarantee used in this work is similar to the standard proofs used in multiplicative-weight update algorithms (see for example [Auer et al., 2002b]). Following [Namkoong et al., 2017] and our work, [Borsos et al., 2018] use a similar bandit formulation of the variance reduction problem, and use an algorithm from the family of Follow-the-Regularized-Leader algorithms to minimize the variance of SGD. In [Borsos et al., 2018], the approximation factor of the bandit algorithm is 1 compared to $1 \leq c \leq 3$ in this work; and the regret scales as $O(T^{2/3})$, compared to $O(T^{1/2})$ in this work. Our smaller regret bound comes therefore at the cost of having a larger approximation factor. The smaller regret bound means that the online algorithm learns faster, hence we expect our algorithm to work better in a setting where the gradients vary rapidly from an iteration to the next, e.g., when the step size is large. On the other hand, when the gradients change smoothly, the algorithm in [Borsos et al., 2018] should converge faster. In addition to [Namkoong et al., 2017] and [Borsos et al., 2018], we prove that the fastest convergence rate of SGD with non-uniform sampling, when the optimal sampling distribution is used, depends linearly on the variance at the optimal coordinate θ^* . We show that SGD in conjunction with MABS can achieve this rate. To prove the fastest achievable convergence rate of SGD with non-uniform sampling, we need to use a biased sampling distribution, which for our setting means that approximation factor should be larger than 1. In our work, we use the smoothness for proving the convergence and bounding the gradient, whereas [Namkoong et al., 2017] and [Borsos et al., 2018] use a projection step to ensure that the gradients are bounded. Using smoothness allows us to naturally bound the gradient and to establish bounds on the minimum achievable cumulative variance.

2.4 Technical Contributions

Before introducing our variance minimization framework, we study the *minimum achievable* cumulative effective variance $\sum_{t=1}^T \min_{\mathbf{p}^t} \mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p}^t)$ which is a function of the sequence of coordinates $\boldsymbol{\theta}^1, \dots, \boldsymbol{\theta}^t$. In Lemma 2.2, we show that if $\boldsymbol{\theta}^t$ converges to $\boldsymbol{\theta}^*$ with rate $\mathbb{E}_{\boldsymbol{\theta}^t} [\|\boldsymbol{\theta}^t - \boldsymbol{\theta}^*\|^2] = O(1/t)$, then an upper and lower bound (depending on $\boldsymbol{\theta}^*$ but not $\boldsymbol{\theta}^t$) on $\sum_{t=1}^T \mathbb{E}_{\boldsymbol{\theta}^t} [\min_{\mathbf{p}^t} \mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p}^t)]$ can be established. By minimizing the cumulative effective variance $\sum_{t=1}^T \mathbb{E}_{\boldsymbol{\theta}^t} [\min_{\mathbf{p}^t} \mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p}^t)]$, SGD achieves its fastest rate in (2.7).

Lemma 2.2. *Let ϕ_i s be L_i -smooth and assume that the coordinates $\boldsymbol{\theta}^t$ converge to $\boldsymbol{\theta}^*$ with rate $\mathbb{E}_{\boldsymbol{\theta}^t} [\|\boldsymbol{\theta}^t - \boldsymbol{\theta}^*\|^2] = O(1/t)$. Then, the minimum achievable cumulative effective variance $\sum_{t=1}^T \mathbb{E}_{\boldsymbol{\theta}^t} [\min_{\mathbf{p}^t} \mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p}^t)]$ lies in the range*

$$\begin{aligned} \frac{T}{4} \min_{\mathbf{p}} \mathbb{V}_e(\boldsymbol{\theta}^*, \mathbf{p}) - O\left(\log T \left(\frac{\sum_{i=1}^n L_i}{n}\right)^2\right) &\leq \sum_{t=1}^T \mathbb{E}_{\boldsymbol{\theta}^t} \left[\min_{\mathbf{p}^t} \mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p}^t) \right] \\ &\leq 4T \min_{\mathbf{p}} \mathbb{V}_e(\boldsymbol{\theta}^*, \mathbf{p}) + O\left(\log T \left(\frac{\sum_{i=1}^n L_i}{n}\right)^2\right), \end{aligned} \quad (2.10)$$

where $\min_{\mathbf{p}} \mathbb{V}_e(\boldsymbol{\theta}^*, \mathbf{p})$ is the minimum effective variance at the optimal coordinates $\boldsymbol{\theta}^*$.

Lemma 2.2 shows that when $\min_{\mathbf{p}} \mathbb{V}_e(\boldsymbol{\theta}^*, \mathbf{p}) > 0$, the minimum achievable cumulative effective variance $\sum_{t=1}^T \mathbb{E}_{\boldsymbol{\theta}^t} [\min_{\mathbf{p}^t} \mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p}^t)] = O(T \min_{\mathbf{p}} \mathbb{V}_e(\boldsymbol{\theta}^*, \mathbf{p}))$. As a result, the fastest convergence rate that SGD can achieve in (2.7) is

$$\mathbb{E} [\|\boldsymbol{\theta}^{T+1} - \boldsymbol{\theta}^*\|^2] = O\left(\frac{1}{\mu^2 T} \min_{\mathbf{p}} \mathbb{V}_e(\boldsymbol{\theta}^*, \mathbf{p})\right). \quad (2.11)$$

SGD attains this rate under the condition $\mathbb{E}_{\boldsymbol{\theta}^t} [\|\boldsymbol{\theta}^t - \boldsymbol{\theta}^*\|^2] = O(1/t)$, and we design MABS to be such that this condition is satisfied.

Proof of Lemma 2.2. From the triangle inequality,

$$\begin{aligned} \|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^t)\|^2 &\leq 2\|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^t) - \nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^*)\|^2 + 2\|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^*)\|^2 \\ &\leq 2L_i^2 \|\boldsymbol{\theta}^t - \boldsymbol{\theta}^*\|^2 + 2\|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^*)\|^2, \end{aligned}$$

where the second inequality follows from the L_i -smoothness of ϕ_i . As a result, the effective variance in (2.5) is bounded as

$$\mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p}^t) \leq \frac{2}{n^2} \sum_{i=1}^n \frac{1}{p_i^t} \left(L_i^2 \|\boldsymbol{\theta}^t - \boldsymbol{\theta}^*\|^2 + \|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^*)\|^2 \right),$$

and therefore, for any distribution \mathbf{q} ,

$$\min_{\mathbf{p}^t} \mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p}^t) \leq \frac{2}{n^2} \sum_{i=1}^n \frac{1}{q_i} \left(L_i^2 \|\boldsymbol{\theta}^t - \boldsymbol{\theta}^*\|^2 + \|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^*)\|^2 \right). \quad (2.12)$$

Setting

$$q_i = \frac{1}{2} \frac{L_i}{\sum_{j=1}^n L_j} + \frac{1}{2} \frac{\|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^*)\|}{\sum_{j=1}^n \|\nabla_{\boldsymbol{\theta}} \phi_j(\boldsymbol{\theta}^*)\|}$$

in (2.12), it becomes

$$\begin{aligned} \min_{\mathbf{p}^t} \mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p}^t) &\leq \frac{4}{n^2} \left[\|\boldsymbol{\theta}^t - \boldsymbol{\theta}^*\|^2 \left(\sum_{i=1}^n L_i \right)^2 + \left(\sum_{i=1}^n \|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^*)\| \right)^2 \right] \\ &= 4 \left[\frac{1}{n^2} \|\boldsymbol{\theta}^t - \boldsymbol{\theta}^*\|^2 \left(\sum_{i=1}^n L_i \right)^2 + \min_{\mathbf{p}} \mathbb{V}_e(\boldsymbol{\theta}^*, \mathbf{p}) \right], \end{aligned}$$

where the last equality follows from (2.5) and (2.6). By taking expectations over $\boldsymbol{\theta}^t$, and next by summing the inequality above over all t and finally by using the assumption $\mathbb{E}_{\boldsymbol{\theta}^t} [\|\boldsymbol{\theta}^t - \boldsymbol{\theta}^*\|^2] = O(1/t)$ we obtain the desired upper bound. The lower bound can be proven using the same technique and tools. \square

The values $\{\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^t)\}_{i \in [n]}$ change over time because they depend on $\boldsymbol{\theta}^t$, which itself varies over time during the iterative process, and they are therefore difficult to estimate in the rounds that follow t . As a result, finding $\arg \min_{\mathbf{p}^t} \mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p}^t)$ at each timestep t is hard. Alternatively, we can solve the easier problem of finding the optimal invariant distribution over T timesteps, which we denote by $\hat{\mathbf{p}}$:

$$\hat{\mathbf{p}} := \hat{\mathbf{p}}(T) = \arg \min_{\mathbf{p}} \sum_{t=1}^T \mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p}). \quad (2.13)$$

If $\boldsymbol{\theta}^t$ converges to the optimal coordinates $\boldsymbol{\theta}^*$, the cumulative effective variance of the optimal invariant distribution $\min_{\mathbf{p}} \sum_{t=1}^T \mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p})$ should also be close to the cumulative minimum effective variance $\sum_{t=1}^T \min_{\mathbf{p}^t} \mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p}^t)$. Indeed, Lemma 2.3 shows that we can replace the latter by the former in (2.10).

Lemma 2.3. *Under the assumptions of Lemma 2.2, the minimum cumulative effective variance $\mathbb{E}_{\boldsymbol{\theta}^1, \dots, \boldsymbol{\theta}^T} \left[\min_{\mathbf{p}} \sum_{t=1}^T \mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p}) \right]$ when the optimal invariant distribution $\hat{\mathbf{p}}$ is used lies in the range*

$$\begin{aligned} \frac{T}{4} \min_{\mathbf{p}} \mathbb{V}_e(\boldsymbol{\theta}^*, \mathbf{p}) - O \left(\log T \left(\frac{\sum_{i=1}^n L_i}{n} \right)^2 \right) &\leq \mathbb{E}_{\boldsymbol{\theta}^1, \dots, \boldsymbol{\theta}^T} \left[\min_{\mathbf{p}} \sum_{t=1}^T \mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p}) \right] \\ &\leq 4T \min_{\mathbf{p}} \mathbb{V}_e(\boldsymbol{\theta}^*, \mathbf{p}) + O \left(\log T \left(\frac{\sum_{i=1}^n L_i}{n} \right)^2 \right), \end{aligned} \quad (2.14)$$

where $\min_{\mathbf{p}} \mathbb{V}_e(\boldsymbol{\theta}^*, \mathbf{p})$ is the minimum effective variance at the optimal coordinates $\boldsymbol{\theta}^*$.

As $\min_{\mathbf{p}} \sum_{t=1}^T \mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p}) \geq \sum_{t=1}^T \min_{\mathbf{p}^t} \mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p}^t)$, the lower bound follows automatically from (2.10). The proof of the upper bound is similar to the proof of the upper bound in Lemma 2.2, and is omitted.

Lemmas 2.2 and 2.3 yield that

$$\mathbb{E}_{\boldsymbol{\theta}^1, \dots, \boldsymbol{\theta}^T} \left[\min_{\mathbf{p}} \sum_{t=1}^T \mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p}) \right] \approx \sum_{t=1}^T \mathbb{E}_{\boldsymbol{\theta}^t} \left[\min_{\mathbf{p}^t} \mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p}^t) \right] \in O \left(T \min_{\mathbf{p}} \mathbb{V}_e(\boldsymbol{\theta}^*, \mathbf{p}) \right)$$

asymptotically with T when $\min_{\mathbf{p}} \mathbb{V}_e(\boldsymbol{\theta}^*, \mathbf{p}) > 0$. They make it therefore possible to replace the difficult selection of $\mathbf{p}^{t*} = \arg \min_{\mathbf{p}^t} \mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p}^t)$ at each timestep t , by the easier task of finding only one distribution $\hat{\mathbf{p}}$ as in (2.13).

Note that the optimal sampling distribution \mathbf{p}^* at $\boldsymbol{\theta}^*$ is in general different from the distribution $\hat{\mathbf{p}}$ given by (2.13), and that

$$\frac{1}{T} \sum_{t=1}^T \mathbb{V}_e(\boldsymbol{\theta}^t, \hat{\mathbf{p}}) \leq \frac{1}{T} \sum_{t=1}^T \mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p}^*).$$

Using \mathbf{p}^* at each iteration can easily make the variance large (or even unbounded). As an example, assume $\|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^*)\|^2$ is small for some i compared to the other gradients for $j \neq i$: then because of (2.6), p_i^* is small. However, there is no need for $\|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^t)\|^2$ to be small, which makes $\|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^t)\|^2 / p_i^*$ large. Using a biased distribution as in [Needell et al., 2014] can solve this issue.

Both Lemmas 2.2 and 2.3 require $\boldsymbol{\theta}^t$ to converge to $\boldsymbol{\theta}^*$. Indeed, if this condition is not satisfied, finding an optimal sampling distribution for a sequence of coordinates $\boldsymbol{\theta}^1, \dots, \boldsymbol{\theta}^T$ that does not converge to the optimal coordinates $\boldsymbol{\theta}^*$ is useless. In Section 2.5, we find that the simple constraint $p_i^t \geq \eta/n$ for all $1 \leq t \leq T$ and for some $0 < \eta \leq 1$ guarantees that in expectation $\boldsymbol{\theta}^t$ to converge to $\boldsymbol{\theta}^*$ with the rate $\mathbb{E}[\|\boldsymbol{\theta}^t - \boldsymbol{\theta}^*\|^2] = O(1/t)$. To satisfy this constraint we can simply use a biased distribution p_i^t . A consequence of this biasing is that no algorithm can find the optimal invariant distribution (2.13).

We relax the problem of finding the optimal sampling distribution $\hat{\mathbf{p}}$ in (2.13) to find an approximate solution of (2.13), i.e., a sequence of distributions $\{\mathbf{p}^t\}$ for $t \in [T]$ such that

$$\frac{1}{T} \sum_{t=1}^T \mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p}^t) \leq \frac{c}{T} \sum_{t=1}^T \mathbb{V}_e(\boldsymbol{\theta}^t, \hat{\mathbf{p}})$$

for some constant $1 \leq c \leq \bar{c}$. We design algorithms such that $\bar{c} \leq 3$.

We note that the framework is not limited to the SGD, and that it is general enough to construe the variance of a broader class of stochastic optimization algorithms such as CD, SVRG, and SAGA in a similar way. Using an unbiased estimator for the gradient is the common property that these algorithms share. In Section 2.6, we show how to decompose the variance of the unbiased gradient of SVRG and SAGA into \mathbb{V}_e and \mathbb{V}_c similar to (2.4).

2.4.1 Multi-armed Bandit Sampling

To find a sampling distribution \mathbf{p}^t that can be computed at time t and that approximates $\hat{\mathbf{p}}$ as well as possible, we transform the problem (2.13) into an adversarial multi-armed bandit (MAB) problem (see [Bubeck and Cesa-Bianchi, 20120] and Section 1.2 for more details about MAB). From now on and for the sake of simplicity, we drop the explicit dependence on $\boldsymbol{\theta}^t$ in $\mathbb{V}_e^t(\mathbf{p}^t) = \mathbb{V}_e(\boldsymbol{\theta}^t, \mathbf{p}^t)$ and use the shorthand

$$a_i^t = \frac{1}{n^2} \left\| \nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^t) \right\|^2. \quad (2.15)$$

The effective variance in (2.5) becomes

$$\mathbb{V}_e^t(\mathbf{p}^t) = \sum_{i=1}^n \frac{a_i^t}{p_i^t}. \quad (2.16)$$

To use the classic framework of MAB, we need to formulate the effective variance minimization problem $\min_{\mathbf{p}^t} \sum_t \mathbb{V}_e^t(\mathbf{p}^t)$ as an adversarial MAB with the cost function $C^t = \langle \mathbf{p}^t - \hat{\mathbf{p}}, \mathbf{r}^t \rangle$, where the losses \mathbf{r}^t need to be defined, as a function of the effective variance $\mathbb{V}_e^t(\mathbf{p}^t)$. The following lemma provides the basis for this definition.

Lemma 2.4. *For any real value constant $\zeta \leq 0.5$ and arbitrary sampling distributions \mathbf{p}^1 and \mathbf{p}^2 we have*

$$(1 - 2\zeta)\mathbb{V}_e^t(\mathbf{p}^1) - (1 - \zeta)\mathbb{V}_e^t(\mathbf{p}^2) \leq \langle \mathbf{p}^1 - \mathbf{p}^2, \nabla_{\mathbf{p}} \mathbb{V}_e^t(\mathbf{p}^1) \rangle + \zeta \langle \mathbf{p}^2, \nabla_{\mathbf{p}} \mathbb{V}_e^t(\mathbf{p}^1) \rangle. \quad (2.17)$$

Take $\mathbf{p}^1 = \mathbf{p}^t$ and $\mathbf{p}^2 = \hat{\mathbf{p}}$ in (2.17), then it becomes

$$(1 - 2\zeta)\mathbb{V}_e^t(\mathbf{p}^t) - (1 - \zeta)\mathbb{V}_e^t(\hat{\mathbf{p}}) \leq \langle \mathbf{p}^t - \hat{\mathbf{p}}, \nabla_{\mathbf{p}} \mathbb{V}_e^t(\mathbf{p}^t) \rangle + \zeta \langle \hat{\mathbf{p}}, \nabla_{\mathbf{p}} \mathbb{V}_e^t(\mathbf{p}^t) \rangle. \quad (2.18)$$

After rearranging (2.18) we get

$$\mathbb{V}_e^t(\mathbf{p}^t) \leq \frac{1 - \zeta}{1 - 2\zeta} \mathbb{V}_e^t(\hat{\mathbf{p}}) + \frac{1}{1 - 2\zeta} \langle \mathbf{p}^t - \hat{\mathbf{p}}, \nabla_{\mathbf{p}} \mathbb{V}_e^t(\mathbf{p}^t) \rangle + \frac{\zeta}{1 - 2\zeta} \langle \hat{\mathbf{p}}, \nabla_{\mathbf{p}} \mathbb{V}_e^t(\mathbf{p}^t) \rangle. \quad (2.19)$$

Observe that the first term in the right-hand side of (2.19) is the optimal effective variance we are looking for, the second term is the cost function C^t of an adversarial

MAB with the i^{th} loss

$$r_i^t = \left(\nabla_{\mathbf{p}} \mathbb{V}_e^t(\mathbf{p}^t) \right)_i = -\frac{a_i^t}{(p_i^t)^2}. \quad (2.20)$$

and the last term is a residual term that we can control by lowering ζ . Although the loss (2.20) is a function of \mathbf{p}^t , it is not an issue here, because our MAB algorithm is adversarial, and in an adversarial MAB, the losses can take any arbitrary form, including being dependent on \mathbf{p}^t (see Section 1.2.2). By upper bounding the last two terms in (2.19), we can guarantee the closeness of $\mathbb{V}_e^t(\mathbf{p}^t)$ to the optimal solution $\mathbb{V}_e^t(\hat{\mathbf{p}})$.

Proof of Lemma 2.4. The effective variance $\mathbb{V}_e^t(\mathbf{p})$ is a convex function with respect to \mathbf{p} , hence for any two $\mathbf{p}^1, \mathbf{p}^2$ and any $\zeta < 1$ we have

$$(1 - \zeta)\mathbb{V}_e^t(\mathbf{p}^1) - (1 - \zeta)\mathbb{V}_e^t(\mathbf{p}^2) \leq (1 - \zeta) \left\langle \mathbf{p}^1 - \mathbf{p}^2, \nabla_{\mathbf{p}} \mathbb{V}_e^t(\mathbf{p}^1) \right\rangle. \quad (2.21)$$

By rearranging the terms of (2.21) we get

$$\begin{aligned} (1 - \zeta)\mathbb{V}_e^t(\mathbf{p}^1) + \zeta \left\langle \mathbf{p}^1, \nabla_{\mathbf{p}} \mathbb{V}_e^t(\mathbf{p}^1) \right\rangle - (1 - \zeta)\mathbb{V}_e^t(\mathbf{p}^2) \leq \\ \left\langle \mathbf{p}^1 - \mathbf{p}^2, \nabla_{\mathbf{p}} \mathbb{V}_e^t(\mathbf{p}^1) \right\rangle + \zeta \left\langle \mathbf{p}^2, \nabla_{\mathbf{p}} \mathbb{V}_e^t(\mathbf{p}^1) \right\rangle. \end{aligned} \quad (2.22)$$

Note that (2.16) yields that

$$\left\langle \mathbf{p}^1, \nabla_{\mathbf{p}} \mathbb{V}_e^t(\mathbf{p}^1) \right\rangle = -\sum_{i=1}^n p_i^1 \frac{a_i^t}{(p_i^1)^2} = -\mathbb{V}_e^t(\mathbf{p}^1). \quad (2.23)$$

By plugging (2.23) in (2.22) we obtain (2.17), which concludes the proof. \square

Building on this analogy between MAB and datapoint sampling, we propose an algorithm, based on EXP3 (see [Auer et al., 2002b] and Section 1.2.2), which we call MABS (for Multi-Armed Bandit Sampling). The loss in the MAB problem is given by (2.20) for all arms (datapoints) $i \in [n]$. The MABS algorithm has n weights $\{w_i^t\}_{i \in [n]}$, each initialized to 1. The sum of weights is called *potential function* $W^t = \sum_{j=1}^n w_j^t$. The distribution \mathbf{p}^t is a combination of the distribution $\{w_i^t/W^t\}_{i \in [n]}$ computed from the weights at time t and of the uniform distribution $\{1/n\}_{i \in [n]}$:

$$p_i^t = (1 - \eta) \frac{w_i^t}{W^t} + \eta \frac{1}{n}. \quad (2.24)$$

MABS trades off exploration with exploitation. The sampling distribution $\{w_i^t/W^t\}_{i \in [n]}$ is responsible for exploitation. It increases the sampling probability of the datapoints with

Algorithm 2.1 MABS

```

1: Input:  $\eta$ , and  $\delta$ 
2: Initialize:  $w_i^1 = 1$  ▷ for all  $i \in [n]$ 
3: for  $t = 1 : T$  do
4:    $W^t = \sum_{j=1}^n w_j^t$ 
5:    $p_j^t \leftarrow (1 - \eta) \frac{w_j^t}{W^t} + \eta \frac{1}{n}$  ▷ for all  $j \in [n]$ 
6:   Sample  $i \sim \mathbf{p}^t$ 
7:   Update  $\boldsymbol{\theta}^t$  using  $\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^t)$ 
8:    $w_i^{t+1} = w_i^t \cdot \exp(\frac{\delta a_i^t}{(p_i^t)^3})$ 
9:    $w_j^{t+1} = w_j^t$  ▷ for all  $j \neq i$ 
10: end for

```

a larger a_i^t . The uniform distribution $\{1/n\}_{i \in [n]}$ is responsible for exploration. It ensures that MABS gathers enough information about a_i^t through the course of optimization. The parameter η determines how much exploration is needed and how much p_i^t deviates from the uniform distribution.

MABS updates the weights by using an unbiased estimator for r_i^t , i.e.,

$$\hat{r}_i^t = \begin{cases} \frac{r_i^t}{p_i^t} & \text{if } i \text{ is chosen at time } t, \\ 0 & \text{otherwise,} \end{cases} \quad (2.25)$$

according to the updating rule

$$w_i^{t+1} = w_i^t \exp(-\delta \hat{r}_i^t), \quad (2.26)$$

where δ is a parameter that controls how much w_i^t can change from one iteration t to the next iteration $t + 1$ based on the value of unbiased estimator \hat{r}_i^t . More precisely, MABS only updates the weight $w_{i_t}^t$ of the selected datapoint i_t at timestep t and keeps all the other ones fixed, i.e., $w_i^{t+1} = w_i^t$ for all $i \neq i_t$.

Remark 2.5. *A difference between the variance-reduction problem (2.13) and multi-armed bandits is that in the latter the losses are assumed to be upper bounded almost surely. Whereas in the former, the losses might be unbounded, depending on the distribution \mathbf{p}^t . This occurs if the probability p_i^t is close to 0, making the term a_i^t/p_i^t in (2.16) very large. By taking p_i^t from (2.24), one ensures that $p_i^t \geq \eta/n > 0$, which avoids this problem.*

Theorem 2.6. *Using MABS (Algorithm 2.1) with $0 < \eta < 0.5$ in (2.24) and $\delta = \sqrt{\eta^4 \ln n / (n^5 T a^2)}$ in (2.26) to minimize (2.16) with respect to $\{\mathbf{p}^t\}_{1 \leq t \leq T}$, we have*

$$\sum_{t=1}^T \mathbb{V}_e(\mathbf{p}^t) \leq \frac{1-\eta}{1-2\eta} \sum_{t=1}^T \mathbb{V}_e(\hat{\mathbf{p}}) + \frac{2-\eta}{\eta^2(1-2\eta)} \sqrt{n^5 T a^2 \ln n}, \quad (2.27)$$

where $T \geq \left(n \max_i (a_i)^2 / \eta^2 \sum_j (a_j)^2 \right) n \ln n$ for some $a_i \geq \sup_t \{a_i^t\}$, and some

$$\bar{a}^2 \geq \frac{\sum_{t=1}^T \sum_{i=1}^n (a_i^t)^2}{nT}.$$

The complexity of MABS is $O(\log_2 n)$ per iteration.

The condition $T \geq \left(n \max_i (a_i)^2 / \eta^2 \sum_j (a_j)^2 \right) n \ln n$ ensures that $-\delta \hat{r}_i^t \leq 1$, which we need in the proof. We could also replace a_i^t with $a_i = \sup_t \{a_i^t\}$ in δ and the result of theorem still holds. In Theorem 2.6, the second term of the right-hand side of (2.27) shows how fast the algorithm converges. To understand the effect of η on (2.27), note that a small η in (2.24) pushes the algorithm to exploit more often, which makes the first term of (2.27) smaller, and the second term larger. As an example, if we choose $\eta = 0.2$ in Theorem 2.6, asymptotically as $T \rightarrow \infty$, (2.27) becomes

$$\sum_{t=1}^T \mathbb{V}_e^t(\mathbf{p}^t) \leq 1.4 \sum_{t=1}^T \mathbb{V}_e^t(\hat{\mathbf{p}}) + 75 \sqrt{n^4 \sum_{t=1}^T \sum_{i=1}^n (a_i^t)^2 \ln n}.$$

Finding the optimum η that minimizes the right-hand side of (2.27) is impossible because we do not know $\sum_{t=1}^T \mathbb{V}_e^t(\hat{\mathbf{p}})$ a priori. But it is not necessary either, because tuning η only changes the constants in the convergence guarantee of Theorem 2.10. In experiments, we find that setting $\eta = 0.4$ yields good performance on different datasets. With $\eta = 0.4$, (2.27) then becomes

$$\sum_{t=1}^T \mathbb{V}_e^t(\mathbf{p}^t) \leq 3 \sum_{t=1}^T \mathbb{V}_e^t(\hat{\mathbf{p}}) + 50 \sqrt{n^4 \sum_{t=1}^T \sum_{i=1}^n (a_i^t)^2 \ln n}. \quad (2.28)$$

In SGD, PSGD, SVRG, and SAGA the effective variance $\mathbb{V}_e^t(\mathbf{p}^t)$ scales as $\sqrt{n \ln n}$ because $a_i^t \sim 1/n^2$ and $a_i \sim 1/n^2$. In addition, note that the second term of the right-hand side of (2.28) increases as \sqrt{T} , whence

$$\frac{1}{T} \sum_{t=1}^T \mathbb{V}_e^t(\mathbf{p}^t) \leq \frac{3}{T} \sum_{t=1}^T \mathbb{V}_e^t(\hat{\mathbf{p}}) + O\left(\sqrt{\frac{n \ln n}{T}}\right).$$

Therefore, as T becomes large, the solution \mathbf{p}^t returned by MABS approximates the solution $\hat{\mathbf{p}}$ of (2.13). The computation of the gradient $\nabla_{\theta} \phi_i(\boldsymbol{\theta}^t)$ requires $O(d)$ computations so that the computational overhead of MABS is insignificant if $\log n$ is small compared to the coordinate dimension d . Such is the case for almost all datasets, in particular for the two datasets in Table 2.2 used in the evaluation section (see Section 2.6). The condition $T = O(n \ln n)$ on T might be prohibitive if n is large, but we can relax it at the expense of having a slightly worse bound (see Appendix 2.A).

Chapter 2. Stochastic Gradient Descent with Bandit Sampling

The proof uses Lemma 2.4 to linearize the effective variance $\sum_{t=1}^T \mathbb{V}_e^t(\mathbf{p}^t)$, and next defines a potential function $W^t = \sum_{i=1}^n w_i^t$ and adopts the approach of multiplicative-weight update algorithms (see for example [Auer et al., 2002b]) combined with Lemma 2.4. By upper and lower bounding the potential function θ^{T+1} at iteration $T+1$, we derive the result of Theorem 2.6. We present the full proof below.

Proof of Theorem 2.6. Remember that $r_i^t = -a_i^t/(p_i^t)^2$ is the loss of datapoint i , because of (2.20), that $\hat{r}_i^t = r_i^t \cdot \mathbb{1}_{\{I_t=i\}}/p_i^t$ is an unbiased estimator for r_i^t , because of (2.25), and that the update rule for the weight w_i^t at timestep t is $w_i^{t+1} = w_i^t \cdot \exp(-\delta \hat{r}_i^t)$, because of (2.26). Therefore, $w_i^{T+1} = \exp\left(-\delta \sum_{t=1}^T \hat{r}_i^t\right)$ and the potential function at time $T+1$ is

$$W^{T+1} = \sum_{i=1}^n w_i^{T+1} \geq w_j^{T+1} = \exp\left(-\delta \sum_{t=1}^T \hat{r}_j^t\right)$$

for all $j \in [n]$. Since $W^1 = \sum_{i=1}^n w_i^1 = n$, we get the following lower bound on $\ln W^{T+1}/W^1$,

$$-\delta \sum_{t=1}^T \hat{r}_j^t - \ln n \leq \ln \frac{W^{T+1}}{W^1}. \quad (2.29)$$

Now, let us upper bound W^{T+1} . Observe that

$$\frac{W^{t+1}}{W^t} = \frac{\sum_{i=1}^n w_i^{t+1}}{W^t} = \frac{\sum_{i=1}^n w_i^t \exp(-\delta \hat{r}_i^t)}{W^t} = \sum_{i=1}^n \left(\frac{p_i^t - \eta/n}{1 - \eta}\right) \exp(-\delta \hat{r}_i^t), \quad (2.30)$$

where $w_i^t/W^t = (p_i^t - \eta/n)/(1 - \eta)$ follows from (2.24). If $-\delta \hat{r}_i^t \leq 1$, we can plug the inequality $\exp(x) \leq 1 + x + x^2$, which holds for all $x \leq 1$, in (2.30), and it becomes

$$\frac{W^{t+1}}{W^t} \leq \sum_{i=1}^n \left(\frac{p_i^t - \eta/n}{1 - \eta}\right) \left(1 - \delta \hat{r}_i^t + (\delta \hat{r}_i^t)^2\right) \leq 1 - \frac{\delta}{1 - \eta} \sum_{i=1}^n p_i^t \hat{r}_i^t + \frac{\delta^2}{1 - \eta} \sum_{i=1}^n p_i^t (\hat{r}_i^t)^2. \quad (2.31)$$

We provide later in the proof a condition that ensures $-\delta \hat{r}_i^t \leq 1$. As $-\hat{r}_i^t \geq 0$, all terms of the right-hand side of (2.31) are non-negative, hence we can plug the inequality $\ln(1+x) \leq x$, which holds for all $x \geq 0$, in (2.31) and it becomes

$$\ln \frac{W^{t+1}}{W^t} \leq -\frac{\delta}{1 - \eta} \sum_{i=1}^n p_i^t \hat{r}_i^t + \frac{\delta^2}{1 - \eta} \sum_{i=1}^n p_i^t (\hat{r}_i^t)^2. \quad (2.32)$$

Summing (2.32) for $1 \leq t \leq T$, we get the following upper bound on $\ln W^{T+1}/W^1$

$$\ln \frac{W^{T+1}}{W^1} = \sum_{t=1}^T \ln \frac{W^{t+1}}{W^t} \leq -\frac{\delta}{1 - \eta} \sum_{t=1}^T \sum_{i=1}^n p_i^t \hat{r}_i^t + \frac{\delta^2}{1 - \eta} \sum_{t=1}^T \sum_{i=1}^n p_i^t (\hat{r}_i^t)^2. \quad (2.33)$$

Combining the lower bound (2.29) and the upper bound (2.33), we get

$$-\delta \sum_{t=1}^T \hat{r}_j^t - \ln n \leq -\frac{\delta}{1-\eta} \sum_{t=1}^T \sum_{i=1}^n p_i^t \hat{r}_i^t + \frac{\delta^2}{1-\eta} \sum_{t=1}^T \sum_{i=1}^n p_i^t (\hat{r}_i^t)^2. \quad (2.34)$$

From (2.25), $\mathbb{E}[\hat{r}_i^t] = r_i^t$ and $\mathbb{E}[(\hat{r}_i^t)^2] = (r_i^t)^2/p_i^t$, hence by taking expectations over \mathbf{p}^t in (2.34), we get

$$-\delta \sum_{t=1}^T r_j^t - \ln n \leq -\frac{\delta}{1-\eta} \sum_{t=1}^T \sum_{i=1}^n p_i^t r_i^t + \frac{\delta^2}{1-\eta} \sum_{t=1}^T \sum_{i=1}^n (r_i^t)^2. \quad (2.35)$$

By multiplying (2.35) by \hat{p}_j and summing over j , we get

$$-\delta \sum_{t=1}^T \sum_{j=1}^n \hat{p}_j r_j^t - \ln n \leq -\frac{\delta}{1-\eta} \sum_{t=1}^T \sum_{i=1}^n p_i^t r_i^t + \frac{\delta^2}{1-\eta} \sum_{t=1}^T \sum_{i=1}^n (r_i^t)^2. \quad (2.36)$$

As $r_i^t = -a_i^t/(p_i^t)^2 = (\nabla_{\mathbf{p}} \mathbb{V}_e^t(\mathbf{p}^t))_i$, we have $\sum_{i=1}^n p_i r_i^t = \langle \mathbf{p}, \nabla_{\mathbf{p}} \mathbb{V}_e^t(\mathbf{p}^t) \rangle$ for any distribution \mathbf{p} . By plugging this in (2.36) and rearranging, we find

$$\sum_{t=1}^T \langle \mathbf{p}^t - \hat{\mathbf{p}}, \nabla_{\mathbf{p}} \mathbb{V}_e^t(\mathbf{p}^t) \rangle + \eta \sum_{t=1}^T \langle \hat{\mathbf{p}}, \nabla_{\mathbf{p}} \mathbb{V}_e^t(\mathbf{p}^t) \rangle \leq \frac{1-\eta}{\delta} \ln n + \delta \sum_{t=1}^T \sum_{i=1}^n (r_i^t)^2. \quad (2.37)$$

Setting $\zeta = \eta$ in (2.18) and combining it with (2.37) gives

$$(1-2\eta) \sum_{t=1}^T \mathbb{V}_e^t(\mathbf{p}^t) - (1-\eta) \sum_{t=1}^T \mathbb{V}_e^t(\hat{\mathbf{p}}) \leq \frac{1-\eta}{\delta} \ln n + \delta \sum_{t=1}^T \sum_{i=1}^n (r_i^t)^2, \quad (2.38)$$

and by rearranging the terms of (2.38) and noting that $0 < \eta < 0.5$, we finally get

$$\sum_{t=1}^T \mathbb{V}_e^t(\mathbf{p}^t) \leq \frac{1-\eta}{1-2\eta} \sum_{t=1}^T \mathbb{V}_e^t(\hat{\mathbf{p}}) + \frac{1-\eta}{\delta(1-2\eta)} \ln n + \frac{\delta}{1-2\eta} \sum_{t=1}^T \sum_{i=1}^n (r_i^t)^2. \quad (2.39)$$

Because of (2.20) and (2.24), (2.39) becomes

$$\sum_{t=1}^T \mathbb{V}_e^t(\mathbf{p}^t) \leq \frac{1-\eta}{1-2\eta} \sum_{t=1}^T \mathbb{V}_e^t(\hat{\mathbf{p}}) + \frac{1-\eta}{\delta(1-2\eta)} \ln n + \frac{\delta n^4}{(1-2\eta)\eta^4} \sum_{t=1}^T \sum_{i=1}^n (a_i^t)^2. \quad (2.40)$$

Setting $\delta = \sqrt{\eta^4 \ln n / (n^5 T \bar{a}^2)}$ for some $\bar{a}^2 \geq \{\sum_{t=1}^T \sum_{i=1}^n (a_i^t)^2 / nT\}$ in (2.40) concludes the proof. The condition $T \geq \left(n \max_i (a_i)^2 / \eta^2 \sum_j (a_j)^2 \right) n \ln n$ in the assumptions of Theorem 2.6 ensures that δ is small and $-\delta \hat{r}_i^t \leq 1$, which is needed to use $\exp(x) \leq 1 + x + x^2$ for $x = -\delta \hat{r}_i^t$ in (2.31).

Chapter 2. Stochastic Gradient Descent with Bandit Sampling

Algorithm 2.2 MABS2

- 1: **Input:** η, δ , and a_i \triangleright for all $i \in [n]$
 - 2: **Initialize:** $q_i = |a_i|^{2/5} / (\sum_{j=1}^n |a_j|^{2/5})$ and $w_i^1 = 1$ \triangleright for all $i \in [n]$
 - 3: **for** $t = 1 : T$ **do** $\boldsymbol{\theta}^t = \sum_{j=1}^n w_j^t$
 - 4: $p_j^t \leftarrow (1 - \eta) \frac{w_j^t}{W^t} + \eta q_j$ \triangleright for all $j \in [n]$
 - 5: Sample $i \sim \mathbf{p}^t$
 - 6: Update $\boldsymbol{\theta}^t$ using $\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^t)$
 - 7: $w_i^{t+1} = w_i^t \cdot \exp(\frac{\delta a_i^t}{(p_i^t)^3})$
 - 8: $w_j^{t+1} = w_j^t$ \triangleright for all $j \neq i$
 - 9: **end for**
-

Finally, as in Section A.4 of [Salehi et al., 2017b], with a tree structure (similar to the interval tree) we can update w_{i_t} and sample from p^t in $O(\log_2 n)$ computations per step. \square

Remark 2.7. *If we know $a_i = \sup_t \{a_i^t\}$, then we can refine MABS and improve the bound (2.27). In MABS2 (Algorithm 2.2), instead of combining the distribution $\{w_i^t/W^t\}_{i \in [n]}$ with a uniform distribution, the idea is to combine $\{w_i^t/W^t\}_{i \in [n]}$ with a non-uniform distribution $\mathbf{q} = \{q_i\}_{i \in [n]}$. In other words (2.24) is replaced by*

$$p_i^t = (1 - \eta) \frac{w_i^t}{W^t} + \eta q_i, \quad (2.41)$$

where distribution \mathbf{q} should be such that if a_j is large for some $j \in [n]$, then q_j is large as well. This way the worst-case guarantee on a_i^t/p_i^t can be strengthened, because the lower bound ηq_i on p_i^t is larger for a datapoint i for which a_i is large as well. In Corollary 2.12 (in Appendix 2.A.2), we find that the optimal distribution is

$$q_i = \frac{a_i^{2/5}}{\sum_{j=1}^n a_j^{2/5}} \quad \text{for all } i \in [n].$$

Remark 2.8. *The idea of decoupling exploration and exploitation from Avner et al. [2012] can also be used to obtain a lower effective variance \mathbb{V}_e^t . The decoupling of exploration and exploitation enables us to explore better while exploiting the existing information and achieve better performance. More precisely, in MABS3 (Algorithm 2.3), a datapoint i is selected with probability p_i^t for updating the coordinate $\boldsymbol{\theta}^t$, and another datapoint j is selected with probability q_j^t for gathering information about the loss r_j^t and for updating \mathbf{p}^t . An estimator for $r_i^t = -a_i^t/(p_i^t)^2$ is built from \mathbf{q}^t as*

$$\hat{r}_i^t = \begin{cases} r_i^t/q_i^t & \text{if } i \text{ is chosen at time } t, \\ 0 & \text{otherwise,} \end{cases} \quad (2.42)$$

Algorithm 2.3 MABS3

- 1: **Input:** η , δ , and a_i ▷ for all $i \in [n]$
 - 2: **Initialize:** $z_i = a_i^{2/5}/(\sum_{j=1}^n a_j^{2/5})$ and $w_i^1 = 1$ ▷ for all $i \in [n]$
 - 3: **for** $t = 1 : T$ **do** $\theta^t = \sum_{j=1}^n w_j^t$
 - 4: $p_j^t \leftarrow (1 - \eta) \frac{w_j^t}{W^t} + \eta z_i$ ▷ for all $j \in [n]$
 - 5: $q_i^t = \frac{a_i/(p_i^t)^{1.5}}{\sum_{j=1}^n a_j/(p_j^t)^{1.5}}$ ▷ for all $i \in [n]$
 - 6: Sample $i \sim \mathbf{p}^t$
 - 7: Update θ^t using $\nabla_{\theta} \phi_i(\theta^t)$
 - 8: Sample $i \sim \mathbf{q}^t$
 - 9: $w_i^{t+1} = w_i^t \cdot \exp(\frac{\delta \|a_i^t\|^2}{(p_i^t)^2 q_i^t})$
 - 10: $w_j^{t+1} = w_j^t$ ▷ for all $j \neq i$
 - 11: **end for**
-

where in Corollary 2.9, \mathbf{q}^t is set as

$$q_i^t = \frac{a_i/(p_i^t)^{1.5}}{\sum_{j=1}^n a_j/(p_j^t)^{1.5}},$$

to minimize the variance of the estimator \hat{r}_i^t for r_i^t .

Corollary 2.9. Using MABS3 with $0 < \eta < 0.5$ and $\delta = \sqrt{\eta^3 \ln n / \left(T \left(\sum_{i=1}^n a_i^{2/5} \right)^5 \right)}$ to minimize (2.16) with respect to $\{\mathbf{p}^t\}_{1 \leq t \leq T}$, we have

$$\sum_{t=1}^T \mathbb{V}_e^t(\mathbf{p}^t) \leq \frac{1 - \eta}{1 - 2\eta} \sum_{t=1}^T \mathbb{V}_e^t(\hat{\mathbf{p}}) + \frac{2 - \eta}{\eta^{1.5} (1 - 2\eta)} \sqrt{T \left(\sum_{i=1}^n a_i^{2/5} \right)^5 \ln n}. \quad (2.43)$$

where $T \geq \sqrt{\left(\sum_{i=1}^n a_i^{2/5} \right) / \min_i a_i^{1/5} \ln n}$ for some $a_i \geq \sup_t \{a_i^t\}$. The complexity of MABS3 is $O(n)$ per iteration.

The condition $T \geq \sqrt{\left(\sum_{i=1}^n a_i^{2/5} \right) / \min_i a_i^{1/5} \ln n}$ ensures that $-\delta \hat{r}_i^t \leq 1$, which is needed in the proof. Comparing (2.43) to (2.27), we observe that for the same η (that yields the same approximation factor) MABS3 finds the approximate distribution faster, because the second term in the upper bound in (2.43) is smaller than the second term in the upper bound in (2.27), even if both terms increase as \sqrt{T} . In particular, if a_i s vary a lot across $i \in [n]$ (for example, if $\max_i a_i \geq c \sum_{j=1}^n a_j$ for some $1/n \leq c \leq 1$), after dropping the constants, the second term in the upper bound in (2.43) is $(cn)^2 / \sqrt{\eta}$ times smaller than the second term in the upper bound in (2.27). The proof is similar to the proof of Theorem 2.6 and uses the idea of decoupling from [Avner et al., 2012]. We present the full proof below.

Chapter 2. Stochastic Gradient Descent with Bandit Sampling

Proof of Corollary 2.9 . Similar to the proof of Theorem 2.6, we define the potential function $W^t = \sum_{i=1}^n w_i^t$. Remember that the loss of the datapoint i is $r_i^t = -a_i^t/(p_i^t)^2$, because of (2.20), that $\hat{r}_i^t = r_i^t \cdot \mathbb{1}_{\{I_t=i\}}/q_i^t$ is an unbiased estimator for r_i^t , because of (2.42), and that the update rule for the weight w_i^t is $w_i^{t+1} = w_i^t \cdot \exp(-\delta \hat{r}_i^t)$, because of (2.26). Following the same steps as the proof of Theorem 2.6, if $-\delta \hat{r}_i^t \leq 1$ by lower and upper bounding W^{T+1} as in (2.34), we get

$$-\delta \sum_{t=1}^T \hat{r}_j^t - \ln n \leq -\frac{\delta}{1-\eta} \sum_{t=1}^T \sum_{i=1}^n p_i^t \hat{r}_i^t + \frac{\delta^2}{1-\eta} \sum_{t=1}^T \sum_{i=1}^n p_i^t (\hat{r}_i^t)^2, \quad (2.44)$$

we provide later in the proof a condition that ensures $-\delta \hat{r}_i^t \leq 1$. From (2.42), $\mathbb{E}[\hat{r}_i^t] = r_i^t$ and $\mathbb{E}[(\hat{r}_i^t)^2] = (r_i^t)^2/q_i^t$, hence by taking expectations over \mathbf{q}^t in (2.44), we get

$$-\delta \sum_{t=1}^T r_j^t - \ln n \leq -\frac{\delta}{1-\eta} \sum_{t=1}^T \sum_{i=1}^n p_i^t r_i^t + \frac{\delta^2}{1-\eta} \sum_{t=1}^T \sum_{i=1}^n \frac{p_i^t}{q_i^t} (r_i^t)^2. \quad (2.45)$$

With $(r_i^t)^2 = (a_i^t)^2/(p_i^t)^4$ because of (2.20) and $a_i \geq \sup_t \{a_i^t\}$, (2.45) becomes

$$-\delta \sum_{t=1}^T r_j^t - \ln n \leq -\frac{\delta}{1-\eta} \sum_{t=1}^T \sum_{i=1}^n p_i^t r_i^t + \frac{\delta^2}{1-\eta} \sum_{t=1}^T \sum_{i=1}^n \frac{(a_i)^2}{(p_i^t)^3 q_i^t}. \quad (2.46)$$

The only term in (2.46) that is a function of \mathbf{q}^t is $\sum_{i=1}^n (a_i)^2/(p_i^t)^3 q_i^t$, so we choose q_i^t such that $\sum_{i=1}^n (a_i)^2/(p_i^t)^3 q_i^t$ is minimized, which is

$$q_i^t = \frac{a_i/(p_i^t)^{1.5}}{\sum_{j=1}^n a_j/(p_j^t)^{1.5}}. \quad (2.47)$$

Plugging (4.4) in (2.46) yields

$$-\delta \sum_{t=1}^T r_j^t - \ln n \leq -\frac{\delta}{1-\eta} \sum_{t=1}^T \sum_{i=1}^n p_i^t r_i^t + \frac{\delta^2}{1-\eta} \sum_{t=1}^T \left(\sum_{i=1}^n \frac{a_i}{(p_i^t)^{3/2}} \right)^2. \quad (2.48)$$

Recall that

$$p_j^t = (1-\eta) \frac{w_j^t}{W^t} + \eta z_j \quad (2.49)$$

in MABS3, where z_j is a fixed distribution over the datapoints. Because of (2.49), the right-hand side of (2.48) is upper bounded as

$$-\delta \sum_{t=1}^T r_j^t - \ln n \leq -\frac{\delta}{1-\eta} \sum_{t=1}^T \sum_{i=1}^n p_i^t r_i^t + \frac{\delta^2}{\eta^3(1-\eta)} T \left(\sum_{i=1}^n \frac{a_i}{(z_i)^{3/2}} \right)^2. \quad (2.50)$$

We choose z_i such that the upper bound in (2.50) is minimized, which is $z_i = a_i^{2/5}/\sum_{j=1}^n a_j^{2/5}$. By plugging $z_i = a_i^{2/5}/\sum_{j=1}^n a_j^{2/5}$ in (2.50), we find

$$-\delta \sum_{t=1}^T r_j^t - \ln n \leq -\frac{\delta}{1-\eta} \sum_{t=1}^T \sum_{i=1}^n p_i^t r_i^t + \frac{\delta^2}{\eta^3(1-\eta)} T \left(\sum_{i=1}^n a_i^{2/5} \right)^5. \quad (2.51)$$

By multiplying (2.51) by \hat{p}_j and summing over j , we get

$$-\delta \sum_{t=1}^T \sum_{j=1}^n \hat{p}_j r_j^t - \ln n \leq -\frac{\delta}{1-\eta} \sum_{t=1}^T \sum_{i=1}^n p_i^t r_i^t + \frac{\delta^2}{\eta^3(1-\eta)} T \left(\sum_{i=1}^n a_i^{2/5} \right)^5. \quad (2.52)$$

As $r_i^t = -a_i^t/(p_i^t)^2 = (\nabla_{\mathbf{p}} \mathbb{V}_e^t(\mathbf{p}^t))_i$, we have $\sum_{i=1}^n p_i r_i^t = \langle \mathbf{p}, \nabla_{\mathbf{p}} \mathbb{V}_e^t(\mathbf{p}^t) \rangle$ for any distribution \mathbf{p} , by plugging this in (2.52) and rearranging it, we find

$$\sum_{t=1}^T \langle \mathbf{p}^t - \hat{\mathbf{p}}, \nabla_{\mathbf{p}} \mathbb{V}_e^t(\mathbf{p}^t) \rangle + \eta \sum_{t=1}^T \langle \hat{\mathbf{p}}, \nabla_{\mathbf{p}} \mathbb{V}_e^t(\mathbf{p}^t) \rangle \leq \frac{1-\eta}{\delta} \ln n + \delta \frac{T}{\eta^3} \left(\sum_{i=1}^n a_i^{2/5} \right)^5. \quad (2.53)$$

Setting $\zeta = \eta$ in (2.18) and combining it with (2.53) gives

$$(1-2\eta) \sum_{t=1}^T \mathbb{V}_e^t(\mathbf{p}^t) - (1-\eta) \sum_{t=1}^T \mathbb{V}_e^t(\hat{\mathbf{p}}) \leq \frac{1-\eta}{\delta} \ln n + \delta \frac{T}{\eta^3} \left(\sum_{i=1}^n a_i^{2/5} \right)^5, \quad (2.54)$$

and by rearranging the terms of (4.10) and noting that $0 < \eta < 0.5$, we finally get

$$\sum_{t=1}^T \mathbb{V}_e^t(\mathbf{p}^t) \leq \frac{1-\eta}{1-2\eta} \sum_{t=1}^T \mathbb{V}_e^t(\hat{\mathbf{p}}) + \frac{1-\eta}{\delta(1-2\eta)} \ln n + \frac{\delta}{1-2\eta} \frac{T}{\eta^3} \left(\sum_{i=1}^n a_i^{2/5} \right)^5. \quad (2.55)$$

Setting $\delta = \sqrt{\eta^3 \ln n / \left(T \left(\sum_{i=1}^n a_i^{2/5} \right)^5 \right)}$ in (2.55) concludes the proof. The condition $T \geq \sqrt{\left(\sum_{i=1}^n a_i^{2/5} \right) / \min_i a_i^{1/5} \ln n}$ in Corollary 2.9 ensures that δ is small and $-\delta \hat{r}_i^t \leq 1$ which is needed to use $\exp(x) \leq 1 + x + x^2$ with $x = -\delta \hat{r}_i^t$ in the derivation of (2.44). \square

Table 2.1 summarizes the performance and running time of three versions of MABS (MABS in Theorem 2.6, MABS2 in Corollary 2.12 in Appendix 2.A.2, and MABS3 in Corollary 2.9). To ease comparison, we recast (2.27) and (2.43) under the more general form

$$\frac{1}{n^2} \sum_{t=1}^T \mathbb{V}_e^t(\mathbf{p}^t) \leq \text{approx} \cdot \frac{1}{n^2} \sum_{t=1}^T \mathbb{V}_e^t(\hat{\mathbf{p}}) + R, \quad (2.56)$$

where *approx* is the approximation factor and R is the regret, both of which are listed in Table 2.1 for the three versions of MABS. In uniform sampling and importance sampling

Table 2.1 – Performance of the different versions of MABS. In the performance metrics, approx is the approximation factor of the algorithms and running_time is the total computational complexity of the algorithms per iteration.

method	approx	Regret	running_time
MABS	$\frac{1-\eta}{1-2\eta}$	$O\left(\frac{2-\eta}{\eta^2(1-2\eta)}\sqrt{T\sum_{i=1}^n a_i^2 \ln n}\right)$	$O(\log n)$
MABS2	$\frac{1-\eta}{1-2\eta}$	$O\left(\frac{2-\eta}{\eta^2(1-2\eta)}\sqrt{T\frac{(\sum_{i=1}^n a_i^{2/5})^5}{n^4} \ln n}\right)$	$O(\log n)$
MABS3	$\frac{1-\eta}{1-2\eta}$	$O\left(\frac{2-\eta}{\eta^{1.5}(1-2\eta)}\sqrt{T\frac{(\sum_{i=1}^n a_i^{2/5})^5}{n^4} \ln n}\right)$	$O(n)$

methods, the approximation factor and R are unknown, hence they are not included in Table 2.1.

In a classic bandit problem, it is important to perform (in hindsight) as well as the optimal solution, i.e., to have approx = 1. Otherwise, the difference between the optimal solution and the solution found by a bandit algorithm is $O(T)$. In this work, however, recall that the end goal of minimizing the effective variance with a bandit algorithm is to have a better estimator for the gradient, and as a result to converge faster to the optimal coordinates θ^* . In the next section, we can guarantee the convergence of SGD, if η in (2.24) is a positive constant. Therefore, an approximation factor larger than 1 (approx > 1) is needed in this setting, yet it only appears as a constant factor in the final convergence rate, and it might be more important to have a smaller R and to find a good distribution fast, especially in a scenario where the norm of the gradients varies a lot from one iteration to the next.

2.5 Combining MABS with Stochastic Optimization Algorithms

In this section, we provide some intuition behind the improvement of the convergence rate brought by the reduction of the effective variance $\nabla_e^t(\mathbf{p}^t)$. We derive (in Section 2.5.1) the convergence guarantee for SGD in conjunction with MABS in order to highlight the impact of the effective variance $\nabla_e^t(\mathbf{p}^t)$ on it. In particular, let the *sub-optimality gap* be $\epsilon(\theta) = F(\theta) - F(\theta^*)$, i.e., the difference between the cost function F at coordinate θ and the minimum cost function F reached at the optimal coordinate θ^* . We show (in Section 2.5.1) that the convergence guarantee on $\epsilon(\theta)$ for SGD is linearly proportional to the expected sum of effective variances over T iterations $\mathbb{E}_{\theta^1, \dots, \theta^T} \left[\sum_{t=1}^T \nabla_e^t(\mathbf{p}^t) \right]$, and we derive the convergence rate of SGD in conjunction with MABS. In Section 2.5.2, we derive a general upper bound on the per-iteration convergence rate for first-order stochastic

optimization algorithms that use an unbiased estimator $\hat{g}(\boldsymbol{\theta}^t, \boldsymbol{\theta}^t)$ for the gradient $\nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta})$, that is reduced together with the effective variance $\mathbb{V}_e^t(\mathbf{p}^t)$. In this section, we use $\hat{g}(\boldsymbol{\theta}^t) = \hat{g}(\boldsymbol{\theta}^t, \mathbf{p}^t)$ and we drop the explicit dependence on \mathbf{p}^t .

2.5.1 SGD

We start with SGD and show that the convergence guarantee on $\epsilon(\boldsymbol{\theta}) = F(\boldsymbol{\theta}) - F(\boldsymbol{\theta}^*)$ is directly proportional to the expected cumulative effective variance $\mathbb{E}_{\boldsymbol{\theta}^1, \dots, \boldsymbol{\theta}^T} \left[\sum_{t=1}^T \mathbb{V}_e^t(\mathbf{p}^t) \right]$. In SGD, given $\boldsymbol{\theta}^t$, the unbiased estimator for the gradient $\nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta})$ is $\hat{g}(\boldsymbol{\theta}^t) = \nabla_{\phi_{i_t}(\boldsymbol{\theta}^t)}/n p_{i_t}$ and $\mathbb{V}_e^t(\mathbf{p}^t) = \mathbb{E}_{\mathbf{p}^t} [\|\hat{g}(\boldsymbol{\theta}^t)\|^2 | \boldsymbol{\theta}^t]$. In the seminal work of [Robbins and Monro, 1951], it has been shown that in order for SGD to converge the step size γ_t should be decreasing in time, and the step size should satisfy

$$\sum_{t=1}^T \gamma_t = \infty \quad \text{and} \quad \sum_{t=1}^T \gamma_t^2 < \infty.$$

We can use SGD in conjunction with MABS to minimize the cumulative effective variance $\mathbb{E}_{\boldsymbol{\theta}^1, \dots, \boldsymbol{\theta}^T} \left[\sum_{t=1}^T \mathbb{V}_e^t(\mathbf{p}^t) \right]$. However, as $\sum_{t=1}^T \mathbb{V}_e^t(\mathbf{p}^t)$ depends on the trajectory of the coordinates $\boldsymbol{\theta}^1, \dots, \boldsymbol{\theta}^T$, it is not clear what the convergence bound in (2.7) becomes, unless we upper bound $\sum_{t=1}^T \mathbb{V}_e^t(\mathbf{p}^t)$. In particular, if \mathbf{p}^t becomes close to 0, then $\mathbb{V}_e^t(\mathbf{p}^t)$ can be very large because of (2.5). Trivial bounds on $\mathbb{V}_e^t(\mathbf{p}^t)$ can be established by assuming that the gradients $\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^t)$ are bounded. This can be achieved by assuming that the optimal coordinate $\boldsymbol{\theta}^*$ is contained in a bounded set $\mathcal{X} \in \mathbb{R}^d$, and by adding a projection step to SGD in order to establish a bound on $\|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^t)\|$.

Here, we assume that ϕ_i s are L_i -smooth, and use the smoothness to bound $\|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^t)\|$ as is done in [Needell et al., 2014, Csiba et al., 2015, Schmidt et al., 2015a]. For simplicity, we assume that there is no regularizer in $F(\boldsymbol{\theta})$. Similar results hold when a convex regularizer $r(\boldsymbol{\theta})$ is added to the cost function $F(\boldsymbol{\theta})$.

Theorem 2.10. *Assume that $F(\boldsymbol{\theta})$ is μ -strongly convex and let each ϕ_i be convex and L_i -smooth. Also, assume that $\min_{\mathbf{p}} \mathbb{V}_e(\boldsymbol{\theta}^*, \mathbf{p}) > 0$. Then, if $\gamma_t = 2/\mu(t+t_0)$ in (2.2), the following convergence guarantee holds for any $T \geq 2.5 (\max_i(G_i)^2/(G^2)) n \ln n$ in SGD with MABS:*

$$\begin{aligned} \mathbb{E} \left[\|\boldsymbol{\theta}^{T+1} - \boldsymbol{\theta}^*\|^2 \right] &\leq \frac{24}{\mu^2 n^2 T} \left(\sum_{i=1}^n \|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^*)\| \right)^2 + \frac{20}{\mu^2 n T^2} \alpha \sum_{i=1}^n L_i + \frac{t_0^2}{T^2} \|\boldsymbol{\theta}^0 - \boldsymbol{\theta}^*\|^2 \\ &\quad + \frac{200 t_0}{\mu^2 T^2} \sqrt{2 \ln n \left(T \left(\sum_{i=1}^n \|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^*)\|^2 \right) + \alpha \sum_{i=1}^n L_i \right)} \quad (2.57) \\ &= O \left(\frac{1}{\mu^2 T} \min_{\mathbf{p}} \mathbb{V}_e(\boldsymbol{\theta}^*, \mathbf{p}) \right) = O \left(\frac{1}{\mu^2 n^2 T} \left(\sum_{i=1}^n \|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^*)\| \right)^2 \right), \end{aligned}$$

Chapter 2. Stochastic Gradient Descent with Bandit Sampling

for some $G_i \geq \max_{\{\boldsymbol{\theta}^1, \dots, \boldsymbol{\theta}^t\}} \|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^t)\|^2$, where the expectations are over the sequence of the updated coordinates $\{\boldsymbol{\theta}^t\}_{t \in [T]}$, $t_0 \geq \max\{1, 4 \sup L_i / \mu\}$, $\alpha = \mu^2 t_0 \|\boldsymbol{\theta}^0 - \boldsymbol{\theta}^*\|^2 + 20\sigma_u^2 / \mu^2 \log(T + t_0)$ and $\sigma_u^2 = \mathbb{E}_{i \sim U[1, n]} [\|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^*)\|^2]$.

The constants in (2.57) are not optimized and better rates can be found by tuning η in (2.24). The first term in (2.57) scales as $O(1/T)$, the second term scales as $O(\log T / T^2)$, the third term scales as $O(1/T^2)$ and the last term scales as $O(1/T^{3/2})$. So asymptotically for large T , the first term is the most important term in the convergence rate, which is asymptotically

$$\mathbb{E} [\|\boldsymbol{\theta}^{T+1} - \boldsymbol{\theta}^*\|^2] = O\left(\frac{1}{\mu^2 T} \min_{\mathbf{p}} \mathbb{V}_e(\boldsymbol{\theta}^*, \mathbf{p})\right) = O\left(\frac{1}{\mu^2 n^2 T} \left(\sum_{i=1}^n \|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^*)\|\right)^2\right),$$

and which is of the same order as the fastest convergence rate (2.11). The full proof of Theorem 2.10 is presented in Appendix 2.A.3. First, in Lemma 2.13 in Appendix 2.A.3 we show that $\boldsymbol{\theta}^t$ converges to $\boldsymbol{\theta}^*$ when SGD with $p_i^t > 0.4/n$ is used, so that the conditions of Lemma 2.3 hold. The analysis of the theorem is then based on showing that the convergence rate depends linearly on the cumulative effective variance $\sum_{t=1}^T \mathbb{V}_e^t(\mathbf{p}^t)$. Therefore, by using MABS and the results of Lemma 2.13 we can directly bound the sum $\sum_{t=1}^T \mathbb{V}_e^t(\mathbf{p}^t)$.

In addition, when the effective variance $\mathbb{V}_e^t(\mathbf{p}^t)$ is small (meaning that (2.3) is a good estimator), we expect a more stable algorithm, i.e., we can choose a larger step size γ_t without diverging. Assume that the cost function $F(\boldsymbol{\theta})$ is L -smooth. Using the smoothness property (see Definition 2.C in Appendix where $h(\cdot) = F(\cdot)$, $y = \boldsymbol{\theta}^{t+1}$ and $x = \boldsymbol{\theta}^t$), we get

$$F(\boldsymbol{\theta}^{t+1}) - F(\boldsymbol{\theta}^t) \leq \left\langle \nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta}^t), \boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^t \right\rangle + \frac{L}{2} \|\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^t\|^2. \quad (2.58)$$

Plugging the update rule (2.2) of SGD in (2.58) yields

$$F(\boldsymbol{\theta}^{t+1}) - F(\boldsymbol{\theta}^t) \leq -\gamma \left\langle \nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta}^t), \frac{\nabla_{\boldsymbol{\theta}} \phi_{i_t}(\boldsymbol{\theta}^t)}{np_{i_t}^t} \right\rangle + \gamma^2 \frac{L}{2} \left\| \frac{\nabla_{\boldsymbol{\theta}} \phi_{i_t}(\boldsymbol{\theta}^t)}{np_{i_t}^t} \right\|^2. \quad (2.59)$$

By taking expectations over \mathbf{p}^t , conditionally on $\boldsymbol{\theta}^t$, we obtain

$$\mathbb{E}_{\mathbf{p}^t} [F(\boldsymbol{\theta}^{t+1}) | \boldsymbol{\theta}^t] - F(\boldsymbol{\theta}^t) \leq -\gamma \|\nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta}^t)\|^2 + \gamma^2 \frac{L}{2} \mathbb{V}_e^t(\mathbf{p}^t).$$

To guarantee that the cost function F decreases (in expectation), we need to have $\gamma \leq 2\|\nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta}^t)\|^2 / (L \cdot \mathbb{V}_e^t(\mathbf{p}^t))$. Therefore, by lowering $\mathbb{V}_e^t(\mathbf{p}^t)$, we can afford a larger step size γ .

2.5. Combining MABS with Stochastic Optimization Algorithms

A similar analysis holds for PSGD, that updates $\boldsymbol{\theta}$ according to

$$\boldsymbol{\theta}^{t+1} = \arg \min_{\boldsymbol{\theta}} \left[\langle \nabla_{\boldsymbol{\theta}} \phi_{i_t}(\boldsymbol{\theta}^t), \boldsymbol{\theta} \rangle + \lambda r(\boldsymbol{\theta}) + \frac{1}{\gamma_t} \mathcal{B}_{\psi}(\boldsymbol{\theta}, \boldsymbol{\theta}^t) \right],$$

where $r(\boldsymbol{\theta})$ is a convex regularizer. We defer the explanation of PSGD to Appendix 2.B. Deriving the convergence results for SVRG and SAGA in conjunction with MABS is left for future work, below we describe why reducing the variance with a non-uniform sampling distribution should improve the convergence rate of first-order optimization algorithms (such as SVRG and SAGA).

2.5.2 First-order Algorithms

Let $\boldsymbol{\theta}^t$ be the coordinate at the current iteration t , and $\boldsymbol{\theta}^{t+1}$ be the coordinate at the next iteration $t + 1$ that is reached by using a first-order optimization algorithm. Let us also consider a more general cost function

$$F(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^n \phi_i(\boldsymbol{\theta}) + \lambda r(\boldsymbol{\theta}),$$

where $r(\cdot)$ is a convex regularizer. Consider the following updating rule

$$\boldsymbol{\theta}^{t+1} = \text{prox}_{\gamma}^{\lambda r} \left(\boldsymbol{\theta}^t - \gamma \hat{g}(\boldsymbol{\theta}^t) \right),$$

where $\hat{g}(\boldsymbol{\theta})$ is an unbiased estimator for the gradient $g(\boldsymbol{\theta}) = \sum_{i=1}^n \nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta})/n$, where γ is the step size, and where

$$\text{prox}_{\gamma}^{\lambda r}(y) = \arg \min_x \left\{ \frac{1}{2\gamma} \|x - y\|^2 + \lambda r(x) \right\} \quad (2.60)$$

is the proximal operator. We can then upper bound $\|\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^*\|^2$ by using a similar technique as in [Defazio et al., 2014]

$$\begin{aligned} \|\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^*\|^2 &= \left\| \text{prox}_{\gamma}^{\lambda r} \left(\boldsymbol{\theta}^t - \gamma \hat{g}(\boldsymbol{\theta}^t) \right) - \text{prox}_{\gamma}^{\lambda r} \left(\boldsymbol{\theta}^* - \gamma g(\boldsymbol{\theta}^*) \right) \right\|^2 \\ &\leq \left\| \boldsymbol{\theta}^t - \boldsymbol{\theta}^* + \gamma \left(g(\boldsymbol{\theta}^*) - \hat{g}(\boldsymbol{\theta}^t) \right) \right\|^2, \end{aligned} \quad (2.61)$$

where the inequality follows from the non-expansiveness of the proximal operator, i.e., $\left\| \text{prox}_{\gamma}^{\lambda r}(x) - \text{prox}_{\gamma}^{\lambda r}(y) \right\|^2 \leq \|x - y\|^2$ for any x and $y \in \mathbb{R}^d$. Next, by taking expectations of the right-hand side of (2.61) over \boldsymbol{p}^t , conditionally on $\boldsymbol{\theta}^t$, we have

$$\begin{aligned} \mathbb{E} \left[\left\| \boldsymbol{\theta}^t - \boldsymbol{\theta}^* + \gamma \left(g(\boldsymbol{\theta}^*) - \hat{g}(\boldsymbol{\theta}^t) \right) \right\|^2 \middle| \boldsymbol{\theta}^t \right] &= \|\boldsymbol{\theta}^t - \boldsymbol{\theta}^*\|^2 + \gamma^2 \|g(\boldsymbol{\theta}^*)\|^2 + \gamma^2 \mathbb{E} \left[\|\hat{g}(\boldsymbol{\theta}^t)\|^2 \middle| \boldsymbol{\theta}^t \right] \\ &\quad + 2\gamma (\boldsymbol{\theta}^t - \boldsymbol{\theta}^*)^\top \left(g(\boldsymbol{\theta}^*) - g(\boldsymbol{\theta}^t) \right) - 2\gamma^2 g(\boldsymbol{\theta}^*)^\top g(\boldsymbol{\theta}^t), \end{aligned} \quad (2.62)$$

because $\mathbb{E} [\hat{g}(\boldsymbol{\theta}^t) | \boldsymbol{\theta}^t] = g(\boldsymbol{\theta}^t)$. Combining (2.61) and (2.62), we get

$$\begin{aligned} & \mathbb{E} \left[\left\| \boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^* \right\|^2 \middle| \boldsymbol{\theta}^t \right] - \left\| \boldsymbol{\theta}^t - \boldsymbol{\theta}^* \right\|^2 \leq \\ & \gamma^2 \|g(\boldsymbol{\theta}^*)\|^2 + \gamma^2 \mathbb{E} \left[\|\hat{g}(\boldsymbol{\theta}^t)\|^2 \middle| \boldsymbol{\theta}^t \right] + 2\gamma (\boldsymbol{\theta}^t - \boldsymbol{\theta}^*)^\top \left(g(\boldsymbol{\theta}^*) - g(\boldsymbol{\theta}^t) \right) - 2\gamma^2 g(\boldsymbol{\theta}^*)^\top g(\boldsymbol{\theta}^t). \end{aligned} \quad (2.63)$$

From (2.63), it is clear that, given $\boldsymbol{\theta}^t$, all terms except $\mathbb{E} [\|\hat{g}(\boldsymbol{\theta}^t)\|^2 | \boldsymbol{\theta}^t]$ are constant with respect to \mathbf{p}^t . In many optimization algorithms such as SGD, PSGD, SAGA, and SVRG the effective variance $\mathbb{V}_e^t(\mathbf{p}^t) = \mathbb{E} [\|\hat{g}(\boldsymbol{\theta}^t)\|^2 | \boldsymbol{\theta}^t] + \mathbb{V}_g \left(\{i_\ell, \boldsymbol{\theta}^\ell\}_{1 \leq \ell \leq t} \right)$, where $\mathbb{V}_g \left(\{i_\ell, \boldsymbol{\theta}^\ell\}_{1 \leq \ell \leq t} \right)$ is a function of the history of the algorithm (past coordinates $\boldsymbol{\theta}^\ell$ and sampled datapoints i_ℓ for $1 \leq \ell \leq t$), but not of \mathbf{p}^t . Therefore, to bring $\boldsymbol{\theta}^{t+1}$ closer in expectations to $\boldsymbol{\theta}^*$, minimizing $\mathbb{V}_e^t(\mathbf{p}^t)$ over \mathbf{p}^t amounts to minimizing $\mathbb{E} [\|\hat{g}(\boldsymbol{\theta}^t)\|^2 | \boldsymbol{\theta}^t]$. As a result, by minimizing $\mathbb{V}_e^t(\mathbf{p}^t)$ (hence $\mathbb{E} [\|\hat{g}(\boldsymbol{\theta}^t)\|^2 | \boldsymbol{\theta}^t]$) in (2.63) we can afford a larger step size γ without diverging. In Section 2.6.4, the stability of the SGD, SAGA and SVRG in conjunction with MABS, for a range of step sizes γ , is tested, and we observe a significant improvement in the stability compared to the corresponding algorithms with uniform sampling.

2.6 Empirical Evaluation

We evaluate the performance of MABS in conjunction with several stochastic optimization algorithms and address the question: *How much can bandit-based sampling improve the convergence rate?* Towards this goal, we compare the performance of several stochastic optimization algorithms that use uniform sampling, importance sampling (IS) and MABS. We conduct two types of experiments.

In the first one, in Section 2.6.2, we create a set of synthetic datasets. Each dataset has a different ratio τ of maximum smoothness $L_m = \max_{i \in [n]} \{L_i\}$ to the average-smoothness $\bar{L} = \sum_{i=1}^n L_i/n$, where L_i is the smoothness parameter of the sub-cost function ϕ_i in (2.1). We refer to the ratio $\tau = L_m/\bar{L}$ as *smoothness ratio*. It is observed in several works (e.g., [Zhao and Zhang, 2015a]) that a non-uniform sampling technique speeds up the convergence rate more for a dataset with larger τ . With a synthetic dataset, we can vary τ and compare the convergence rate of different sampling methods for a range of values of τ .

In the second one, in Section 2.6.3, we compare the convergence rate of different sampling methods on real datasets to observe how the algorithms works in practice. We address the question: *How stable is a stochastic optimization algorithm in conjunction with different sampling methods?* To answer this question, in Section 2.6.4, we vary the step size (a.k.a., learning rate) in the stochastic optimization algorithms and find empirically the maximum step size for different sampling methods for which the stochastic optimization algorithm

sill converges to the optimum coordinates $\boldsymbol{\theta}^*$. Finally, we address the question: *How much can bandit-based sampling improve the convergence rate in terms of the wall-clock time?* Towards this goal in Section 2.6.5, we report the convergence rate of SGD with different sampling methods as a function of the wall-clock time, instead of epochs.

2.6.1 Experimental Setup

In this paper, we consider a broader class of stochastic optimization algorithms, which includes not only SGD but also other optimization algorithms that use

$$\hat{g}(\boldsymbol{\theta}^t, \mathbf{p}^t) = \frac{b_{i_t}(\boldsymbol{\theta}^t)}{p_{i_t}^t} + \bar{g}(\{i_\ell, \boldsymbol{\theta}^\ell\}_{1 \leq \ell \leq t}) \quad (2.64)$$

as an *unbiased* estimator for $\nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta}^t)$, where $b_{i_t}(\boldsymbol{\theta}^t)$ is a function of $\boldsymbol{\theta}^t$ but not of \mathbf{p}^t , and where $\bar{g}(\{i_\ell, \boldsymbol{\theta}^\ell\}_{1 \leq \ell \leq t})$ is a function of the history of the algorithm (past coordinates $\boldsymbol{\theta}^\ell$ and sampled datapoints i_ℓ for $1 \leq \ell \leq t$), but not of \mathbf{p}^t . The exact expressions of $b_{i_t}(\cdot)$ and $\bar{g}(\cdot)$ depend on the stochastic optimization algorithm (we present some examples next) and the cost function F . For example, if a smooth convex regularizer $\lambda r(\boldsymbol{\theta})$ is used in the cost function, then $b_{i_t} = \nabla_{\boldsymbol{\theta}} \phi_{i_t}(\boldsymbol{\theta}^t)/n$ and $\bar{g}(\{i_\ell, \boldsymbol{\theta}^\ell\}_{1 \leq \ell \leq t}) = \lambda \nabla_{\boldsymbol{\theta}} r(\boldsymbol{\theta}^t)$ in SGD. This class includes not only SGD, but also PSGD, SVRG, and SAGA, and is characterized by the property that the variance $\mathbb{V}(\boldsymbol{\theta}^t, \mathbf{p}^t)$ of $\hat{g}(\boldsymbol{\theta}^t, \mathbf{p}^t)$ can be written as in (2.4), as the difference of the effective variance \mathbb{V}_e that takes the form

$$\mathbb{V}_e(\mathbf{p}^t) = \sum_{i=1}^n \frac{a_i^t}{p_i^t}, \quad (2.65)$$

where

$$a_i^t = \|b_i(\boldsymbol{\theta}^t)\|^2, \quad (2.66)$$

and of a term $\mathbb{V}_c(\{i_\ell, \boldsymbol{\theta}^\ell\}_{1 \leq \ell \leq t})$ that does not depend on \mathbf{p}^t . If $\hat{g}(\boldsymbol{\theta}^t, \mathbf{p}^t)$ is given by (2.3) (as in SGD), then the effective variance $\mathbb{V}_e^t(\mathbf{p}^t)$ is given by (2.5) and

$$a_i^t = \frac{1}{n^2} \|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^t)\|^2$$

in (2.65). We first define the appropriate unbiased estimator $\hat{g}(\boldsymbol{\theta}^t)$ for $\nabla_{\boldsymbol{\theta}} F(\boldsymbol{\theta}^t)$ and a_i^t in (2.65) for each algorithm. As the goal of the experiments is to show the advantage of MABS over other sampling methods, in each experiment, we fix the optimization algorithm and change the sampling method. In particular, we assume that the cost function

$$F(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^n \phi_i(\boldsymbol{\theta}) + \lambda r(\boldsymbol{\theta}),$$

where $r(\boldsymbol{\theta})$ is a smooth convex regularizer. We compare the following algorithms and present below the necessary definitions for MABS:

- **Stochastic Gradient Descent (SGD):**

$$\hat{g}(\boldsymbol{\theta}^t) = \frac{\nabla_{\boldsymbol{\theta}}\phi_{i_t}(\boldsymbol{\theta}^t)}{np_{i_t}^t} + \lambda\nabla_{\boldsymbol{\theta}}r(\boldsymbol{\theta}^t) \quad \text{and} \quad a_i^t = \frac{1}{n^2} \left\| \nabla_{\boldsymbol{\theta}}\phi_i(\boldsymbol{\theta}^t) \right\|^2.$$

- **Stochastic Variance-Reduced Gradient (SVRG):**

$$\hat{g}(\boldsymbol{\theta}^t) = \frac{\nabla_{\boldsymbol{\theta}}\phi_{i_t}(\boldsymbol{\theta}^t) - \nabla_{\boldsymbol{\theta}}\phi_{i_t}(\hat{\boldsymbol{\theta}})}{np_{i_t}^t} + \frac{\sum_{i=1}^n \nabla_{\boldsymbol{\theta}}\phi_i(\hat{\boldsymbol{\theta}})}{n} + \lambda\nabla_{\boldsymbol{\theta}}r(\boldsymbol{\theta}^t)$$

and

$$a_i^t = \frac{1}{n^2} \left\| \nabla_{\boldsymbol{\theta}}\phi_i(\boldsymbol{\theta}^t) - \nabla_{\boldsymbol{\theta}}\phi_i(\hat{\boldsymbol{\theta}}) \right\|^2,$$

where $\hat{\boldsymbol{\theta}}$ is defined as follows. Time is divided into bins of size n , and at the beginning of each bin c (in the c^{th} bin $cn \leq t < (c+1)n$) $\hat{\boldsymbol{\theta}}$ is updated as

$$\hat{\boldsymbol{\theta}} = \frac{\sum_{\ell=(c-1)n}^{cn} \boldsymbol{\theta}^\ell}{n},$$

see [Xiao and Zhang, 2014] for more details and [Allen-Zhu and Yuan, 2016, Kern and György, 2016] for improved versions of the algorithm.

- **SAGA:**

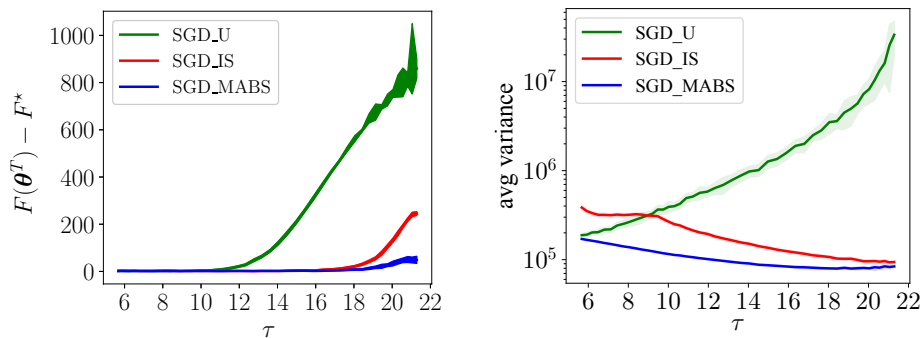
$$\hat{g}(\boldsymbol{\theta}^t) = \frac{\nabla_{\boldsymbol{\theta}}\phi_{i_t}(\boldsymbol{\theta}^t) - \nabla_{\boldsymbol{\theta}}\phi_{i_t}(\hat{\boldsymbol{\theta}}_{i_t})}{np_{i_t}^t} + \frac{\sum_{i=1}^n \nabla_{\boldsymbol{\theta}}\phi_i(\hat{\boldsymbol{\theta}}_i)}{n} + \lambda\nabla_{\boldsymbol{\theta}}r(\boldsymbol{\theta}^t)$$

and

$$a_i^t = \frac{1}{n^2} \left\| \nabla_{\boldsymbol{\theta}}\phi_i(\boldsymbol{\theta}^t) - \nabla_{\boldsymbol{\theta}}\phi_i(\hat{\boldsymbol{\theta}}_i) \right\|^2,$$

where $\nabla_{\boldsymbol{\theta}}\phi_i(\hat{\boldsymbol{\theta}}_i)$ is the gradient of the sub-cost function ϕ_i at the last time that datapoint i was chosen (see Defazio et al. [2014] for more details).

For each stochastic optimization algorithm, we use three sampling methods: (1) uniform sampling (denoted by suffix `_U`), (2) IS (denoted by suffix `_IS`), and (3) MABS (denoted by suffix `_MABS`). If the regularizer $\lambda r(\boldsymbol{\theta})$ is not smooth (as for L_1 -penalized logistic regression in Section 2.6.4), we will use the proximal operator to update the coordinates $\boldsymbol{\theta}^t$.



(a) The difference between the cost function $F(\theta^T)$ found by SGD with different sampling methods and the cost function F^* found by gradient descent method for different smoothness ratios τ .

(b) The average effective variance $\sum_{t=1}^T \nabla_e^t(\theta^t)/T$ for different smoothness ratios τ .

Figure 2.2 – We study SGD for minimizing mean squared error with different sampling methods by comparing the convergence and the effective variance as a function of smoothness ratio $\tau = \max_i\{L_i\}/\sum_{j=1}^n L_j/n$ (a measure of the dissimilarity of the $\nabla_{\theta}\phi_i$ s). We observe that both are lowest when MABS is used. The standard deviation is also depicted in the plots.

2.6.2 Empirical Results for Different Smoothness Ratios τ

As discussed in Section 2.1, the benefit of MABS (and of non-uniform sampling more generally) will depend on how *dissimilar* the $\nabla_{\theta}\phi_i$ s are. Recall that the smoothness ratio τ is the ratio between the maximum-smoothness $L_m = \max_{i \in [n]}\{L_i\}$ and the average-smoothness $\bar{L} = \sum_{i=1}^n L_i/n$. As observed in [Zhao and Zhang, 2015a], when the smoothness ratio τ is large, we expect non-uniform sampling (and in particular MABS) to be more advantageous. To study this effect, we present results on synthetic datasets with different smoothness ratios τ using SGD_U, SGD_IS, and SGD_MABS.⁴

Dataset. The datasets have $n = 101$ datapoints and $d = 5$ features.⁵ The labels are defined to be $y_i \triangleq \langle \mathbf{x}_i, \boldsymbol{\beta} \rangle + N_i$, where $\boldsymbol{\beta} \in \mathbb{R}^5$ is the coefficient of the hyperplane generated from a Gaussian distribution with mean 0 and standard deviation 10, and N_i is a Gaussian noise with mean 0 and variance 1. The features $\mathbf{x}_i \in \mathbb{R}^5$ are generated from a Gaussian distribution whose mean and variance are generated randomly. In order to obtain different smoothness ratios τ , we choose the datapoint m with the largest smoothness L_m and multiply its entire feature vector \mathbf{x}_m by a number $c > 1$, whereas all labels and all other features remain fixed. This increases L_m , hence the smoothness ratio τ and we scale down the learning rate by c for all algorithms. The sub-cost function

⁴In SGD_IS, the sampling distribution is $p_i = L_i/(\sum_{j=1}^n L_j)$ (see [Zhao and Zhang, 2015a]).

⁵Similar results are obtained for different values of n and d .

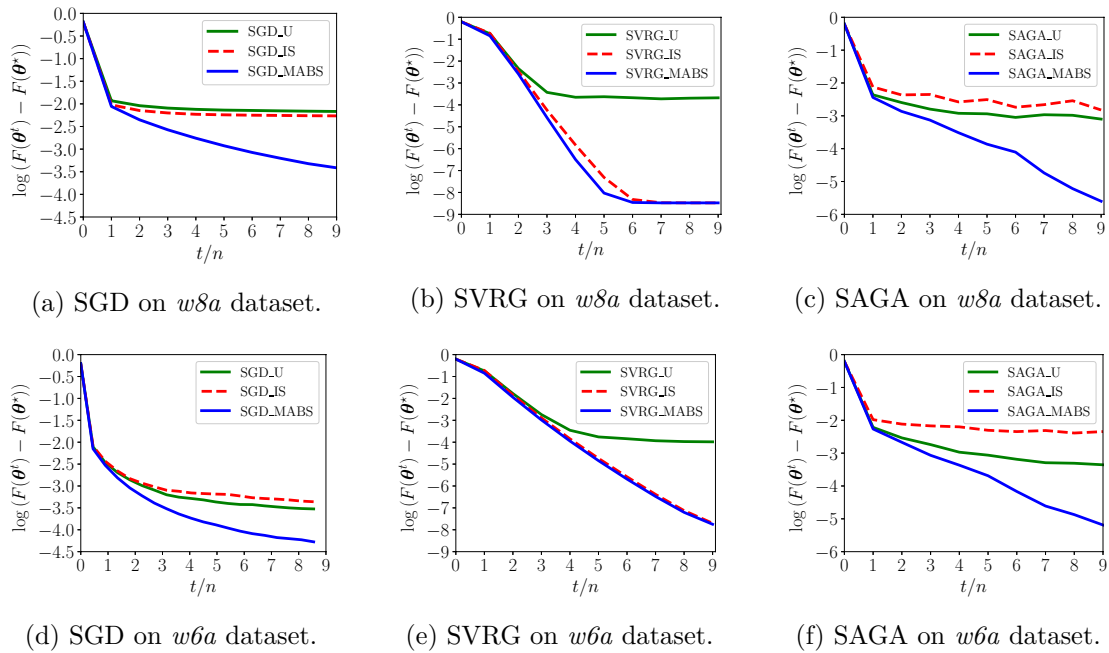


Figure 2.3 – Comparison of three different stochastic optimization algorithms (SGD, SVRG, and SAGA) on two datasets (*w8a* and *w6a*) when using different sampling methods. MABS is never suboptimal and often significantly outperforms the other sampling methods.

used here is $\phi_i(\boldsymbol{\theta}) = (\langle \mathbf{x}_i, \boldsymbol{\theta} \rangle - y_i)^2/2$, i.e., mean square error. Each experiment is run for 2000 epochs and repeated $k = 20$ times. We report the difference of values $F(\boldsymbol{\theta}^t)$ found by three sampling versions of SGD and the value F^* found by gradient descent at the final iteration T , this way we compare the stochastic algorithms (SGD_U, SGD_IS, and SGD_MABS) to the ideal gradient descent algorithm.

Results. In Figure 2.2a, we observe that MABS has the best performance of all three sampling methods as the value of $F(\boldsymbol{\theta}^t)$ for SGD_MABS is the closest to F^* for all smoothness ratios τ . Additionally, as the smoothness ratio τ increases, the performance of SGD_MABS further improves, thus confirming that when there is a datapoint with large gradient, the convergence of MABS to the optimal sampling distribution is faster. As expected, the performance of SGD_U degrades in τ , as SGD_U needs more iterations to converge. As SGD_IS appears to be less affected by τ , the advantage of MABS over IS is stronger for large τ . Figure 2.2b depicts the average effective variance $\sum_{t=1}^T \mathbb{V}_e^t(\boldsymbol{\theta}^t)/T$ as a function of the smoothness ratio τ , and similar observations can be made. In particular, the average effective variance of SGD_MABS is the lowest, and the average effective variance of SGD_MABS and SGD_IS are decreasing in τ , whereas the effective variance of SGD_U is increasing in τ .

Table 2.2 – Statistics of the datasets used in the experiments.

Dataset	n	d	τ
synthetic	101	5	3.7-83.9
w6a	17188	300	9.08
w8a	49749	300	9.79
ijcnn1	49990	22	2.61

2.6.3 Empirical Results on Real-World Data

We consider two binary classification datasets, *w8a* with the smoothness ratio $\tau = 9.8$ and *w6a* with the smoothness ratio $\tau = 9.1$ from [Chang and Lin, 2011] (see Table 2.2). For each of SGD, SVRG, and SAGA, we compare the effect of different sampling methods. We report the log of the sub-optimality gap ($\log \epsilon(\boldsymbol{\theta}) = \log (F(\boldsymbol{\theta}^t) - F(\boldsymbol{\theta}^*))$) reached by the three sampling versions of stochastic optimization algorithms, as a function of the number of iterations t .

First, as a cost function $F(\boldsymbol{\theta})$ we use L_2 -penalized logistic regression ($r(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_2^2$ and $\phi_i(\boldsymbol{\theta}) = \log(1 + \exp(-y_i \langle \mathbf{x}_i, \boldsymbol{\theta} \rangle))$) with the regularization parameter $\lambda = 10^{-4}$. The regularization parameter λ is chosen such that the test error and the train error are comparable. Each experiment is run for $T = 10n$ iterations and repeated 100 times. In the experiments for SGD, the step size $\gamma = 1/\max_i\{L_i\}$ (recall that L_i is the smoothness of ϕ_i); this choice of step size performs the best in our experiments). In the experiments for SAGA, we choose the larger step size $\gamma = 1/(3\bar{L})$ as in [Hofmann et al., 2015], where \bar{L} is the average smoothness. In the experiments for SVRG, the step size $\gamma = 1/\bar{L}$ as in [Xiao and Zhang, 2014]. In SGD and SAGA, using MABS benefits more as it consistently improves the convergence rates (see Figures 2.3a, 2.3d, 2.3c, and 2.3f). In SVRG, MABS improves the convergence over uniform sampling, and the convergence rates of MABS and IS are similar (see Figures 2.3b and 2.3e).

2.6.4 Stability

Following the discussion in Section 2.5, we study the robustness of SGD, SVRG, and SAGA when using a large step-size γ . In particular, we consider the *w8a* dataset and L_1 -penalized logistic regression. We use the proximal operator (2.60) to update the coordinates $\boldsymbol{\theta}^t$. We run different stochastic optimization algorithms in conjunction with different sampling methods and with different step sizes γ , and we collect the value $F(\boldsymbol{\theta}^t)$ at final iteration $T = 60n$. Each experiment is repeated 50 times.

The results are depicted in Figure 2.4 and show that MABS is indeed a more robust sampling method: SGD_MABS is able to find the optimal coordinate $\boldsymbol{\theta}^*$ with step sizes up to $\gamma = 5$ (see Figure 2.4a), whereas SGD_U and SGD_IS diverge after $\gamma = 0.5$. In

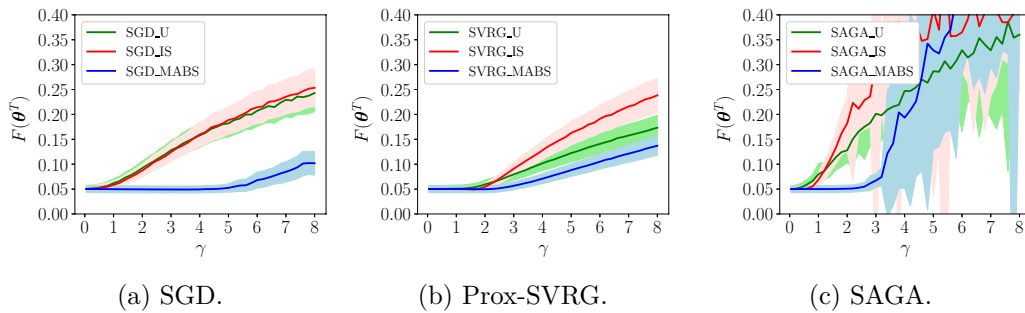


Figure 2.4 – Comparison of three different stochastic optimization algorithms (SGD, SVRG, and SAGA) on *w8a* dataset when using different sampling methods and different step sizes γ . The standard deviation is also depicted in the plots. MABS significantly outperforms the other methods and is able to find the optimal value even for a large γ .

Figure 2.4b, the difference between the three sampling methods is less significant, but SVRG_MABS still slightly outperforms the others. SAGA_MABS is also more robust than SAGA with other sampling methods, and it is able to find the optimal coordinate θ^* with step sizes up to $\gamma = 2.5$, whereas SAGA_U and SAGA_IS diverge after $\gamma = 0.7$ (see Figure 2.4c).

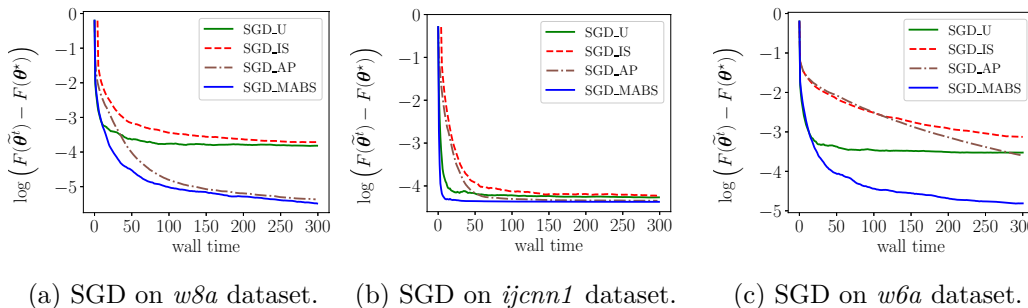


Figure 2.5 – Comparison of the convergence of SGD when using different sampling methods as a function of the wall-clock time in seconds.

2.6.5 Training Time

We note that adding MABS does not cost much with respect to training time. For example, given high-dimensional data with $d = 4000$ and $n = 50000$, we find empirically that SGD_MABS uses only 10% more clock-time than SGD_U.

We compare the convergence of SGD, SVRG, and SAGA as a function of the wall-clock time in seconds. For SGD, in addition to the uniform and importance samplings, we consider the adaptive sampling method in [Papa et al., 2015]. In the adaptive sampling method (AP), at the beginning of an epoch, the full gradient $\nabla_{\theta} \phi_i(\theta^t)$ is computed, then the sampling probability \mathbf{p} is computed according to (2.6) and is used for sampling the

datapoints in that epoch. We test the algorithms with different sampling methods and with weighted averaging as in [Lacoste-Julien et al., 2012] and Theorem 2.10, i.e.,

$$\tilde{\theta}^t = \frac{2}{t(t+1)} \sum_{j=1}^t j \cdot \theta^j. \quad (2.67)$$

The cost function $F(\theta)$ is the same as the cost function in Section 2.6.3 (i.e., L_2 -penalized logistic regression). The simulations are done in Python on an OS X with 2.9 GHz Intel Core i5 processor and 16 GB 2133 MHz LPDDR3 memory. The results are shown in Figures 3.4 and 2.6. As shown in Figure 3.4, MABS and AP sampling methods converge faster than other algorithms. The difference on *w8a* and *w6a* datasets is larger than on the *ijcnn1* dataset because τ is larger in *w8a* and *w6a* (it is 9 in *w8a* and *w6a* compared to 2.6 in *ijcnn1*), thus non-uniform sampling is more effective for *w8a* and *w6a* dataset. Figure 2.6 also shows that MABS is a better sampling method, and improves the convergence speed. We also see that the running average (2.67) improves the convergence rate compared to the results with θ^t in Figure 2.3.

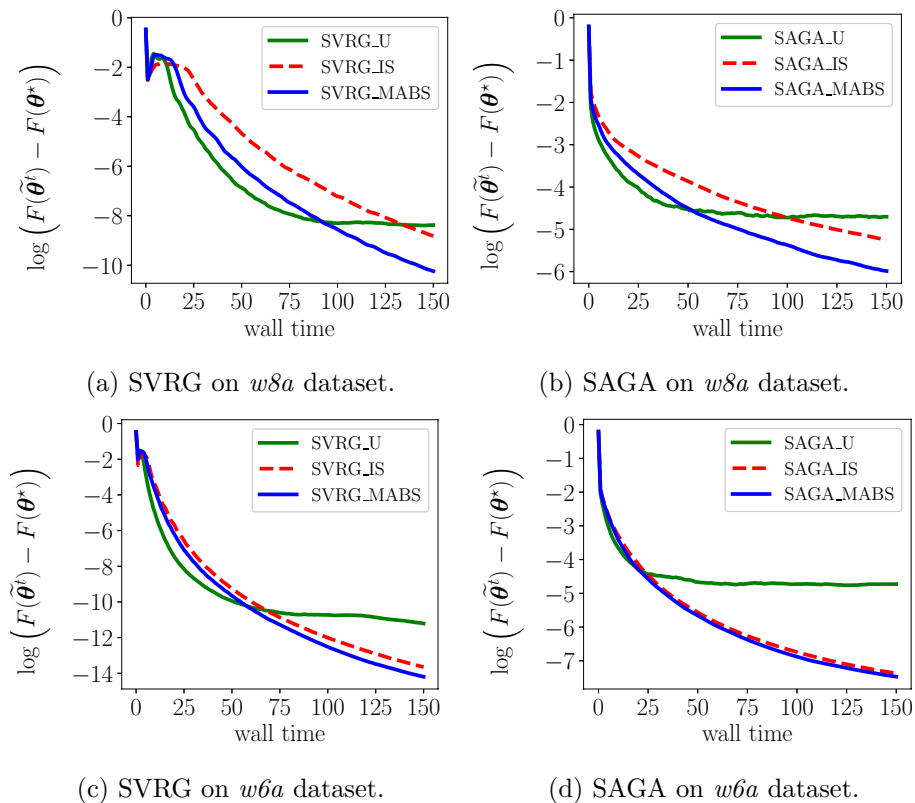


Figure 2.6 – Comparison of the convergence of SGD when using different sampling methods as a function of the wall-clock time in seconds.

2.7 Summary

In this chapter, a novel sampling method (called MABS) is presented to reduce the variance of gradient estimation. The method is inspired by multi-armed bandit algorithms (in particular EXP3) and does not require any preprocessing. First, the variance of the unbiased estimator of the gradient at iteration t is defined as a function of the sampling distribution \mathbf{p}^t and of the gradients of sub-cost functions $\nabla_{\boldsymbol{\theta}}\phi_i(\boldsymbol{\theta}^t)$. Next, using the past information, MABS minimizes this cost function by appropriately updating the distribution \mathbf{p}^t , and learns the optimal distribution given the set of selected datapoints $\{i_t\}_{1 \leq t \leq T}$ and gradients $\{\nabla_{\boldsymbol{\theta}}\phi_{i_t}(\boldsymbol{\theta}^t)\}_{1 \leq t \leq T}$. We have shown that under a natural assumption (bounded gradients) MABS can asymptotically approximate the optimal variance within a factor of 3. Moreover, MABS combined with three stochastic optimization algorithms (SGD, Prox_SVRG, and SAGA) is tested on real data. We observe its effectiveness on variance reduction and the rate of convergence of these optimization algorithms as compared to other sampling approaches. Furthermore, MABS is tested on synthetic datasets, and its effectiveness is observed for a large range of τ (i.e., the ratio of maximum smoothness to the average smoothness). It is also observed that SGD_MABS is significantly more stable than SGD with other sampling methods. Several important directions remain open. First, one would like to improve the constants in the bound in Theorem 2.6. Secondly, although we observe robustness, finding the optimal step size γ for Prox_SVRG and SAGA remains open. Lastly, it could be of interest to extend the work to other stochastic optimization methods, both by providing theoretical guarantees and observing their performance in practice.

Appendix

2.A Proofs

2.A.1 Omitted Proofs

Proof of Theorem 2.1 The starting point is the inequality

$$t \left(\mathbb{E}[F(\boldsymbol{\theta}^t)] - F(\boldsymbol{\theta}^*) \right) \leq \frac{\mathbb{E}[\mathbb{V}_e^t(\mathbf{p}^t)]}{\mu} + \frac{\mu}{4} \left(t(t-1) \mathbb{E} \left[\|\boldsymbol{\theta}^t - \boldsymbol{\theta}^*\|^2 \right] - t(t+1) \mathbb{E} \left[\|\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^*\|^2 \right] \right) \quad (2.68)$$

where the expectations are over the randomness of SGD, (2.68) holds for all \mathbf{p}^t and it is established in [Lacoste-Julien et al., 2012]. By summing (2.68) over $t = 1, \dots, T$ we find

$$\sum_{t=1}^T t \left(\mathbb{E}[F(\boldsymbol{\theta}^t)] - F(\boldsymbol{\theta}^*) \right) \leq \frac{1}{\mu} \mathbb{E} \left[\sum_{t=1}^T \mathbb{V}_e^t(\mathbf{p}^t) \right] - \frac{\mu}{4} T(T+1) \mathbb{E} \left[\|\boldsymbol{\theta}^{T+1} - \boldsymbol{\theta}^*\|^2 \right], \quad (2.69)$$

which as $t \left(\mathbb{E}[F(\boldsymbol{\theta}^t)] - F(\boldsymbol{\theta}^*) \right) \geq 0$ implies

$$\mathbb{E} \left[\|\boldsymbol{\theta}^{T+1} - \boldsymbol{\theta}^*\|^2 \right] \leq \frac{4}{\mu^2 T(T+1)} \mathbb{E} \left[\sum_{t=1}^T \mathbb{V}_e^t(\mathbf{p}^t) \right].$$

In addition, as the cost function F is convex, Jensen's inequality yields

$$\mathbb{E} \left[F \left(\frac{1}{\sum_{t=1}^T t} \sum_{t=1}^T t \cdot \boldsymbol{\theta}^t \right) \right] - F(\boldsymbol{\theta}^*) \leq \frac{1}{\sum_{t=1}^T t} \sum_{t=1}^T t \left(\mathbb{E}[F(\boldsymbol{\theta}^t)] - F(\boldsymbol{\theta}^*) \right). \quad (2.70)$$

Noting that $\sum_{t=1}^T t = T(T+1)/2$, plugging (2.70) in (2.69) yields

$$\mathbb{E} \left[F \left(\frac{2}{T(T+1)} \sum_{t=1}^T t \cdot \boldsymbol{\theta}^t \right) \right] - F(\boldsymbol{\theta}^*) \leq \frac{2}{\mu T(T+1)} \mathbb{E} \left[\sum_{t=1}^T \mathbb{V}_e^t(\mathbf{p}^t) \right].$$

That concludes the proof. \square

Corollary 2.11. *Using MABS with $0 < \eta < 0.5$ in (2.24) and $\delta = 1/c\sqrt{\eta^4 \ln n / (Tn^4 \sum_{i=1}^n a_i^2)}$ in (2.26), for some $c > 1$, to minimize (2.16) with respect to $\{\mathbf{p}^t\}_{1 \leq t \leq T}$, we have*

$$\sum_{t=1}^T \mathbb{V}_e^t(\mathbf{p}^t) \leq \frac{1-\eta}{1-2\eta} \sum_{t=1}^T \mathbb{V}_e^t(\mathbf{p}^*) + \frac{c(1-\eta) + 1/c}{\eta^2(1-2\eta)} \sqrt{n^4 T \sum_{i=1}^n a_i^2 \ln n}, \quad (2.71)$$

where $T \geq n \ln n \cdot \max_i (a_i)^2 / (\eta^2 c^2 \overline{a^2})$, for some $a_i \geq \sup_t \{a_i^t\}$, and where $\overline{a^2} = \sum_{i=1}^n a_i^2 / n$. The complexity of MABS is $O(\log n)$ per iteration.

Proof. Following the same steps as Theorem 2.6, we have (2.39), where by plugging the new $\delta = 1/c\sqrt{\eta^4 \ln n / (Tn^4 \sum_{i=1}^n a_i^2)}$ in (2.39) we get (2.71). Recall that to get (2.31) and hence (2.39), we need to have $-\delta \hat{r}_i^t \leq 1$, where now by choosing a smaller δ we can decrease the minimum acceptable T . By choosing $\delta = 1/c\sqrt{\eta^4 \ln n / (Tn^4 \sum_{i=1}^n a_i^2)}$ the constraint on T becomes $T \geq n \ln n \cdot \max_i (a_i)^2 / (\eta^2 c^2 \overline{a^2})$, which allows us to use a c^2 times smaller T than the one used in Theorem 2.6. Changing δ does not affect the running time of MABS in Theorem 2.6, which is $O(\log n)$ per iteration. \square

2.A.2 MABS with IS

Similar to IS, assume that we can compute the bounds $a_i = \sup_t \{a_i^t\}$ exactly, then we can refine the algorithm and improve the results. The idea is that, instead of mixing the distribution $\{w_i^t / W^t\}_{1 \leq i \leq n}$ with a uniform distribution, we mix $\{w_i^t / W^t\}_{1 \leq i \leq n}$ with distribution $\{a_i^{2/5} / \sum_{j=1}^n a_j^{2/5}\}_{1 \leq i \leq n}$:

$$p_i^t = (1-\eta) \frac{w_i^t}{W^t} + \eta \frac{a_i^{2/5}}{\sum_{j=1}^n a_j^{2/5}}.$$

Corollary 2.12. *Using MABS2 (Algorithm 2.2) with $0 < \eta < 0.5$ in (2.41) and $\delta = \sqrt{\eta^4 \ln n / \left(T \left(\sum_{i=1}^n a_i^{2/5} \right)^5 \right)}$ in (2.26) to minimize (2.16) with respect to $\{\mathbf{p}^t\}_{1 \leq t \leq T}$, we have*

$$\sum_{t=1}^T \mathbb{V}_e^t(\mathbf{p}^t) \leq \frac{1-\eta}{1-2\eta} \sum_{t=1}^T \mathbb{V}_e^t(\mathbf{p}^*) + \frac{2-\eta}{\eta^2(1-2\eta)} \sqrt{T \left(\sum_{i=1}^n a_i^{2/5} \right)^5 \ln n}, \quad (2.72)$$

where $T \geq \ln n \cdot \overline{(a^{2/5})}/(\eta^2 \cdot \min_i a_i^{2/5})$ for $a_i = \sup_t \{a_i^t\}$, with $\overline{(a^{2/5})} = \sum_{i=1}^n a_i^{2/5}/n$. The complexity of MABS2 is $O(\log n)$ per iteration.

Proof. Following the same steps as Theorem 2.6, we have (2.39). With $a_i \geq \sup_t \{a_i^t\}$ and with $p_i^t \geq \eta q_i$ because of (2.41) we can minimize the upper bound on $\sum_{i=1}^n (r_i^t)^2$ in (2.39) as

$$\sum_{i=1}^n (r_i^t)^2 = \sum_{i=1}^n \frac{(a_i^t)^2}{(p_i^t)^4} \leq \frac{1}{\eta^4} \sum_{i=1}^n \frac{(a_i)^2}{(q_i)^4}. \quad (2.73)$$

The right-hand side of (2.73) reaches its minimum for $q_i = a_i^{2/5}/\sum_{j=1}^n a_j^{2/5}$, and it is

$$\sum_{i=1}^n (r_i^t)^2 \leq \frac{1}{\eta^4} \left(\sum_{i=1}^n (a_i)^{2/5} \right)^5. \quad (2.74)$$

Plugging the upper bound (2.74) in (2.39) with $\delta = \sqrt{\eta^4 \ln n / T \left(\sum_{i=1}^n a_i^{2/5} \right)^5}$ concludes the proof. Note that to get (2.39), we need to have $-\delta \hat{r}_i^t \leq 1$, where given $\delta = \sqrt{\eta^4 \ln n / T \left(\sum_{i=1}^n a_i^{2/5} \right)^5}$ – to have $-\delta \hat{r}_i^t \leq 1$ – the constraint on T becomes $T \geq \ln n \cdot \overline{(a^{2/5})}/(\eta \cdot \min_i a_i^{2/5})$. Finally, as in Section A.4 of [Salehi et al., 2017b], with a tree structure (similar to the interval tree) we can update w_{i_t} and sample from \mathbf{p}^t in $O(\log n)$ per iteration. \square

With the same line of reasoning as in Section 2.1, MABS2 can reduce the second term of the right-hand side of (2.27) by n^2 in extreme cases (that happens when one of the a_i is very large compared to the rest).

2.A.3 Omitted Proofs of Section 2.5

Lemma 2.13. *Assume that $F(\boldsymbol{\theta})$ is μ -strongly convex and let each ϕ_i be convex and L_i -smooth. Then, if $\gamma_t = 2/\mu(t+t_0)$ in (2.2) and the sampling distributions $p_i^t \geq 0.4/n$ for all t , the SGD iterates satisfy:*

$$\mathbb{E} \left[\|\boldsymbol{\theta}^t - \boldsymbol{\theta}^*\|^2 \right] \leq \frac{\mu^2 t_0 \|\boldsymbol{\theta}^0 - \boldsymbol{\theta}^*\|^2 + 20\sigma_u^2}{\mu^2(t_0 + t)}, \quad (2.75)$$

where

$$\sigma_u^2 = \mathbb{E}_{i \sim U[1,n]} [\|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^*)\|^2]$$

and

$$t_0 \geq \max \left\{ 1, \frac{4 \sup L_i}{\mu} \right\}. \quad (2.76)$$

Proof The proof is based on Theorem 2.1 of [Needell et al., 2014]. Needell et al. [2014] established that

$$\mathbb{E} \left[\|\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^*\|^2 \right] \leq (1 - 2\gamma_t \mu (1 - \gamma_t \sup L_i)) \mathbb{E} \left[\|\boldsymbol{\theta}^t - \boldsymbol{\theta}^*\|^2 \right] + 2\gamma_t^2 \sigma^2, \quad (2.77)$$

where

$$\sigma^2 = \mathbb{E}_{i \sim \mathbf{p}^t} \left[\left\| \frac{\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^*)}{np_i^t} \right\|^2 \right].$$

For $p_i^t \geq 0.4/n$, we have

$$\sigma^2 = \mathbb{E}_{i \sim \mathbf{p}^t} \left[\left\| \frac{\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^*)}{np_i^t} \right\|^2 \right] \leq 2.5 \mathbb{E}_{i \sim U[1, n]} \left[\|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^*)\|^2 \right]. \quad (2.78)$$

where $\sigma_u^2 = \mathbb{E}_{i \sim U[1, n]} [\|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^*)\|^2]$. Note that σ_u^2 is fixed and does not change with time t . Plugging (2.78) in (2.77) yields

$$\mathbb{E} \left[\|\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^*\|^2 \right] \leq (1 - 2\gamma_t \mu (1 - \gamma_t \sup L_i)) \mathbb{E} \left[\|\boldsymbol{\theta}^t - \boldsymbol{\theta}^*\|^2 \right] + 5\gamma_t^2 \sigma_u^2. \quad (2.79)$$

We prove the lemma by induction. Let (2.75) hold for some $t \in [T]$, we validate the guarantee for $t + 1$. Plugging (2.75) in (2.79) yields

$$\begin{aligned} \mathbb{E} \left[\|\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^*\|^2 \right] &\leq (1 - 2\gamma_t \mu (1 - \gamma_t \sup L_i)) \frac{\mu^2 t_0 \|\boldsymbol{\theta}^0 - \boldsymbol{\theta}^*\|^2 + 20\sigma_u^2}{\mu^2(t_0 + t)} + 5\gamma_t^2 \sigma_u^2 \\ &\leq \left(1 - \frac{2}{t + t_0} \right) \frac{\mu^2 t_0 \|\boldsymbol{\theta}^0 - \boldsymbol{\theta}^*\|^2 + 20\sigma_u^2}{\mu^2(t_0 + t)} + 5\gamma_t^2 \sigma_u^2, \end{aligned} \quad (2.80)$$

where the last inequality follows from the definition of t_0 in (2.76). After some algebraic manipulations it can be shown that the right-hand side of (2.80) is less than $\mu^2 t_0 \|\boldsymbol{\theta}^0 - \boldsymbol{\theta}^*\|^2 + 20\sigma_u^2 / \mu^2(t_0 + t + 1)$, which validates the hypothesis for $t + 1$. To prove the base of induction, set $t = 0$ in (2.79) and get $\|\boldsymbol{\theta}^0 - \boldsymbol{\theta}^*\|^2 \leq \|\boldsymbol{\theta}^0 - \boldsymbol{\theta}^*\|^2 + 20\sigma_u^2 / \mu^2 t_0$, that is simply correct. \square

Proof of Theorem 2.10 The starting point is the inequality

$$\mathbb{E}[F(\boldsymbol{\theta}^t)] - F(\boldsymbol{\theta}^*) \leq \frac{\gamma_t \mathbb{E}[\mathbb{V}_\epsilon^t(\mathbf{p}^t)]}{2} + \frac{\gamma_t^{-1} - \mu}{2} \mathbb{E} \left[\|\boldsymbol{\theta}^t - \boldsymbol{\theta}^*\|^2 \right] - \frac{\gamma_t^{-1}}{2} \mathbb{E} \left[\|\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^*\|^2 \right], \quad (2.81)$$

where the expectations are over the randomness of SGD, (2.81) holds for all sampling distributions \mathbf{p}^t and it is established in [Lacoste-Julien et al., 2012]. Substituting $\gamma_t = 2/\mu(t+t_0)$ into (2.81) gives

$$\mathbb{E}[F(\boldsymbol{\theta}^t)] - F(\boldsymbol{\theta}^*) \leq \frac{\mathbb{E}[\mathbb{V}_e^t(\mathbf{p}^t)]}{\mu(t+t_0)} + \frac{\mu(t+t_0-2)}{4} \mathbb{E}[\|\boldsymbol{\theta}^t - \boldsymbol{\theta}^*\|^2] - \frac{\mu(t+t_0)}{4} \mathbb{E}[\|\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^*\|^2]. \quad (2.82)$$

Multiply (2.82) by $(t+t_0-1)$ and sum it over $t = 1, \dots, T$

$$\begin{aligned} & \sum_{t=1}^T (t+t_0-1) \left(\mathbb{E}[F(\boldsymbol{\theta}^t)] - F(\boldsymbol{\theta}^*) \right) \leq \\ & \sum_{t=1}^T \frac{\mathbb{E}[\mathbb{V}_e^t(\mathbf{p}^t)]}{\mu} + \frac{\mu(t_0-2)(t_0-1)}{4} \|\boldsymbol{\theta}^0 - \boldsymbol{\theta}^*\|^2 - \frac{\mu(T+t_0)(T+t_0-1)}{2} \mathbb{E}[\|\boldsymbol{\theta}^{T+1} - \boldsymbol{\theta}^*\|^2]. \end{aligned}$$

Invoking Jensen's inequality yields

$$\begin{aligned} & \mathbb{E} \left[F \left(\frac{2}{T(T+2t_0-1)} \sum_{t=1}^T (t+t_0-1) \boldsymbol{\theta}^t \right) \right] - F(\boldsymbol{\theta}^*) \leq \\ & \frac{2}{T(T+2t_0-1)} \left[\sum_{t=1}^T \frac{\mathbb{E}[\mathbb{V}_e^t(\mathbf{p}^t)]}{\mu} + \frac{\mu(t_0-2)(t_0-1)}{4} \|\boldsymbol{\theta}^0 - \boldsymbol{\theta}^*\|^2 \right] - \frac{\mu}{2} \mathbb{E}[\|\boldsymbol{\theta}^{T+1} - \boldsymbol{\theta}^*\|^2], \end{aligned}$$

which implies

$$\begin{aligned} & \mathbb{E} \left[F \left(\frac{2}{T(T+2t_0-1)} \sum_{t=1}^T (t+t_0-1) \boldsymbol{\theta}^t \right) \right] - F(\boldsymbol{\theta}^*) \leq \\ & \frac{2}{T(T+2t_0-1)} \left[\sum_{t=1}^T \frac{\mathbb{E}[\mathbb{V}_e^t(\mathbf{p}^t)]}{\mu} + \frac{\mu(t_0-2)(t_0-1)}{4} \|\boldsymbol{\theta}^0 - \boldsymbol{\theta}^*\|^2 \right] \end{aligned} \quad (2.83)$$

and

$$\mathbb{E}[\|\boldsymbol{\theta}^{T+1} - \boldsymbol{\theta}^*\|^2] \leq \frac{4}{\mu T(T+2t_0-1)} \left[\sum_{t=1}^T \frac{\mathbb{E}[\mathbb{V}_e^t(\mathbf{p}^t)]}{\mu} + \frac{\mu(t_0-2)(t_0-1)}{4} \|\boldsymbol{\theta}^0 - \boldsymbol{\theta}^*\|^2 \right]. \quad (2.84)$$

Next, let us bound $\sum_{t=1}^T \mathbb{E}[\mathbb{V}_e^t(\mathbf{p}^t)]$. MABS (with $\eta = 0.4$) enjoys the performance guarantee in (2.27), that is

$$\sum_{t=1}^T \mathbb{E}[\mathbb{V}_e^t(\mathbf{p}^t)] \leq 3 \min_{\mathbf{p}} \sum_{t=1}^T \mathbb{E}[\mathbb{V}_e^t(\mathbf{p})] + 50 \mathbb{E} \left[\sqrt{nTG^2 \ln n} \right], \quad (2.85)$$

Chapter 2. Stochastic Gradient Descent with Bandit Sampling

for some $\overline{G^2} \geq \sum_{t=1}^T \sum_{i=1}^n \|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^t)\|^2 / (Tn)$. From the triangle inequality,

$$\begin{aligned} \|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^t)\|^2 &\leq 2\|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^*)\|^2 + 2\|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^*)\|^2 \\ &\leq 2L_i \|\boldsymbol{\theta}^t - \boldsymbol{\theta}^*\|^2 + 2\|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^*)\|^2, \end{aligned} \quad (2.86)$$

where the second inequality follows from the L_i -smoothness of ϕ_i . In SGD, the effective variance $\mathbb{V}_e^t(\mathbf{p}) = \sum_{i=1}^n \|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^t)\|^2 / n^2 p_i$, plugging (2.86) in the effective variance yields

$$\mathbb{V}_e^t(\mathbf{p}) \leq \frac{2}{n^2} \sum_{i=1}^n \left[\frac{\|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^*)\|^2}{p_i} + \frac{L_i \|\boldsymbol{\theta}^t - \boldsymbol{\theta}^*\|^2}{p_i} \right].$$

As in MABS $p_i^t \geq 0.4/n$ we have

$$\mathbb{V}_e^t(\mathbf{p}) \leq \frac{2}{n^2} \sum_{i=1}^n \frac{\|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^*)\|^2}{p_i} + \frac{5}{n} \|\boldsymbol{\theta}^t - \boldsymbol{\theta}^*\|^2 \sum_{i=1}^n L_i.$$

Thus

$$\begin{aligned} 3 \min_{\mathbf{p}} \sum_{t=1}^T \mathbb{E}[\mathbb{V}_e^t(\mathbf{p})] &\leq \frac{6}{n^2} \min_{\mathbf{p}} \left[\sum_{t=1}^T \sum_{i=1}^n \frac{\|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^*)\|^2}{p_i} \right] + \frac{5}{n} \left(\sum_{i=1}^n L_i \right) \sum_{t=1}^T \mathbb{E} \left[\|\boldsymbol{\theta}^t - \boldsymbol{\theta}^*\|^2 \right] \\ &= \frac{6T}{n^2} \left(\sum_{i=1}^n \|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^*)\| \right)^2 + \frac{5}{n} \left(\sum_{i=1}^n L_i \right) \sum_{t=1}^T \mathbb{E} \left[\|\boldsymbol{\theta}^t - \boldsymbol{\theta}^*\|^2 \right]. \end{aligned} \quad (2.87)$$

As conditions of Lemma 2.13 hold, we can use (2.75) in Lemma 2.13 to bound $\sum_{t=1}^T \mathbb{E} \left[\|\boldsymbol{\theta}^t - \boldsymbol{\theta}^*\|^2 \right]$,

$$\begin{aligned} \sum_{t=1}^T \mathbb{E} \left[\|\boldsymbol{\theta}^t - \boldsymbol{\theta}^*\|^2 \right] &\leq \sum_{t=1}^T \frac{\mu^2 t_0 \|\boldsymbol{\theta}^0 - \boldsymbol{\theta}^*\|^2 + 20\sigma_u^2}{\mu^2 (t_0 + t)} \\ &\leq \frac{\mu^2 t_0 \|\boldsymbol{\theta}^0 - \boldsymbol{\theta}^*\|^2 + 20\sigma_u^2}{\mu^2} \log(T + t_0), \end{aligned}$$

Let

$$\alpha := \frac{\mu^2 t_0 \|\boldsymbol{\theta}^0 - \boldsymbol{\theta}^*\|^2 + 20\sigma_u^2}{\mu^2} \log(T + t_0).$$

Combining the bound above with (2.87) gives

$$3 \min_{\mathbf{p}} \sum_{t=1}^T \mathbb{E}[\mathbb{V}_e^t(\mathbf{p})] \leq \frac{6T}{n^2} \left(\sum_{i=1}^n \|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^*)\| \right)^2 + \frac{5}{n} \alpha \sum_{i=1}^n L_i. \quad (2.88)$$

Similarly the second term in the right hand side of (2.85) can be upper bounded as

$$\begin{aligned} \mathbb{E} \left[\sqrt{\ln n \sum_{t=1}^T \sum_{i=1}^n \|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^t)\|^2} \right] &\leq \sqrt{\mathbb{E} \left[\ln n \sum_{t=1}^T \sum_{i=1}^n \mathbb{E} \|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^t)\|^2 \right]} \\ &\leq \sqrt{2 \ln n \left(T \left(\sum_{i=1}^n \|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^*)\|^2 \right) + \alpha \sum_{i=1}^n L_i \right)}, \end{aligned} \quad (2.89)$$

where the first inequality follows from the Jensen's inequality. Plugging (2.89), (2.87) and (2.85) in (2.84) yields

$$\begin{aligned} \mathbb{E} \left[\|\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^*\|^2 \right] &\leq \frac{24}{\mu^2 n^2 T} \left(\sum_{i=1}^n \|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^*)\| \right)^2 + \frac{20}{\mu^2 n T^2} \alpha \sum_{i=1}^n L_i \\ &\quad + \frac{200t_0}{\mu^2 T^2} \sqrt{2 \ln n \left(T \left(\sum_{i=1}^n \|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^*)\|^2 \right) + \alpha \sum_{i=1}^n L_i \right)} \\ &\quad + \frac{t_0^2}{T^2} \|\boldsymbol{\theta}^0 - \boldsymbol{\theta}^*\|^2. \end{aligned} \quad (2.90)$$

Note that the first term in (2.90) scales as $O(1/T)$, the second term scales as $O(\log T/T^2)$, the third term scales as $O(1/T^{3/2})$ and the last term scales as $O(1/T^2)$. So asymptotically, the first term is the most important term in the convergence rate and we conclude the proof. □

2.B PSGD

For PSGD, let $\sum_{i=1}^n \phi_i(\boldsymbol{\theta}^t)/n$ be μ -strongly convex and L -smooth with respect to ψ , a continuously differentiable function, and let $\mathcal{B}_{\psi}(w_1, w_2)$ be the Bregman divergence associated with the function ψ (see Appendix 2.C for a summary of these standard definitions). Consider the cost function

$$F(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^n \phi_i(\boldsymbol{\theta}^t) + \lambda r(\boldsymbol{\theta}). \quad (2.91)$$

PSGD updates $\boldsymbol{\theta}$ according to

$$\boldsymbol{\theta}^{t+1} = \arg \min_{\boldsymbol{\theta}} \left[\langle \nabla_{\boldsymbol{\theta}} \phi_{i_t}(\boldsymbol{\theta}^t), \boldsymbol{\theta} \rangle + \lambda r(\boldsymbol{\theta}) + \frac{1}{\gamma_t} \mathcal{B}_{\psi}(\boldsymbol{\theta}, \boldsymbol{\theta}^t) \right]. \quad (2.92)$$

Chapter 2. Stochastic Gradient Descent with Bandit Sampling

Intuitively, this method works by minimizing the first order approximation of the sub cost-function ϕ_{i_t} in (2.91) plus the regularizer $\lambda r(\boldsymbol{\theta})$. In the non-uniform version of this algorithm, $\nabla_{\boldsymbol{\theta}} \phi_{i_t}(\boldsymbol{\theta}^t)$ is replaced by $\nabla_{\boldsymbol{\theta}} \phi_{i_t}(\boldsymbol{\theta}^t)/(np_{i_t}^t)$, see Zhao and Zhang [2015a]. In PSGD, we also use the same step size as in SGD in Theorem 2.1.

Theorem 2.14. *Assume that $\sum_{i=1}^n \phi_i(\boldsymbol{\theta}^t)/n$ is μ -strongly convex and L -smooth with respect to ψ where ψ is a σ -strongly convex function, and that the regularizer $r(\boldsymbol{\theta})$ in (2.91) is convex. Then, if $\gamma_t = 2/\mu(1+t)$ in (2.92), the following inequality holds for any $T \geq (\max_i(G_i)^2/\eta^2(\overline{G^2})) n \ln n$ in PSGD with MABS:*

$$\mathbb{E} \left[\mathcal{B}_{\psi}(\boldsymbol{\theta}^*, \boldsymbol{\theta}^{T+1}) \right] = O \left(\frac{1}{\mu^2 \sigma T^2} \left(\mathbb{E} \left[\sum_{t=1}^T \mathbb{V}_e^t(\hat{\mathbf{p}}) \right] + \sqrt{T \sum_{i=1}^n G_i^2 \ln n} \right) \right), \quad (2.93)$$

and

$$\begin{aligned} \mathbb{E} \left[\epsilon \left(\frac{2}{T(T+1)} \cdot \boldsymbol{\theta}^t \right) \right] &= \mathbb{E} \left[F \left(\frac{2}{T(T+1)} \sum_{t=1}^T t \cdot \boldsymbol{\theta}^{t+1} \right) \right] - F(\boldsymbol{\theta}^*) \\ &= O \left(\frac{1}{\mu \sigma T^2} \left(\mathbb{E} \left[\sum_{t=1}^T \mathbb{V}_e^t(\hat{\mathbf{p}}) \right] + \sqrt{T \sum_{i=1}^n G_i^2 \ln n} \right) \right) \end{aligned} \quad (2.94)$$

for some $G_i \geq \sup_{\boldsymbol{\theta}^1, \dots, \boldsymbol{\theta}^T} \|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^t)\|^2$, where $\hat{\mathbf{p}} = \arg \min_{\mathbf{p}} \sum_{t=1}^T \sum_{i=1}^n \|\nabla_{\boldsymbol{\theta}} \phi_i(\boldsymbol{\theta}^t)\|^2/p_i$, and where the expectations are over the sequence of the coordinates $\{w^t\}_{t \in [T]}$ that are updated.

If $\psi(\mathbf{x}) = \|\mathbf{x}\|^2$, then $\mathcal{B}_{\psi}(\boldsymbol{\theta}^1, \boldsymbol{\theta}^2) = \|\boldsymbol{\theta}^1 - \boldsymbol{\theta}^2\|^2$ and the convergence rates (2.93) and (2.94) in this theorem are $O(1/T)$, compared to $O(\ln T/T)$ in Zhao and Zhang [2015a]. The proof is an extension of Theorem 1 in Zhao and Zhang [2015a] where we also use the analysis in Lacoste-Julien et al. [2012]. For completeness, we present the proof below.

Proof Our starting point is the inequality

$$\mathbb{E}[F(\boldsymbol{\theta}^{t+1}) - F(\boldsymbol{\theta}^*)] \leq \frac{\gamma_t}{\sigma} \mathbb{E} \left[\mathbb{V}_e^t(\mathbf{p}^t) \right] + \left(\frac{1}{\gamma_t} - \mu \right) \mathbb{E} \left[\mathcal{B}_{\psi}(\boldsymbol{\theta}^*, \boldsymbol{\theta}^t) \right] - \frac{1}{\gamma_t} \mathbb{E} \left[\mathcal{B}_{\psi}(\boldsymbol{\theta}^*, \boldsymbol{\theta}^{t+1}) \right], \quad (2.95)$$

where the expectations are over the randomness of PSGD, (2.95) holds for all \mathbf{p}^t and it is established in Lemma 1 of [Zhao and Zhang, 2015a]. Next, inspired by Lacoste-Julien et al. [2012], let $\gamma_t = 2/\mu(1+t)$. Plugging this γ_t in (2.95) yields

$$\begin{aligned} t \left(\mathbb{E}[F(\boldsymbol{\theta}^{t+1}) - F(\boldsymbol{\theta}^*)] \right) &\leq \frac{2t}{\sigma \mu(t+1)} \mathbb{E} \left[\mathbb{V}_e^t(\mathbf{p}^t) \right] + \frac{\mu(t-1)t}{2} \mathbb{E} \left[\mathcal{B}_{\psi}(\boldsymbol{\theta}^*, \boldsymbol{\theta}^t) \right] \\ &\quad - \frac{\mu t(t+1)}{2} \mathbb{E} \left[\mathcal{B}_{\psi}(\boldsymbol{\theta}^*, \boldsymbol{\theta}^{t+1}) \right]. \end{aligned} \quad (2.96)$$

By summing (2.96) over $t = 1, \dots, T$ we get

$$\sum_{t=1}^T t \left(\mathbb{E}[F(\boldsymbol{\theta}^{t+1}) - F(\boldsymbol{\theta}^*)] \right) \leq \frac{2}{\sigma\mu} \mathbb{E} \left[\sum_{t=1}^T \mathbb{V}_e^t(\mathbf{p}^t) \right] - \frac{\mu T(T+1)}{2} \mathbb{E} \left[\mathcal{B}_\psi(\boldsymbol{\theta}^*, \boldsymbol{\theta}^{T+1}) \right], \quad (2.97)$$

which as $t \left(\mathbb{E}[F(\boldsymbol{\theta}^t)] - F(\boldsymbol{\theta}^*) \right) \geq 0$ implies

$$\mathbb{E} \left[\mathcal{B}_\psi(\boldsymbol{\theta}^*, \boldsymbol{\theta}^{T+1}) \right] \leq \frac{4}{\sigma\mu^2 T(T+1)} \mathbb{E} \left[\sum_{t=1}^T \mathbb{V}_e^t(\mathbf{p}^t) \right]. \quad (2.98)$$

In addition, as the cost function F is convex, Jensen's inequality yields

$$\mathbb{E} \left[F \left(\frac{1}{\sum_{t=1}^T t} \sum_{t=1}^T t \cdot \boldsymbol{\theta}^t \right) \right] - F(\boldsymbol{\theta}^*) \leq \frac{1}{\sum_{t=1}^T t} \sum_{t=1}^T t \left(\mathbb{E}[F(\boldsymbol{\theta}^t)] - F(\boldsymbol{\theta}^*) \right). \quad (2.99)$$

Noting that $\sum_{t=1}^T t = T(T+1)/2$, plugging (2.99) in (2.97) yields

$$\mathbb{E} \left[F \left(\frac{2}{T(T+1)} \sum_{t=1}^T t \cdot \boldsymbol{\theta}^{t+1} \right) \right] - F(\boldsymbol{\theta}^*) \leq \frac{4}{\sigma\mu T(T+1)} \mathbb{E} \left[\sum_{t=1}^T \mathbb{V}_e^t(\mathbf{p}^t) \right]. \quad (2.100)$$

Finally, plugging (2.27) in (2.98) and (2.100) concludes the proof. \square

2.C Definitions

Definition (L -smooth). Let $L > 0$. Function $h(\cdot)$ is L -smooth if for any \mathbf{x} and $\mathbf{y} \in \mathbb{R}^d$

$$h(\mathbf{y}) \leq h(\mathbf{x}) + \langle \nabla h(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{L}{2} \|\mathbf{x} - \mathbf{y}\|^2. \quad (2.101)$$

Definition (μ -strongly convex). Let $\mu > 0$. Function $h(\cdot)$ is μ -strongly convex if for any \mathbf{x} and $\mathbf{y} \in \mathbb{R}^d$

$$h(\mathbf{y}) \geq h(\mathbf{x}) + \langle \nabla h(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{\mu}{2} \|\mathbf{x} - \mathbf{y}\|^2. \quad (2.102)$$

Definition (Bregman divergence). Let $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2 \in \mathbb{R}^d$. The Bregman divergence associated with the function ψ is

$$\mathcal{B}_\psi(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = \psi(\boldsymbol{\theta}_1) - \psi(\boldsymbol{\theta}_2) - \langle \nabla \psi(\boldsymbol{\theta}_2), \boldsymbol{\theta}_1 - \boldsymbol{\theta}_2 \rangle. \quad (2.103)$$

Chapter 2. Stochastic Gradient Descent with Bandit Sampling

Definition (μ -strongly convex with respect to ψ). Let $\mu > 0$. Function $f(\cdot)$ is μ -strongly convex with respect to a differentiable function ψ if for any $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2 \in \mathbb{R}^d$

$$f(\boldsymbol{\theta}_1) \geq f(\boldsymbol{\theta}_2) + \langle \nabla \psi(\boldsymbol{\theta}_2), \boldsymbol{\theta}_1 - \boldsymbol{\theta}_2 \rangle + \mu \mathcal{B}_\psi(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2). \quad (2.104)$$

3 Coordinate Descent with Bandit Sampling

In the previous chapter, we saw how to accelerate the convergence rate of stochastic gradient descent by developing a bandit algorithm to select datapoints for updating the model. In this chapter¹, we shift our attention from stochastic gradient descent (SGD) to the coordinate descent (CD) method. The former (SGD) minimizes a cost function by using the gradient of one of the datapoints sampled at random, whereas the latter (CD) usually minimizes a cost function by updating a decision variable (corresponding to one coordinate) at a time. Ideally, we would update the decision variable that yields the largest decrease in the cost function. However, finding this coordinate would require checking all of them, which would effectively negate the improvement in computational tractability that coordinate descent is intended to afford.

To address this, we take a similar approach as in Chapter 2; we study the coordinate-selection module of CD in the multi-armed bandit setting. More precisely, first, we find a lower bound on the amount the cost function decreases when a coordinate is updated. Next, we use a stochastic multi-armed bandit algorithm (see Section 1.2.3 for details about the stochastic multi-armed bandit setting) to learn which coordinates result in the largest lower bound by interleaving this learning with conventional CD updates except that the coordinate is selected proportionately to the expected decrease. We show that our approach improves the convergence of coordinate descent methods both theoretically and experimentally.

3.1 Introduction

As we explained in Chapter 2, most supervised learning algorithms minimize an empirical risk cost function over a dataset. Here, we rewrite the cost function (2.1) in a form that

¹This chapter is based on [Salehi et al., 2018].

is more appropriate for coordinate descent methods,

$$F(\boldsymbol{\theta}) = f(\mathbf{A}\boldsymbol{\theta}) + \sum_{i=1}^d g_i(\theta_i), \quad (3.1)$$

where $f(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}$ is a smooth convex function, d is the number of decision variables (coordinates) on which the cost function is minimized, which are gathered in vector $\boldsymbol{\theta} \in \mathbb{R}^d$, $g_i(\cdot) : \mathbb{R} \rightarrow \mathbb{R}$ are convex functions for all $i \in [d]$, and $\mathbf{A} \in \mathbb{R}^{n \times d}$ is the data matrix. As a running example, consider Lasso: if $\mathbf{y} \in \mathbb{R}^n$ is the vector of labels,

$$f(\mathbf{A}\boldsymbol{\theta}) = \frac{1}{2n} \|\mathbf{y} - \mathbf{A}\boldsymbol{\theta}\|^2,$$

where $\|\cdot\|$ stands for the Euclidean norm, and $g_i(\theta_i) = \lambda|\theta_i|$. Often the particular form of the cost function in (3.1) is used to represent the dual of empirical risk cost functions. Note that when the primal of a cost function is minimized, d is the number of features, whereas when the dual of a cost function is minimized, d is the number of datapoints. For example, the dual of L_2 -regularized linear regression in Section 2.1 is

$$F(\boldsymbol{\theta}) = \frac{1}{2\lambda d^2} \|\mathbf{A}\boldsymbol{\theta}\|^2 + \frac{1}{d} \sum_{i=1}^d \left(\frac{\theta_i^2}{4} - \theta_i y_i \right),$$

where $\mathbf{y} \in \mathbb{R}^d$, $f(\mathbf{A}\boldsymbol{\theta}) = 1/2\lambda d^2 \|\mathbf{A}\boldsymbol{\theta}\|^2$, and $g_i(\theta_i) = 1/d (\theta_i^2/4 - \theta_i y_i)$.

To bypass the computational intractability of gradient descent, coordinate descent (CD) selects one *coordinate* θ_i to optimize over at each timestep. When CD was first introduced, algorithms did not differentiate between coordinates; each coordinate $i \in [d]$ was selected uniformly at random at each time step (see, e.g., [Shalev-Shwartz and Zhang, 2013a,b]). Recent works (see, e.g., [Glasmachers and Dogan, 2013, Zhao and Zhang, 2015a, Perekrestenko et al., 2017]) have shown that exploiting the structure of the data and sampling the coordinates from an appropriate non-uniform distribution can result in better convergence guarantees, both in theory and practice. The challenge is to find the appropriate non-uniform sampling distribution with a lightweight mechanism that maintains the computational tractability of CD.

Our Contributions

In this chapter, we propose a novel adaptive non-uniform coordinate selection method that can be applied to both the primal and dual forms of a cost function. The method exploits the structure of the data to optimize the model by finding and frequently updating the most predictive decision variables. In particular, for each $i \in [d]$ at time t , a lower bound r_i^t is derived (which we call the *marginal decrease*) on the amount by which the cost function will decrease when only the i^{th} coordinate is updated.

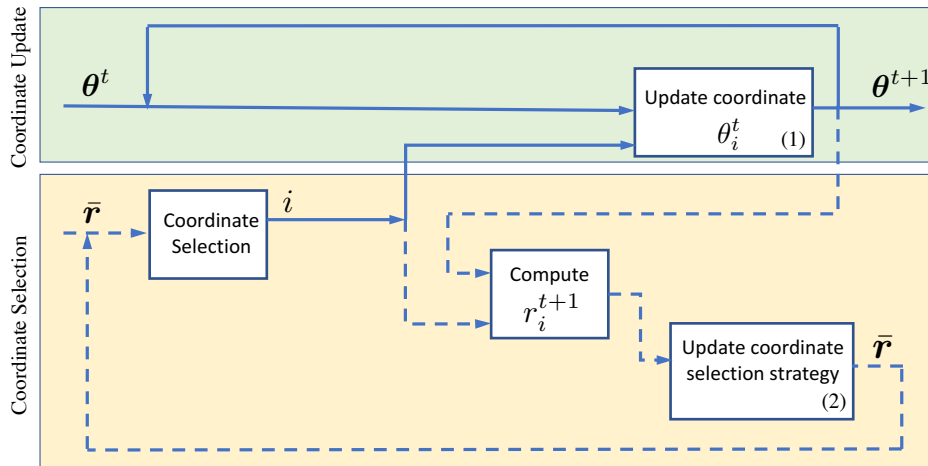


Figure 3.1 – Our approach for coordinate descent. The top (green) part handles the updates to the decision variable θ_i^t (using whichever CD update is desired); our theoretical results hold for updates in the class \mathcal{H} in Definition 3.4.1 in the supplementary materials. The bottom (yellow) part of the approach handles the selection of $i \in [d]$ according to a coordinate selection strategy which is updated via bandit optimization (using whichever bandit algorithm is desired) from r_i^{t+1} .

The marginal decrease r_i^t quantifies by how much updating the i^{th} coordinate is guaranteed to improve the model. The coordinate i with the largest r_i^t is then the one that is updated by the algorithm `max_r`, described in Section 3.4.2. This approach is particularly beneficial when the distribution of r_i^t s has a high variance across i ; in such cases updating different coordinates can yield very different decreases in the cost function. For example, if the distribution of r_i^t s has a high variance across i , `max_r` is up to d^2 times better than uniform sampling, whereas state-of-the-art methods can be at most $d^{3/2}$ better than uniform sampling in such cases (see Theorem 3.6 in Section 3.4.2). More precisely, in `max_r` the convergence speed is proportional to the ratio of the duality gap to the maximum coordinate-wise duality gap. `max_r` is able to outperform existing adaptive methods because it explicitly finds the coordinates that yield a large decrease of the cost function, instead of computing a distribution over coordinates based on an approximation of the marginal decreases.

However, the computation of the marginal decrease r_i^t for all $i \in [d]$ may still be computationally prohibitive. To bypass this obstacle, we adopt in Section 3.4.3 a principled approach (`B_max_r`) for learning the best r_i^t s, instead of explicitly computing all of them: At each time t , we choose a single coordinate i and update it. Next, we compute the marginal decrease r_i^t of the selected coordinate i and use it as feedback to adapt our coordinate selection strategy using a bandit framework. Thus, in effect, we learn estimates of the r_i^t s and simultaneously optimize the cost function (see Figure 3.1). We prove that this approach can perform almost as well as `max_r`, yet decreases the number of calculations required by a factor of d (see Proposition 3.8).

We test this approach on several standard datasets, using different cost functions (including Lasso, logistic and ridge regression) and for both the adaptive setting (the first approach) and the bandit setting (the second approach). We observe that the bandit coordinate selection approach accelerates the convergence of a variety of CD methods (e.g., StingyCD [Johnson and Guestrin, 2017] for Lasso in Figure 3.2, dual CD [Shalev-Shwartz and Tewari, 2011] for L_1 -regularized logistic-regression in Figure 3.3, and dual CD [Nutini et al., 2015] for ridge-regression in Figure 3.3). Furthermore, we observe that in most of the experiments `B_max_r` (the second approach) converges as fast as `max_r` (the first approach), while it has the same computational complexity as CD with uniform sampling (see Section 3.5).

3.2 Preliminaries

Consider the following primal-dual optimization pairs

$$\begin{aligned} \min_{\boldsymbol{\theta} \in \mathbb{R}^d} F(\boldsymbol{\theta}) &= f(\mathbf{A}\boldsymbol{\theta}) + \sum_{i=1}^d g_i(\theta_i), \\ \min_{\mathbf{w} \in \mathbb{R}^n} F_D(\mathbf{w}) &= f^*(\mathbf{w}) + \sum_{i=1}^d g_i^*(-\mathbf{a}_i^\top \mathbf{w}), \end{aligned} \quad (3.2)$$

where $A = [\mathbf{a}_1, \dots, \mathbf{a}_d]$, $\mathbf{a}_i \in \mathbb{R}^n$, and f^* and g_i^* are the convex conjugates of f and g_i , respectively. The convex conjugate of a function $h(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}$ is

$$h^*(\boldsymbol{\theta}) = \sup_{\mathbf{v} \in \mathbb{R}^d} \{\boldsymbol{\theta}^\top \mathbf{v} - h(\mathbf{v})\}.$$

The goal is to find $\bar{\boldsymbol{\theta}} := \arg \min_{\boldsymbol{\theta} \in \mathbb{R}^d} F(\boldsymbol{\theta})$. We denote by $\epsilon(\boldsymbol{\theta}) = F(\boldsymbol{\theta}) - F(\bar{\boldsymbol{\theta}})$ the *sub-optimality gap* of $F(\boldsymbol{\theta})$.

The optimal primal-dual pair $(\bar{\boldsymbol{\theta}}, \bar{\mathbf{w}})$ is reached when the following optimality conditions are satisfied (see [Bauschke and Combettes, 2011]):

$$\begin{aligned} \mathbf{w} &\in \partial f(\mathbf{A}\bar{\boldsymbol{\theta}}), & -\mathbf{a}_i^\top \bar{\mathbf{w}} &\in \partial g_i(\bar{\theta}_i) \text{ for all } i \in [d], \\ \mathbf{A}\bar{\boldsymbol{\theta}} &\in \partial f^*(\bar{\mathbf{w}}), & \bar{\theta}_i &\in g_i^*(-\mathbf{a}_i^\top \bar{\mathbf{w}}) \text{ for all } i \in [d]. \end{aligned}$$

The *duality gap* $G(\boldsymbol{\theta}, \mathbf{w})$ is the difference between the primal and the dual solutions,

$$G(\boldsymbol{\theta}, \mathbf{w}) = F(\boldsymbol{\theta}) - (-F_D(\mathbf{w})),$$

and it is an upper bound on $\epsilon(\boldsymbol{\theta})$ for all $\boldsymbol{\theta} \in \mathbb{R}^d$. If the algorithm sets $\mathbf{w} = \nabla f(\mathbf{A}\boldsymbol{\theta})$, then the Fenchel-Young property for $\mathbf{w} = \nabla f(\mathbf{A}\boldsymbol{\theta})$ yields that $f(\mathbf{A}\boldsymbol{\theta}) + f^*(\mathbf{w}) = (\mathbf{A}\boldsymbol{\theta})^\top \mathbf{w}$,

which in turn implies that

$$G(\boldsymbol{\theta}, \mathbf{w}) = \sum_{i=1}^d \left(g_i^*(-\mathbf{a}_i^\top \mathbf{w}) + g_i(\theta_i) + \theta_i \mathbf{a}_i^\top \mathbf{w} \right).$$

In rest of the chapter, we further use the shorthand $G(\boldsymbol{\theta})$ for $G(\boldsymbol{\theta}, \mathbf{w})$ when $\mathbf{w} = \nabla f(\mathbf{A}\boldsymbol{\theta})$. We call $G_i(\boldsymbol{\theta}) = \left(g_i^*(-\mathbf{a}_i^\top \mathbf{w}) + g_i(\theta_i) + \theta_i \mathbf{a}_i^\top \mathbf{w} \right)$ the i^{th} *coordinate-wise duality gap*. Finally, we denote by $\kappa_i = \bar{u} - \theta_i$ the i^{th} *dual residue* where $\bar{u} = \arg \min_{u \in \partial g_i^*(-\mathbf{a}_i^\top \mathbf{w})} |u - \theta_i|$ with $\mathbf{w} = \nabla f(\mathbf{A}\boldsymbol{\theta})$. Dual residues were introduced in [Csiba et al., 2015] and similar notions appear in [Perekrestenko et al., 2017, Shalev-Shwartz and Zhang, 2013b].

3.3 Related Work

Non-uniform coordinate selection has been proposed first for constant (non-adaptive) probability distributions \mathbf{p} over $[d]$. In [Zhao and Zhang, 2015a], p_i is proportional to the Lipschitz constant of g_i^* . Similar distributions are used in [Allen-Zhu et al., 2016, Zhang and Gu, 2016] for strongly convex f in (3.1).

Time varying (adaptive) distributions, such as $p_i^t = |\kappa_i^t| / (\sum_{j=1}^d |\kappa_j^t|)$ [Csiba et al., 2015], and $p_i^t = G_i(\boldsymbol{\theta}^t) / G(\boldsymbol{\theta}^t)$ [Perekrestenko et al., 2017, Osokin et al., 2016], have also been considered. In all these cases, the full information setting is used, which requires the computation of the distribution \mathbf{p}^t ($\Omega(nd)$ calculations) at each step. To bypass this problem, heuristics are often used; e.g., \mathbf{p}^t is calculated once at the beginning of an epoch of length E and is left unchanged throughout the remainder of that epoch. This heuristic approach does not work well in a scenario where $G_i(\boldsymbol{\theta}^t)$ varies significantly. In [Dünner et al., 2017] a similar idea to `max_r` is used with r_i replaced by G_i , but only in the full information setting. Because of the update rule used in [Dünner et al., 2017], the convergence rate is $O(d \cdot \max G_i(\boldsymbol{\theta}^t) / G(\boldsymbol{\theta}^t))$ times slower than Theorem 3.6 (see also the comparison at the end of Section 3.4.2). The Gauss-Southwell rule (GS) is another coordinate selection strategy for smooth cost functions [Shi et al., 2016] and its convergence is studied in [Nutini et al., 2015] and [Stich et al., 2017]. GS selects the coordinate to update as the one that maximizes $|\nabla_i F(\boldsymbol{\theta}^t)|$ at time t . The algorithm `max_r` can be seen as an extension of GS to a broader class of cost functions (see Lemma 3.9 in Appendix 3.B). Furthermore, when only sub-gradients are defined for $g_i(\cdot)$, GS needs to solve a proximal problem. To address the computational tractability of GS, in [Stich et al., 2017], lower and upper bounds on the gradients are computed (instead of computing the gradient itself) and used for selecting the coordinates, but these lower and upper bounds might be loose and/or difficult to find. For example, without a heavy pre-processing of the data, ASCD in [Stich et al., 2017] converges with the same rate as uniform sampling when the data is normalized and $f(\mathbf{A}\boldsymbol{\theta}) = \|\mathbf{A}\boldsymbol{\theta} - \mathbf{Y}\|^2$.

In contrast, our principled approach leverages a bandit algorithm to learn a good estimate of r_i^t ; this allows for theoretical guarantees and outperforms the state-of-the-art methods, as we will see in Section 3.5. Furthermore, our approach does not require the cost function to be strongly convex (contrary to e.g., [Csiba et al., 2015, Nutini et al., 2015])

3.4 Technical Contributions

3.4.1 Marginal Decreases

Our coordinate selection approach works for a class \mathcal{H} of update rules for the decision variable θ_i . The update rule should be able to capture the non-optimality along different coordinates, i.e., it should attain a larger decrease in the cost function for a non-optimal decision variable θ_i^t compared to updating an already close to the optimal decision variable θ_j^t . This essentially allows to search for a decision variable θ_i^t that is far from optimum. For example, the update rule in [Shalev-Shwartz and Tewari, 2011] for lasso, the update rules in [Shalev-Shwartz and Zhang, 2013b] for hinge-loss SVM and ridge regression, the update rule in [Csiba et al., 2015], in addition to the update rule in (3.3), belong to this class \mathcal{H} .

Definition (\mathcal{H}). In (3.1), let $f(\cdot)$ be $1/\beta$ -smooth and each $g_i(\cdot)$ be μ_i -strongly convex with convexity parameter $\mu_i \geq 0 \forall i \in [d]$. For $\mu_i = 0$, we assume that g_i has a L_i -bounded support. Let $\hat{h} : \mathbb{R}^n \times [n] \rightarrow \mathbb{R}^n$ be the update rule, i.e., $\boldsymbol{\theta}^{t+1} = \hat{h}(\boldsymbol{\theta}^t, i)$, for the decision variables $\boldsymbol{\theta}^t$ whose j^{th} entry is

$$\hat{h}_j(\boldsymbol{\theta}, i) = \begin{cases} \theta_j + s_j \kappa_j & \text{if } j = i, \\ \theta_j & \text{if } j \neq i, \end{cases} \quad (3.3)$$

where

$$s_i = \min \left\{ 1, \frac{G_i(\boldsymbol{\theta}) + \mu_i |\kappa_i|^2 / 2}{|\kappa_i|^2 (\mu_i + \|\mathbf{a}_i\|^2 / \beta)} \right\}. \quad (3.4)$$

We use the update \hat{h} as a baseline to define \mathcal{H} . \mathcal{H} is the class of all update rules $h : \mathbb{R}^n \times [n] \rightarrow \mathbb{R}^n$ such that $\forall \boldsymbol{\theta} \in \mathbb{R}^n$ and $i \in [d]$,

$$F(h(\boldsymbol{\theta}, i)) \leq F(\hat{h}(\boldsymbol{\theta}, i)), \text{ or} \quad (3.5)$$

$$\hat{F}_P(\boldsymbol{\theta}, h(\boldsymbol{\theta}, i)) \leq \hat{F}_P(\boldsymbol{\theta}, \hat{h}(\boldsymbol{\theta}, i)), \quad (3.6)$$

where

$$\hat{F}_P(\boldsymbol{\theta}, \boldsymbol{\theta}') = \sum_{i=1}^d \left((\nabla f(A\boldsymbol{\theta})^\top \mathbf{a}_i) (\theta'_i - \theta_i) + \frac{1}{2\beta} \|\mathbf{a}_i\|^2 (\theta'_i - \theta_i)^2 + g_i(\theta'_i) - g_i(\theta_i) \right). \quad (3.7)$$

Intuitively, $\widehat{F}_P(\boldsymbol{\theta}, \boldsymbol{\theta}')$ approximates the difference of the cost function evaluated at $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$, which follows from the smoothness property of f :

$$\begin{aligned} F(\boldsymbol{\theta}') - F(\boldsymbol{\theta}) &\leq \nabla f(A\boldsymbol{\theta})^\top \left(\sum_{i=1}^d \mathbf{a}_i(\theta'_i - \theta_i) \right) + \frac{1}{2\beta} \left\| \sum_{i=1}^d \mathbf{a}_i(\theta'_i - \theta_i) \right\|^2 + \sum_{i=1}^d g_i(\theta'_i) - g_i(\theta_i) \\ &\leq \sum_{i=1}^d \left((\nabla f(A\boldsymbol{\theta})^\top \mathbf{a}_i) (\theta'_i - \theta_i) + \frac{1}{2\beta} \|\mathbf{a}_i\|^2 (\theta'_i - \theta_i)^2 + g_i(\theta'_i) - g_i(\theta_i) \right), \end{aligned}$$

where the first inequality follows from the smoothness property of f and the last inequality follows from the triangle inequality.

We begin our analysis with a lemma that provides the marginal decrease r_i^t of updating a coordinate $i \in [d]$ according to any update rule in the class \mathcal{H} .

Lemma 3.1. *In (3.1), let f be $1/\beta$ -smooth and each g_i be μ_i -strongly convex with convexity parameter $\mu_i \geq 0 \forall i \in [d]$. For $\mu_i = 0$, we assume that g_i has a L -bounded support. After selecting the coordinate $i \in [d]$ and updating θ_i^t with an update rule in \mathcal{H} , we have the following guarantee:*

$$F(\boldsymbol{\theta}^{t+1}) \leq F(\boldsymbol{\theta}^t) - r_i^t, \quad (3.8)$$

where

$$r_i^t = \begin{cases} G_i^t - \frac{\|\mathbf{a}_i\|^2 |\kappa_i^t|^2}{2\beta} & \text{if } s_i^t = 1, \\ \frac{s_i^t (G_i^t + \mu_i |\kappa_i^t|^2 / 2)}{2} & \text{otherwise,} \end{cases} \quad (3.9)$$

and where s_i^t is given by (3.4).

In the proof of Lemma 3.1, the decrease of the cost function is upper-bounded using the smoothness property of $f(\cdot)$ and the convexity of $g_i(\cdot)$ for any update rule in the class \mathcal{H} . We present the proof below.

Proof of Lemma 3.1. We first prove the claim for the update rule \widehat{h} given by (3.3) in part (i), and next extend it to any update rule in \mathcal{H} in part (ii).

(i) Our starting point is the inequality

$$F(\boldsymbol{\theta}^{t+1}) \leq F(\boldsymbol{\theta}^t) - s_i^t G_i(\boldsymbol{\theta}^t) - \left(\frac{\mu_i (s_i^t - (s_i^t)^2)}{2} - \frac{(s_i^t)^2 \|\mathbf{a}_i\|^2}{2\beta} \right) |\kappa_i^t|^2, \quad (3.10)$$

which holds for $s_i^t \in [0, 1]$, for all $i \in [d]$ and which follows from Lemma 3.1 of [Perekrestenko et al., 2017].² After minimizing the right-hand side of (3.10) with respect to s_i^t , we attain the desired bound (3.8) for s_i^t as in (3.4).

(ii) We now extend (i) to any update rule in \mathcal{H} . If the update rule $h(\boldsymbol{\theta}^t, i)$ satisfies (3.5), we can easily recover (3.8) because

$$F(h(\boldsymbol{\theta}^t, i)) \leq F(\widehat{h}(\boldsymbol{\theta}^t, i)) \leq F(\boldsymbol{\theta}^t) - r_i^t.$$

If the update rule satisfies (3.6), we have

$$F(h(\boldsymbol{\theta}^t, i)) \leq F(\boldsymbol{\theta}^t) + \widehat{F}_P(\boldsymbol{\theta}^t, h(\boldsymbol{\theta}^t, i)) \quad (3.11)$$

$$\leq F(\boldsymbol{\theta}^t) + \widehat{F}_P(\boldsymbol{\theta}^t, \widehat{h}(\boldsymbol{\theta}^t, i)) \quad (3.12)$$

$$\leq F(\boldsymbol{\theta}^t) - r_i^t, \quad (3.13)$$

where (3.11) follows from the $1/\beta$ -smoothness of f and (3.7), (3.12) follows from (3.6), and (3.13) follows from the μ_i -strong convexity of g_i . More precisely, by plugging

$$\begin{aligned} g_i(\boldsymbol{\theta}_i^t + s_i^t \kappa_i^t) &= g_i(\boldsymbol{\theta}_i^t + s_i^t(u^t - x_i^t)) \leq \\ & s_i^t g_i(u^t) + (1 - s_i^t) g_i(\boldsymbol{\theta}_i^t) - \frac{\mu_i}{2} s_i^t (1 - s_i^t) (\kappa_i^t)^2 \end{aligned} \quad (3.14)$$

into (3.12), and using the Fenchel-Young property, we recover (3.10). Then, by setting s_i^t as in (3.4) we recover (3.13). \square

Remark 3.2. *In the well-known SGD, the cost function $F(\boldsymbol{\theta}^t)$ might increase at some iteration t . In contrast, if we use CD with an update rule in \mathcal{H} , it follows from (3.9) and (3.4) that $r_i^t \geq 0$ for all t , and from (3.8) that the cost function $F(\boldsymbol{\theta}^t)$ never increases. This property provides a strong stability guarantee, and explains (in part) the good performance observed in the experiments in Section 3.5.*

3.4.2 Greedy Algorithms (Full Information Setting)

In first setting, which we call full information setting, we assume that we have computed r_i^t for all $i \in [d]$ and all t (we will relax this assumption in Section 3.4.3). Our first algorithm `max_r` makes then a greedy use of Lemma 3.1, by simply choosing at time t the coordinate i with the largest r_i^t .

Proposition 3.3 (`max_r`). *Under the assumptions of Lemma 3.1, the optimal coordinate i_t for minimizing the right-hand side of (3.8) at time t is $i_t = \arg \max_{j \in [d]} r_j^t$.*

²This inequality improves variants in Theorem 2 of [Shalev-Shwartz and Zhang, 2013b], Lemma 2 of [Zhao and Zhang, 2015a] and Lemma 3 of [Csiba et al., 2015].

Remark 3.4. This rule can be seen as an extension of the Gauss-Southwell rule [Nutini et al., 2015] for the class of cost functions that the gradient does not exist, which selects the coordinate whose gradient has the largest magnitude (when $\nabla_i F(\boldsymbol{\theta})$ exists), i.e., $i_t = \arg \max_{i \in [d]} |\nabla_i F(\boldsymbol{\theta})|$. Indeed, Lemma 3.9 in Appendix 3.B shows that for the particular case of L_2 -regularized cost functions $F(\boldsymbol{\theta})$, the Gauss-Southwell rule and \max_r are equivalent.

If functions $g_i(\cdot)$ are strongly convex (i.e., $\mu_i > 0$), then \max_r results in a linear convergence rate and matches the lower bound in [Arjevani and Shamir, 2016].

Theorem 3.5. Let g_i in (3.1) be μ_i -strongly convex with $\mu_i > 0$ for all $i \in [d]$. Under the assumptions of Lemma 3.1, we have the following linear convergence guarantee:

$$\epsilon(\boldsymbol{\theta}^t) \leq \epsilon(\boldsymbol{\theta}^0) \prod_{l=1}^t \left(1 - \max_{i \in [d]} \frac{G_i(\boldsymbol{\theta}^l) \mu_i}{G(\boldsymbol{\theta}^l) \left(\mu_i + \frac{\|\mathbf{a}_i\|^2}{\beta} \right)} \right), \quad (3.15)$$

for all $t > 0$, where $\epsilon(\boldsymbol{\theta}^0)$ is the sub-optimality gap at $t = 0$.

The result is proven by induction. We distinguish the two cases in Lemma 3.1: $s_i^t = 1$ and $s_i^t < 1$. For both cases we show that the induction hypothesis holds. The complete proof is given below.

Proof of Theorem 3.5. According to Proposition 3.3, we know that the selection rule \max_r is optimal for the bound (3.8). Therefore, if we prove the convergence results using (3.8) for another selection rule, then the same convergence result holds for \max_r . For this proof, we use the following selection rule: At time t , we choose the coordinate i with the largest $G_i(\boldsymbol{\theta}^t) \mu_i / (\mu_i + \|\mathbf{a}_i\|^2 / \beta)$, which we denote by i^* .

First, we show that $r_{i^*}^t$ in (3.9) is lower bounded as follows

$$r_{i^*}^t \geq G_{i^*}(\boldsymbol{\theta}^t) \frac{\mu_{i^*}}{\mu_{i^*} + \frac{\|\mathbf{a}_{i^*}\|^2}{\beta}}. \quad (3.16)$$

We prove (3.16) for two cases $s_{i^*}^t = 1$ and $s_{i^*}^t < 1$ separately, where $s_{i^*}^t$ is defined in (3.4) for $i \in [d]$.

(a) If $s_{i^*}^t = 1$, according to (3.9) we have

$$r_{i^*}^t = G_{i^*}(\boldsymbol{\theta}^t) - \frac{\|\mathbf{a}_{i^*}\|^2 |\kappa_{i^*}^t|^2}{2\beta}.$$

Next, we prove (3.16) by showing that $r_{i^*}^t - G_{i^*}(\boldsymbol{\theta}^t) \frac{\mu_{i^*}}{\mu_{i^*} + \|\mathbf{a}_{i^*}\|^2 / \beta} \geq 0$,

$$\begin{aligned}
 r_{i^*}^t - G_{i^*}(\boldsymbol{\theta}^t) &= \frac{\mu_{i^*}}{\mu_{i^*} + \frac{\|\mathbf{a}_{i^*}\|^2}{\beta}} \\
 &= G_{i^*}(\boldsymbol{\theta}^t) \frac{\frac{\|\mathbf{a}_{i^*}\|^2}{\beta}}{\mu_{i^*} + \frac{\|\mathbf{a}_{i^*}\|^2}{\beta}} - \frac{\|\mathbf{a}_{i^*}\|^2 |\kappa_{i^*}^t|^2}{2\beta} \\
 &= \frac{\|\mathbf{a}_{i^*}\|^2}{2\beta} \cdot \frac{2G_{i^*}(\boldsymbol{\theta}^t) - \mu_{i^*} |\kappa_{i^*}^t|^2 - \frac{\|\mathbf{a}_{i^*}\|^2 |\kappa_{i^*}^t|^2}{\beta}}{\mu_{i^*} + \frac{\|\mathbf{a}_{i^*}\|^2}{\beta}} \geq 0,
 \end{aligned} \tag{3.17}$$

where the last inequality follows by setting $s_{i^*}^t = 1$ in (3.4) which then reads:

$$G_{i^*}(\boldsymbol{\theta}^t) - \frac{\mu_{i^*} |\kappa_{i^*}^t|^2}{2} - \frac{\|\mathbf{a}_{i^*}\|^2 |\kappa_{i^*}^t|^2}{\beta} \geq 0.$$

This proves (3.16).

(b) Now, if $s_{i^*}^t < 1$, according to (3.9) we have

$$r_{i^*}^t = \frac{(G_{i^*}(\boldsymbol{\theta}^t) + \mu_{i^*} |\kappa_{i^*}^t|^2 / 2)^2}{2(\mu_{i^*} + \frac{\|\mathbf{a}_{i^*}\|^2}{\beta}) |\kappa_{i^*}^t|^2}. \tag{3.18}$$

With $r_{i^*}^t$ given by (3.18), (3.16) becomes

$$\frac{(G_{i^*}(\boldsymbol{\theta}^t) + \mu_{i^*} |\kappa_{i^*}^t|^2 / 2)^2}{2(\mu_{i^*} + \frac{\|\mathbf{a}_{i^*}\|^2}{\beta}) |\kappa_{i^*}^t|^2} \geq G_{i^*}(\boldsymbol{\theta}^t) \frac{\mu_{i^*}}{\mu_{i^*} + \frac{\|\mathbf{a}_{i^*}\|^2}{\beta}},$$

and rearranging the items, it successively becomes

$$\begin{aligned}
 \frac{(G_{i^*}(\boldsymbol{\theta}^t) + \mu_{i^*} |\kappa_{i^*}^t|^2 / 2)^2}{2|\kappa_{i^*}^t|^2} &\geq G_{i^*}(\boldsymbol{\theta}^t) \mu_{i^*} \\
 (G_{i^*}(\boldsymbol{\theta}^t) + \mu_{i^*} |\kappa_{i^*}^t|^2 / 2)^2 &\geq 2G_{i^*}(\boldsymbol{\theta}^t) |\kappa_{i^*}^t|^2 \mu_{i^*} \\
 G_{i^*}(\boldsymbol{\theta}^t)^2 + (\mu_{i^*} |\kappa_{i^*}^t|^2 / 2)^2 - G_{i^*}(\boldsymbol{\theta}^t) |\kappa_{i^*}^t|^2 \mu_{i^*} &\geq 0 \\
 (G_{i^*}(\boldsymbol{\theta}^t) - \mu_{i^*} |\kappa_{i^*}^t|^2 / 2)^2 &\geq 0,
 \end{aligned}$$

which always holds and therefore recovers the claim, i.e., (3.16).

Hence in both cases (3.16) holds. Now, plugging (3.16) and $G(\boldsymbol{\theta}^t) \geq \epsilon(\boldsymbol{\theta}^t)$ in (3.8) yields

$$\begin{aligned} \epsilon(\boldsymbol{\theta}^{t+1}) - \epsilon(\boldsymbol{\theta}^t) &= F(\boldsymbol{\theta}^{t+1}) - F(\boldsymbol{\theta}^t) \leq -r_{i^*}^t \\ &\leq -G(\boldsymbol{\theta}^t) \max_{i \in [d]} \frac{G_i(\boldsymbol{\theta}^t) \mu_i}{G(\boldsymbol{\theta}^t) \left(\mu_i + \frac{\|\mathbf{a}_i\|^2}{\beta} \right)} \leq -\epsilon(\boldsymbol{\theta}^t) \max_{i \in [d]} \frac{G_i(\boldsymbol{\theta}^t) \mu_i}{G(\boldsymbol{\theta}^t) \left(\mu_i + \frac{\|\mathbf{a}_i\|^2}{\beta} \right)}, \end{aligned} \quad (3.19)$$

that results in

$$\epsilon(\boldsymbol{\theta}^{t+1}) \leq \epsilon(\boldsymbol{\theta}^t) - \epsilon(\boldsymbol{\theta}^t) \max_{i \in [d]} \frac{G_i(\boldsymbol{\theta}^t) \mu_i}{G(\boldsymbol{\theta}^t) \left(\mu_i + \frac{\|\mathbf{a}_i\|^2}{\beta} \right)}, \quad (3.20)$$

which gives

$$\epsilon(\boldsymbol{\theta}^{t+1}) \leq \epsilon(\boldsymbol{\theta}^t) \left(1 - \max_{i \in [d]} \frac{G_i(\boldsymbol{\theta}^t) \mu_i}{G(\boldsymbol{\theta}^t) \left(\mu_i + \frac{\|\mathbf{a}_i\|^2}{\beta} \right)} \right). \quad (3.21)$$

As (3.21) holds for all t , we conclude the proof. \square

Now, if functions $g_i(\cdot)$ are not necessary strongly convex (i.e., $\mu_i = 0$), `max_r` is also very effective and outperforms the state-of-the-art.

Theorem 3.6. *Under the assumptions of Lemma 3.1, let $\mu_i \geq 0$ for all $i \in [d]$. Then,*

$$\epsilon(\boldsymbol{\theta}^t) \leq \frac{8L^2\eta^2/\beta}{2d+t-t_0} \quad (3.22)$$

for all $t \geq t_0$, where $t_0 = \max\{1, 2d \log^{d\beta\epsilon(\boldsymbol{\theta}^0)}/4L^2\eta^2\}$, $\epsilon(\boldsymbol{\theta}^0)$ is the sub-optimality gap at $t = 0$ and $\eta = O(d)$ is an upper bound on $\min_{i \in [d]} G(\boldsymbol{\theta}^t) \|\mathbf{a}_i\|/G_i(\boldsymbol{\theta}^t)$ for all iterations $l \in [t]$.

The result is proven by induction. We distinguish the two cases in Lemma 3.1: $s_i^t = 1$ and $s_i^t < 1$. When $s_i^t = 1$, we show that $\epsilon(\boldsymbol{\theta}^t)$ decreases by a factor $1 - 1/2d$ (i.e., a linear convergence) and when $s_i^t < 1$, we lower bound r_i^t , next we validate the induction hypothesis in both cases. The complete proof is given below.

Proof of Theorem 3.6. Similar to the proof of Theorem 3.5, we prove the theorem for the following selection rule: At time t , the coordinate i with the largest $G_i(\boldsymbol{\theta}^t)$ is chosen. Since the optimal selection rule for minimizing the bound in Lemma 3.1 is to select the coordinate i with the largest r_i^t in (3.8), as shown by Proposition 3.3, the convergence guarantees provided here holds for `max_r` as well.

The bound (3.22) is proven by using induction.

Chapter 3. Coordinate Descent with Bandit Sampling

Suppose that (3.22) holds for some $t \geq t_0$. We want to verify it for $t + 1$. Let $i^* = \operatorname{argmax}_{i \in [d]} G_i(\boldsymbol{\theta}^t)$. We study two cases $s_{i^*}^t = 1$ and $s_{i^*}^t < 1$ separately, where s_i^t is defined in (3.4) for $i \in [d]$.

(a) If $s_{i^*}^t = 1$, then first we show that

$$\epsilon(\boldsymbol{\theta}^{t+1}) \leq \epsilon(\boldsymbol{\theta}^t) \cdot \left(1 - \frac{1}{2d}\right), \quad (3.23)$$

second we show that induction hypothesis (3.22) holds. Since $s_{i^*}^t = 1$, (3.4) yields that

$$G_{i^*}(\boldsymbol{\theta}^t) \geq \frac{|\kappa_{i^*}^t|^2 \|\mathbf{a}_{i^*}\|^2}{\beta} + \frac{\mu_{i^*} |\kappa_{i^*}^t|}{2},$$

that gives

$$G_{i^*}(\boldsymbol{\theta}^t) \geq \frac{|\kappa_{i^*}^t|^2 \|\mathbf{a}_{i^*}\|^2}{\beta},$$

which, combined with (3.9), implies that

$$r_{i^*}^t = G_{i^*}(\boldsymbol{\theta}^t) - \frac{|\kappa_{i^*}^t|^2 \|\mathbf{a}_{i^*}\|^2}{2\beta} \geq \frac{G_{i^*}(\boldsymbol{\theta}^t)}{2}. \quad (3.24)$$

Using $F(x^{t+1}) - F(x^t) = \epsilon(\boldsymbol{\theta}^{t+1}) - \epsilon(\boldsymbol{\theta}^t)$ and (3.24), we can rewrite (3.8) as

$$\epsilon(\boldsymbol{\theta}^{t+1}) - \epsilon(\boldsymbol{\theta}^t) \leq -\frac{G_{i^*}(\boldsymbol{\theta}^t)}{2}.$$

As i^* is the coordinate with the largest $G_i(\boldsymbol{\theta}^t)$, we have

$$\epsilon(\boldsymbol{\theta}^{t+1}) - \epsilon(\boldsymbol{\theta}^t) \leq -\frac{G_{i^*}(\boldsymbol{\theta}^t)}{2} \leq -\frac{G(\boldsymbol{\theta}^t)}{2d}. \quad (3.25)$$

According to weak duality, $\epsilon(\boldsymbol{\theta}^t) \leq G(\boldsymbol{\theta}^t)$. Plugging this in (4.4) yields

$$\epsilon(\boldsymbol{\theta}^{t+1}) - \epsilon(\boldsymbol{\theta}^t) \leq -\frac{G(\boldsymbol{\theta}^t)}{2d} \leq -\frac{\epsilon(\boldsymbol{\theta}^t)}{2d}, \quad (3.26)$$

and therefore

$$\epsilon(\boldsymbol{\theta}^{t+1}) \leq \epsilon(\boldsymbol{\theta}^t) \cdot \left(1 - \frac{1}{2d}\right). \quad (3.27)$$

Now, by plugging (3.22) in (3.27) we prove the inductive step at time $l + 1$:

$$\begin{aligned}\epsilon(\boldsymbol{\theta}^{t+1}) &\leq \frac{\frac{8L^2\eta^2}{\beta}}{2d+t-t_0} \left(1 - \frac{1}{2d}\right) \\ &\leq \frac{\frac{8L^2\eta^2}{\beta}}{2d+t+1-t_0}.\end{aligned}$$

(b) If $s_{i^*}^t < 1$, the marginal decreases in (3.9) becomes

$$r_{i^*}^t = \frac{\left(G_{i^*}(\boldsymbol{\theta}^t) + \mu_{i^*} \frac{|\kappa_{i^*}^t|^2}{2}\right)^2}{2|\kappa_{i^*}^t|^2 \left(\mu_{i^*} + \frac{\|\mathbf{a}_{i^*}\|^2}{\beta}\right)}. \quad (3.28)$$

Next, we show that

$$r_{i^*}^t \geq \frac{G_{i^*}^2(\boldsymbol{\theta}^t)\beta}{2|\kappa_{i^*}^t|^2 \|\mathbf{a}_{i^*}\|^2}. \quad (3.29)$$

To prove (3.29), we plug (3.28) in (3.29) and rearrange the terms which gives

$$\frac{\|\mathbf{a}_{i^*}\|^2}{\beta} \left(\mu_{i^*}^2 \frac{|\kappa_{i^*}^t|^4}{4} + G_{i^*}(\boldsymbol{\theta}^t) \mu_{i^*} |\kappa_{i^*}^t|^2 \right) \geq \mu_{i^*} G_{i^*}^2(\boldsymbol{\theta}^t), \quad (3.30)$$

(3.30) holds because of (3.4). More precisely, if we plug the value of $s_{i^*}^t < 1$ in (3.4) we get

$$G_{i^*}(\boldsymbol{\theta}^t) \leq \frac{|\kappa_{i^*}^t|^2 \|\mathbf{a}_{i^*}\|^2}{\beta} + \frac{\mu_{i^*} |\kappa_{i^*}^t|}{2}, \quad (3.31)$$

which shows the correctness of (3.30), hence (3.29).

According to Lemma 22 of [Shalev-Shwartz and Zhang, 2013b] or Lemma 2.7 of [Perekrestenko et al., 2017] we know $|\kappa_i^t| \leq 2L$. Plugging $|\kappa_i^t| \leq 2L$ in (3.29) yields

$$r_{i^*}^t \geq \frac{G_{i^*}^2(\boldsymbol{\theta}^t)\beta}{8L^2 \|\mathbf{a}_{i^*}\|^2}.$$

Next, using weak duality and the definition of η in Theorem 3.6, we lower bound $r_{i^*}^t$ by

$$\begin{aligned}r_{i^*}^t &\geq \left(\frac{G_{i^*}(\boldsymbol{\theta}^t)}{G(\boldsymbol{\theta}^t) \|\mathbf{a}_{i^*}\|} \right)^2 \frac{G^2(\boldsymbol{\theta}^t)\beta}{8L^2} \\ &\geq \frac{\epsilon^2(\boldsymbol{\theta}^t)\beta}{8L^2\eta^2}.\end{aligned} \quad (3.32)$$

Hence we have

$$\epsilon(\boldsymbol{\theta}^{t+1}) - \epsilon(\boldsymbol{\theta}^t) \leq -r_{i^*}^t \leq -\frac{\epsilon^2(\boldsymbol{\theta}^t)\beta}{8L^2\eta^2},$$

and therefore

$$\epsilon(\boldsymbol{\theta}^{t+1}) \leq \epsilon(\boldsymbol{\theta}^t) \left(1 - \frac{\epsilon(\boldsymbol{\theta}^t)\beta}{8L^2\eta^2}\right). \quad (3.33)$$

Let $f(y) = y(1 - y\beta/8L^2\eta^2)$, as $f'(y) > 0$ for $y < 4L^2\eta^2/\beta$, plugging (3.22) in (3.33) yields

$$\epsilon(\boldsymbol{\theta}^t) \left(1 - \frac{\epsilon(\boldsymbol{\theta}^t)\beta}{8L^2\eta^2}\right) \leq \frac{\frac{8L^2\eta^2}{\beta}}{2d+t-t_0} \left(1 - \frac{\frac{8L^2\eta^2}{\beta}}{2d+t-t_0} \frac{\beta}{8L^2\eta^2}\right). \quad (3.34)$$

Now, we prove the inductive step at time $t+1$ by using (3.34):

$$\begin{aligned} \epsilon(\boldsymbol{\theta}^{t+1}) &\leq \frac{\frac{8L^2\eta^2}{\beta}}{2d+t-t_0} \cdot \left(1 - \frac{\frac{8L^2\eta^2}{\beta}}{2d+t-t_0} \frac{\beta}{8L^2\eta^2}\right) \\ &\leq \frac{\frac{8L^2\eta^2}{\beta}}{2d+t+1-t_0}. \end{aligned}$$

To conclude the proof, we need to show that the induction base case is correct, i.e., we need to show that

$$\epsilon(\boldsymbol{\theta}^{t_0}) \leq \frac{4L^2\eta^2}{\beta d}. \quad (3.35)$$

First, we rewrite (3.9) using $r_{i^*}^t \geq \epsilon^2(\boldsymbol{\theta}^t)\beta/8L^2\eta^2$ for $s_{i^*}^t < 1$ and $r_{i^*}^t \geq \epsilon(\boldsymbol{\theta}^t)/2d$ for $s_{i^*}^t = 1$ as

$$\epsilon(\boldsymbol{\theta}^{t+1}) - \epsilon(\boldsymbol{\theta}^t) \leq -r_{i^*}^t \leq -1\{s_{i^*}^t = 1\} \frac{\epsilon(\boldsymbol{\theta}^t)}{2d} - 1\{s_{i^*}^t < 1\} \frac{\epsilon^2(\boldsymbol{\theta}^t)\beta}{8L^2\eta^2}. \quad (3.36)$$

From (3.36), for $l < t_0$ we have

$$\begin{aligned} \epsilon(\boldsymbol{\theta}^{t+1}) &\leq \epsilon(\boldsymbol{\theta}^t) \left(1 - 1\{s_{i^*}^t = 1\} \frac{1}{2d} - 1\{s_{i^*}^t < 1\} \frac{\epsilon(\boldsymbol{\theta}^t)\beta}{8L^2\eta^2}\right) \\ &\leq \epsilon(\boldsymbol{\theta}^t) \left(1 - \min\left\{\frac{1}{2d}, \frac{\epsilon(\boldsymbol{\theta}^t)\beta}{8L^2\eta^2}\right\}\right) \\ &\leq \epsilon(\boldsymbol{\theta}^t) \left(1 - \min\left\{\frac{1}{2d}, \frac{\epsilon(\boldsymbol{\theta}^{t_0})\beta}{8L^2\eta^2}\right\}\right), \end{aligned} \quad (3.37)$$

where (3.37) holds because for $t \leq t_0$ we know that $\epsilon(\boldsymbol{\theta}^{t_0}) \leq \epsilon(\boldsymbol{\theta}^t)$. We use the proof by contradiction to check the induction base, i.e., we show that assuming $\epsilon(\boldsymbol{\theta}^{t_0}) > 4L^2\eta^2/\beta d$

results in a contradiction. If $\epsilon(\boldsymbol{\theta}^{t_0}) > 4L^2\eta^2/\beta d$, then

$$\frac{1}{2d} = \min \left\{ \frac{1}{2d}, \frac{\epsilon(\boldsymbol{\theta}^{t_0})\beta}{8L^2\eta^2} \right\}. \quad (3.38)$$

From (3.37) and (3.38) we get

$$\epsilon(\boldsymbol{\theta}^{t_0}) \leq \epsilon(\boldsymbol{\theta}^0) \left(1 - \frac{1}{2d}\right)^{t_0}. \quad (3.39)$$

Using the inequality $1 + y < \exp(y)$ for $y < 1$ we have

$$\begin{aligned} \epsilon(\boldsymbol{\theta}^{t_0}) &\leq \epsilon(\boldsymbol{\theta}^0) \exp\left(-\frac{t_0}{2d}\right) = \epsilon(\boldsymbol{\theta}^0) \exp\left(-\log \frac{d\beta\epsilon(\boldsymbol{\theta}^0)}{4L^2\eta^2}\right) \\ &= \epsilon(\boldsymbol{\theta}^0) \frac{4L^2\eta^2}{\beta d\epsilon(\boldsymbol{\theta}^0)} = \frac{4L^2\eta^2}{\beta d}, \end{aligned}$$

which shows that the induction base holds and this concludes the proof. \square

To make the convergence bounds (3.15) and (3.22) easier to understand, assume that $\mu_i = \mu_1$ and that the data is normalized, so that $\|\mathbf{a}_i\| = 1$ for all $i \in [d]$. First, by letting $\eta = O(d)$ be an upper bound on $\min_{i \in [d]} G_i(\boldsymbol{\theta}^l)/G_i(\boldsymbol{\theta}^l)$ for all iterations $l \in [t]$, Theorem 3.5 results in a linear convergence rate, i.e., $\epsilon(\boldsymbol{\theta}^t) = O(\exp(-c_1 t/\eta))$ for some constant $c_1 > 0$ that depends on μ_1 and β , whereas Theorem 3.6 provides a sublinear convergence guarantee, i.e., $\epsilon(\boldsymbol{\theta}^t) = O(\eta^2/t)$.

Second, note that in both convergence guarantees, we would like to have a small η . The ratio η can be as large as d , when the different coordinate-wise gaps $G_i(\boldsymbol{\theta}^t)$ are equal. In this case, non-uniform sampling does not bring any advantage over uniform sampling, as expected. In contrast, if for instance $c \cdot G(\boldsymbol{\theta}^t) \leq \max_{i \in [d]} G_i(\boldsymbol{\theta}^t)$ for some constant $1/d \leq c \leq 1$, then choosing the coordinate with the largest r_i^t results in a decrease in the cost function, that is $1 \leq c \cdot d$ times larger compared to uniform sampling.

Finally, let us compare the bound of `max_r` given in Theorem 3.6 with the state-of-the-art bounds of `ada_gap` in Theorem 3.7 of [Perekrestenko et al., 2017] and of CD algorithm in Theorem 2 of [Dünner et al., 2017]. For the sake of simplicity, assume that $\|\mathbf{a}_i\| = 1$ for all $i \in [d]$. When $c \cdot G(\boldsymbol{\theta}^t) \leq \max_{i \in [d]} G_i(\boldsymbol{\theta}^t)$ and some constant $1/d \leq c \leq 1$, the convergence guarantee for `ada_gap` is $\mathbb{E}[\epsilon(\boldsymbol{\theta}^t)] = O\left(\sqrt{d}L^2/\beta(c^2+1/d)^{3/2}(2d+t)\right)$ and the convergence guarantee of the CD algorithm in [Dünner et al., 2017] is $\mathbb{E}[\epsilon(\boldsymbol{\theta}^t)] = O(dL^2/\beta c(2d+t))$, which are much tighter than the convergence guarantee of CD with uniform sampling $\mathbb{E}[\epsilon(\boldsymbol{\theta}^t)] = O(d^2L^2/\beta(2d+t))$. In contrast, the convergence guarantee of `max_r` is $\epsilon(\boldsymbol{\theta}^t) = O(L^2/\beta c^2(2d+t))$, which is \sqrt{d}/c times better than `ada_gap`, cd times better than the CD algorithm in [Dünner et al., 2017] and c^2d^2 times better than uniform sampling for the same constant $c \geq 1/d$.

Algorithm 3.1 B_max_r

```

1: Input:  $\theta^0$ ,  $\varepsilon$  and  $E$ 
2: Initialize: set  $\bar{r}_i^0 = r_i^0$  for all  $i \in [d]$ 
3: for  $t = 1$  to  $T$  do
4:   if  $t \bmod E == 0$  then
5:     set  $\bar{r}_i^t = r_i^t$  for all  $i \in [d]$ 
6:   end if
7:   Generate  $K \sim \text{Bern}(\varepsilon)$ 
8:   if  $K == 1$  then
9:     Select  $i_t \in [d]$  uniformly at random
10:  else
11:    Select  $i_t = \arg \max_{i \in [d]} \bar{r}_i^t$ 
12:  end if
13:  Update  $\theta_{i_t}^t$  by an update rule in  $\mathcal{H}$ 
14:  Set  $\bar{r}_{i_t}^{t+1} = r_{i_t}^{t+1}$  and  $\bar{r}_i^{t+1} = \bar{r}_i^t$  for all  $i \neq i_t$ 
15: end for

```

Remark 3.7. *There is no randomness in the selection rule used in max_r (beyond tie breaking), hence the convergence results given in Theorems 3.5 and 3.6 a.s. hold for all t .*

3.4.3 Bandit Algorithms (Partial Information Setting)

State-of-the-art algorithms and max_r require knowing a sub-optimality metric (e.g., G_i^t in [Perekrestenko et al., 2017, Dünner et al., 2017], the norm of gradient $\nabla_i F(\theta^t)$ in [Nutini et al., 2015], the marginal decreases r_i^t in this work) for all coordinates $i \in [d]$, which can be computationally expensive if the number of coordinates d is large. To overcome this problem, we use a novel approach inspired by the bandit framework that *learns* the best coordinates over time from the partial information it receives during the training.

In our second algorithm B_max_r, the marginal decreases r_i^t computed for all $i \in [d]$ at each round t by max_r are replaced by estimates \bar{r}_i computed by a multi-armed bandit algorithm (MAB) as follows. First, time is divided into bins of size E . At the beginning of a bin t_e , the marginal decreases $r_i^{t_e}$ of all coordinates $i \in [d]$ are computed, and the estimates are set to these values ($\bar{r}_i^t = r_i^{t_e}$ for all $i \in [d]$). At each iteration $t_e \leq t \leq t_e + E$ within that bin, with probability ε a coordinate $i_t \in [d]$ is selected uniformly at random, and otherwise (with probability $(1 - \varepsilon)$) the coordinate with the largest \bar{r}_i^t is selected. Coordinate i_t is next updated, as well as the estimate of the marginal decrease $\bar{r}_{i_t}^{t+1} = r_{i_t}^{t+1}$, whereas the other estimates \bar{r}_j^{t+1} remain unchanged for $j \neq i_t$. The algorithm can be seen as a modified version of ε -greedy that is developed for the setting where the reward of arms follow a fixed probability distribution, ε -greedy uses the empirical mean of the observed rewards as an estimate of the rewards (see Section 1.2.3 for more details about ε -greedy). In contrast, in our setting, the rewards do not follow

such a fixed probability distribution and the most recently observed reward is the best estimate of the reward that we could have. In `B_max_r`, we choose E not too large and ε large enough such that every arm (coordinate) is sampled often enough to maintain an accurate estimate of the rewards r_i^t (we use $E = O(d)$ and $\varepsilon = 1/2$ in the experiments of Section 3.5).

The next proposition shows the effect of the estimation error on the convergence rate.

Proposition 3.8. *Consider the same assumptions as Lemma 3.1 and Theorem 3.6. For simplicity, let $\|\mathbf{a}_i\| = \|\mathbf{a}_1\|$ for all $i \in [d]$ and $\epsilon(\boldsymbol{\theta}^0) \leq \sqrt{2\alpha L^2 \|\mathbf{a}_1\|^2 / \beta (\varepsilon/d + 1 - \varepsilon/c)} = O(d)$.³ Let $j_*^t = \arg \max_{i \in [d]} \bar{r}_i^t$. If $\max_{i \in [d]} r_i^t / r_{j_*^t}^t \leq c(E, \varepsilon)$ for some finite constant $c = c(E, \varepsilon)$, then by using `B_max_r` (with bin size E and exploration parameter ε) we have*

$$\mathbb{E} \left[\epsilon(\boldsymbol{\theta}^t) \right] \leq \frac{\alpha}{2 + t - t_0}, \quad \text{where } \alpha = \frac{8L^2 \|\mathbf{a}_1\|^2}{\beta (\varepsilon/d^2 + (1-\varepsilon)/\eta^2 c)}, \quad (3.40)$$

for all $t \geq t_0 = \max \{1, 4\epsilon(\boldsymbol{\theta}^0)/\alpha \log(2\epsilon(\boldsymbol{\theta}^0)/\alpha)\} = O(d)$ and where η is an upper bound on $\min_{i \in [d]} G_i(\boldsymbol{\theta}^l)/G_i(\boldsymbol{\theta}^t)$ for iterations $l \in [t]$.

What is the effect of $c(E, \varepsilon)$? In Proposition 3.8, $c = c(E, \varepsilon)$ upper bounds the estimation error of the marginal decreases r_i^t . To make the effect of $c(E, \varepsilon)$ on the convergence bound (3.40) easier to understand, let $\varepsilon = 1/2$, then $\alpha \sim 1/(1/d^2 + 1/\eta^2 c)$. We can see from the convergence bound (3.40) and the value of α that if c is large, the convergence rate is proportional to d^2 similarly to uniform sampling (i.e., $\epsilon(\boldsymbol{\theta}^t) \in O(d^2/t)$). Otherwise, if c is small, the convergence rate is similar to `max_r` ($\epsilon(\boldsymbol{\theta}^t) \in O(\eta^2/t)$, see Theorem 3.6).

How to control $c = c(E, \varepsilon)$? We can control the value of c by varying the bin size E . Doing so, there is a trade-off between the value of c and the average computational cost of an iteration. On the one hand, if we set the bin size to $E = 1$ (i.e., full information setting), then $c = 1$ and `B_max_r` boils down to `max_r`, while the average computational cost of an iteration is $O(nd)$. On the other hand, if $E > 1$ (i.e., partial information setting), then $c \geq 1$, while the average computational complexity of an iteration is $O(nd/E)$. In our experiments, we find that by setting $d/2 \leq E \leq d$, `B_max_r` converges faster than uniform sampling (and other state-of-the-art methods) while the average computational cost of an iteration is $O(n + \log d)$, similarly to the computational cost of an iteration of CD with uniform sampling ($O(n)$), see Figures 3.2 and 3.3. We also find that any exploration parameter $\varepsilon \in [0.2, 0.7]$ in `B_max_r` works reasonably well. The proof of Proposition 3.8 is similar to the proof of Theorem 3.6 and is given in Appendix 3.B.

³These assumptions are not necessary but they make the analysis simpler. For example, even if $\epsilon(\boldsymbol{\theta}^0)$ does not satisfy the required condition, we can scale down $F(\boldsymbol{\theta})$ by m so that $F(\boldsymbol{\theta})/m$ is minimized. The new sub-optimality gap becomes $\epsilon(\boldsymbol{\theta}^0)/m$, and for a sufficiently large m the initial condition is satisfied.

Table 3.1 – The shaded rows correspond to the algorithms introduced in this work. \bar{z} denotes the number of non-zero entries of the data matrix A . The numbers below the column dataset/cost are the clock time (in seconds) needed for the algorithms to reach a sub-optimality gap of $\epsilon(\theta^t) = \exp(-5)$.

method	computational cost (per epoch)	dataset/cost		
		aloi/Lasso	a9a/log reg	usps/ridge reg
uniform	$O(\bar{z})$	27.8	11.8	1
ada_gap	$O(d \cdot \bar{z})$	52.8	42.4	88
max_r	$O(d \cdot \bar{z})$	6.2	4.5	9.5
gap_per_epoch	$O(\bar{z} + d \log d)$	75	11.1	300
Approx	$O(\bar{z} + d \log d)$	16.3	2.3	-
NUACDM	$O(\bar{z} + d \log d)$	-	-	6
B_max_r	$O(\bar{z} + d \log d)$	11	1.9	1

Comparing CD to SGD with multi-armed bandit sampling In Chapter 2, we developed a multi-armed bandit algorithm for selecting datapoints for stochastic gradient descent algorithm (SGD). The datapoint-selection algorithm for SGD is inspired by adversarial multi-armed bandit algorithms, whereas the coordinate-selection algorithm for CD is inspired by stochastic multi-armed bandit algorithms. In SGD, an update of the parameters θ changes the rewards for all datapoints, and this change of the rewards does not follow any simple pattern. As a result, we study the datapoint-selection in an adversarial setting. Contrary to SGD, in CD update of a coordinate θ_i does not change the rewards of other coordinates much. As a result, we study the coordinate-selection in a stochastic setting. We also notice that for the cost functions that have the form in (3.1) CD is more effective than SGD, because CD uses a better update rule (3.3) than SGD’s update rule (2.2). However, we note that SGD is applicable to a broader set of cost functions (most of the finite-sum cost functions for which gradients could be computed).

3.5 Empirical Evaluation

We compare the algorithms from this work with the state-of-the-art approaches, in two ways. First, we compare the algorithm (max_r) for full information setting as in Section 3.4.2 against other state-of-the-art methods that similarly use $O(d \cdot \bar{z})$ computations per epoch of size d , where \bar{z} denotes the number of non-zero elements of A . Next, we compare the algorithm for partial information setting as in Section 3.4.3 (B_max_r) against other methods with appropriate heuristic modifications that also allow them to use $O(\bar{z})$ computations per epoch. The datasets we use are found in [Chang and Lin, 2011]; we consider usps, aloi and protein for regression, and w8a and a9a for binary classification (see Table 3.2 for statistics about these datasets).

Table 3.2 – Statistics of the datasets. The first three datasets are used for regression and the last two for binary classification.

	#classes	#datapoints	#features	%nonzero
usps	10	7291	256	100%
aloi	1000	108000	128	24%
protein	3	17766	357	29%
w8a	2	49749	300	4%
a9a	2	32561	123	11%

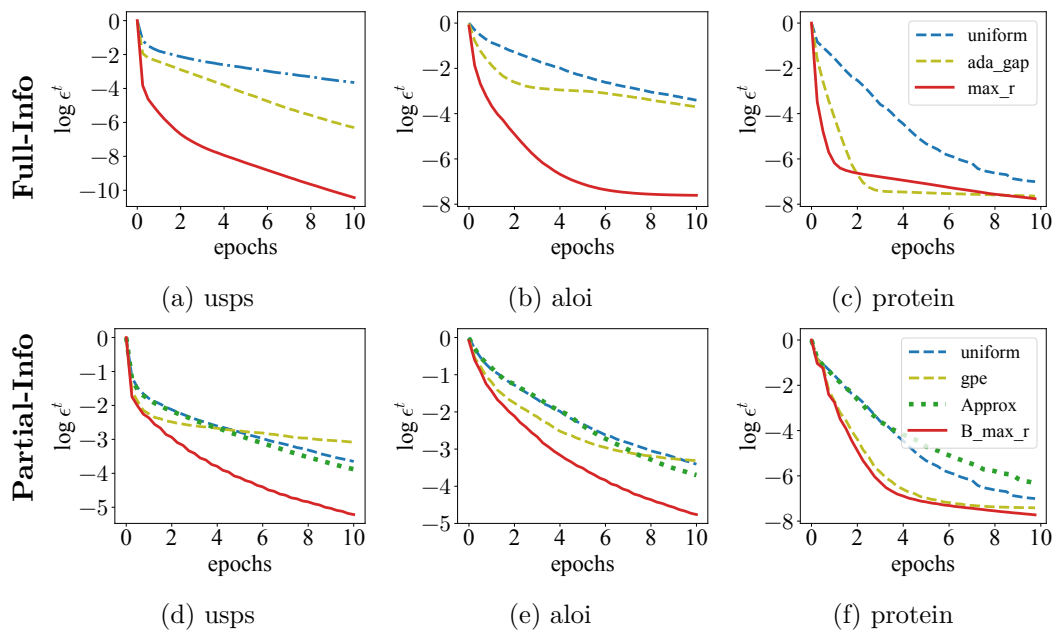


Figure 3.2 – CD for regression using Lasso (i.e., a non-smooth cost function). Y-axis is the log of sub-optimality gap and x-axis is the number of epochs. The algorithms presented in this work (max_r, B_max_r) outperform the state-of-the-art across the board.

Various cost functions are considered for the experiments, including a strongly convex cost function (ridge regression) and non-smooth cost functions (Lasso and L_1 -regularized logistic regression). These cost functions are optimized using different algorithms, which minimize either the primal or the dual cost function. The convergence time is the metric that we use to evaluate different algorithms.

3.5.1 Experimental Setup

Benchmarks for Adaptive Algorithm (max_r):

- **uniform** [Shalev-Shwartz and Tewari, 2011]: Sample a coordinate $i \in [n]$ uniformly at random.⁴
- **ada_gap** [Perekrestenko et al., 2017]: Sample a coordinate $i \in [n]$ with probability $G_i(\boldsymbol{\theta}^t)/G(\boldsymbol{\theta}^t)$.

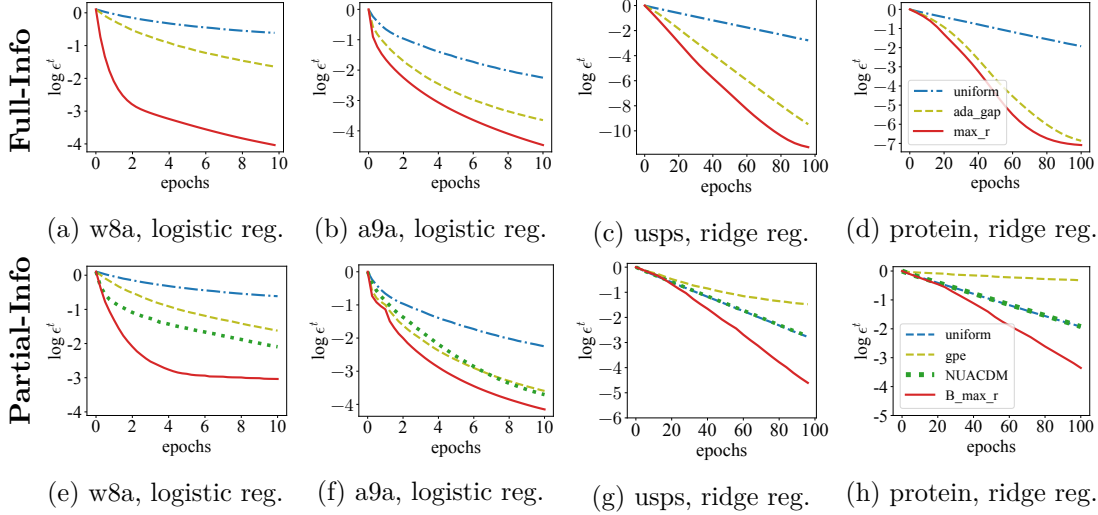


Figure 3.3 – CD for binary Classification using L_1 -regularized logistic regression and CD for regression using Lasso. The algorithms presented in this work (`max_r` and `B_max_r`) outperform the state-of-the-art across the board.

Benchmarks for Adaptive-Bandit Algorithm (`B_max_r`):

For comparison, in addition to the uniform sampling, we consider the coordinate selection method that has the best performance empirically in [Perekrestenko et al., 2017] and two accelerated CD methods NUACDM in [Allen-Zhu et al., 2016] and Approx in [Ferroq and Richtárik, 2015].

- **gap_per_epoch** [Perekrestenko et al., 2017]: This algorithm is a heuristic version of `ada_gap`, where the sampling probability $p_i^t = G_i(\boldsymbol{\theta}^t)/G(\boldsymbol{\theta}^t)$ for $i \in [d]$ is re-computed once at the beginning of each bin of length E .
- **NUACDM** [Allen-Zhu et al., 2016]: Sample a coordinate $i \in [d]$ with probability proportional to the square root of smoothness of the cost function along the i^{th} coordinate, then use an unbiased estimator for the gradient to update the decision variables. NUACDM is the state-of-the-art accelerated CD method (see Figures 2 and 3 in [Allen-Zhu et al., 2016]) and was proposed for smooth cost functions.
- **Approx** [Ferroq and Richtárik, 2015]: Sample a coordinate $i \in [d]$ uniformly at random, then use an unbiased estimator for the gradient to update the decision variables. Approx is an accelerated CD method and was proposed for cost functions

⁴If $\|\mathbf{a}_i\| = \|\mathbf{a}_j\| \forall i, j \in [n]$, importance sampling method in [Zhao and Zhang, 2015a] is equivalent to uniform in Lasso and logistic regression.

that have non-smooth $g_i(\cdot)$ in (3.1). We implemented Approx for such cost functions in Lasso and L_1 -regularized logistic regression.

We also implemented Approx for ridge-regression but NUACDM converged faster in our setting, whereas for the smoothen version of Lasso but Approx converged faster than NUACDM in our setting. There are other adaptive sampling methods in [Perekrestenko et al., 2017, Stich et al., 2017, Rakotomamonjy et al., 2017], but the ones compared above yield the best performance. The origin of the computational cost is two-fold: Sampling a coordinate i and updating it. The average computational cost of the algorithms for $E = d/2$ is depicted in Table 3.1. Next, we explain the setups and update rules used in the experiments.

Setup and update rule for Lasso: For Lasso

$$F(\boldsymbol{\theta}) = 1/2n\|\mathbf{Y} - A\boldsymbol{\theta}\|^2 + \sum_{i=1}^n \lambda|\theta_i|.$$

We consider the stingyCD update proposed in [Johnson and Guestrin, 2017]:

$$\theta_i^{t+1} = \arg \min_z \left[f(A\boldsymbol{\theta}^t + (z - \theta_i^t)\mathbf{a}_i) \right] + g_i(\theta_i).$$

This update rule belongs to the class \mathcal{H} . In Lasso, the g_i s are not strongly convex ($\mu_i = 0$). Therefore, for computing the dual residue, the Lipschitzing technique in [Dünner et al., 2016] is used, i.e., $g_i(\cdot)$ is assumed to have bounded support of size $B = F(\boldsymbol{\theta}^0)/\lambda$ and $g_i^*(u_i) = B \max\{|u_i| - \lambda, 0\}$. For both aloi and protein $\lambda = 10^{-5}$, and for usps $\lambda = 10^{-3}$. The total number of iterations is $T = 10d$.

Setup and update rule for logistic regression: For logistic regression

$$F(\boldsymbol{\theta}) = 1/n \sum_{i=1}^n \log \left(1 + \exp(-y_i \cdot \boldsymbol{\theta}^\top \mathbf{a}_i) \right) + \sum_{i=1}^n \lambda|\theta_i|.$$

We consider the update rule proposed in [Shalev-Shwartz and Tewari, 2011]:

$$\theta_i^{t+1} = s_{4\lambda}(\theta_i^t - 4\partial f(A\boldsymbol{\theta}^t)/\partial \theta_i),$$

where $s_\lambda(q) = \text{sign}(q) \max\{|q| - \lambda, 0\}$. This update rule also belongs to \mathcal{H} . For both datasets w8a and a9a, we again have $\lambda = 10^{-3}$ and $T = 10d$.

Setup and update rule for ridge regression: For ridge regression

$$F(\boldsymbol{\theta}) = 1/n\|\mathbf{Y} - A\boldsymbol{\theta}\|^2 + \lambda/2\|\boldsymbol{\theta}\|^2$$

and it is strongly convex. We consider the update proposed for the dual of ridge regression in [Shalev-Shwartz and Zhang, 2013b], hence B_max_r and other adaptive methods

select one of the dual decision variables to update. This update rule also belongs to \mathcal{H} . For both datasets usps and protein, we have $\lambda = 2 \times 10^{-5}$ and $T = 100d$.

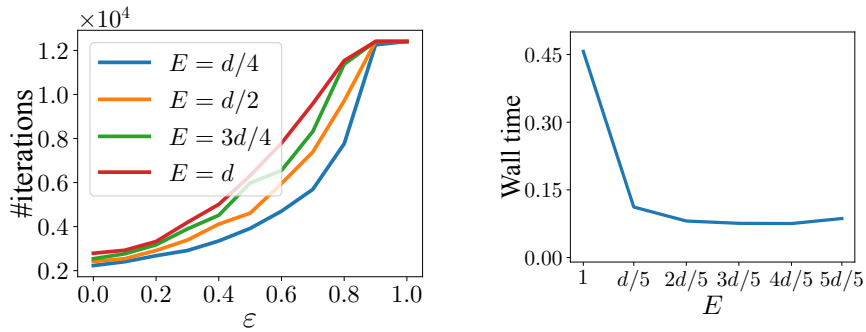
In all experiments, λ s are chosen such that the test and train errors are comparable. In addition, in all experiments, $E = d/2$ in B_max_r and gap_per_epoch. Recall that when minimizing the primal, d is the number of features and when minimizing the dual, d is the number of datapoints. In all three cost functions, recall that d is the number of features and \mathbf{a}_i are normalized to the same value $\|\mathbf{a}_i\|$.

3.5.2 Empirical Results

Figure 3.2 shows the result for Lasso. Among the adaptive algorithms, max_r outperforms the state-of-the-art (see Figures 3.2a, 3.2b and 3.2c). Among the adaptive-bandit algorithms, B_max_r outperforms the benchmarks (see Figures 3.2d, 3.2e and 3.2f). We also see that B_max_r converges slower than max_r for the same number of iterations, but we note that an iteration of B_max_r is $O(d)$ times cheaper than max_r. For logistic regression, see Figures 3.3a, 3.3b, 3.3e and 3.3f. Again, those algorithms outperform the state-of-the-art. We also see that B_max_r converges with the same rate as max_r. We see that the accelerated CD method Approx converges faster than uniform sampling and gap_per_epoch, but using B_max_r improves the convergence rate and reaches a lower sub-optimality gap ϵ with the same number of iterations. For ridge regression, we see in Figures 3.3c, 3.3d that max_r converges faster than the state-of-the-art ada-gap. We also see in Figures 3.3g, 3.3h that B_max_r converges faster than other algorithms. gap_per_epoch performs poorly because it is unable to adapt to the variability of the coordinate-wise duality gaps G_i that vary a lot from one iteration to the next. In contrast, this variation slows down the convergence of B_max_r compared to max_r, but B_max_r is still able to cope with this change by exploring and updating the estimations of the marginal decreases. In the experiments we report the sub-optimality gap as a function of the number of iterations, but the results are also favourable when we report them as a function of actual time. To clarify, we compare the clock time⁵ needed by each algorithm to reach a sub-optimality gap $\epsilon(\theta^t) = \exp(-5)$ in Table 3.1.

Next, we study the choice of parameters ε and E in Algorithm 3.1. As explained in Section 3.4.3 the choice of these two parameters affect c in Proposition 3.8, hence the convergence rate. To test the effect of ε and E on the convergence rate, we choose a9a dataset and perform a binary classification on it by using the logistic regression cost function. Figure 3.4a depicts the number of iterations required to reach the log-suboptimality gap $\log \epsilon$ of -5 . In the top-right corner, $\varepsilon = 1$ and B_max_r becomes CD with uniform sampling (for any value of E). As expected, for any ε , the smaller

⁵In our numerical experiments, all algorithms are optimized as much as possible by avoiding any unnecessary computations, by using efficient data structures for sampling, by reusing the computed values from past iterations and (if possible) by writing the computations in efficient matrix form.



(a) Number of iterations to reach $\log \epsilon(\theta^t) = -5$.

(b) Per-epoch clock time for different values of E .

Figure 3.4 – Analysis of the running time of `B_max_r` for different values of ϵ and E . A smaller E results in fewer iterations, and results in larger clock time per epoch (an epoch is d iterations of CD).

E , the smaller the number of iterations to reach the log-suboptimality gap of -5 . This means that $c(\epsilon, E)$ is a decreasing function of E . Also, we see that as ϵ increases, the convergence becomes slower. That implies that for this dataset and cost function $c(\epsilon, E)$ is close to 1 for all ϵ hence there is no need for exploration and a smaller value for ϵ can be chosen. Figure 3.4b depicts the per epoch clock time for $\epsilon = 0.5$ and different values of E . Note that the clock time is not a function of ϵ . As expected, a smaller bin size E results in a larger clock time, because we need to compute the marginal decreases for all coordinates more often. After $E = 2d/5$ we see that clock time does not decrease much, this can be explained by the fact that for large enough E computing the gradient takes more clock time than computing the marginal decreases.

3.6 Summary

In this chapter, we propose a new approach to select the coordinates to update in CD methods. We derive a lower bound on the decrease of the cost function in Lemma 3.1, i.e., the marginal decrease, when a coordinate is updated, for a large class of update methods \mathcal{H} . We use the marginal decreases to quantify how much updating a coordinate improves the model. Next, we use a bandit algorithm to *learn* which coordinates decrease the cost function significantly throughout the course of the optimization algorithm by using the marginal decreases as feedback (see Figure 3.1). We show that the approach converges faster than state-of-the-art approaches both theoretically and empirically. We emphasize that our coordinate selection approach is quite general and works for a large class of update rules \mathcal{H} , which includes Lasso, SVM, ridge and logistic regression, and a large class of bandit algorithms that select the coordinate to update.

The bandit algorithm `B_max_r` uses only the marginal decrease of the selected coordinate to update the estimations of the marginal decreases. An important open question is to

Chapter 3. Coordinate Descent with Bandit Sampling

understand the effect of having additional budget to choose multiple coordinates at each time t . The challenge lies in designing appropriate algorithms to invest this budget to update the coordinate selection strategy such that B_{\max_r} performance becomes even closer to \max_r .

Appendix

3.A Basic Definitions

For completeness, in this section we recall a variety of standard definitions.

3.A.1 Basic Definitions

Definition ($1/\beta$ -smooth). A function $f(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}$ is $1/\beta$ -smooth if for any $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^n$

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) + \frac{1}{2\beta} \|\mathbf{x} - \mathbf{y}\|^2.$$

Definition (μ -strongly convex). A function $f(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}$ is μ -strongly convex if for any $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^n$

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) + \frac{\mu}{2} \|\mathbf{x} - \mathbf{y}\|^2.$$

Definition (L -bounded support). A function $f(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}$ has L -bounded support if there exists a euclidean ball with radius L such

$$f(\boldsymbol{\theta}) < \infty \Rightarrow \|\boldsymbol{\theta}\| \leq L.$$

3.B Proofs

In this section, we present the proofs of the results in Sections 3.4.1, 3.4.2 and 3.4.3.

Lemma 3.9. *Under the assumptions of Lemma 3.1, if $g_i(\theta_i) = \lambda \cdot (\theta_i)^2$ in (3.1) and $\|\mathbf{a}_i\| = 1$ for all $i \in [d]$, then the Gauss-Southwell rule and \max_r are equivalent.*

Chapter 3. Coordinate Descent with Bandit Sampling

Proof of Lemma 3.9. We prove the lemma for $\boldsymbol{\theta} \in \mathbb{R}^n$ and drop the dependence on t throughout the proof. First, we show that $G_i \sim (\nabla_i F(\boldsymbol{\theta}))^2$. The function $g_i(\theta_i) = \lambda \cdot (\theta_i)^2$ is 2λ strongly convex for all $i \in [d]$, i.e., $\mu_i = \mu = 2\lambda$. The dual convex conjugate of the function $g_i(\theta_i) = \lambda \cdot (\theta_i)^2$ is

$$g^*(z) = \frac{z^2}{4\lambda}.$$

Then, for $\mathbf{w} = \nabla f(A\boldsymbol{\theta})$, $G_i(\boldsymbol{\theta}) = g_i^*(-\mathbf{a}_i^\top \mathbf{w}) + g_i(\theta_i) + \theta_i \mathbf{a}_i^\top \mathbf{w}$ becomes

$$G_i(\boldsymbol{\theta}) = \frac{(\mathbf{a}_i^\top \mathbf{w})^2}{4\lambda} + \lambda(\theta_i)^2 + \theta_i \mathbf{a}_i^\top \mathbf{w} = \frac{(\mathbf{a}_i^\top \mathbf{w} + 2\lambda\theta_i)^2}{4\lambda}.$$

As $\nabla_i F(\boldsymbol{\theta}) = \mathbf{a}_i^\top \nabla f(A\boldsymbol{\theta}) + 2\lambda\theta_i = \mathbf{a}_i^\top \mathbf{w} + 2\lambda\theta_i$, we have

$$G_i(\boldsymbol{\theta}) = \frac{(\nabla_i F(\boldsymbol{\theta}))^2}{4\lambda}.$$

Next, note that

$$\kappa_i = \partial g_i^*(-\mathbf{a}_i^\top \mathbf{w}) - \theta_i = \frac{-\mathbf{a}_i^\top \mathbf{w}}{2\lambda} - \theta_i = -\frac{\nabla_i F(\boldsymbol{\theta})}{2\lambda}.$$

Next, plugging $G_i(\boldsymbol{\theta}) = (\nabla_i F(\boldsymbol{\theta}))^2/4\lambda$ and $\kappa_i = -\nabla_i F(\boldsymbol{\theta})/2\lambda$ in (3.4) yields

$$s_i = \min \left\{ 1, \frac{3\lambda}{2\lambda + \frac{1}{\beta}} \right\} \text{ for all } i \in [d].$$

Hence, r_i in (3.9) becomes

$$r_i = \begin{cases} \frac{(\nabla_i F(\boldsymbol{\theta}))^2}{8\lambda^2\beta} (2\lambda\beta - 1) & \text{if } \lambda \geq \frac{1}{\beta}, \\ \frac{3}{4} \frac{(\nabla_i F(\boldsymbol{\theta}))^2}{2\lambda + \frac{1}{\beta}} & \text{otherwise.} \end{cases}$$

As a result, $\arg \max_{i \in [d]} r_i = \arg \max_{i \in [d]} (\nabla_i F(\boldsymbol{\theta}))^2$. In Gauss-Southwell rule, we choose the coordinate whose gradient has the largest magnitude, i.e., $\arg \max_{i \in [d]} |\nabla_i F(\boldsymbol{\theta})|$. Therefore, the selection rules \max_r and Gauss-Southwell rule are equivalent. \square

Proof of Proposition 3.8. The proof is similar to the proof of Theorem 3.6 and it uses induction. We highlight the differences here. To make the proof easier, we simplify the definition of s_i^t in (3.4) and the marginal decrease r_i^t in (3.9) by using the upper bound $|\kappa_i^t| \leq 2L$ (recall that $L = L_i$ for all i in Proposition 3.8). The upper bound $|\kappa_i^t| \leq 2L$ follows from Lemma 22 of [Shalev-Shwartz and Zhang, 2013b]. The starting point of the

proof is the following equation

$$F(\boldsymbol{\theta}^{t+1}) \leq F(\boldsymbol{\theta}^t) - s_i^t G_i(\boldsymbol{\theta}^t) + 2 \frac{(s_i^t)^2 \|\mathbf{a}_1\|^2}{\beta} L^2, \quad (3.41)$$

which is derived by upper bounding (3.10) using $|\kappa_i^t| \leq 2L$ and setting $\mu_i = 0$ for all $i \in [d]$, which holds since $g_i(\cdot)$ are not strongly convex. Equation (3.41) holds for $s_i^t \in [0, 1]$, and for all $i \in [d]$. After minimizing the right-hand side of (3.41) with respect to s_i^t , we attain the following *new* s_i^t and the *new* marginal decrease r_i^t :

$$s_i^t = \min \left\{ 1, \frac{G_i^t}{4L^2 \|\mathbf{a}_1\|^2 / \beta} \right\}, \quad (3.42)$$

and

$$r_i^t = \begin{cases} G_i^t - \frac{2\|\mathbf{a}_1\|^2 L^2}{\beta} & \text{if } s_i^t = 1, \\ \frac{(G_i^t)^2}{8L^2 \|\mathbf{a}_1\|^2 / \beta} & \text{otherwise.} \end{cases} \quad (3.43)$$

Hereafter, let

$$\alpha = \frac{8L^2 \|\mathbf{a}_1\|^2}{\beta (\varepsilon/d^2 + (1-\varepsilon)\eta^2/c)}$$

as defined in Proposition 3.8.

Now, suppose that (3.22) holds for some $t \geq t_0$. We want to verify it for $t + 1$. We start the analysis by computing the expected marginal decrease for ε in Algorithm 3.1,

$$\mathbb{E} [r_i^t | \boldsymbol{\theta}^t] \geq \frac{\varepsilon}{d} \left(\sum_{s_i^t=1} r_i^t + \sum_{s_i^t < 1} r_i^t \right) + (1 - \varepsilon) \frac{r_{i^*}^t}{c}, \quad (3.44)$$

where c is a finite constant in Proposition 3.8 and $i^* = \arg \max_{i \in [d]} r_i^t$. The expectation is with respect to the random choice of the algorithm.

When $s_i^t = 1$, from (3.42) we have $G_i^t \geq 4L^2 \|\mathbf{a}_1\|^2 / \beta$ and from (3.43) we have $r_i^t \geq 2L^2 \|\mathbf{a}_1\|^2 / \beta$. Plugging $r_i^t \geq 2L^2 \|\mathbf{a}_1\|^2 / \beta$ when $s_i^t = 1$ in (3.44) yields

$$\mathbb{E} [r_i^t | \boldsymbol{\theta}^t] \geq \frac{\varepsilon}{d} \left(\sum_{s_i^t=1} \frac{2L^2 \|\mathbf{a}_1\|^2}{\beta} + \sum_{s_i^t < 1} \frac{G_i^2(\boldsymbol{\theta}^t) \beta}{8L^2 \|\mathbf{a}_1\|^2} \right) + (1 - \varepsilon) \frac{r_{i^*}^t}{c}, \quad (3.45)$$

(a) If $s_{i^*}^t = 1$, then the cost function decreases at least by

$$\mathbb{E} \left[r_i^t | \boldsymbol{\theta}^t, s_{i^*}(\boldsymbol{\theta}^t) = 1 \right] \geq \frac{2L^2 \|\mathbf{a}_1\|^2}{\beta} \left(\frac{\varepsilon}{d} + \frac{1-\varepsilon}{c} \right). \quad (3.46)$$

(b) Let $s_{i^*}^t < 1$, from the definition of $s_{i^*}^t$ we know that $G_i(\boldsymbol{\theta}^t) \leq G_{i^*}(\boldsymbol{\theta}^t)$ for all $i \in [d]$, hence we deduce that $s_i^t < 1$ for all $i \in [d]$, then (3.45) reads as

$$\begin{aligned} \mathbb{E} \left[r_i^t | \boldsymbol{\theta}^t, s_{i^*}(\boldsymbol{\theta}^t) < 1 \right] &\geq \frac{\varepsilon}{d} \left(\sum_{i=1}^d \frac{G_i^2(\boldsymbol{\theta}^t) \beta}{8L^2 \|\mathbf{a}_1\|^2} \right) + (1-\varepsilon) \frac{G_{i^*}^2(\boldsymbol{\theta}^t) \beta}{8L^2 c \|\mathbf{a}_1\|^2} \\ &\geq \frac{\beta}{8L^2 \|\mathbf{a}_1\|^2} \left(\varepsilon \frac{\left(\sum_{i=1}^d G_i(\boldsymbol{\theta}^t) \right)^2}{d^2} + (1-\varepsilon) \frac{G_{i^*}^2(\boldsymbol{\theta}^t)}{c} \right) \\ &\geq \frac{\beta}{8L^2 \|\mathbf{a}_1\|^2} \left(\varepsilon \frac{G^2(\boldsymbol{\theta}^t)}{d^2} + (1-\varepsilon) \frac{G^2(\boldsymbol{\theta}^t)}{\eta^2 c} \right), \end{aligned} \quad (3.47)$$

where (3.47) follows from the assumption $G(\boldsymbol{\theta}^t) \leq \eta G_{i^*}(\boldsymbol{\theta}^t)$ in Proposition 3.8. Similar to the proof of Theorem 3.6, we plug the inequality $\epsilon(\boldsymbol{\theta}^t) < G(\boldsymbol{\theta}^t)$ in (3.47) and get

$$\mathbb{E} \left[r_i^t | \boldsymbol{\theta}^t, s_{i^*}(\boldsymbol{\theta}^t) < 1 \right] \geq \frac{\beta \epsilon^2(\boldsymbol{\theta}^t)}{8L^2 \|\mathbf{a}_1\|^2} \left(\frac{\varepsilon}{d^2} + \frac{(1-\varepsilon)}{\eta^2 c} \right) = \frac{\epsilon^2(\boldsymbol{\theta}^t)}{\alpha}. \quad (3.48)$$

Next, we use (3.46), (3.48) and use the tower property to check the induction hypothesis

$$\begin{aligned} \mathbb{E}[\epsilon(\boldsymbol{\theta}^{t+1})] - \mathbb{E}[\epsilon(\boldsymbol{\theta}^t)] &\leq \mathbb{E} \left[\mathbf{1}\{s_{i^*}^t = 1\} \mathbb{E} \left[r_i^t | \boldsymbol{\theta}^t, s_{i^*}^t = 1 \right] + \mathbf{1}\{s_{i^*}^t < 1\} \mathbb{E} \left[r_i^t | \boldsymbol{\theta}^t, s_{i^*}^t < 1 \right] \right] \\ &\leq -\mathbb{E} \left[\mathbf{1}\{s_{i^*}^t = 1\} \frac{2L^2 \|\mathbf{a}_1\|^2}{\beta} \left(\frac{\varepsilon}{d} + \frac{1-\varepsilon}{c} \right) + \mathbf{1}\{s_{i^*}^t < 1\} \frac{\epsilon^2(\boldsymbol{\theta}^t)}{\alpha} \right]. \end{aligned} \quad (3.49)$$

As we assumed

$$\epsilon^2(\boldsymbol{\theta}^t) \leq \epsilon^2(\boldsymbol{\theta}^0) \leq \frac{2\alpha L^2 \|\mathbf{a}_1\|^2}{\beta} \left(\frac{\varepsilon}{d} + \frac{1-\varepsilon}{c} \right)$$

in Proposition 3.8, we have

$$\min \left\{ \frac{2L^2 \|\mathbf{a}_1\|^2}{\beta} \left(\frac{\varepsilon}{d} + \frac{1-\varepsilon}{c} \right), \frac{\epsilon^2(\boldsymbol{\theta}^t)}{\alpha} \right\} = \frac{\epsilon^2(\boldsymbol{\theta}^t)}{\alpha}.$$

Hence, (3.49) becomes

$$\mathbb{E}[\epsilon(\boldsymbol{\theta}^{t+1})] - \mathbb{E}[\epsilon(\boldsymbol{\theta}^t)] \leq -\mathbb{E} \left[\frac{\epsilon^2(\boldsymbol{\theta}^t)}{\alpha} \right] \leq -\frac{\mathbb{E}[\epsilon(\boldsymbol{\theta}^t)]^2}{\alpha}, \quad (3.50)$$

where the last inequality is because of the Jensen's inequality (i.e., $\mathbb{E}[\epsilon(\boldsymbol{\theta}^t)]^2 \leq \mathbb{E}[\epsilon^2(\boldsymbol{\theta}^t)]$). By rearranging the terms in (3.50) we get

$$\mathbb{E}[\epsilon(\boldsymbol{\theta}^{t+1})] \leq \mathbb{E}[\epsilon(\boldsymbol{\theta}^t)] \left(1 - \frac{\mathbb{E}[\epsilon(\boldsymbol{\theta}^t)]}{\alpha}\right) \quad (3.51)$$

Now, let $f(y) = y(1 - \frac{y}{\alpha})$, as $f'(y) > 0$ for $y < \alpha/2$, we can plug (3.40) in (3.51) and prove the inductive step at time $t + 1$;

$$\begin{aligned} \mathbb{E}[\epsilon(\boldsymbol{\theta}^{t+1})] &\leq \mathbb{E}[\epsilon(\boldsymbol{\theta}^t)] \left(1 - \frac{\mathbb{E}[\epsilon(\boldsymbol{\theta}^t)]}{\alpha}\right) \\ &\leq \frac{\alpha}{2+t-t_0} \cdot \left(1 - \frac{1}{2+t-t_0}\right) \leq \frac{\alpha}{2+t+1-t_0}. \end{aligned} \quad (3.52)$$

Finally, we need to show that the induction basis indeed is correct. By using the inequality (3.50) for $t = 1, \dots, t_0$ we get

$$\mathbb{E}[\epsilon(\boldsymbol{\theta}^{t_0})] \leq \epsilon(\boldsymbol{\theta}^0) - \sum_{t=0}^{t_0-1} \frac{\mathbb{E}[\epsilon(\boldsymbol{\theta}^t)]^2}{\alpha}, \quad (3.53)$$

since at each iteration the cost function decreases, we have $\epsilon(\boldsymbol{\theta}^{t+1}) \leq \epsilon(\boldsymbol{\theta}^t)$ for all $t \geq 0$. Therefore, if $\mathbb{E}[\epsilon(\boldsymbol{\theta}^t)] \leq \alpha/2$ for any $0 \leq t \leq t_0$, we can conclude that $\mathbb{E}[\epsilon(\boldsymbol{\theta}^{t_0})] \leq \alpha/2$. We prove the induction hypothesis by showing that $\mathbb{E}[\epsilon(\boldsymbol{\theta}^{t_0})] > \alpha/2$ results in a contradiction. With this assumption, (3.53) becomes

$$\mathbb{E}[\epsilon(\boldsymbol{\theta}^{t_0})] \leq \epsilon(\boldsymbol{\theta}^0) - t_0 \frac{\alpha}{4} = \epsilon(\boldsymbol{\theta}^0) \left(1 - t_0 \frac{\alpha}{4\epsilon(\boldsymbol{\theta}^0)}\right), \quad (3.54)$$

Next, we use the inequality $1 + y \leq \exp(y)$ with (3.54)

$$\mathbb{E}[\epsilon(\boldsymbol{\theta}^{t_0})] \leq \epsilon(\boldsymbol{\theta}^0) \exp\left(-t_0 \frac{\alpha}{4\epsilon(\boldsymbol{\theta}^0)}\right). \quad (3.55)$$

Plugging

$$t_0 = \frac{4\epsilon(\boldsymbol{\theta}^0)}{\alpha} \log\left(\frac{2\epsilon(\boldsymbol{\theta}^0)}{\alpha}\right)$$

in (3.55) yields

$$\mathbb{E}[\epsilon(\boldsymbol{\theta}^{t_0})] \leq \frac{\alpha}{2}, \tag{3.56}$$

which proves the induction basis and concludes the proof. □

4 Controlling Polarization in Personalization

In Chapters 2 and 3, we have studied how multi-armed bandit can improve machine learning algorithms by improving two of most widely used optimization algorithms. In this chapter¹, we see how personalized multi-armed bandit algorithms in the online ad space can affect humans, and how to address that. Personalization is pervasive in the online ad space as it leads to higher efficiency for the user and higher revenue for the platform by individualizing the most relevant content for each user. However, recent studies suggest that such personalization can learn and propagate systemic biases and polarize opinions; this has led to calls for regulatory mechanisms and algorithms that are constrained to combat bias and the resulting echo-chamber effect. We propose a versatile framework that enables for the possibility to reduce polarization in personalized systems by allowing the user to constrain the distribution from which content is selected. We then present a scalable bandit algorithm with provable guarantees that satisfies the given constraints on the types of the content that can be displayed to a user, but – subject to these constraints – will continue to learn and personalize the content in order to maximize utility. We illustrate this framework on a curated dataset of online news articles that are conservative or liberal, show that it can control polarization, and we examine the trade-off between decreasing polarization and the resulting loss to revenue. We further exhibit the flexibility and scalability of our approach. We frame the problem in terms of the more general diverse content selection problem, and we test it empirically on both a News dataset and the MovieLens dataset.

4.1 Introduction

News and social media feeds, product recommendation, online advertising and other media that pervades the Internet is increasingly personalized. Content selection algorithms consider a user’s properties and past behavior in order to produce a personalized list of content to display [Goldfarb and Tucker, 2011, Liu et al., 2010]. This personalization

¹This chapter is based on [Celis et al., 2019].

leads to higher utility and efficiency both for the platform, and for the user, who sees content more directly related to their interests [Fox-Brewster, 2017, Farahat and Bailey, 2012]. However, it is now known that such personalization may result in propagating or even creating biases that can influence decisions and opinions. In an important study, Epstein and Robertson [2015] showed that user opinions about political candidates, and hence elections, can be manipulated by changing the personalized rankings of search results. Other studies show that allowing for personalization of news and other sources of information can result in a “filter bubble” [Pariser, 2011] which results in a type of tunnel vision, effectively isolating people into their own cultural or ideological bubbles; e.g., enabled by polarized information, many people did not expect a Brexit vote or Trump election [Baer, 2016]. This phenomenon has been observed on many social media platforms (see, e.g., [Hong and Kim, 2016, Conover et al., 2011, Weber et al., 2013]), and studies have shown that over the past eight years polarization has increased constantly [Garimella and Weber, 2017].

Polarization, and the need to combat it, was raised as a problem in [Robertson et al., 2018], where it was shown that Google search results differ significantly based on political preferences in the month following the 2016 elections in the United States. In a different setting, the ease with which algorithmic bias can be introduced and the need for solutions was highlighted in [Speicher et al., 2018] where it was shown that it is very easy to target people on platforms such as Facebook in a discriminatory fashion. Several approaches to quantify bias and polarization of online media have now been developed [Ribeiro et al., 2018], and interventions for fighting polarization have been proposed [Bozdag and van den Hoven, 2015]. One approach to counter such polarization would be to hide certain user properties so that they cannot be used for personalization. However, this could come at a loss to the utility for both the user and the platform – the content displayed would be less relevant and result in decreased attention from the user and less revenue for the platform (see, e.g., [Sakulkar and Krishnamachari, 2016]).

4.1.1 Groups and Polarization

Often, content is classified into different *groups* which are defined by one or more multi-valued *sensitive attributes*; for instance, news stories can have a political leaning (e.g., conservative or liberal), and a topic (e.g., politics, business or entertainment). More generally, search engines and other platforms and applications maintain topic models over their content (see e.g., [Alghamdi and Alfalqi, 2015]). At every time-step, the algorithm must select a piece of content to display to a given user,² and feedback is obtained in the form of whether they click on, purchase or hover over the item. The goal of the content selection algorithm is to select content for each user in order to maximize the

²In order to create a complete feed, content can simply be selected repeatedly in this manner to fill the screen as the user scrolls down; for ease of exposition, we describe the one-step process of selecting a single piece of content.

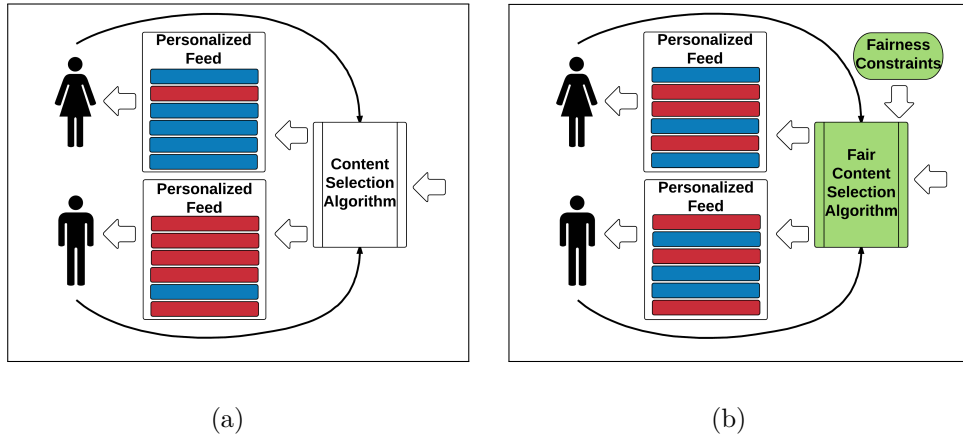


Figure 4.1 – **Unfiltered vs. balanced content delivery engines.** (a) polarization can occur on using personalized platforms, e.g., primarily showing ads for high-paying jobs (in red) to men and ads for low-paying jobs (in blue) to women (see [Datta et al., 2015]). (b) With constraints on the extent to which the feeds can differ, our model displays a more balanced feed.

positive feedback (and hence revenue) received; to do so, it must learn about the topics or groups the user is most interested in. Thus, as this optimal topic is a-priori unknown, the process is often modeled as a multi-armed bandit problem (see Section 1.2) in which a user-specific probability distribution (from which one selects content) is maintained and updated according to feedback given [Pandey and Olston, 2006]. As the content selection algorithm learns more about a user, the corresponding probability distribution begins to concentrate the mass on a small subset of topics; this results in polarization where the feed is primarily composed of a single type of content.

Our Contributions

To counter polarization, we introduce a simple framework which allows us to place *constraints* on the probability distribution from which content is sampled. The goal is to control polarization on the content displayed *at all time steps* (see Section 4.2.2) and ensure that the given recommendations do not specialize to a *single* group. Our constraints are linear and limit the total expected weight that can be allocated to a given group through lower and upper bound parameters on each group. These polarization constraints are taken as input and can be set according to the context or application. Importantly, though simple, these constraints are versatile enough to control polarization with respect to a variety of metrics which can measure the extent of polarization, or lack thereof, in a given algorithm. This is due to the fact that several fairness metrics depend, e.g., on the ratio or difference between the probability mass on two groups, hence can be implemented by picking appropriate lower/upper bound parameters for the constraints in our setting to give an immediate fairness guarantee (such reductions

can be formalized following standard techniques, see, e.g., [Celis et al., 2018]). Thus, by placing such constraints the content shown to different types of users is varied, and polarization is controlled.

While there are several polynomial time algorithms for similar settings, the challenge is to come up with a *scalable* content selection algorithm for the resulting optimization problem of maximizing revenue (via personalization) subject to satisfying the polarization constraints. We show how an adaptation of the existing algorithm ε -GREEDY (see Section 1.2.3) for the unconstrained bandit setting, along with the special structure of our constraints, can lead to a scalable algorithm with provable guarantees for this constrained optimization problem (see Theorem 4.1). We evaluate this framework and our algorithm on a curated dataset of online news articles that are conservative or liberal, show that it can control polarization, and examine the trade-off between decreasing polarization and the resulting loss to revenue. We further illustrate the flexibility and scalability of this approach by considering the problem of diverse content selection, and evaluate our algorithm on the MovieLens dataset for diverse movie recommendation as well as the YOW dataset for diverse article recommendation. To the best of our knowledge, this is the first algorithm to control polarization in personalized settings that comes with provable guarantees, allows for the specification of general constraints, and is viable in practice.

4.2 Preliminaries

4.2.1 Polarization in Existing Models

Algorithms for the general (unconstrained, and hence potentially biased) problem of displaying personalized content are often developed in the multi-armed bandit setting (see e.g., [Li et al., 2010, 2016]). For an overview of the multi-armed bandit setting and algorithms developed for this setting, see Section 1.2. Below, we explain how the problem of displaying personalized content can be formulated as a multi-armed bandit problem.

At each time step $t = 1, \dots, T$, a user views a page (e.g., Facebook, Twitter or Google News), and one piece of content (or *arm*) $a \in [K]$ must be selected to be displayed. A random *reward* r_a^t , which depends on the selected content is then received by the content selection algorithm. This reward captures resulting clicks, purchases, or time spent viewing the given content. The rewards r_a^t are assumed to be drawn independently across a and t from an unknown distribution \mathcal{D} . Hence the problem falls under stochastic multi-armed bandit category (see Section 1.2.3), and regret takes the form

$$\bar{R}_T := \max_{a \in [K]} \mathbb{E}_{r^t \sim \mathcal{D}} \left[\sum_{t=1}^T r_a^t - \sum_{t=1}^T r_{a_t}^t \right],$$

where a_t is the chosen arm at time step t . As explained in Section 1.2.1, the regret metric is used to study the efficacy of a multi-armed bandit algorithm.

The problem with this approach is that, bandit algorithms, by optimizing for that “ideal” arm $a^* = \arg \max_{a \in [K]} \mathbb{E}_{r^t \sim \mathcal{D}}[r_a^t]$, *by definition* strive for polarization. To understand how, let $G_1, \dots, G_g \subseteq [K]$ be g groups of arms which correspond to different types of content across which we do not want to polarize. In the simplest setting, the G_i s form a partition (e.g., conservative and liberal news articles when the arms represent news stories), but in general the group structure can be arbitrary. A feature of bandit algorithms is that the probability distribution on the arms, that the algorithm is learning, converges to the action with the best expected reward; i.e., the entire probability mass ends up on the single arm a^* , and hence in a single group – causing polarization (see e.g., [Li et al., 2010]).

4.2.2 Constraint setting

We would like an approach that can control polarization with respect to the groups that the selected arms belong to. Towards this, for each group G_i , let ℓ_i be a lower bound and u_i be an upper bound on the amount of weighted probability mass that we allow the content selection algorithm to place on this group. Formally, we impose the following constraints:

$$\ell_i \leq \sum_{a \in G_i} w_a(G_i) \cdot p_a^t \leq u_i \quad \forall i \in [g], \forall t \in [T], \quad (4.1)$$

where $w_a(G_i) \in [0, 1]$ represents the group weight of arm a on group G_i and p_a^t is the probability of selecting arm a at time t .

The group weight $w_a(G_i)$ denotes the similarity between arm a and group G_i . For instance, following our earlier discussion on news articles, a conservative leaning news article might have a group weight of 0.9 for the *conservative articles* group and a group weight of 0.1 for the *liberal articles* group, whereas a neutral article might have both of these weights as 0.5. In case of categorical groups (e.g., men vs. women), the group weight can take a binary value. For more general cases (see, e.g., Section 4.5), this weight can take a real value between 0 and 1. The values for $w_a(G_i)$ s can be set using various methods depending on the application and can also take into account error bounds for classifiers that decide whether a given $a \in G_i$ or not. For the case of text documents (e.g., news and scientific articles [Wang and Blei, 2011]), the weights can be set using techniques like topic modeling, which give us the percentage of a document that corresponds to a certain topic.

The bounds ℓ_i s and u_i s provide a handle with which we can ensure that the weighted probability mass placed on any given group is neither too high nor too low at each

Chapter 4. Controlling Polarization in Personalization

Algorithm 4.1 CONS- ϵ -GREEDY

- 1: **Input:** Constraint set \mathcal{C} , a constrained probability distribution $\mathbf{q}_f \in \{\mathbf{q} : B_\infty(\mathbf{q}, \eta) \subset \mathcal{C}\}$, a positive integer T , a constant L that controls the exploration
 - 2: **Initialize:** $\bar{\boldsymbol{\mu}}_1 := 0$
 - 3: **for** $t = 1, \dots, T$ **do**
 - 4: Update $\epsilon_t := \min\{1, 4/(\eta L^2 t)\}$
 - 5: Compute $\mathbf{p}^t := \arg \max_{\mathbf{p} \in \mathcal{C}} \bar{\boldsymbol{\mu}}_t^\top \mathbf{p}$
 - 6: Sample a from the probability distribution $(1 - \epsilon_t)\mathbf{p}^t + \epsilon_t \mathbf{q}_f$
 - 7: Observe reward $r_t = r_a^t$
 - 8: Update empirical mean $\bar{\boldsymbol{\mu}}_{t+1}$
 - 9: **end for**
-

time step. Rather than fixing the values of u_i s and ℓ_i s, we allow them to be specified as input. This allows one to control the extent of polarization of content depending on the application, and hence (indirectly) encode bounds on a wide variety of existing metrics for different notions of *group fairness* which, in effect, encode the extent of polarization. This requires translating the metric parameters into concrete values of ℓ_i s and u_i s. For instance, given $\beta > 0$, by setting u_i s and ℓ_i s such that $u_i - \ell_i \leq \beta$ for all i , we can ensure that the *risk difference* is bounded by β . An additional feature of our model is that no matter what the group structures, or the lower and upper bounds are, the constraints are always linear.

Importantly, note that unlike ignoring user preferences entirely as in [Pariser, 2011], the constraints still allow for personalization *across* groups. For instance, if the groups are conservative-leaning (C) vs liberal-leaning (L) articles, and the users are known conservatives or liberals, we may require that $w(\text{C}) \cdot p_{\text{C}}^t \leq 0.75$ and $w(\text{L}) \cdot p_{\text{L}}^t \leq 0.75$ for all t . This ensures that extreme polarization cannot occur – at least 25% of the content a conservative is presented with will be liberal-leaning. Despite these constraints, personalization at the group level can still occur, e.g., by letting $w(\text{C}) \cdot p_{\text{C}}^t = 0.75$ and $w(\text{L}) \cdot p_{\text{L}}^t = 0.25$ for a conservative-leaning user. Furthermore, this framework allows for complete personalization *within* a group; e.g., the conservative-leaning articles shown to conservatives and liberals may differ. This is crucial as the utility maximizing conservative-leaning articles for a conservative may differ from the utility maximizing conservative-leaning articles for a liberal.

In the unconstrained setting, the performance of an algorithm is measured with respect to the (unknown) optimal solution (see the definition of regret in Section 1.2.1). The next question we address is how to measure an algorithm’s performance against the best *constrained* solution. We say that a probability distribution \mathbf{p} on $[K]$ is *constrained* if it satisfies the upper and lower bound constraints in (4.1). Let \mathcal{C} be the set of all such probability distributions. Note that given the linear nature of the constraints, the set \mathcal{C} is a polytope (an intersection of a set of half spaces), and hence we can formulate the problem of finding \mathbf{v}^* as a linear programming problem.

Algorithm	Per iteration Running time	Regret Bound
CONF-BALL ₂ [Dani et al., 2008]	NP-Hard problem	$O\left(\frac{K^2}{\gamma} \log^3 T\right)$
OFUL [Abbasi-Yadkori et al., 2011]	NP-Hard problem	$\tilde{O}\left(\frac{1}{\gamma} \left(K^2 + \log^2 T\right)\right)$
CONF-BALL ₁ [Dani et al., 2008]	$O(K^\omega) + 2K$ LP-s	$O\left(\frac{K^3}{\gamma} \log^3 T\right)$
CONS- ε -GREEDY (Algorithm 4.1)	$O(1) + 1$ LP	$O\left(\frac{K}{\gamma^2} \log T\right)$
CONS- L_1 -OFUL (Algorithm 4.2)	$O(K^\omega) + 2K$ LP-s	$\tilde{O}\left(\frac{K}{\gamma} \left(K^2 + \log^2 T\right)\right)$

Table 4.1 – The complexity and problem-dependent regret bounds for various algorithms when the decision set is a polytope.

An algorithm is said to be constrained if it only selects $\mathbf{p}^t \in \mathcal{C}$. The *constrained regret* for such an algorithm can be defined as

$$\text{CRegret}_T := \mathbb{E}_{\mathbf{r}^t \sim \mathcal{D}, \tilde{\mathbf{a}} \sim \mathbf{v}^*} \left[\sum_{t=1}^T r_{\tilde{\mathbf{a}}}^t - \sum_{t=1}^T r_{\mathbf{a}^t}^t \right],$$

where $\mathbf{v}^* \in \mathcal{C}$ represents a point in the constraint set \mathcal{C} with the highest expected reward: $\mathbf{v}^* := \arg \max_{\mathbf{p} \in \mathcal{C}} \mathbb{E}_{\mathbf{r} \sim \mathcal{D}, \tilde{\mathbf{a}} \sim \mathbf{p}} [r_{\tilde{\mathbf{a}}}]$.

4.3 Related Work

Approaches to Curtail Polarization. There is a large body of work studying the effects of polarization, and ways in which we can combat it. A significant portion of this literature considers interventions to inform or educate users on the effects of personalization and is orthogonal to our work. Pariser, who coined the term “filter bubble”, proposes that we simply remove personalization entirely [Pariser, 2011]. However, this would come at a complete loss to the utility and efficiency that personalization can bring to both the user and the platform. In contrast, our approach does allow for personalization – up to a point. It ensures that the content is not polarized beyond the given constraints, but within that personalizes in order to maintain high utility. Another approach would be to manipulate the user ratings (e.g., by adding noise or a regularizer to the recommender algorithm) in order to have only approximate preferences; this has been shown to help reduce polarization [Adomavicius et al., 2014, Adomavicius and Kwon, 2012, Wasilewski and Hurley, 2016]. We compare against such an approach in our empirical results (CONS-RAN), and observe that our algorithm significantly outperforms this method. The key difference is that such an approach adds noise to attain de-polarization, while our approach de-polarizes in an informed manner that personalizes content as much as possible subject to the polarization constraints.

Algorithms for Constrained Bandit Optimization. Constrained bandit optimization is a broad field that has arisen in the consideration of a variety of problems unrelated

to polarization. For example, knapsack-like constraints on bandit optimization is studied in [Agrawal and Devanur, 2016]; however, this work only considers constraints that are placed on the final probability vector \mathbf{p}^T , whereas in our setting it is important to satisfy fairness constraints at every time step $\{\mathbf{p}^t\}_{t=1}^T$. A different line of work [Joseph et al., 2016] considers *online individual fairness* constraints which require that the probability of selecting all *arms* be approximately equal until enough information is gathered to confidently know which arm is the best. In a similar vein, another work [Ding et al., 2013] considered budgets on the number of times that any given arm can be selected. Both of these results can be loosely interpreted as working with the special case of our model in which each arm belongs to its own group; their results cannot be applied to our more general setting or be used to curtail polarization.

4.4 Technical Contributions

For each arm $a \in [K]$, let its mean reward be μ_a and $\boldsymbol{\mu} = [\mu_1, \dots, \mu_K]$ be the vector of all arms mean rewards. In this case, the unknown parameters are the expectations of each arm μ_a for $a \in [K]$. We assume that the reward for the t -th time step is sampled from a Bernoulli distribution with probability of success μ_a . For a probability distribution $\mathbf{q} \in \mathcal{C}$ and a small enough constant $\eta > 0$, we define $B_\infty(\mathbf{q}, \eta)$ to be the set of all probability distributions that lie inside \mathcal{C} , such that a probability distribution $\mathbf{q}_f \in B_\infty(\mathbf{q}, \eta)$ has at least η probability mass on each arm. More formally, $B_\infty(\mathbf{q}, \eta) \in \mathcal{C}$ is an ℓ_∞ -ball of radius η centered at \mathbf{q} . Let $V(\mathcal{C})$ denote the set of vertices of \mathcal{C} and $\mathbf{v}^* := \arg \max_{\mathbf{v} \in V(\mathcal{C})} \sum_{a \in [K]} \mu_a v_a$.

Theorem 4.1. *Let $\eta > 0$ be a small enough constant. Given the description of \mathcal{C} , any probability distribution $\mathbf{q}_f \in \{\mathbf{q} : B_\infty(\mathbf{q}, \eta) \subset \mathcal{C}\}$ that lies in the constrained region, and the sequence of rewards, the CONS- ϵ -GREEDY algorithm (Algorithm 4.1), run for T iterations, has the following constrained regret bound:*

$$\mathbb{E}[\text{CRegret}_T] = O\left(\frac{\ln T}{\eta\gamma^2}\right), \quad (4.2)$$

where $\epsilon_t = \min\{1, 4/(\eta d^2 t)\}$ and $d = \min\{\gamma, 1/2\}$. The algorithm works for any lower bound L on γ , with a L instead of γ in the regret bound (4.2). Here γ is the difference between the maximum and the second maximum expected rewards over the vertices of the polytope \mathcal{C} . More formally, $\gamma := \sum_{a \in [K]} \mu_a v_a^* - \max_{\mathbf{v} \in V(\mathcal{C}) \setminus \mathbf{v}^*} \sum_{a \in [K]} \mu_a v_a$.

Before we present the formal details, we first highlight some key aspects of the algorithm, theorem and proofs.

For general convex sets, γ can be 0 and the regret bound can at best only be $O(\sqrt{T})$ [Dani et al., 2008]. As our constraints result in a constraint set \mathcal{C} which is a polytope,

unless there are degeneracies, γ is non-zero. In general, γ may be hard to estimate theoretically. However, for the settings in which we conduct our experiments, we observe that the value of γ is reasonably large.

When the probability space is unconstrained, we can use any algorithm developed for the stochastic multi-armed bandit setting (see Section 1.2.3). In this case, it suffices to solve $\arg \max_{a \in [K]} \bar{\mu}_a$, where $\bar{\mu}_a$ is an estimate for the mean reward of the a -th arm. When the probability distribution is constrained to lie in a polytope \mathcal{C} , instead of a maximum over the arm mean estimates, we need to solve $\arg \max_{\mathbf{p} \in \mathcal{C}} \bar{\boldsymbol{\mu}}^\top \mathbf{p}$. This necessitates the use of a linear program for any algorithm operating in this fashion. At every iteration, `CONS- ϵ -GREEDY` solves one LP. We can speed up the LP computation considerably in practice by using the interior points method and warm starting the LP solver from the optimal \mathbf{p} found in the previous iteration (see Section 4.5.1).

4.4.1 Overview of Algorithm 4.1: Constrained- ϵ -Greedy

The algorithm, with probability $1 - \epsilon_t$ chooses the distribution

$$\mathbf{p}^t = \arg \max_{\mathbf{p} \in \mathcal{C}} \bar{\boldsymbol{\mu}}^\top \mathbf{p},$$

and with probability ϵ_t it samples from a feasible constrained distribution $\mathbf{q}_f \in \mathcal{C}$ in $B_\infty(\mathbf{q}, \eta)$, i.e., there is at least η probability mass on each arm. The reward at each time step t is generated as $r_t \sim \text{Bernoulli}(\mu_a)$, where $a \sim (1 - \epsilon_t)\mathbf{p}^t + \epsilon_t\mathbf{q}_f$ is the arm the algorithm chooses at the t^{th} time step. The algorithm observes this reward and updates its estimate to $\bar{\boldsymbol{\mu}}_{t+1}$ for the next time-step appropriately. `CONS- ϵ -GREEDY` is a variant of the classic ϵ -`GREEDY` approach [Auer et al., 2002a] (see also Section 1.2.3). Recall that in our setting, an arm is an article (corner of the K -dimensional simplex) and not a vertex of the polytope \mathcal{C} . The polytope \mathcal{C} sits inside this simplex and may have exponentially many vertices. This is not that case in the setting of [Dani et al., 2008, Abbasi-Yadkori et al., 2011] – there may not be any ambient simplex in which their polytope sits, and even if there is, they do not use this additional information about which vertex of the simplex was chosen at each time t . Thus, while they are forced to maintain confidence intervals of rewards for all the points in \mathcal{C} , this specialty in our model allows us to get away by maintaining confidence intervals only for the K arms (vertices of the simplex) and then use these intervals to obtain a confidence interval for any point in \mathcal{C} . Similar to ϵ -`GREEDY`, if we choose each arm enough number of times, we can build a good confidence interval around the mean of the reward for each arm. The difference is that instead of converging to the optimal arm, our constraints maintain the point inside \mathcal{C} and the point converges to a vertex of \mathcal{C} . To ensure that we choose each arm with high probability, we fix a constrained point $\mathbf{q}_f \in B_\infty(\mathbf{q}, \eta)$ of \mathcal{C} and sample from the point

Chapter 4. Controlling Polarization in Personalization

Dataset	#Arms (K)	#Instances	#Iterations (T)	#Groups (g)
PoliticalNews	1356 (avg.)	30 (# days)	10,000	2
MovieLens	25	943 (# users)	1000	19
YOW	81	21 (# users)	10,000	7

Table 4.2 – Overview of datasets used in the empirical results in Section 4.5.1.

$(1 - \epsilon_t)\mathbf{p}^t + \epsilon_t\mathbf{q}_f$. Then, as in ε -GREEDY, we proceed by bounding the regret showing that if the confidence-interval is tight enough, the optimal of LP with true mean $\boldsymbol{\mu}$ and LP with the empirical mean $\bar{\boldsymbol{\mu}}$ does not change.

Proof of Theorem 4.1. Let $\mathbf{v}^* = [v_1^*, \dots, v_K^*] \in \mathcal{C}$ be the optimal probability distribution. Conditioned on the history at time t , the expected regret of CONS- ε -GREEDY at iteration t can be bounded as follows

$$\begin{aligned}
 R(t) &= \boldsymbol{\mu}^\top \mathbf{v}^* - \left((1 - \epsilon_t)\boldsymbol{\mu}^\top \bar{\mathbf{v}}^t + \epsilon_t \sum_{a=1}^K q_{a,f} \boldsymbol{\mu}_a \right) \\
 &\leq (1 - \epsilon_t) \boldsymbol{\mu}^\top (\mathbf{v}^* - \bar{\mathbf{v}}^t) + \epsilon_t \boldsymbol{\mu}^\top \mathbf{v}^* \\
 &\leq (1 - \epsilon_t) \boldsymbol{\mu}^\top \mathbf{v}^* \mathbb{1} \{ \bar{\mathbf{v}}^t \neq \mathbf{v}^* \} + \epsilon_t \boldsymbol{\mu}^\top \mathbf{v}^*,
 \end{aligned}$$

where $\bar{\mathbf{v}}^t = \arg \max_{\mathbf{p} \in \mathcal{C}} \bar{\boldsymbol{\mu}}^\top \mathbf{p}$.

Let $n = 4/(\eta d^2)$. For $t \leq n$, since $\epsilon_t = \min\{1, 4/(\eta L^2 t)\}$ we have $\epsilon_t = 1$. The expected regret of the CONS- ε -GREEDY is

$$\mathbb{E} [\text{CRegret}_T] \leq \boldsymbol{\mu}^\top \mathbf{v}^* \sum_{t=n+1}^T \mathbb{P}(\bar{\mathbf{v}}^t \neq \mathbf{v}^*) + \boldsymbol{\mu}^\top \mathbf{v}^* \sum_{t=1}^T \epsilon_t. \quad (4.3)$$

Let $\Delta \boldsymbol{\mu} = \bar{\boldsymbol{\mu}} - \boldsymbol{\mu}$. Without loss of generality, let $\boldsymbol{\mu}^\top \mathbf{v}_i > \boldsymbol{\mu}^\top \mathbf{v}_j$ for any $\mathbf{v}_i, \mathbf{v}_j \in V(C)$ with $i < j$. Hence, $\mathbf{v}_1 = \mathbf{v}^*$. Let $\Delta_i = \boldsymbol{\mu}^\top (\mathbf{v}_1 - \mathbf{v}_i)$. As a result $\Delta_1 = 0$ and $\Delta_2 = \gamma$. The event $\bar{\mathbf{v}}^t \neq \mathbf{v}^*$ happens when $\bar{\boldsymbol{\mu}}_t^\top \mathbf{v}_i > \bar{\boldsymbol{\mu}}_t^\top \mathbf{v}_1$ for some $i > 1$, that is,

$$(\boldsymbol{\mu} + \Delta \boldsymbol{\mu}_t)^\top (\mathbf{v}_i - \mathbf{v}_1) = -\Delta_i + \Delta \boldsymbol{\mu}_t^\top (\mathbf{v}_i - \mathbf{v}_1) \geq 0.$$

As a result, we have

$$\begin{aligned}
 \mathbb{P}(\bar{\mathbf{v}}^t \neq \mathbf{v}^*) &= \mathbb{P}\left(\bigcup_{\mathbf{v}_i \in V(\mathcal{C}) \setminus \mathbf{v}_1} \Delta \boldsymbol{\mu}_t^\top (\mathbf{v}_i - \mathbf{v}_1) \geq \Delta_i\right) \\
 &\leq \mathbb{P}\left(\bigcup_{\mathbf{v}_i \in V(\mathcal{C}) \setminus \mathbf{v}_1} \|\Delta \boldsymbol{\mu}_t\|_\infty \|\mathbf{v}_i - \mathbf{v}_1\|_1 \geq \Delta_i\right) \\
 &\leq \mathbb{P}\left(\bigcup_{\mathbf{v}_i \in V(\mathcal{C}) \setminus \mathbf{v}_1} \|\Delta \boldsymbol{\mu}_t\|_\infty \geq \frac{\Delta_i}{2}\right) \\
 &= \mathbb{P}\left(\|\Delta \boldsymbol{\mu}_t\|_\infty \geq \frac{\gamma}{2}\right) = \mathbb{P}\left(\bigcup_{j \in [K]} |\Delta \mu_{t,j}| \geq \frac{\gamma}{2}\right) \\
 &\leq \sum_{j \in [K]} \mathbb{P}\left(|\Delta \mu_{t,j}| \geq \frac{\gamma}{2}\right).
 \end{aligned} \tag{4.4}$$

In (4.4) we use Holder's inequality. Let $E_t = \eta \sum_{\tau=1}^t \epsilon_\tau / 2$ and let $N_{t,j}$ be the number of times that we have chosen arm j up to time t . Next, we bound $\mathbb{P}\{|\Delta \mu_{t,j}| \geq \frac{\gamma}{2}\}$.

$$\begin{aligned}
 &\mathbb{P}\left(|\Delta \mu_{t,j}| \geq \frac{\gamma}{2}\right) \\
 &= \mathbb{P}\left(|\Delta \mu_{t,j}| \geq \frac{\gamma}{2} \mid N_{t,j} \geq E_t\right) \mathbb{P}(N_{t,j} \geq E_t) \\
 &\quad + \mathbb{P}\left(|\Delta \mu_{t,j}| \geq \frac{\gamma}{2} \mid N_{t,j} < E_t\right) \mathbb{P}(N_{t,j} < E_t) \\
 &\leq \mathbb{P}\left(|\Delta \mu_{t,j}| \geq \frac{\gamma}{2} \mid N_{t,j} \geq E_t\right) + \mathbb{P}(N_{t,j} < E_t).
 \end{aligned} \tag{4.6}$$

As $\mathbf{q}_f \in \{\mathbf{q} : B_\infty(\mathbf{q}, \eta) \subset \mathcal{C}\}$, we have $q_{a,f} > \eta$, i.e., the probability of selecting an arm a is at least $\epsilon_t q_{a,f}$. Next, we bound each term of (4.6). First, using Chernoff-Hoeffding bound we have

$$\mathbb{P}\left(|\Delta \mu_{t,j}| \geq \frac{\gamma}{2} \mid N_{t,j} \geq E_t\right) \leq 2 \exp\left(-\frac{E_t \gamma^2}{2}\right). \tag{4.7}$$

Using the Bernstein inequality [Sridharan, 2002], we have

$$\mathbb{P}(N_{t,j} < E_t) \leq \exp\left(-\frac{E_t}{5}\right). \tag{4.8}$$

For $t \leq n$, $\epsilon_t = 1$ and $E_t = \eta t/2$. For $t > n$ we have

$$\begin{aligned} E_t &= \frac{\eta \cdot n}{2} + \sum_{i=n+1}^t \frac{2}{d^2 i} \geq \frac{2}{d^2} + \frac{2}{d^2} \ln \left(\frac{t}{n} \right) \\ &= \frac{2}{d^2} \ln \left(\frac{et}{n} \right). \end{aligned} \quad (4.9)$$

By plugging (4.7), (4.8) and (4.9) in (4.6) and noting that $\gamma < 1/2$ we get

$$\mathbb{P} \left(|\Delta \mu_{t,j}| \geq \frac{\gamma}{2} \right) \leq \left(\frac{n}{et} \right)^{\frac{\gamma^2}{d^2}} + \left(\frac{n}{et} \right)^{\frac{4}{10d^2}} \leq \left(\frac{n}{et} \right) + \left(\frac{n}{et} \right)^{\frac{4}{10d^2}} \leq 2 \left(\frac{n}{et} \right). \quad (4.10)$$

Plugging (4.10) in (4.3) yields

$$\mathbb{E} [\text{CRegret}_T] \leq \boldsymbol{\mu}^\top \mathbf{v}^* \left(\left(1 + \frac{2n}{e}\right) \ln T + n \right). \quad (4.11)$$

By substituting $n = 4/(\eta d^2)$ in the regret above and noting that $\gamma \leq 2d$ we conclude the proof

$$\mathbb{E} [\text{CRegret}_T] \leq \boldsymbol{\mu}^\top \mathbf{v}^* \left(\left(1 + \frac{4}{\eta d^2}\right) \ln T + \frac{4}{\eta d^2} \right) = O \left(\frac{\ln T}{\eta \gamma^2} \right).$$

□

4.4.2 Alternate Approaches and Special Cases

In this section, we briefly outline an alternate approach for solving this problem that results in a different regret / runtime guarantee (see Table 4.1). We further show that, for certain special cases of the group structure, e.g., if the groups perfectly partition the arms, one can design even faster solutions to the LP.

Algorithm 4.2: Cons- L_1 -OFUL

Any algorithm for solving the linear bandit problem with an infinite, continuous set of arms can be adapted to solve the constrained multi-armed bandit problem. The constrained multi-armed bandit problem can be thought of as a special case of this type of linear bandit problem, where the continuous space of arms is simply the probability simplex over our discrete arms. Thus, each arm increases the dimensionality of the linear bandit problem by one, and the continuous arm selected at time t corresponds to the probability distribution we select at time t . The difference between these settings is that while one gets rewards for *points in the simplex* in the case of linear bandit problems, we get rewards for the arms themselves (i.e. the vertices of the simplex) in the constrained

multi-armed bandit problems.³ Using these algorithms as a black-box can be inefficient, and does not allow us to come up with practical algorithms for real-world applications.

However, in some cases, we can adapt algorithms for linear bandits to our constrained setting in a way that makes the computations efficient. Consider the OFUL algorithm that appeared in [Abbasi-Yadkori et al., 2011]; we will adapt this algorithm to our constrained setting, and we call the adapted algorithm $\text{CONS-}L_1\text{-OFUL}$. $\text{CONS-}L_1\text{-OFUL}$ is an example of algorithms for linear bandits being used to solve the constrained multi-armed bandit problem. The key difference between $\text{CONS-}L_1\text{-OFUL}$ and OFUL is that instead of using a scaled L_2 -ball in each iteration, we use a scaled L_1 -ball, which makes $\text{CONS-}L_1\text{-OFUL}$ efficient; without this adaptation the equivalent step in our setting required solving an NP-hard and nonconvex optimization problem.⁴ $\text{CONS-}L_1\text{-OFUL}$ incurs $\tilde{O}\left(\frac{K}{\gamma}\left(K^2 + \log^2 T\right)\right)$ regret (see Theorem 4.3). This gives a worse dependence on K but a better dependence on γ as compared with $\text{CONS-}\varepsilon\text{-GREEDY}$ (see Table 4.1), and hence could be beneficial in some settings. However, the runtime is considerably slower than $\text{CONS-}\varepsilon\text{-GREEDY}$. Instead of maintaining a least-squares estimate of the optimal reward vector, $\text{CONS-}\varepsilon\text{-GREEDY}$ maintains an empirical mean estimate of it denoted by $\bar{\mu}_t$, which is computationally cheaper per iteration. It also solves only one linear program instead of $2K$ linear programs at every iteration. Both of these factors together cause a significant decrease in running time compared to $\text{CONS-}L_1\text{-OFUL}$. Thus, while $\text{CONS-}L_1\text{-OFUL}$ theoretically achieves lower regret than $\text{CONS-}\varepsilon\text{-GREEDY}$ in terms of γ , it is not as computationally efficient, and performs worse in practice.

More Efficient LP Solvers for Special Group Structures

For the special case where group weights are binary, i.e.,

$$w_a(G_i) \in \{0, 1\} \quad \forall a \in [K], \quad i \in [g],$$

and the constraint set have some special structure, we can solve the LP efficiently:

Single Partition. If the groups in the constraint set form a partition, one can solve the linear program in $O(K)$ time via a simple greedy algorithm. Since each part is separate, we can simply put the minimum probability mass as required by the constraints on the best arm of each group, and then put the maximum possible probability mass on arms in descending order of arm utility. This gives a probability vector that satisfies the constraints and is optimal with respect to the reward.

Laminar Constraints. Let the groups $G_1, \dots, G_g \subseteq [K]$ be such that: $G_i \cap G_j \neq \emptyset$ implies $G_i \subseteq G_j$ or $G_j \subseteq G_i$. The groups form a tree-like data structure, where the

³This is also what allows us to get fast and efficient algorithms like $\text{CONS-}\varepsilon\text{-GREEDY}$ for the constrained multi-armed bandit setting.

⁴This is similar in spirit to how CONF-BALL_2 can be adapted to CONF-BALL_1 in [Dani et al., 2008].

children are the largest groups that are subset of the parents. In this case, the LP can be solved efficiently by a greedy algorithm, and we can solve the LP step in $O(gk)$ time exactly. For the sake of brevity and clarity, we defer the full explanation to the appendix.

4.5 Empirical Evaluation

In this section we compare the performance of CONS- ϵ -GREEDY to the unconstrained algorithm, the hypothetical *optimal* constrained algorithm (which we could implement if we knew the rewards of the arms a-priori), a smoothed version of the unconstrained algorithm that satisfies the constraints, and a naive baseline that satisfies the constraints but does not aim to optimize the reward.⁵ We briefly outline the experiments and results here, with details in the following subsections.

We conduct counterfactual experiments on three datasets (see Table 4.2). We consider a curated PoliticalNews where the constraints aim to reduce the political polarization of the presented search results. As mentioned above, we can similarly apply these techniques to the diversification of content in areas beyond political polarization. Towards this, we simulate our algorithm on another dataset of news articles Zhang [2005] and strive to diversify across topics (e.g., business, entertainment, and world news), and the MovieLens dataset Harper and Konstan [2015] where we strive to diversify recommendations across genres. In all cases, we find that CONS- ϵ -GREEDY consistently outperforms the smoothed version of the unconstrained algorithm as well as the naive baseline, accumulating much higher reward, while closely approximating the hypothetical optimal. This benefit of CONS- ϵ -GREEDY is most evident when the constraints are the tightest; e.g., CONS- ϵ -GREEDY accumulates twice as much reward as the smoothed version of the unconstrained algorithm on the YOW dataset (see Figure 4.2).

We then compare the polarization and diversification for the constrained and unconstrained algorithms. We aim to reduce polarization by recommending news articles with both, liberal and conservative biases. Similarly, we aim to increase diversity by recommending articles and movies not just from the *best group* in terms of rewards, but from other groups as well. We observe that algorithms in the unconstrained setting quickly converge to the *best group* in terms of rewards, whereas algorithms in the constrained setting always display a certain minimum percentage of content *not* from the best group, hence improving diversification and avoiding polarization.

⁵In our simulations, the regret for CONS- L_1 -OFUL was similar or slightly worse than CONS- ϵ -GREEDY. As CONS- ϵ -GREEDY is also much more efficient we use it as the main comparator, and leave open the question as to if or when CONS- L_1 -OFUL performs better as suggested by the theoretical results.

4.5.1 Experimental Setup

Algorithm and Benchmarks.

In each counterfactual simulation we report the normalized cumulative reward for each of the following algorithms and benchmarks:

Unconstrained-Optimal is the hypothetical *optimal* algorithm when there is no constraint and the expected rewards of all arms $a \in [K]$ are known. It simply chooses the best arm a^* at each step t .

Unconstrained- ε -Greedy is the *unconstrained* ε -GREEDY algorithm, where \mathcal{C} is the set of *all* probability distributions over $[K]$.

Cons-Optimal is the hypothetical *optimal* probability distribution, subject to the polarization constraints, that we could have used if we had known the reward vector μ^* for the arms a-priori.

Constrained- ε -Greedy is our implementation of Algorithm 4.1 with the given polarization constraints as input.⁶

Cons-Ran is a smoothed version of UNCONSTRAINED- ε -GREEDY that satisfies the constraints. At each time step, given the probability distribution \mathbf{p}^t specified by the unconstrained ε -GREEDY algorithm, CONS-RAN takes the largest $\theta \in [0, 1]$ such that selecting an arm with probability $\theta \cdot \mathbf{p}^t$ does not violate the constraints. With the remaining probability $(1 - \theta)$ it follows the same procedure as in CONS-NAIVE to select an arm at *random* subject to the constraints.

Cons-Naive. As a baseline, we consider a simple algorithm that satisfies the constraints as follows: for each group i and arm a , with probability $\frac{\ell_i}{w_a(G_i)}$ it selects an arm at random from G_i , then, with any remaining probability, it selects an arm uniformly at random from the entire collection $[K]$ while respecting the upper bound constraints u_i .

Note that if we know the true rewards of the arms, this optimal distribution is easy to compute via a simple greedy algorithm; it simply places the most probability mass that satisfies the constraints on the best arm, the most probability mass remaining on the second-best arm subject to the constraints, and so on and so forth until the entire probability mass is exhausted. This strategy can be found by solving one LP.

Implementation Details.

Instead of solving an LP from scratch at each iteration (in step 4 of Algorithm 4.1), we warm start the LP by using the solution of the LP from the previous iteration as the starting point for our solver. We modified an implementation of an LP solver Yan which uses the interior points method. “Warm-starting” the LP solver in this way speeds up the LP computation considerably in practice and allows efficient implementation of the algorithm even when there are many groups that do not form a partition and hence many

⁶We set $\epsilon_t = \min(1, 10/t)$. Tuning ϵ_t could give even better results.

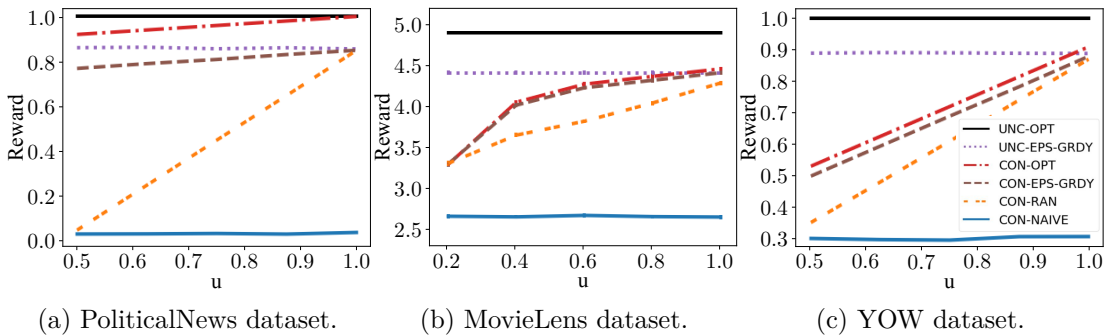


Figure 4.2 – **Effect of Polarization Constraints (u) on Reward.** The normalized cumulative reward attained as a function of the strength of the upper-bound constraints is reported for the three datasets in figures (a), (b) and (c). In all cases, our algorithm $\text{CONS-}\epsilon\text{-GREEDY}$ does not allow polarization, and performs near-optimally with respect to the reward. The lower the value of u , the stronger are the constraints.

nontrivial constraints. For certain special cases, provably fast algorithms for solving the LP also exist (see Section 4.4.2), however we did not employ these techniques in the simulations.

Note that $\text{CONS-}\epsilon\text{-GREEDY}$, CONS-RAN and $\text{UNCONSTRAINED-}\epsilon\text{-GREEDY}$ implementations all use Algorithm 4.1 as a subroutine; however $\text{CONS-}\epsilon\text{-GREEDY}$ and CONS-RAN take the constraints as input, with CONS-RAN satisfying the polarization constraints via smoothing the probability distribution, and $\text{UNCONSTRAINED-}\epsilon\text{-GREEDY}$ need not satisfy the constraints at all.

Description of Datasets and Group Weights.

PoliticalNews. We curate this dataset by using a large scale web-crawler Web to collect online news articles over a span of 30 days (23rd July – 21st August, 2018), along with the number of Facebook likes that each article received as of 22nd August, 2018. We look at the political leaning of each article’s publisher as determined by AllSides All, which provides labels left, left-leaning, neutral, right-leaning or right for a wide set of publishers. We discard any articles that remain unlabelled or have fewer than 10 likes. This results in a dataset consisting of an average of 1356 articles each day, of which 15% are right, 7% are right-leaning, 31% are neutral, 34% are left-leaning and 13% are left. On average, the most-liked right article has 42,293 likes, the most-liked right-leaning article has 144,624 likes, the most-liked neutral article has 48,647 likes, the most-liked left-leaning article has 117,267 likes and the most-liked left article has 107,497 likes. For each day, we encode each article as an arm with Bernoulli reward with mean proportional to the number of likes on Facebook (normalized to lie in the range $[0, 1]$).

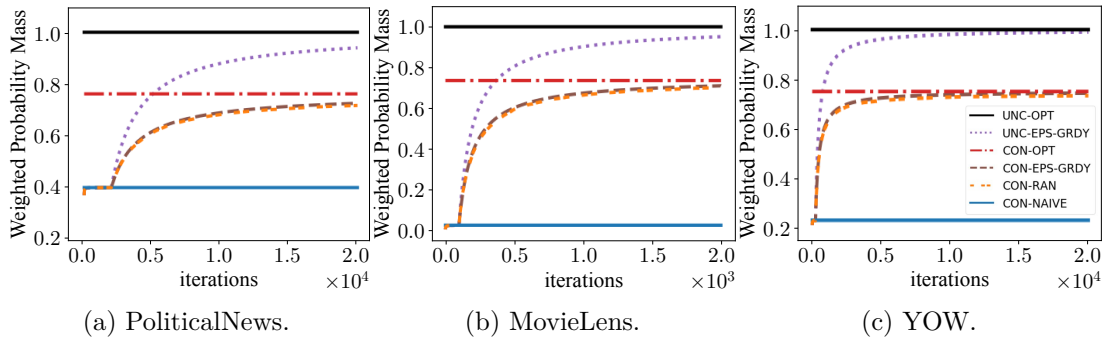


Figure 4.3 – **Visualizing Polarization and Diversification.** The weighted probability mass on the *best group* is reported against the number of iterations. While the unconstrained algorithm converges quickly to placing all of its probability mass on the optimal group, the constrained algorithm – by definition – maintains some weight on the non-optimal groups. This is what ensures diversification across content and avoids polarization.

We place a group weight of 0, 0.25, 0.5, 0.75 and 1 on right, right-leaning, neutral, left-leaning and left articles respectively for the *liberal* group ($w(L)$). Similarly, we place a group weight of 1, 0.75, 0.5, 0.25 and 0 on right, right-leaning, neutral, left-leaning and left articles respectively for the *conservative* group ($w(C)$).

MovieLens. We consider the MovieLens dataset Harper and Konstan [2015], which consists of 100,000 ratings from 943 users across 1,682 movies; each user rated at least 20 movies on a scale of 1 – 5. Each movie is also affiliated with one or more of 19 genres (e.g., sci-fi, romance, thriller). As some genres have significant overlap (e.g., thriller and horror), while others have different meanings at their intersections (e.g., romance vs rom-com vs comedy), we first cluster the movies into different meta-categories based on their genres using a black-box k -means clustering algorithm with $K = 25$ Pedregosa et al. [2011].⁷ We use the cluster centres as representative *arms*, and associate all movies in that cluster to that arm. For a given user, the reward associated with an arm is given by a Gaussian where the mean is the average rating the user gave to movies associated with the arm, and standard deviation $\sigma = 0.1$.

For a genre i ($i \in [19]$) and movie category a , the group weight $w_a(G_i)$ is set to be the i^{th} coordinate of the cluster centre of movie category a found by the k -means clustering.

YOW. We consider the YOW dataset Zhang [2005] which contains data from a collection of 24 paid users who read 5921 unique articles over a 4 week time period. The dataset contains the time at which each user read an article, a [0-5] rating for each article read by each user, and (optional) user-generated categories of articles viewed. We use this data to construct reward distributions for each user on a set of arms that one can expect to see from the real world.

⁷We determine $K = 25$ using the graph of silhouette values Rousseeuw [1987] vs. K .

We create a simple ontology to categorize the 10010 user-generated labels into a total of $g = 7$ groups of content: Science, Entertainment, Business, World, Politics, Sports, and USA. On average there are $K = 81$ unique articles in a day. We take this to be the number of *arms* in this experiment. Similar to the MovieLens experiments, we cluster the articles into 81 arms based on the news categories they belong to, using k -means clustering ($K = 81$). We use the cluster centres as representative *arms*, and associate all articles in that cluster to that arm. For a given user, the reward associated with that arm is given by a Gaussian where the mean is the average rating the user gave to articles associated with the arm, and standard deviation $\sigma = 0.1$.

For a news category i ($i \in [7]$) and article a , the group weight $w_a(G_i)$ is set to be the i^{th} coordinate of the cluster centre found by k -means clustering.

4.5.2 Empirical Results on Effect of Reducing Polarization on the Reward

We vary the tightness of upper bound constraints on the probability mass of displaying arms of a given group, and report the normalized cumulative reward.

PoliticalNews.

For this dataset, there are only two groups: either left or right. However, a news article may have weight on both groups, and it is these weights that determine how right- or left-leaning an article is, and hence how much they contribute towards polarization in a given direction. We simulated each of the 30 days separately, resulting in $n = 30$ datapoints. We report the normalized cumulative reward after $T = 10,000$ iterations, averaged over experiments from all 30 days. As there are only two groups, setting a lower bound constraint $\ell_1 = \zeta$ is equivalent to setting an upper bound constraint $u_2 = 1 - \zeta$. Hence, it suffices to see the effect as we vary the upper bounds. We vary $u_1 = u_2 = u$ from 0.5 to 1; i.e., from a fully constrained one in which each group has exactly 50% weighted probability of being selected to a completely unconstrained setting. We observe in Figure 4.2a that, even for very large values of u (i.e., when the constraints are loose), the CONS- ϵ -GREEDY algorithm significantly outperforms CONS-RAN with respect to regret, and is only worse than the unconstrained (and hence polarized) algorithm by an additive factor of approximately $\frac{1-u}{5}$ (i.e., less than 10%).

MovieLens.

For this dataset, a group corresponds to a *genre*. Note that a movie can belong to multiple genres with varying weights which may not add up to one. We report the normalized cumulative reward averaged across all 943 users after $T = 1000$ iterations. Error bars

depict the standard error of the mean. We observe in Figure 4.2b that CONS- ϵ -GREEDY significantly outperforms the CONS-NAIVE and CONS-RAN algorithms across constraints. Additionally, as there are fewer arms in the MovieLens dataset as compared to the PoliticalNews dataset, the learning cost is lower and hence the CONS- ϵ -GREEDY performs essentially as well as the (unattainable) CONS-OPTIMAL algorithm.

YOW

For this dataset, a group corresponds to an *article category*. Note that an article can belong to multiple categories (e.g., science and business) simultaneously, with varying weights across each category. We report the normalized cumulative reward averaged across all 21 users after $T = 10,000$ iterations. Error bars depict the standard error of the mean. As before, we observe in Figure 4.2c that CONS- ϵ -GREEDY significantly outperforms the CONS-NAIVE and CONS-RAN algorithms across constraints, and performs almost as well as the (unattainable) CONS-OPTIMAL algorithm.

4.5.3 Empirical Results on Polarization Over Time

In order to see how polarization can be avoided and diversification can be enforced using our framework, for each dataset we plot the normalized cumulative weighted probability mass on the *best group* for each datapoint against the number of iterations, with the $u = 0.75$. Initially, the unconstrained and constrained algorithms have the same weighted probability mass for the best group, because the algorithms are simply exploring the arms. However, the difference between the algorithms becomes very apparent once the algorithm begin to learn. Due to a larger number of arms in the PoliticalNews dataset, this process takes longer as compared to the other two datasets. UNCONSTRAINED- ϵ -GREEDY quickly polarizes almost-entirely to display content only from the best group. This depicts the necessity for such constraints. However, CONS- ϵ -GREEDY maintains at least $1 - u$ of its weighted probability mass on content not belonging to the best group, increasing diversification and avoiding polarization.

4.6 Summary

In this chapter, we initiate a formal study of combating polarization in personalization algorithms that learn user behavior. We present a general framework that allows one to prevent polarization by ensuring that a balanced set of items are displayed to each user. We show how one can modify a simple bandit algorithm in order to perform well with respect to regret subject to satisfying the polarization constraints, improving the regret bound over the state-of-the-art. Empirically, we observe that the CONS- ϵ -GREEDY algorithm performs well; it not only converges quickly to the theoretical optimum, but this optimum, even for the tightest constraints on the arm values selected ($u = 0.2$

for MovieLens, $u = 0.5$ for PoliticalNews), is within a factor of 2 of the unconstrained rewards. Furthermore, CONS- ϵ -GREEDY is fast and we expect it to scale well in web-level applications.

With regard to future work, a limitation of our algorithms is the fact that they assume we are given the group labels and weights for each piece of content. These labels would either need to be inferred from the data, which could bring with it additional bias associated with this learning algorithm, or would need to be self-reported, which can lead to adversarial manipulation. Additionally, it would be important to extend this work to a dynamic setting in which the type of content changes over time, e.g., using restless bandit techniques. From an experimental standpoint, testing this algorithm in the field, in particular to measure user satisfaction given diversified news feeds, would be of significant interest. Such an experiment would give deeper insight into the benefits and tradeoffs between personalization and the diversification of content, which could then be leveraged to determine which kind of constraints can prevent polarization not just of the items in the feed, but of the beliefs and opinions of those viewing them.

Appendix

4.A Constrained- L_1 -OFUL.

As explained in Remark 4.4.2, at any given time t , CONS- L_1 -OFUL maintains a regularized least-squares estimate for the mean reward vector $\boldsymbol{\mu}$, which is denoted by $\hat{\boldsymbol{\mu}}_t$. At each time step t , the algorithm first constructs a suitable confidence set B_t^1 around $\hat{\boldsymbol{\mu}}_t$. Roughly, the definition of this set ensures that the confidence ball is “flatter” in the directions already explored by the algorithm so it has more likelihood of picking a probability vector from unexplored directions. The algorithm chooses a probability distribution \mathbf{p}^t by solving a linear program on each of the $2k$ vertices of this confidence set, and plays an arm $a^t \sim \mathbf{p}^t$. Recall that for each arm $a \in [K]$, the mean reward is $\mu_a \in [0, 1]$. The reward for each time step is generated as $r_t \sim \text{Bernoulli}(\mu_{a^t})$, where $a^t \sim \mathbf{p}^t$ is the arm the algorithm chooses at the t^{th} time instant. The algorithm observes this reward and updates its estimate to $\hat{\boldsymbol{\mu}}_{t+1}$ for the next time-step appropriately.

CONS- L_1 -OFUL (Algorithm 4.2) is an adaptation of the OFUL algorithm that appeared in [Abbasi-Yadkori et al., 2011]. The key difference is that instead of using a scaled L_2 -ball in each iteration, we use a scaled L_1 -ball (Step 4 in Algorithm 4.2). As we explain below, this makes Step 5 of our algorithm efficient as opposed to that of Abbasi-Yadkori et al. [2011] where the equivalent step required solving a NP-hard and nonconvex optimization problem. This idea is similar to how CONF-BALL₂ was adapted to CONF-BALL₁ in [Dani et al., 2008]. In particular, our algorithm improves, by a multiplicative factor of $O(\log T)$, the regret bound of $\left(O\left(\frac{K^3}{\gamma} \log^3 T\right)\right)$ of CONF-BALL₁ in [Dani et al., 2008], see Table 4.1.

Next, we prove the regret guarantee of CONS- L_1 -OFUL.

Theorem 4.2. *Given the description of \mathcal{C} and the sequence of rewards drawn from a $O(1)$ -subgaussian distribution with the expectation vector $\boldsymbol{\mu}$. Assume that $\|\boldsymbol{\mu}\|_2 \leq \sigma$ for*

Chapter 4. Controlling Polarization in Personalization

some $\sigma \geq 1$. Then, with probability at least $1 - \delta$, the regret of CONS- L_1 -OFUL after time T is:

$$\begin{aligned} \text{CRegret}_T &\leq \frac{8k\sigma^2}{\gamma} \left(\log T + (K-1) \log \frac{64\sigma^2}{\gamma^2} + \right. \\ &\quad \left. 2(K-1) \log (K \log (1 + T/K) + 2 \log (1/\delta)) + 2 \log (1/\delta) \right)^2. \end{aligned}$$

Notations for the proof. For a positive definite matrix $\mathbf{A} \in \mathbb{R}^{K \times K}$, the weighted 1-norm and 2-norm of a vector $\mathbf{x} \in \mathbb{R}^K$ is defined by

$$\|\mathbf{x}\|_{1,\mathbf{A}} := \sum_{i=1}^K |\mathbf{A}^{1/2} \mathbf{x}|_i \text{ and } \|\mathbf{x}\|_{2,\mathbf{A}} := \sqrt{\mathbf{x}^\top \mathbf{A} \mathbf{x}}.$$

With some abuse of notations, let $\mathbf{p}^* := \arg \max_{\mathbf{p} \in \mathcal{C}} \boldsymbol{\mu}^\top \mathbf{p}$. Let the instantaneous regret R_t at time t of CONS- L_1 -OFUL be defined as the difference between the expected values of the reward received for \mathbf{p}^* and the chosen probability \mathbf{p}^t :

$$R_t = \boldsymbol{\mu}^\top (\mathbf{p}^* - \mathbf{p}^t).$$

The cumulative regret until time T , CRegret_T , is defined as $\sum_{t=1}^T R_t$. Recall that r^t is the reward that the algorithm receives at the t -th time instance. Note that the expected value of reward r^t is $\boldsymbol{\mu}^\top \mathbf{p}^t$. Let

$$\eta_t := r^t - \boldsymbol{\mu}^\top \mathbf{p}^t.$$

The fact that r^t is $O(1)$ -subgaussian implies that η_t is also $O(1)$ -subgaussian. Finally, recall that we denote our estimate of $\boldsymbol{\mu}$ at the t -th iteration by $\hat{\boldsymbol{\mu}}_t$.

We assume that we have an upper bound σ for the value of $\|\boldsymbol{\mu}\|_2$, i.e. $\|\boldsymbol{\mu}\|_2 \leq \sigma$ for some $\sigma \geq 1$.

Theorem 4.3 requires η_t to R -sub-Gaussian for a fixed $R \geq 0$. Formally, this means:

$$\forall \lambda \in \mathbb{R}, \quad \mathbb{E} \left[e^{\lambda \eta_t} \mid \mathbf{p}^1, \dots, \mathbf{p}^t, \eta_1, \dots, \eta_{t-1} \right] \leq \exp \left(\frac{\lambda^2 R^2}{2} \right).$$

We can prove that η_t is 1-sub-Gaussian if r^t is sampled from a Bernoulli distribution with probability of success μ_{a^t} (for $a^t \sim \mathbf{p}^t$) by showing that η_t is a zero-mean noise that lies in $[-1, 1]$. Since every bounded zero-mean noise lying in an interval of length at most $2R$ is R -sub-Gaussian, this would prove that η_t is 1-sub-Gaussian. Note that:

$$\begin{aligned} &\mathbb{E}[\eta_t \mid \mathbf{p}^1, \dots, \mathbf{p}^t, \eta_1, \dots, \eta_{t-1}] \\ &= \mathbb{E}_{a^t \sim \mathbf{p}^t} \mathbb{E}[\eta_t \mid a^t, \mathbf{p}^1, \dots, \mathbf{p}^t, \eta_1, \dots, \eta_{t-1}] \\ &= 0, \end{aligned}$$

Algorithm 4.2 CONS- L_1 -OFUL

Require: Constraint set \mathcal{C} , maximum failure probability δ , an L_2 -norm bound on $\boldsymbol{\mu}$:

- $\|\boldsymbol{\mu}\|_2 \leq \sigma$ and a positive integer T
- 1: Initialize $V_1 := \mathcal{I}$, $\hat{\boldsymbol{\mu}}_1 := 0$, and $\mathbf{b}_1 := 0$
 - 2: **for** $t = 1, \dots, T$ **do**
 - 3: Compute $\beta_t(\delta) := \left(\sqrt{2 \log \left(\frac{\det(V_t)}{\delta} \right)} + \sigma \right)^2$
 - 4: Denote $B_t^1 := \left\{ \boldsymbol{\mu} : \|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}_t\|_{1, V_t} \leq \sqrt{K \beta_t(\delta)} \right\}$
 - 5: Compute $\mathbf{p}^t := \arg \max_{\mathbf{p} \in \mathcal{C}} \max_{\boldsymbol{\mu} \in B_t^1} \boldsymbol{\mu}^\top \mathbf{p}$
 - 6: Sample a from the probability distribution \mathbf{p}^t
 - 7: Observe reward $r_t = r_a^t$
 - 8: Update $\mathbf{V}_{t+1} := \mathbf{V}_t + \mathbf{p}^t \mathbf{p}^{t \top}$
 - 9: Update $\mathbf{b}_{t+1} := \mathbf{b}_t + r_t \mathbf{p}^t$
 - 10: Update $\hat{\boldsymbol{\mu}}_{t+1} := \mathbf{V}_{t+1}^{-1} \mathbf{b}_{t+1}$
 - 11: **end for**
-

where the first equality follows from the law of iterated expectations and the second equality follows from simple arithmetic.

Further, if $r^t = 1$, $\eta_t = 1 - \boldsymbol{\mu}^\top \mathbf{p}^t$. Since $0 \leq \mu_a \leq 1$ for all $a \in [K]$, the value of η_t is upper bounded by 1. Similarly, if $r^t = 0$, $\eta_t = -\boldsymbol{\mu}^\top \mathbf{p}^t$. Since $0 \leq \mu_a \leq 1$ for all $a \in [K]$, the value of η_t is lower bounded by -1. Thus, η_t satisfies the R -sub-Gaussian condition required by Theorem 4.3 (Theorem 2 in [Abbasi-Yadkori et al., 2011]), with the value of $R = 1$. The same arguments hold for any zero-mean bounded noise.

Technical lemmas. Towards the proof of Theorem 4.2, we need some results from Dani et al. [2008] and Abbasi-Yadkori et al. [2011] that we restate in our setting. The first is a theorem from Abbasi-Yadkori et al. [2011] which helps us to prove that $\boldsymbol{\mu}$ lies in the confidence set B_t^1 at each time-step with high probability.

Theorem 4.3 (Theorem 2 in [Abbasi-Yadkori et al., 2011]). *Assume that the rewards are drawn from an $O(1)$ -subgaussian distribution with the expectation vector $\boldsymbol{\mu}$. Then, for any $0 < \delta < 1$, with probability at least $1 - \delta$, for all $t \geq 0$, $\boldsymbol{\mu}$ lies in the set*

$$B_t^2 := \left\{ \boldsymbol{\mu} \in \mathbb{R}^K : \|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}_t\|_{2, V_t} \leq \sqrt{\beta_t(\delta)} \right\}$$

where β_t is defined in Step 5 of the CONS- L_1 -OFUL algorithm.

As a simple consequence of this theorem we prove that $\boldsymbol{\mu}$ lies inside B_t^1 with high probability.

Chapter 4. Controlling Polarization in Personalization

Lemma 4.4. $\boldsymbol{\mu}$ lies in the confidence set B_t^1 with a probability at least $1 - \delta$ for all $t \in T$.

Proof.

$$\|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}_t\|_{1, \mathbf{V}_t} \leq \sqrt{K} \|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}_t\|_{2, \mathbf{V}_t} \leq \sqrt{K \beta_t(\delta)},$$

Here, the first inequality follows from Cauchy-Schwarz and the second inequality holds with probability at least $1 - \delta$ for all t due to Theorem 4.3. \square

The following four lemmas would be required in the proof of our theorem.

Lemma 4.5 (Lemma 7 in [Dani et al., 2008]). *For all $\boldsymbol{\mu} \in B_t^1$ (as defined in Step 6 of CONS- L_1 -OFUL) and all $\mathbf{p} \in \mathcal{C}$, we have:*

$$|(\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}_t)^\top \mathbf{p}| \leq \sqrt{K \beta_t(\delta) \mathbf{p}^\top \mathbf{V}_t^{-1} \mathbf{p}}$$

where \mathbf{V}_t is defined in Step 10 of the CONS- L_1 -OFUL algorithm.

Lemma 4.6 (Lemma 8 in [Dani et al., 2008]). *If $\boldsymbol{\mu} \in B_t^1$, then*

$$R_t \leq 2 \min \left(\sqrt{K \beta_t(\delta) \mathbf{p}^{t\top} \mathbf{V}_t^{-1} \mathbf{p}^t}, 1 \right).$$

Lemma 4.7 (Lemma 11 in [Abbasi-Yadkori et al., 2011]). *Let $\{\mathbf{p}^t\}_{t=1}^T$ be a sequence in \mathbb{R}^K , $\mathbf{V} = \mathcal{I}_K$ be the $K \times K$ identity matrix, and define $\mathbf{V}_t := \mathbf{V} + \sum_{\tau=1}^t \mathbf{p}^\tau \mathbf{p}^{\tau\top}$. Then, we have that:*

$$\log \det(\mathbf{V}_t) \leq \sum_{t=1}^T \|\mathbf{p}^t\|_{\mathbf{V}_{t-1}^{-1}}^2 \leq 2 \log \det(\mathbf{V}_t).$$

Finally, we state another result from the proof of Theorem 5 in [Abbasi-Yadkori et al., 2011].

Lemma 4.8. *For any T , we have the following upper bound on the value of $\frac{\beta_T(\delta)}{\gamma} \log \det(\mathbf{V}_t)$:*

$$\begin{aligned} \frac{\beta_T(\delta)}{\gamma} \log \det(\mathbf{V}_t) &\leq \frac{\sigma^2}{\gamma} \left(\log T + (K-1) \log \frac{64\sigma^2}{\gamma^2} \right. \\ &\quad \left. + 2(K-1) \log(K \log(1 + T/K) + 2 \log(1/\delta)) + 2 \log(1/\delta) \right)^2 \end{aligned}$$

where σ, δ are as in Theorem 4.2.

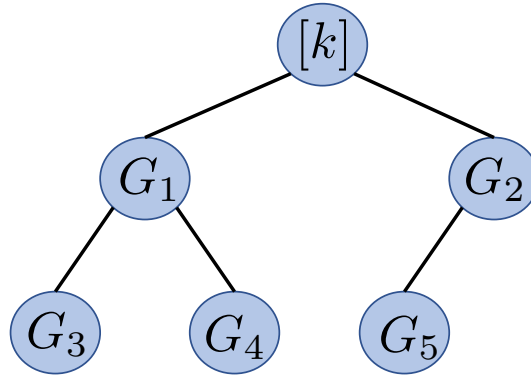


Figure 4.4 – Laminar Group structure.

4.B Laminar Constraints

In this section, we consider a laminar type of constraints. Let the Groups $G_1, \dots, G_g \subseteq [K]$ be such that: $G_i \cap G_j \neq \emptyset$ implies $G_i \subseteq G_j$ or $G_j \subseteq G_i$. We assume that the group weights are *binary*, i.e. an item either belongs to a group or doesn't.

In this case, the LP can be solved efficiently by a greedy algorithm. The groups form a tree-like data structure, where the children are the largest groups that are subset of the parents. For example in Figure 4.4, the groups G_1 and G_2 are subsets of the arms $[K]$ and $G_1 \cap G_2 = \emptyset$. Similarly, the groups G_3 and G_4 are subsets of the group G_1 and $G_3 \cap G_4 = \emptyset$. G_5 is a subset of G_2 .

If the lower bound ℓ_i for a group G_i is smaller than the sum of the lower bounds for the children groups, then we increase it to the sum of of the lower bounds for the children groups. For example in Figure 4.4, if $\ell_1 < \ell_3 + \ell_4$, then we increase it to $\ell_3 + \ell_4$. This is because satisfying the lower bound of G_3 and G_4 automatically satisfies the probability of G_1 . Similarly, if the upper bound u_i for a group G_i is larger than the sum of the lower bounds for the children groups, then we decrease it to the sum of of the upper bounds for the children groups. For example in Figure 4.4, if $u_1 > u_3 + u_4$, then we decrease it to $u_3 + u_4$. This is again because the total probability that an arm in group G_i is selected cannot be larger than the upper bounds of its children. This change of the upper and lower bounds does not change the optimum of the LP problem.

In the greedy algorithm, first we satisfy the lower bounds, then we allocate the remaining probability such that the upper bounds are not violated.

In satisfying the lower bounds, we take a bottom-up approach. We start from the leaves and satisfy the lower bound by giving the item to the arm a with the largest reward in the group, i.e., $\arg \max_{a \in G_i} \mu_a$.

In our example, we set the probability of $\arg \max_{a \in G_3} \mu_a$ to ℓ_3 , the probability of

arg max $_{a \in G_4} \mu_a$ to ℓ_4 and the probability of arg max $_{a \in G_5} \mu_a$ to ℓ_5 . Next, we proceed with satisfying the lower bound for the parents. In our example, we add the probability of $\ell_1 - (\ell_3 + \ell_4)$ to arg max $_{a \in G_1} \mu_a$, and the probability of $\ell_2 - \ell_5$ to arg max $_{a \in G_2} \mu_a$. We continue the process until no group remains infeasible. Finally, we assign the remaining probability to the arm with the largest reward. In our example, we add the probability of $1 - (\ell_1 + \ell_2)$ to arg max $_{a \in [K]} \mu_a$.

The remaining probability is first allocated to arg max $_{a \in [K]} \mu_a$ until we reach the probability for one of the upper bound constraints. Then, we eliminate the arms inside that group, and we allocate some probability to the arm with the maximum reward until another upper bound constraint is reached. We continue this process until either our distribution over arms is a probability distribution or we cannot allocate more probability to any arm without violating a constraint.

Let the probability that an arm from the group G_i is selected be $\sum_{a \in G_i} v_a = q_i$ and the children of group G_i be $G_{i1}, G_{i2}, \dots, G_{im}$. We denote the optimal allocation (subject to the constraints) at node G_i be $OPT(G_i, q_i, \ell, u)$. Then, we have

$$OPT(G_i, q_i, \ell, u) = \sum_{j=1}^m OPT(G_{ij}, \ell_{ij}, \ell, 1) + OPT(G_i, q_i - \sum_{j=1}^m \ell_{ij}, 0, u). \quad (4.12)$$

In satisfying the lower bounds (i.e., first term in (4.12)) if we do not change the probability of selecting an arm inside a group G_i , i.e., $\sum_{a \in G_i} v_a$, then it does not effect the parent groups, hence we can locally optimize the problem beginning from the smaller groups.

We can show by contradiction that the procedure for allocating the remaining probability (i.e., second term in (4.12)) is optimal. This is because, if the probability of arg max $_{a \in [K]} v_a$ can be increased without violating an upper bound or we can increase it by reducing another arm's probability of winning, then the current probability allocation is not optimal.

The running time of the algorithm is linear in the number of the arms k and height of the tree. Given that height of the tree is less than g , the total running time becomes $O(gk)$.

4.B.1 Budget Type Constraints

Remark 4.9. *The constraints on the probabilities p_a^t can be easily translated to long term budget type constraints*

$$\ell_i \leq \sum_{a \in G_i} w_a(G_i) \cdot \frac{n_a^T}{T} \leq u_i \quad \forall i \in [g], \quad (4.13)$$

where n_a^T is the number of times that content a is selected. Because a simple application of Hoeffding's inequality yields that

$$\mathbb{P} \left[\left| \frac{n_a^T}{T} - p_a^T \right| \geq \epsilon \right] = O(-\epsilon^2 T^2). \quad (4.14)$$

Hence, if we satisfy the constraint (4.1) with a ϵ margin, then with high probability $1 - O(\epsilon^2 T^2)$, the budget constraint (4.13) is also satisfied. Yet, the question remains: is it better to satisfy a probability constraint as in Inequality (4.1) in each round t or a global budget constraint as in Inequality (4.13)? Indeed, the following simple lemma shows that, in expectation, satisfying a probability constraint yields at least as much utility as if we satisfy a global budget constraint.

Lemma 4.10. *The maximum utility gained by solving the following problem*

$$\begin{aligned} & \max_{p \in \mathcal{C}} \mathbb{E}_{r \sim \mathcal{D}, \tilde{a} \sim p} \left[\sum_{t=1}^T r_{\tilde{a}}^t \right], \\ & \text{S.t. } \ell_i \leq \sum_{a \in G_i} w_a(G_i) \cdot p_a^t \leq u_i \quad \forall i \in [g], \forall t \in [T], \end{aligned}$$

is at least as large as the utility gained by solving the following problem

$$\begin{aligned} & \max_{p \in \mathcal{C}} \mathbb{E}_{r \sim \mathcal{D}, \tilde{a} \sim p} \left[\sum_{t=1}^T r_{\tilde{a}}^t \right], \\ & \text{S.t. } \ell_i \leq \sum_{a \in G_i} w_a(G_i) \cdot \frac{n_a^T}{T} \leq u_i \quad \forall i \in [g]. \end{aligned}$$

The proof is simple and removed for brevity.

5 Learn from Thy Neighbor

In previous chapters, we have studied how machine learning algorithms can benefit from multi-armed bandit algorithms, and how multi-armed bandit algorithms can affect humans. In this chapter¹, we study how human interactions can be studied in the MAB framework. An individual's decisions are often guided by *those of his or her peers*, i.e., neighbors in a social network. Presumably, being privy to the experiences of others aids in learning and decision making, but how much advantage does an individual gain by observing her neighbors? Such problems make appearances in sociology and economics and, in this chapter, we present a novel model to capture such decision-making processes and appeal to the classic multi-armed bandit framework to analyze it. Each individual, in addition to her own actions, can observe the actions and rewards obtained by her neighbors, and can use all of this information in order to minimize her own regret. We provide algorithms for this setting, both for stochastic and adversarial bandits, and show that their regret smoothly interpolates between the regret in the classical bandit setting and that of the full-information setting as a function of the neighbors' exploration. In the stochastic setting the additional information must simply be incorporated into the usual estimation of the rewards, while in the adversarial setting this is attained by constructing a new unbiased estimator for the rewards and appropriately bounding the amount of additional information provided by the neighbors. Further, we show via empirical simulations that our algorithms, often significantly, outperform existing algorithms that one could apply to this setting.

5.1 Introduction

Individuals often have access to information, via their social or economic network, that they can use to make improved decisions. This phenomenon has been observed widely in the social and natural sciences. For instance, a recent work [Yoo, 2012] studies farmers who, every year, have to decide which kind of seed to plant (not just what kind of crop,

¹This chapter is based on [Celis and Salehi, 2017].

but which variety of seed) in order to attain the most profit (i.e., revenue - cost). In their study, Yoo [2012] finds that farmers' decisions are based on (i) their own experience in previous years of how different varieties performed, and (ii) the experiences of peers attained either directly (explicitly via conversations with social contacts) or indirectly (implicitly by observing the farming practices of peers). Moreover, the information farmers used is primarily from peers in their *physical neighborhood* – not only because these are where their contacts are most likely to be, but also because the profit is correlated due to similar soil and weather conditions. These connections between peers then form a network of farmers across the country, where locally, each farmer is trying to learn the best seed for their farm using her own information and that of her neighbors. As another example, consider WI-FI networks in which nodes want to send their data across the best frequency band. Nodes could obtain the current quality of the band their peers are using indirectly through capacity estimation or directly by message passing, and use this information to determine which band to use. Similar social learning phenomena appear in many other areas in various disguises – e.g., in the acquisition of consumer products by individuals, the adoption of new technologies, the prevalence and spread of corruption, and in the behavior of animals such as squirrels; see [Lazarsfeld et al., 1948, Katz and Lazarsfeld, 1955, Zhang et al., 2007, Sanditov, 2006, Accinelli and Sánchez-Carrera, 2012].

Consider the following formulation geared towards capturing the type of settings mentioned above (see also Figure 5.1): at each time step each individual selects one of K possible actions, observes the value or reward of selecting that action, and observes the actions, values and/or decision process of their neighbors in the network. This selection and observation is repeated again and again, and each individual has the end goal of identifying the action $a^* \in K$ that brings them the best value over all time steps; i.e., minimizing the *regret*. This formulation seems to suggest that the problem is suited for study using the multi-armed bandit optimization framework (detailed in Section 1.2), except that now there is additional information available to an individual via her neighbors.

Towards this, one approach could be to consider the framework of bandits with *side observations* for which, in the adversarial setting, variants of the multiplicative-weight update algorithm have been developed with success. Informally, side observations just mean that at each time step, in addition to observing the reward of a selected action a_t , one may observe (but not receive) rewards from a set of other actions $S(t)$. A recent body of work has explored how to minimize regret for various different models of $S(t)$. In the *free observation model*, the individual is allowed to select $S(t)$ up to some cardinality (e.g., Amin et al. [2015]). However, if one tried to apply such algorithms to the social settings considered above, it would require an individual to decide which actions her neighbors should take, and hence is not feasible as a solution in this setting. In another line of work (e.g., Alon et al. [2015]) an *action-network model* has been studied: Here, the actions form a network and the individual observes the rewards of the neighbors of the *action* she selects (as opposed to the rewards of the actions that *her* neighbors select). The action-network is often taken to be exogenous and can be changing over

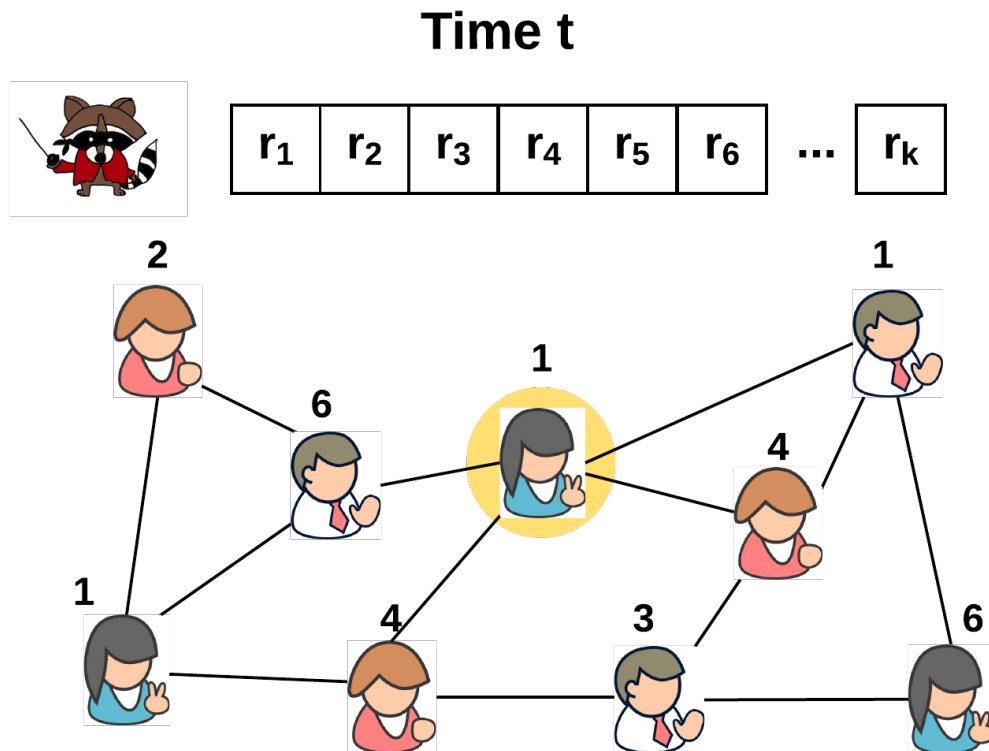


Figure 5.1 – A social network in which all individuals play against the same bandit, i.e., if two individuals select the same arm at the same time step, they observe the same loss (up to noise). At each time t , each individual selects an arm (shown), and then observes its loss along with the actions/losses of her neighbors. E.g., the yellow circled individual would observe the loss of actions 1, 4 and 6 in this time step.

time. Thus, one can apply the algorithms developed in the action-network setting to the social setting above by defining $S(t)$ to be the set of actions selected by the individual's neighbors; however, this may not always be optimal for the social setting as neighbors can provide even more information (see Section 5.5.2).

The above approaches have been developed in independent contexts and hence geared towards different settings. Towards obtaining optimal results in the social setting described above, the challenge is to adequately model the information from neighbors that can aid in learning, in leverage it appropriately, and in quantify the advantage it provides.

Our Contributions

In this work, we show how one can incorporate additional social information in order to obtain optimal results. More specifically, in the stochastic setting, we show that incorporating side information in a simple way gives rise to a near-optimal algorithm. Our

UCBN algorithm extends the classic UCB algorithm by incorporating all observed samples indiscriminately. We show that this suffices to improve performance, often dramatically, both asymptotically and in silico. We show that the regret of our algorithm interpolates between $O(1)$ and the $O(K \ln T)$ regret for the classic bandit setting depending on the amount of exploration conducted by the neighbors (see Theorem 5.1), and these bounds are asymptotically optimal (see Theorem 5.13). The theoretical results are presented in Section 5.4 and Appendix 5.B, and the empirical results are presented in Section 5.6.3. In the adversarial setting, we present a modified multiplicative-weight update algorithm that uses a new unbiased estimator to incorporate this side information appropriately into the estimation of the reward of each action. We show that the regret of our algorithm provably interpolates between the $O(\sqrt{KT \ln K})$ regret for the classical bandit setting and the $O(\sqrt{T \ln K})$ regret for the full-information setting (where all the K rewards are observed at each time step) depending on the amount of exploration conducted by the neighbors (see Theorem 5.2). The proofs requires us to overcome some additional hurdles in order to bound the amount of information gleaned from the neighbors and attain optimal bounds on the regret. The theoretical results are presented in Section 5.5 and Appendix 5.A, and the empirical results are presented in Section 5.6.1.

5.2 Preliminaries

The Model. Firstly, we assume all individuals play against the same multi-armed bandit: in the stochastic setting, the reward distribution \mathcal{D} is the same for arm i and for all individuals (although the realizations at any given time may differ), and in the adversarial setting the reward vector \mathbf{r}^t selected by the adversary at time t is the same for all individuals. Clearly, there must be some similarity in the rewards between neighbors for social learning to occur. Our results also extend to the setting in which the distributions or rewards are correlated, e.g., via 0-mean noise (i.e., each individual i receives the reward as above + individual noise); for ease of presentation we omit this extension, the proofs follow analogously. Secondly, we assume that each individual can observe the following for all *neighbors* i :

- (1) the actions a_i^t ,
- (2) the rewards $r_{a_i^t}^t$, and
- (3) (for the adversarial setting only) the probability distribution each neighbor used to select an arm at *the previous* time step.

Assumptions (1) and (2) are natural and directly inspired by applications such as those mentioned in the introduction; individuals either directly or indirectly observe their neighbors' actions and rewards. (3) additionally assumes a limited knowledge of how neighbors made their decisions on a step-by-step basis, without having to assume we know their overall algorithm or restricting their behavior in any way. All individuals are

free to select their probability distributions arbitrarily (and depend arbitrarily on each other), and each one can draw her decision independently of the rest. While it would be nice to drop Assumption (3) entirely, this would prevent us from attaining optimal regret bounds.²

Importantly, an individual

- (4) *does not* know about the actions and rewards of individuals beyond her neighbors,
- (5) *does not* know any global properties of the network,
- (6) *does not* know which algorithm other individuals (including neighbors) are using, and
- (7) *cannot* dictate or coerce other individuals to act a certain way.

Removing any of Assumptions (4)-(7) would be unnatural for the social learning setting described above: If (4) does not hold, we would simply consider such an individual a neighbor, removing (5) is unnatural as the network can be very large and we cannot expect to have knowledge of distant individuals, removing (6) seems impractical as it would mean that individuals have a detailed knowledge of how neighbors select actions, and allowing individuals to be coerced as in (7) would be in conflict with the idea that every individual seeks to improve her own performance. Hence, given (1)-(7), any improvement in the individual's regret arises solely from *passive* observation of *local* information.

5.3 Related Work

Distributed learning in a network is a broad topic and has been studied under various names in several disciplines. However, to the best of our knowledge, our model for learning from neighbors, along with its assumptions and non-assumptions (1)-(7) which are motivated by relevant settings in sociology and economics, is novel. Here we briefly survey the closest relatives to our work.

In the study of *non-strategic learning on networks*, individuals are connected via a network, and each individual has a finite set of actions with probabilistic rewards whose distributions depend on the *state of the world* (see [Goyal, 2005]). Indeed, this setting is similar to our model in the stochastic setting. However, work in this area has focused on studying variants of a *greedy* algorithm, and answering the question of whether learning (i.e., discovery of the state of the world, and hence convergence to the best action) occurs asymptotically (see, e.g., [Bala and Goyal, 1998, Ellison and Fudenberg, 1993, Bala and

²Alternatively, under different assumptions (e.g., if we assume the neighbors are using an algorithm such as EXP3) we can estimate these distributions which would suffice.

Algorithm 5.1 UCBN

```

1: Input:  $\alpha$ 
2: Initialize: empirical means  $\bar{\mu}(1) := 0$  and counts  $\mathbf{n}(1) := 0$ 
3: for  $t = 1, \dots, T$  do
4:   Choose  $i = \arg \max_{j \in [K]} \bar{\mu}_j(t) + \sqrt{\frac{\alpha \ln t}{2n_i(t)}}$ 
5:   Observe all rewards (itself and neighbors)
6:   Update empirical means  $\bar{\mu}$  and counts  $\mathbf{n}$ 
7: end for

```

Goyal, 2001, Gale and Kariv, 2003, Golub and Jackson, 2010]). Instead, we are concerned with *regret*, which could be loosely interpreted as the *rate of convergence*.

Recall that in models of bandits with *side observations*, in addition to observing $r_{a^t}^t$, one may *observe* (but not receive) additional rewards $r_{S(t)}^t$. The set of arms $S(t)$ depends on the particular model of side observations. In the free observations model, the individual can select B additional arms to observe at each time step; i.e., $|S(t)| = B$ and the individual selects $S(t)$ for all t . Such models have been studied both for stochastic [Yu and Mannor, 2009] and adversarial [Avner et al., 2012, Amin et al., 2015] bandits. Without Assumption (7), we could apply such algorithms directly because an individual could dictate which actions her neighbors should take. In the social setting we cannot hope to control our neighbor’s decisions in this manner. Still, we show that the performance of our algorithm is equivalent empirically to such algorithms (see Section 5.6). In the arm-network (or action-network) setting, the individual observes the rewards of the neighbors of the arm she selects. Such stochastic [Caron et al., 2012, Buccapatnam et al., 2014] and adversarial [Mannor and Shamir, 2011, Alon et al., 2013, Kocák et al., 2014, Alon et al., 2015] bandit settings have been studied. While one could apply these algorithms to the social setting, some social information, in particular from Assumption (3), is left on the table. Leveraging this allows us to provably outperform such approaches (see Section 5.5.2), and empirically the difference can be dramatic (see Figure 5.3).

Other work has considered bounding the *cumulative* regret of all individuals, rather than individuals minimizing their own regret. Towards this, centralized algorithms for various versions of stochastic bandits have been studied, in particular for the complete graph [Buccapatnam et al., 2013, Szorenyi et al., 2013, Cesa-Bianchi et al., 2013]. Although the centralized setting is not the object of our study, as a corollary, we obtain a centralized algorithm for adversarial bandits that is optimal on the complete network (see Section 5.5.3).

5.4 Technical Contributions for the Stochastic Setting

The algorithm for stochastic setting is based on UCB, see Section 1.2.3 for a review of the UCB algorithm and its performance guarantees. We make a simple extension to UCB for an agent on a network: the agent simply incorporates *all* samples and *all* rewards into n_j and $\bar{\mu}_j$ regardless of whether it came from her action or was observed from one of her neighbors. We denote this algorithm by UCBN, and we note that it can be implemented by an individual irrespective of the graph structure and the algorithm(s) her neighbors may employ. The regret of UCBN algorithm is upper bounded as follows.

Theorem 5.1. *Consider an agent with neighbors who play arbitrarily. Let m_i^t be the number of times arm i has been selected by one of her neighbors by time t . Then, the regret of UCBN is*

$$\bar{R}_{\text{UCBN}}^T \leq \sum_{i, \Delta_i > 0} \left(\max \left\{ \max_{t=1, \dots, T} \left\{ \frac{2\alpha \ln t}{\Delta_i} - m_i^t \Delta_i \right\}, 0 \right\} + \frac{\alpha}{\alpha - 2} \right), \quad (5.1)$$

where Δ_i is the difference between the expected reward of the best arm μ_{i^*} and the expected reward of a sub-optimal arm μ_i , and $\alpha > 2$ is a constant that depends on the variance of the rewards for which (1.5) holds.

This result is asymptotically optimal (see Theorem 5.13). The regret differs from the regret of the classic UCB regret (1.5) (in Section 1.2.3) by the $-m_j^T \Delta_j$ term, and, depending on the behavior of the neighbors, can potentially take the agent from logarithmic to a constant regret.

Clearly, the performance of an agent must depend on the behavior of her neighbors. In the worst case, if there are *clumsy* agents who always select the same arm, then our regret is not improved much. However, as long as the agent has at least one neighbor who explores an arm uniformly at random with probability $\varepsilon^t \in \Omega(\frac{K \ln t}{t})$ at time t (e.g., this occurs if a neighbor uses an adaptive greedy algorithm), then the regret is $O(1)$! Hence, this allows us to interpret neighbor behavior to our regret seamlessly. As an instructive example, consider the setting where all agents use UCBN in a complete graph. The regret in this setting is $O(\frac{K \ln T}{b})$, where b is the number of nodes. In other words, the regret of an agent using UCBN is a factor $O(1/b)$ less than that of an agent using UCB – indeed we cannot hope to do better, even in a completely centralized setting. The proof parallels the proofs for the original UCB results (see, e.g., [Bubeck and Cesa-Bianchi, 20120] for a template), and can be found along with further discussion in Appendix 5.B. While the story for the stochastic setting turns out to be simple and easy to manage, the adversarial setting, as we see below, turns out to be more challenging.

5.5 Technical Contributions for the Adversarial Setting

In the adversarial setting, to ease the presentation of the algorithm and results, we use the loss $l_i^t = 1 - r_i^t \in [0, 1]$ instead of rewards.

We call our algorithm in the adversarial setting EXPN. Recall that p_j^t is the probability that an individual selects arm j at time t . Let $q_j^{i,t}$ be the probability that her neighbor i selects arm j at time t . We denote the number of an individual's neighbors by b . Note that the number of nodes in a network, denoted by N , may be much larger, but the remaining network does not play a role in the algorithm or main results.

Theorem 5.2. *Given an individual with b neighbors who are playing arbitrarily, the regret when using EXPN (Algorithm 5.2) is*

$$\bar{R}_{\text{EXPN}}^T \leq \mathbb{E} \left[2 \sqrt{\left(T + \sum_{t=1}^T \gamma^t \right) \ln K} \right], \quad (5.2)$$

where $\gamma^t = \sum_{j=1}^K \frac{p_j^t}{p_j^t + \sum_{\ell=1}^b q_j^{\ell,t}}$.

For ease of presentation, momentarily assume that for all arms $j \in [K]$, neighbors $i \in [b]$, and times $t \in [T]$ we have that $q_j^{i,t} \geq \frac{\varepsilon_i}{K}$ for some $\varepsilon_i \in (0, 1]$.³ We can then reinterpret the regret of EXPN in (5.2) \bar{R}_{EXPN}^T as a function of the bandit regret (\bar{R}_{EXP3}^T) as follows:

$$\bar{R}_{\text{EXPN}}^T = \begin{cases} \bar{R}_{\text{EXP3}}^T = O(\sqrt{TK \log K}) & \text{if } \sum_{i=1}^b \varepsilon_i = O(1), \\ \bar{R}_{\text{EXP3}}^T / \sqrt{\varepsilon_i} = O(\sqrt{T \sqrt{K} \log K}) & \text{if } \sum_{i=1}^b \varepsilon_i = \Theta(\sqrt{K}), \\ \bar{R}_{\text{EXP3}}^T / \sqrt{\varepsilon_i} = O(\sqrt{T \log K}) & \text{if } \sum_{i=1}^b \varepsilon_i = \Theta(K). \end{cases} \quad (5.3)$$

In particular, note that when $\sum_{i=1}^b \varepsilon_i = O(1)$, none of the individual's neighbors maintain a probability distribution that is bounded away from 0 for all arms. In other words, the neighbors are not exploring effectively. In this case, the regret is $\bar{R}_{\text{EXPN}}^T \in O(\sqrt{TK \ln K})$, the same as in the classic bandit setting. On the other hand, for example, when $\sum_{i=1}^b \varepsilon_i = \Theta(K)$, then the regret is $\bar{R}_{\text{EXPN}}^T \in O(\sqrt{T \ln K})$, the same as in the full-information setting. Hence, this algorithm smoothly interpolates between bandit regret and full information regret as a function of the neighbors' exploration.

At first, the proof of Theorem 5.2 parallels standard approaches to analyze the multiplicative-weight update method; the crucial difference is a new unbiased estimator that is used in order to incorporate the neighbors' information (see Section 5.5.1). This leads to the

³This assumption is not required for the proof of Theorem 5.2, and it is only used for the ease of interpretation in (5.3). Note that if a neighbor is running any variant of the multiplicative weight update method, this condition is satisfied. Removing this assumption requires the *number* of non-zero ε_j to be tracked for each j , and these numbers would appear in the regret bound.

following bound on the regret:

$$\bar{R}_{\text{EXP3}}^T \leq \frac{\ln K}{\delta} + \delta T \sum_{j=1}^K \frac{p_j^t}{p'_j(t)}.$$

The technical obstacle then becomes attaining tight bounds on the $\sum_{j=1}^K \frac{p_j^t}{p'_j(t)}$ term (see Lemma 5.3).

5.5.1 The EXPN Algorithm

EXPN is a multiplicative weight-update algorithm (see Section 1.2.2 for details about the multiplicative weight-update algorithms), key to our EXPN algorithm is the following new unbiased estimator for the losses:

$$\hat{l}_j^t = \begin{cases} \frac{l_j^t}{p'_j(t)} & \text{if some individuals select action } j \text{ at time } t \\ 0 & \text{otherwise,} \end{cases} \quad (5.4)$$

where $p'_j(t)$ is the probability that at least one individual selects action j , i.e.,

$$p'_j(t) \stackrel{\text{def}}{=} 1 - (1 - p_j^t)(1 - q_j^{1,t}) \cdots (1 - q_j^{b,t}). \quad (5.5)$$

The algorithm then updates the weights according to

$$w_j(t+1) = w_j^0 e^{-\delta \sum_{s=1}^t \hat{l}_j^s},$$

where $w_j^0 = 1$, and updates the probability distributions according to $p_j^t = w_j^t / W^t$ where $W^t = \sum_j w_j^t$. Note that this algorithm can be implemented irrespective of the network structure and depends only on the information obtained locally from neighbors as defined in our model. In essence, the key to our algorithm is two fold:

1. *Design* a new unbiased estimator \hat{l}_j^t that incorporates the side observations obtained from neighbors: Unlike for stochastic bandits, naïve estimators do not suffice, and a new approach is required.⁴
2. *Decouple* the exploration and exploitation parameters: When an individual's neighbors explore a lot, she could benefit by *free-riding* off of the exploration of her neighbors; this is accomplished by decreasing her exploration parameter. However, if we take δ fixed as in EXP3, this dampens our updates. Hence we need δ to depend on the strategy of the neighbors.

⁴We must ensure that in bounding $\mathbb{E}[(\hat{l}_j^t)^2]$, we get some improvement over the usual bandit setting; it is easy to verify that such bounds do not hold for naïve estimators such as the average of the neighbors' estimators.

Algorithm 5.2 EXPN

1: **Initialize:** $\hat{L}_j^1 = 0$ and $p_j^1 = \frac{1}{K}$ \triangleright for all $j \in [K]$
2: **for** $t = 1 : T$ **do**
3: Compute $\gamma^t = \sum_{j=1}^K \frac{p_j^{t-1}}{p_j^{t-1} + \sum_{\ell=1}^b q_j^{\ell, t-1}}$
4: Update $\delta^t = \sqrt{\frac{\ln K}{\sum_{\tau=1}^t (1 + \gamma^\tau)}}$
5: Sample $j \sim \mathbf{p}^t$
6: Compute the unbiased estimator \hat{l}^t (5.4) for the losses
7: $\hat{L}_j^{t+1} = \hat{L}_j^t + \hat{l}_j^t$ \triangleright for all $j \in [K]$
8: $w_j^{t+1} = \exp(-\delta^t \hat{L}_j^{t+1})$ \triangleright for all $j \in [K]$
9: $W^{t+1} = \sum_{j=1}^n w_j^{t+1}$
10: $p_j^{t+1} \leftarrow \frac{w_j^{t+1}}{W^{t+1}}$, \triangleright for all $j \in [K]$
11: **end for**

Proof of Theorem 5.2. The first part of the proof (up to (5.12)) parallels the traditional analysis of multiplicative weight update algorithms; for completeness we present the steps without going into the details (see [Bubeck and Cesa-Bianchi, 20120] for an exposition). The expected loss \bar{l}^t at iteration t is equal to $\sum_{j=1}^K p_j^t l_j^t$. We can write the expected loss as

$$\bar{l}^t = \mathbb{E}_{j \sim \mathbf{p}^t} \left[\mathbb{E}_{a^t \sim \mathbf{p}'(t)} [\hat{l}_j^t] \right],$$

where $\mathbb{E}_{a^t \sim \mathbf{p}'(t)} [\hat{l}_j^t] = l_j^t$ and $\mathbb{E}_{j \sim \mathbf{p}^t} [l_j^t] = \bar{l}^t$. We first rewrite the expected loss \bar{l}^t as follows:

$$\mathbb{E}_{j \sim \mathbf{p}^t} \left[\mathbb{E}_{a^t \sim \mathbf{p}'(t)} [\hat{l}_j^t] \right] = \frac{1}{\delta^t} \ln \mathbb{E} \left[\exp \left(-\delta^t (\hat{l}_j^t - \mathbb{E}[\hat{l}_j^t]) \right) \right] - \frac{1}{\delta^t} \ln \mathbb{E} \left[\exp(-\delta^t \hat{l}_j^t) \right], \quad (5.6)$$

where the expectation is over the randomness of the estimator ($a^t \sim \mathbf{p}'(t)$) and choice of the arm ($j \sim \mathbf{p}^t$): $\mathbb{E}_{j \sim \mathbf{p}^t} [\mathbb{E}_{a^t \sim \mathbf{p}'(t)} [\cdot]]$. We will now consider the right-hand side of (5.6) and upper bound the two terms separately.

$$\begin{aligned} \frac{1}{\delta^t} \ln \mathbb{E} \left[\exp \left(-\delta^t (\hat{l}_j^t - \mathbb{E}[\hat{l}_j^t]) \right) \right] &= \frac{1}{\delta^t} \ln \mathbb{E} \left[\exp(-\delta^t \hat{l}_j^t) \right] + \mathbb{E}[\hat{l}_j^t] \\ &\leq \frac{1}{\delta^t} \mathbb{E} \left[\exp(-\delta^t \hat{l}_j^t) - 1 + \delta^t \hat{l}_j^t \right] \\ &\leq \frac{\delta^t}{2} \mathbb{E}[(\hat{l}_j^t)^2], \end{aligned} \quad (5.7)$$

where we use the inequalities $\ln x \leq x - 1$ and $\exp(-x) - 1 + x \leq x^2/2$ for $x \geq 0$. Now, we bound the second term in (5.6). By invoking Jensen's inequality we get

$$-\frac{1}{\delta^t} \ln \mathbb{E}_{a^t \sim \mathbf{p}'(t)} \left[\mathbb{E}_{j \sim \mathbf{p}^t} [\exp(-\delta^t \hat{l}_j^t)] \right] \leq -\frac{1}{\delta^t} \mathbb{E}_{a^t \sim \mathbf{p}'(t)} \left[\ln \mathbb{E}_{j \sim \mathbf{p}^t} [\exp(-\delta^t \hat{l}_j^t)] \right]. \quad (5.8)$$

5.5. Technical Contributions for the Adversarial Setting

Let $\hat{L}_i^t = \sum_{\tau=1}^t \hat{l}_i^\tau$ and let $\psi_t(\delta) = \frac{1}{\delta} \ln \left[\frac{1}{K} \sum_{i=1}^K \exp(-\delta \hat{L}_i^t) \right]$, (5.8) becomes

$$\begin{aligned} -\frac{1}{\delta^t} \mathbb{E}_{a^t \sim \mathbf{p}'(t)} \left[\ln \mathbb{E}_{j \sim \mathbf{p}^t} [\exp(-\delta^t \hat{l}_j^t)] \right] &= -\frac{1}{\delta^t} \mathbb{E}_{a^t \sim \mathbf{p}'(t)} \left[\ln \left(\frac{\sum_{i=1}^K \exp(-\delta^t \hat{L}_i^t)}{\sum_{i=1}^K \exp(-\delta^t \hat{L}_i^{t-1})} \right) \right] \\ &= \mathbb{E}_{a^t \sim \mathbf{p}'(t)} [\psi_{t-1}(\delta^t) - \psi_t(\delta^t)]. \end{aligned} \quad (5.9)$$

By summing up the terms in (5.7) and (5.9) over all t and plugging them in (5.6) we obtain

$$\sum_{t=1}^T \bar{l}^t \leq \sum_{t=1}^T \frac{\delta^t}{2} \mathbb{E}[(\hat{l}_j^t)^2] + \mathbb{E}_{a^t \sim \mathbf{p}'(t)} \left[\sum_{t=1}^T \psi_{t-1}(\delta^t) - \psi_t(\delta^t) \right]. \quad (5.10)$$

The function $\psi_t(\delta)$ is an increasing function in δ , thus $\psi_t(\delta^{t+1}) - \psi_t(\delta^t) < 0$, because $\delta^{t+1} < \delta^t$. Therefore,

$$\begin{aligned} \sum_{t=1}^T [\psi_{t-1}(\delta^t) - \psi_t(\delta^t)] &= \psi_0(\delta^1) - \psi_T(\delta^T) + \sum_{t=1}^{T-1} [\psi_t(\delta^{t+1}) - \psi_t(\delta^t)] \\ &\leq \psi_0(\delta^1) - \psi_T(\delta^T). \end{aligned} \quad (5.11)$$

According to the definition of $\psi_t(\delta)$, we have $\psi_0(\delta^1) = 0$ and we can simply bound $-\psi_T(\delta^T)$ as

$$-\psi_T(\delta^T) \leq \frac{\ln K}{\delta^T} + \hat{L}_i^T,$$

which holds for all $i \in [K]$. Combining the inequality above with (5.10), results in

$$\sum_{t=1}^T \bar{l}^t \leq \sum_{t=1}^T \frac{\delta^t}{2} \mathbb{E}[(\hat{l}_j^t)^2] + \frac{\ln K}{\delta^T} + \sum_{t=1}^T \mathbb{E}_{a^t \sim \mathbf{p}'(t)} [\hat{l}_i^t]. \quad (5.12)$$

We note that given \mathbf{p}^t and $\mathbf{p}'(t)$ at time t , the following expectations hold

$$\mathbb{E}_{a^t \sim \mathbf{p}'(t)} [\hat{l}_j^t] = l_j^t, \quad (5.13a)$$

$$\mathbb{E}_{j \sim \mathbf{p}^t} \left[\mathbb{E}_{a^t \sim \mathbf{p}'(t)} [(\hat{l}_j^t)^2] \right] = \sum_{j=1}^K \frac{p_j^t}{p'_j(t)} (l_j^t)^2 \leq \sum_{j=1}^K \frac{p_j^t}{p'_j(t)}, \quad (5.13b)$$

where in the last inequality above we use $0 \leq l_j^t \leq 1$. Computing the expectations in (5.12) yields

$$\sum_{t=1}^T \bar{l}^t \leq \sum_{t=1}^T \frac{\delta^t}{2} \sum_{j=1}^K \frac{p_j^t}{p'_j(t)} + \frac{\ln K}{\delta^T} + \sum_{t=1}^T l_i^t. \quad (5.14)$$

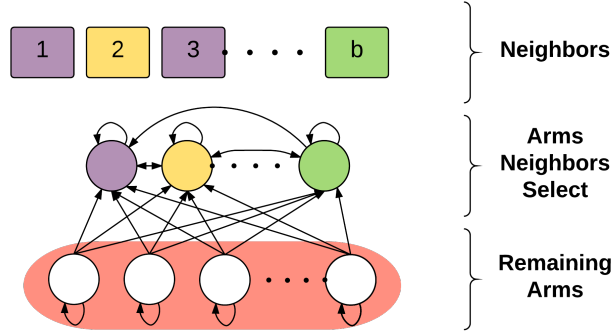


Figure 5.2 – The arm-network has an edge from arm u to arm v if, having selected arm u , we observe the reward of arm v . Arms selected by neighbors in the social networks form a clique, and the remaining arms have self loops and edges to all selected arms.

Since (5.14) holds for all i , we can upper bound regret as

$$\bar{R}_{\text{EXP N}} \leq \frac{\ln K}{\delta^T} + \sum_{t=1}^T \frac{\delta^t}{2} \sum_{j=1}^K \frac{p_j^t}{p_j^t(t)}. \quad (5.15)$$

Now, attaining a good bound on the regret boils down to attaining a good bound on $\sum_{j=1}^K \frac{p_j^t}{p_j^t(t)}$. Towards this, we need a technical lemma that, in effect, allows to bound the amount of information received from neighbors.

Lemma 5.3. $\sum_{j=1}^K \frac{p_j}{1-(1-p_j)(1-q_j^1)\cdots(1-q_j^b)} \leq \sum_{j=1}^K \frac{p_j}{p_j+q_j^1+q_j^2+\cdots+q_j^b} + 1.$

The proof is presented in Appendix 5.A. Using this lemma and combining all of the above, we get

$$\bar{R}_{\text{EXP N}} \leq \frac{\ln K}{\delta^T} + \frac{1}{2} \sum_{t=1}^T \delta^t (1 + \gamma^t). \quad (5.16)$$

Recall that $\delta^t = \sqrt{\frac{\ln K}{\sum_{\tau=1}^t (1+\gamma^\tau)}}$, then use Lemma 3.5 of [Auer et al., 2002c] (for completeness, the lemma is presented in Appendix 5.A) to conclude the proof. \square

5.5.2 Comparison to Alternate Approaches

Instead of developing a new algorithm, we could have attempted to leverage an existing one. The most natural one to try is from the arm-network setting which is as follows: there is a single individual and the bandit’s arms form an arm-network which can change over time. An edge from arm u to arm v means that by choosing arm u we observe the reward of arm v . Thus we could, in retrospect at each time step, recreate an arm-

5.5. Technical Contributions for the Adversarial Setting

network (see Figure 5.2) and apply an arm-network algorithm. We consider EXP3G [Alon et al., 2015], which is the state-of-the-art solution for such problems, and performed best amongst arm-network algorithms in our empirical simulations. However, we prove that our algorithm is at least as good.

Proposition 5.4. $\bar{R}_{\text{EXP3G}}^T = O(\bar{R}_{\text{EXP3G}}^T)$.

Proof of Proposition 5.4. First, let us introduce some notations regarding the arm-network setting. The players choose arms according to their algorithm, and after selecting and revealing the rewards. Each player i individually can construct an arm-network. We denote the arms selected by neighbors of player i by $\mathcal{A}[N(i)]$, thus in arm-network there is an edge from all arms (vertices) to $\mathcal{A}[N(i)]$. In addition, if we denote the cardinality of set $\mathcal{A}[N(i)]$ by C_i , then the maximum independence number ν of the underlying graph is $K + 1 - C_i$.

We show that our algorithm's regret is at most that of EXP3G, the state-of-the-art algorithm for the arm-network setting. From Theorem 5.2, we know that the regret of our algorithm is $\tilde{O}\left(\sqrt{T + \sum_{t=1}^T \gamma^t}\right)$ and from Alon et al. [2015] the regret of EXP3G is $\tilde{O}\left(\sqrt{\sum_{t=1}^T \nu^t}\right)$, where ν^t is the maximum independence number at time t .

Let us focus on one of the nodes. Let \mathcal{A} be the arms chosen by its neighbors, let C be its cardinality, and let $\mathbb{1}\{j \in \mathcal{A}\}$ be an indicator random variable that is 1 if and only if j is in \mathcal{A} . Lastly, let $\mathbb{1}\{j = a_\ell^t\}$ be an indicator random variable that is 1 if and only if a player ℓ (one of the neighbors) chooses arm j at round t . First, we lower bound ν using the indicator random variable defined above, then we show that in expectation this term is greater than γ^t .

Lemma 5.5. *The independence number ν^t is lower bounded as follows*

$$\nu^t \geq \sum_{j=1}^K \frac{p_j^t}{p_j^t + \sum_{\ell=1}^b \mathbb{1}\{j = a_\ell^t\}}. \quad (5.17)$$

Proof. First, we show the following

$$\nu^t \geq \sum_{j=1}^K \frac{p_j^t}{p_j^t + \mathbb{1}\{j \in \mathcal{A}\}}. \quad (5.18)$$

Decomposing the sum we have

$$\sum_{j=1}^K \frac{p_j^t}{p_j^t + \mathbb{1}\{j \in \mathcal{A}\}} = \sum_{j \in \mathcal{A}} \frac{p_j^t}{p_j^t + 1} + \sum_{j \notin \mathcal{A}} \frac{p_j^t}{p_j^t}, \quad (5.19)$$

and plugging the cardinality of \mathcal{A} we get

$$\sum_{j=1}^K \frac{p_j^t}{p_j^t + \mathbb{1}\{j \in \mathcal{A}\}} = \sum_{j \in \mathcal{A}} \frac{p_j^t}{p_j^t + 1} + K - C, \quad (5.20)$$

we know $\nu^t = 1 + K - C$ (this comes from the fact that arms in \mathcal{A} are connected to all arms, see Figure 5.2), knowing that $\sum_{j \in \mathcal{A}} \frac{p_j^t}{p_j^t + 1}$ is less than 1 completes the first part of lemma.

Second, we have $\mathbb{1}\{j \in \mathcal{A}\} \leq \sum_{\ell=1}^b \mathbb{1}\{j = a_\ell^t\}$, which yields the lemma. \square

In Lemma 5.5, the term $\mathbb{1}\{j = a_\ell^t\}$ can be seen as an unbiased estimator for $q_j^{\ell,t}$ (we denote it by $\hat{q}_j^{\ell,t}$). As a final step, we show the following lemma.

Lemma 5.6. *For a multinomial distribution \mathbf{q} and its unbiased estimator $\hat{q}_j^{\ell,t} = \mathbb{1}\{j = a_\ell^t\}$, we have*

$$\sqrt{\sum_{t=1}^T (1 + \gamma^t)} \leq \mathbb{E}_{\hat{\mathbf{q}}} \left[\sqrt{2 \sum_{t=1}^T \nu^t} \right]. \quad (5.21)$$

Because $\nu^t \geq 1$, we know $\sum_{t=1}^T \nu^t \geq T$, and we can conclude that $\sqrt{T + \sum_{t=1}^T \gamma^t} \leq \mathbb{E}_{\hat{\mathbf{q}}} \left[\sqrt{2 \sum_{t=1}^T \nu^t} \right]$.

From Lemma 5.5, we have

$$\mathbb{E}_{\hat{\mathbf{q}}} \left[\sqrt{\sum_{t=1}^T \nu^t} \right] \geq \mathbb{E}_{\hat{\mathbf{q}}} \left[\sqrt{\sum_{t=1}^T \sum_{j=1}^K \frac{p_j^t}{p_j^t + \sum_{\ell=1}^b \hat{q}_j^{\ell,t}}} \right]. \quad (5.22)$$

In the next step, we want to show

$$\mathbb{E}_{\hat{\mathbf{q}}} \left[\sqrt{\sum_{t=1}^T \sum_{j=1}^K \frac{p_j^t}{p_j^t + \sum_{\ell=1}^b \hat{q}_j^{\ell,t}}} \right] \geq \sqrt{\sum_{t=1}^T \gamma^t}.$$

Let $\phi(\hat{\mathbf{q}}) = \phi(\hat{q}^1(1), \hat{q}^2(1), \dots, \hat{q}^b(1), \hat{q}^1(2), \dots, \hat{q}^b(T)) = \sqrt{\sum_{t=1}^T \sum_{j=1}^K \frac{p_j^t}{p_j^t + \sum_{\ell=1}^b \hat{q}_j^{\ell,t}}}$. This function is convex (it is convex along every arbitrary line with positive entries, so it is convex), we can use Jensen's inequality to swap the order of expectation and ϕ to get the following

$$\mathbb{E}_{\hat{\mathbf{q}}}[\phi(\hat{\mathbf{q}})] \geq \phi(\mathbb{E}_{\mathbf{q}}[\hat{\mathbf{q}}]) = \phi(\mathbf{q}) = \sqrt{\sum_{t=1}^T \gamma^t}. \quad (5.23)$$

The first inequality is Jensen's inequality and the last equality comes from definition of γ^t . Combining Inequality (5.23) with (5.22) concludes the proof. \square

Moreover, as we will see in Section 5.6, the regret of EXPN is often drastically better empirically. Because EXP3G and other similar algorithms were developed for different settings in which it is not possible to make use of the probability distributions afforded to us by Assumption (3).

Proposition 5.7. *Let n_t be the size of the set of arms selected (arbitrarily) by all of the individual's neighbors at time t . Then, the regret $R_{\mathcal{A}}$ for any algorithm \mathcal{A} in our setting without Assumption (3) is $\bar{R}_{\mathcal{A}}^T = \Omega\left(\sqrt{T + \sum_{t=1}^T (K - n_t)}\right)$.*

The proof follows from Theorem 5 of [Alon et al., 2014]. Our EXPN algorithm is often able to beat this bound by leveraging Assumption (3). For example, this proposition implies that if we have a complete network on b vertices where $\log K \ll b \ll K$, then $\bar{R}_{\text{EXP3G}}^T = \Omega\left(\sqrt{(K-b)T}\right) = \Omega\left(\sqrt{KT}\right)$ while in our case $\bar{R}_{\text{EXPN}}^T = O\left(\sqrt{\frac{K}{b}T}\right)$ (see Corollary 5.9).

5.5.3 A Centralized Solution for the Network

Our model and algorithm are formulated for an individual because this allows us to draw the most general conclusions – bounding the individual's regret as a *function* of the neighbors' behavior. However, a surprising feature is that it can also be made into a centralized solution. In the general case, this requires assuming there is an external coordinator that can select a maximum-degree individual to lead and direct the rest on how to act as follows: Let v^* be the maximum degree node selected. The coordinator directs v^* to use the EXPN algorithm. The remaining nodes u are each assigned a neighbor v_u that lies on the shortest path between them and v^* , and are directed to copy the probability distribution that v_u used in the previous time step.

Theorem 5.8. *Using the above centralized algorithm, the regret of all individuals is at most*

$$\bar{R}^T = O\left(\kappa + \sqrt{\left(1 + \frac{K}{1 + b_{\max}}\right) T \ln K}\right), \quad (5.24)$$

where b_{\max} is the degree of v^* and κ is the diameter of the network.

The proof follows, with minor modifications, from the proof of Theorem 5.2; the main difference regards accounting for the delay (of at most κ time steps) for the farthest node from v^* to update their probability distribution. By replacing γ^t with $\frac{K}{1+b_{max}}$, this gives us the resulting regret bound. In the simple case of a complete network on N nodes, no coordinator is required, and we obtain the following corollary.

Corollary 5.9. *On a complete network with b nodes, if all nodes use the EXPN algorithm, then they attain $\bar{R}^T = O\left(\sqrt{\left(1 + \frac{K}{b}\right) T \ln K}\right)$, which is optimal (up to log factors) for any centralized solution.*

This again follows from the proof of Theorem 5.2 using the fact that the number of neighbors is $b - 1$ on a complete network, and that a centralized solution has average regret $\Omega\left(\sqrt{\left(1 + \frac{K}{b}\right) T}\right)$ as shown in [Amin et al., 2015].

5.6 Empirical Evaluation

5.6.1 Adversarial Setting: Experimental setup

Benchmarks. We compare our algorithm against the bandit algorithms developed for various settings with side-information, namely EXP3G [Alon et al., 2015], EXP.IX [Kocák et al., 2014] and BEXP [Amin et al., 2015]. The first two are designed for the arm-network setting as described in Section 5.5.2, while the latter is designed for the free-exploration setting described in Section 5.3. Recall that in free-exploration there are no neighbors; rather there is a budget B , and at each time step the individual can choose up to B arms to select. In order to attain a fair comparison, we assume we have budget $B = b + 1$ for BEXP, where b is the number of neighbors.

Experimental Setup. We consider a bandit with Bernoulli rewards that has a single *good* arm with mean 0.7, while the remaining arms have mean 0.5. This is similar to the worst-case (minimax) bandit; the difficulty arises from the fact that it is hard, in an information-theoretic sense, to distinguish the single good arm from the rest with few samples. Indeed the performance for our algorithm in comparison to our benchmarks is only improved for all other settings we attempted.

5.6.2 Adversarial Setting: Empirical Results

Performance in Networks. In addition to exploring the effect of the various algorithms on a single individual, we are able to consider various network topologies and consider the regret as a whole. Towards this goal, in the first set of simulations, all nodes in the specified networks use the same algorithm. We first compare the regret of

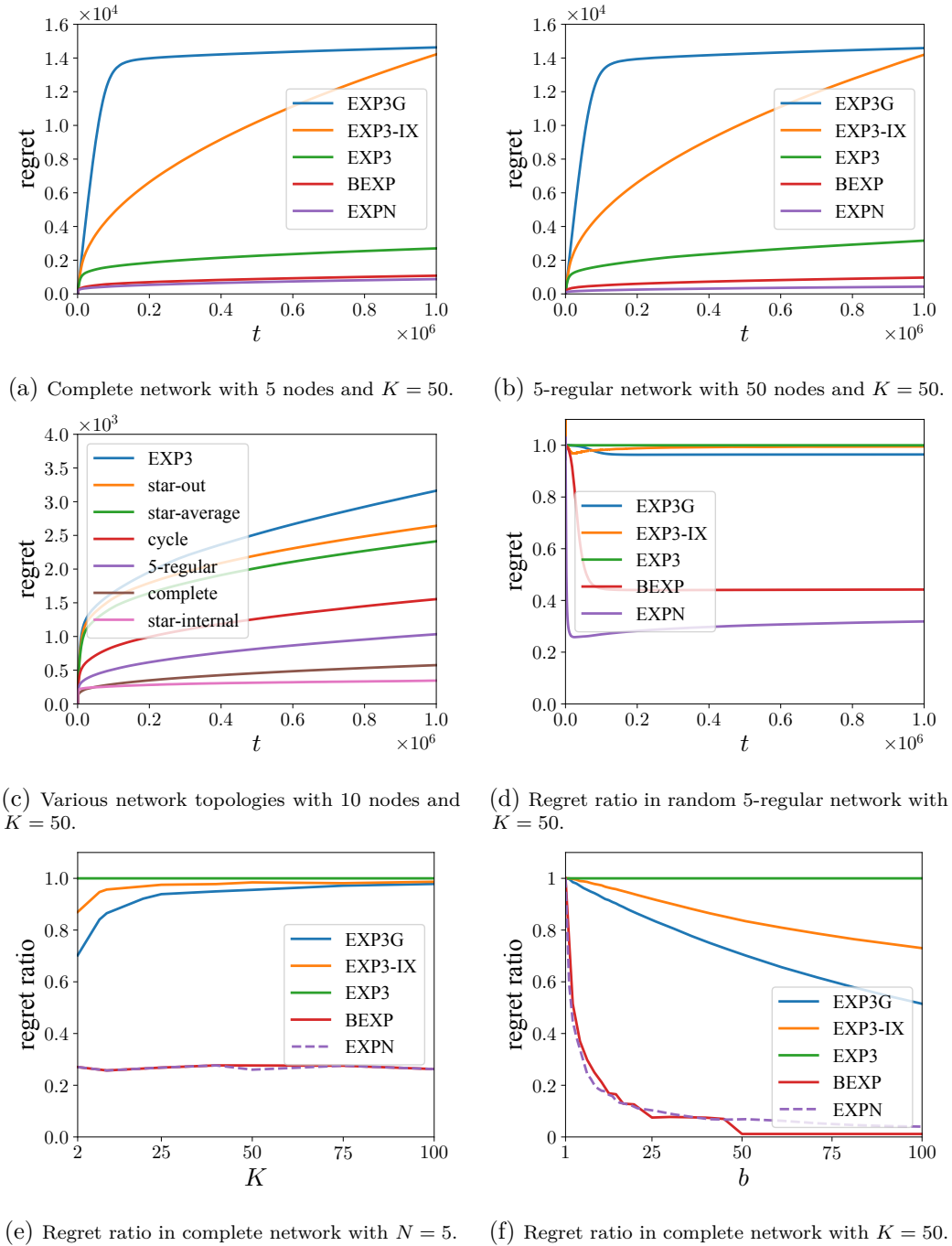


Figure 5.3 – Performance of our algorithm (EXPN) for the adversarial setting against benchmarks. Our algorithm significantly outperforms EXP3, indicating that the presence of neighbors indeed improves learning. It also significantly outperforms arm-network algorithms (EXP3G and EXP.IX) that could be applied in our setting. Surprisingly, its performance is as good as BEXP, which would require a single node to dictate the choices of her neighbors; hence, our distributed algorithm is performing as well as a centralized one. Figures (a)-(c) depict the regret. Figures (d)-(f) depict the regret ratio, i.e., the ratio between an algorithm’s regret with b neighbors over its regret with 0 neighbor (where b depends on the network structure addressed in the corresponding subfigure).

EXPN against the benchmarks in a complete network on 5 nodes (Figure 5.3a); even on such a small network the difference in regret is dramatic.⁵ EXPN significantly outperforms arm-network algorithms (EXP3G and EXP.IX), which empirically are initially worse than even EXP3. Asymptotically EXP3G eventually outperforms EXP3, although EXP.IX does not. Surprisingly, our algorithm performs as well as BEXP, which would be equivalent to identifying a single node as the leader and having them dictate the action of all other nodes. Hence, our distributed algorithm is as good as a centralized one. For comparison, we also consider a random 5-regular graph on 50 nodes (Figure 5.3b), and observe that the performance of all algorithms is roughly equivalent to the complete network on 5 nodes; i.e., the primary determining factor in the regret appears to be the number of neighbors rather than the topology of the network.

We also consider the regret of EXPN on various network topologies on 10 vertices: the complete network, a random 5-regular network, a cycle, and a star network (Figure 5.3c). When the number of neighbors differs in a topology, the regret of the nodes may differ; the star is the extreme example and we depict the minimum (for the center node), maximum (for one of the leaves) and average regret. As expected, the more neighbors one has, the better the regret is, with the internal node of star outperforming all. We also observe that there is an advantage to having neighbors that are not well-connected; despite a node in the complete network having the same degree as the center node of the star, the former has more regret. Because the nodes that are not well-connected receive less information, they must explore more – this is advantageous for their neighbors.

Performance of Individuals. Moving back to analyzing the performance for an individual, consider a setting where her neighbors all use the EXP3 algorithm. We measure the *regret ratio*, i.e., the ratio between the regret of bandit algorithm \mathcal{B} when the node has b neighbors divided by the regret of \mathcal{B} when the node has 0 neighbor. This allows us to better visualize the improvement in regret that each algorithm obtains as a function of the number of neighbors. We vary time T (Figure 5.3d), the number of arms K (Figure 5.3e) and the number of neighbors b (Figure 5.3f). We observe that, in all cases, our EXPN algorithm always matches or outperforms the benchmarks. The fact that the performance of our EXPN is comparable to that of BEXP is surprising, as we could not hope to do any better.

5.6.3 Stochastic Setting: Empirical Results

The setup for the empirical results in this section parallels that of Section 5.6.1. Recall that we make no assumption in our algorithm about our neighbors or how they play. We simply observe their actions and rewards. We let $\alpha = 2.5$ in the UCBN algorithm; the

⁵Indeed, on larger networks the differences are only more pronounced – we present the results on a small network in order to be able to visualize them adequately.

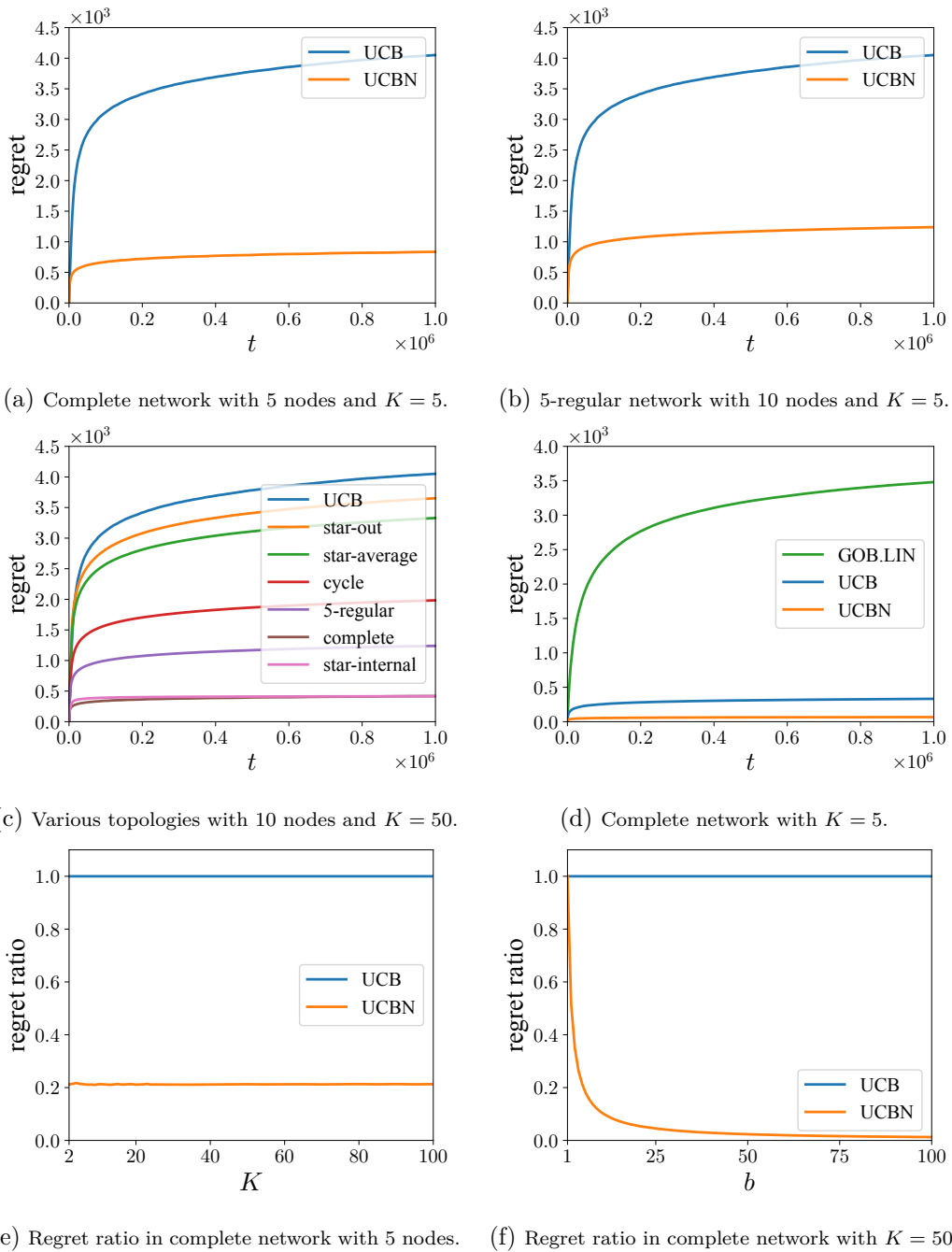


Figure 5.4 – Performance of our algorithm (UCBN) for the stochastic setting against benchmarks. Our algorithm significantly outperforms UCB, indicating that the presence of neighbors indeed improves learning. It also significantly outperforms GOB.LIN that could be applied in our setting (Figure 5.4d). Figures (a)-(d) depict the regret. Figures (e)-(f) depict the regret ratio, i.e., the ratio between an algorithm’s regret with b neighbors over its regret with 0 neighbor (where b depends on the network structure addressed in the corresponding subfigure).

performance could be improved by optimizing α . We first observe that more neighbors leads to less regret (Figure 5.4f).

We then consider the regret of UCBN on various network topologies on 10 vertices: the complete network, a random 5-regular network, and a star network (Figure 5.4c). Similar to the previous experiments, for networks in which all vertices have the same number of neighbors (all but the star network), all agents attain the same regret and hence we report the average regret. However, this is not the case if the number of neighbors differs; the star is the extreme example and we depict the minimum (for the center node), maximum (for one of the leaves) and average regret. As expected, the more neighbors one has, the better the regret is, with the complete network and center of star outperforming all. We also observe that there is an advantage to having neighbors that are not well-connected; despite a node in the complete network having the same degree as the center node of the star, the former has slightly more regret. The reason is that a neighbor with lower degree attains less information from neighbors and explore the suboptimal arms more which itself is in the favor of its neighbors (here the center of star).

We then consider the *regret ratio*, i.e., the ratio between the regret of bandit algorithm \mathcal{B} when the agent has b neighbors divided by the regret of \mathcal{B} when the agent has 0 neighbor. This allows us to better visualize the improvement in regret that each algorithm obtains as a function of the number of neighbors. We vary the number of arms K (Figure 5.4e) and the number of neighbors b (Figure 5.4f). We observe that, in all cases, our algorithm UCBN attains the theoretical regret ratio, i.e, in the complete graph when all agents use UCBN the regret ratio $\mathcal{B}_b \rightarrow 1/b$ as $T \rightarrow \infty$.

Finally, we compare our algorithm to the one proposed in [Cesa-Bianchi et al., 2013] (GOB.LIN); see Figure 5.4d. Although this algorithm is centralized and developed for a different setting (namely, for linear contextual bandits), it can be adapted to our setting by assuming that individuals are cooperative instead of selfish. Despite the centralized nature of GOB.LIN, our algorithm outperforms its regret.

5.7 Summary

In this chapter, we consider a model for social learning that puts the problem in the bandit framework. This model allows the problem to be analyzed both in the stochastic and adversarial bandit settings, and we provide algorithms for both cases. The regret of our algorithms interpolates between the regret of the traditional bandit setting (e.g., when an individual has no neighbors) and the regret of the full information setting (e.g., when the number of neighbors goes to infinity). We show, both theoretically and empirically, that we outperform state-of-the-art bandit algorithms that one could also apply to this setting, and illustrate how our approach could also lead to centralized algorithms of interest.

With respect to improvements to the social learning model, relaxing Assumption (3) would be ideal. As we have shown (see Proposition 5.7), removing it entirely results in strictly weaker regret bounds. Would an alternate relaxed assumption suffice? Lastly, it remains to formally study the effect of arbitrary network topologies on the regret, both for the individual (based on their position in the network) and on average.

Appendix

Appendix

5.A Adversarial Bandits

Lemma 5.10. For $p^i \in [0, 1]$ we want to show that $\prod_{i=1}^b (1 - p^i) \leq \frac{1}{1 + \sum_{i=1}^b p^i}$.

Proof. The following formula holds for all positive p_i ,

$$\prod_{i=1}^b (1 + p^i) \geq 1 + \sum_{i=1}^b p^i, \quad (5.25)$$

which implies

$$\frac{1}{\prod_{i=1}^b (1 + p^i)} \leq \frac{1}{1 + \sum_{i=1}^b p^i}. \quad (5.26)$$

As we have

$$\begin{aligned} \prod_{i=1}^b (1 - p^i) \cdot \prod_{i=1}^b (1 + p^i) \\ = \prod_{i=1}^b (1 - (p^i)^2) \leq 1, \end{aligned} \quad (5.27)$$

We can conclude

$$\prod_{i=1}^b (1 - p^i) \leq \frac{1}{\prod_{i=1}^b (1 + p^i)}. \quad (5.28)$$

From (5.26) and (5.28) we get

$$\begin{aligned} \prod_{i=1}^b (1 - p^i) &\leq \frac{1}{\prod_{i=1}^b (1 + p^i)} \\ &\leq \frac{1}{1 + \sum_{i=1}^b p^i}. \end{aligned} \tag{5.29}$$

□

Proof of Lemma 5.3. From Lemma 5.10 we have

$$(1 - p_j)(1 - q_j^1) \cdots (1 - q_j^b) \leq \frac{1}{1 + p_j + q_j^1 + q_j^2 + \cdots + q_j^b}, \tag{5.30}$$

substituting this bound in the statement of lemma yields

$$\begin{aligned} \sum_{j=1}^K \frac{p_j}{1 - (1 - p_j)(1 - q_j^1) \cdots (1 - q_j^b)} &\leq \sum_{j=1}^K \frac{p_j}{1 - \frac{1}{1 + p_j + q_j^1 + q_j^2 + \cdots + q_j^b}} \\ &\leq \sum_{j=1}^K \frac{p_j}{p_j + q_j^1 + q_j^2 + \cdots + q_j^b} + 1. \end{aligned} \tag{5.31}$$

□

Lemma 5.11 (Lemma 3.5 in [Auer et al., 2002c]). *Let c_1, c_2, \dots, c_T and b be non-negative real numbers. Then*

$$\sum_{t=1}^T \frac{c_t}{\sqrt{b + \sum_{i=1}^t c_i}} \leq 2 \left(\sqrt{b + \sum_{t=1}^T c_t} - \sqrt{b} \right), \tag{5.32}$$

where $0/\sqrt{0} = 0$

5.B Stochastic Bandits

In this section, we first formally state and prove the results for the stochastic setting. In what follows let $\mu_i = \mathbb{E}_{r \sim \mathcal{D}}[r_i]$ and let $\mu_{i^*} = \mu^*$ be the highest expected reward where i^* is the arm with highest expected reward. Also, let $\bar{\mu}$ be the empirical mean of observed rewards. Let us first recall our result:

Theorem 5.1. *Let m_i^t be the number times arm i has been selected by one of her neighbors by time t . Then, the regret of UCBN for any $\alpha > 2$ is*

$$\bar{R}_{\text{UCBN}}^T \leq \sum_{i, \Delta_i > 0} \left(\max \left\{ \max_{t=1, \dots, T} \left\{ \frac{2\alpha \ln t}{\Delta_i} - m_i^t \Delta_i \right\}, 0 \right\} + \frac{\alpha}{\alpha - 2} \right), \tag{5.33}$$

where Δ_i is the difference between μ_{i^*} and μ_i .

Corollary 5.12. *On a complete graph with b nodes, if all agents use UCBN then under the same conditions as in Theorem 5.1, the regret of an agent is*

$$\bar{R}_{\text{UCBN}}^T \leq \sum_{i, \Delta_i > 0} \left(\frac{2\alpha \ln T}{b\Delta_i} + \frac{\alpha}{\alpha - 2} \right) \in O\left(\frac{K \ln T}{b}\right). \quad (5.34)$$

The following lower bound yields same behavior for getting free observation from neighbors.

Theorem 5.13. *Let m_i^t be the number of times a strategy selects an arm i in T rounds. Consider a strategy that satisfies $\mathbb{E}[m_i^t] = o(T^a)$, any arm i with $\Delta_i > 0$, and any $a > 0$. Let m_i^t be the number of times arm i selected (arbitrarily) by all of the agent's neighbors up to time t , then, for any set of Bernoulli reward distributions the following inequality holds*

$$\lim_{T \rightarrow +\infty} \inf \frac{R}{\ln T} \geq \sum_{i, \Delta_i > 0} \frac{1}{2\Delta_i} - \lim_{T \rightarrow +\infty} \inf \frac{\sum_{i, \Delta_i > 0} m_i^T \Delta_i}{\ln T}. \quad (5.35)$$

Our proofs parallel, with additional bookkeeping, the proofs for the original UCB results (see, e.g., [Bubeck and Cesa-Bianchi, 20120] for a template). Note that the results for stochastic bandits hold when the reward distributions satisfy the following standard conditions.

Definition (Conditions on \mathcal{D}). Every reward distribution \mathcal{D} satisfies Hoeffding's lemma, i.e., there exists a convex function ψ on the reals such that, for all $\lambda \geq 0$, we have $\ln \left[\mathbb{E} \left[e^{\lambda|r - \mathbb{E}[r]|} \right] \right] \leq \psi(\lambda)$ where $r \sim \mathcal{D}$.

For example, when $r \in [0, 1]$, one can take $\psi(\lambda) = \frac{\lambda^2}{8}$; indeed the results in the main body of the work take this ψ . The results can be easily generalized for other ψ in the usual manner.

We first state a lemma and its proof from [Bubeck and Cesa-Bianchi, 20120] that will be of assistance in the proof of Theorem 5.1. Recall that a^t is the arm the agent selects at time t .

Lemma 5.14. *If the selected arm $a^t = i$, at least one of the three following inequalities is true:*

$$\bar{\mu}_{i^*, n_{i^*}^{t-1}} + \sqrt{\frac{\alpha \ln t}{2n_{i^*}^{t-1}}} \leq \mu^*, \quad (5.36a)$$

$$\bar{\mu}_{i, n_i^{t-1}} > \mu_i + \sqrt{\frac{\alpha \ln t}{2n_i^{t-1}}}, \quad (5.36b)$$

$$n_i^{t-1} < \frac{2\alpha \ln t}{\Delta_i^2}, \quad (5.36c)$$

where i^* is the arm with the highest expected reward.

Proof. The contrapositive is proved. Assume that $a^t = i$ and that none of the inequalities (5.36a), (5.36b) or (5.36c) are true.

$$\bar{\mu}_{i^*, n_{i^*}^{t-1}} + \sqrt{\frac{\alpha \ln t}{2n_{i^*}^{t-1}}} > \mu^*, \quad (5.37a)$$

$$\bar{\mu}_{i, n_i^{t-1}} < \mu_i + \sqrt{\frac{\alpha \ln t}{2n_i^{t-1}}}, \quad (5.37b)$$

$$n_i^{t-1} > \frac{2\alpha \ln t}{\Delta_i^2}. \quad (5.37c)$$

By plugging $\mu^* = \mu_i + \Delta_i$ in (5.37a) we obtain

$$\bar{\mu}_{i^*, n_{i^*}^{t-1}} + \sqrt{\frac{\alpha \ln t}{2n_{i^*}^{t-1}}} > \mu_i + \Delta_i. \quad (5.38)$$

From (5.37c) we have

$$\mu_i + \Delta_i > \mu_i + \sqrt{\frac{2\alpha \ln t}{n_i^{t-1}}} \quad (5.39)$$

and plugging (5.39) in (5.38) yields

$$\bar{\mu}_{i^*, n_{i^*}^{t-1}} + \sqrt{\frac{\alpha \ln t}{2n_{i^*}^{t-1}}} > \bar{\mu}_{i, n_i^{t-1}} + \sqrt{\frac{\alpha \ln t}{2n_i^{t-1}}}. \quad (5.40)$$

This implies $a^t \neq i$, which negates the assumption $a^t = i$. \square

Proof of Theorem 5.1 . With some abuse of notations, let $n_i^{1:t}$ be number of times the agent selects the arm i . Let

$$n_i^t = n_i^{1,t} + m_i^t, \quad (5.41)$$

where m_i^t is the number of times her neighbors select arm i . Using Lemma 5.14, we will first find an upper bound for m_i^t for a suboptimal arm i .

Lemma 5.14 states that at least one of the three inequalities (5.36a), (5.36b) and (5.36c) must be true. If (5.36c) holds, then from (5.41) we obtain

$$n_i^{1,t} \leq \frac{2\alpha}{\Delta_i^2} \ln t - m_i^t. \quad (5.42)$$

Let U be the maximum of right hand side of (5.42) for $t = 1, \dots, T$:

$$U = \max \left\{ \max_{t=1, \dots, T} \left\{ \frac{2\alpha}{\Delta_i^2} \ln t - m_i^t \right\} \Delta_i, 0 \right\}, \quad (5.43)$$

as a result if the (5.36c) holds for some instance τ , then $n_i^{1,\tau}$ is bounded by U , i.e.,

$$n_i^{1,\tau} \leq U. \quad (5.44)$$

For bounding the regret we find an upper bound on the number of times we select a suboptimal arm i :

$$\mathbb{E}[m_i^t] = \mathbb{E} \left[\sum_{t=1}^T \mathbb{1}_{a^t=i} \right] = \mathbb{E} \left[\sum_{t=1}^U \mathbb{1}_{a^t=i} \right] + \mathbb{E} \left[\sum_{t=U+1}^T \mathbb{1}_{a^t=i} \right].$$

Since $\mathbb{E} \left[\sum_{t=1}^U \mathbb{1}_{a^t=i} \right] \leq U$, we can deduce

$$\mathbb{E}[n_i^{1,t}] \leq U + \mathbb{E} \left[\sum_{t=U+1}^T \mathbb{1}_{a^t=i} \right],$$

as we saw in (5.14), $\mathbb{1}_{\{a^t=i\}} = 1$ requires that at least one of the three (5.36a), (5.36b) and (5.36c) is true. Assume the last time that (5.36c) is true is at time τ , hence

$$n_i^{1,\tau} \leq \frac{2\alpha}{\Delta_i^2} \ln \tau - m_i^\tau,$$

and since τ is the last time that (5.36c) holds, we can upper bound $\mathbb{E}[m_i^t]$ by

$$\mathbb{E}[n_i^{1,\tau}] \leq \frac{2\alpha}{\Delta_i^2} \ln \tau - m_i^\tau + \mathbb{E} \left[\sum_{t=\tau+1}^T \mathbb{1}_{\{(5.36a) \text{ or } (5.36b) \text{ is true and } (5.36c) \text{ is false}\}} \right]. \quad (5.45)$$

According to the definition of U in (5.43) and (5.45) we have

$$\begin{aligned} \mathbb{E}[n_i^{1,\tau}] &\leq U + \mathbb{E} \left[\sum_{t=U+1}^T \mathbb{1}_{\{(5.36a) \text{ or } (5.36b) \text{ is true and } (5.36c) \text{ is false}\}} \right] \\ &\leq U + \sum_{t=U+1}^T \mathbb{P}[(5.36a) \text{ is true}] + \mathbb{P}[(5.36b) \text{ is true}]. \end{aligned}$$

It suffices to bound the probability (5.36a) and (5.36b):

$$\mathbb{P}[(5.36a) \text{ is true}] = \sum_{n_i^t} \mathbb{P}[(5.36a) \text{ is true} | n_i^t] \cdot \mathbb{P}[n_i^t]. \quad (5.46)$$

Let us recall (1.5) from Section 1.2.3 that

$$\mu_i \leq U_i(t) = \bar{\mu}_i^t + \sqrt{\frac{\alpha \ln(t)}{2n_i^t}}$$

holds with probability at least $1 - t^{-\alpha}$. Thus

$$\mathbb{P}[(5.36a) \text{ is true} | n_i^t] \leq \frac{1}{t^\alpha}. \quad (5.47)$$

Plugging (5.47) in (5.46) yields that

$$\mathbb{P}[(5.36a) \text{ is true}] \leq \frac{1}{t^\alpha} \sum_{n_i^t} \mathbb{P}[n_i^t] = \frac{1}{t^\alpha}. \quad (5.48)$$

Then we take integral of $\frac{1}{t^\alpha}$ for t from 1 to T , which is smaller than $\frac{\alpha}{2(\alpha-2)}$. The same upper bound holds for (5.36b).

Thus, the regret is

$$\bar{R}_{\text{UCBN}}^T \leq \sum_{i, \Delta_i > 0} \left(\max \left\{ \max_{t=1, \dots, T} \left\{ \frac{2\alpha}{\Delta_i^2} \ln t - m_i^t \right\} \Delta_i, 0 \right\} + \frac{\alpha}{\alpha - 2} \right)$$

as desired. \square

5.B.1 UCBN on Complete Graphs

In this section, we analyze the regret in a complete graph when all agents use UCBN. The following curious lemma will assist in the proof.

Lemma 5.15. *Given a complete graph of agents, if all agents use UCBN with a deterministic common tie breaking scheme, then in every time step all agents select the same action.*

Proof. Since the graph is complete, all agents see the rewards of other agents at every time step; hence the sample means $\bar{r}_i(t)$ and number of samples n_i^t at time t are the same for all agents. Furthermore, every agent selects an arm that maximizes $\bar{\mu}_i^t + \sqrt{\frac{\alpha \ln(t)}{2n_i^t}}$. Therefore, the arm selected at time t will be same for all agents. \square

Proof of Corollary 5.12. Let

$$U = \left\lceil \frac{2\alpha \ln T}{b \cdot \Delta_i^2} \right\rceil.$$

We bound the number of times action i other than the best arm is selected. Following the proof of Theorem 5.1,

$$\mathbb{E}[m_i^t] \leq U + \sum_{t=U+1}^T \mathbb{P}[(5.36a) \text{ is true}] + \mathbb{P}[(5.36b) \text{ is true}].$$

The upper bound of the probabilities (5.36a) and (5.36b) are same as before. Hence, the regret bound is

$$\bar{R}_{\text{UCBN}}^T \leq \sum_{i, \Delta_i > 0} \left(\frac{2\alpha}{b \cdot \Delta_i} \ln T + \frac{\alpha}{\alpha - 2} \right).$$

\square

5.B.2 Lower Bound

The lower bound for UCB (see [Lai and Robbins, 1985]) is

$$\lim_{T \rightarrow +\infty} \inf \frac{R}{\ln T} \geq \sum_{i, \Delta_i > 0} \frac{\Delta_i}{KL(\mu_i, \mu^*)}.$$

Our proof follows the same template.

Proof of Theorem 5.13. As in [Lai and Robbins, 1985], we assume the rewards are drawn from a Bernoulli distribution. From their proof it follows that the expected number of times that a suboptimal arm must be selected in order to distinguish between best arm and other arms is at least

$$\mathbb{E}[n_i^{1,T}] + m_i^T \geq (1 + o(1)) \frac{1 - \varepsilon}{1 + \varepsilon} \frac{\ln T}{KL(\mu_i, \mu^*)}.$$

Chapter 5. Learn from Thy Neighbor

where the second term is the information coming from the neighbors (that is, the number of times neighbors selected arm i up to time t), μ^* is the mean of the best arm, and $KL(p, q)$ is the Kullback-Leibler divergence between a Bernoulli variable with parameter p and a Bernoulli variable with parameter q , defined to be

$$KL(p, q) \stackrel{\text{def}}{=} p \ln \left(\frac{p}{q} \right) + (1 - p) \ln \left(\frac{1 - p}{1 - q} \right).$$

As the number of rounds increases, ε can be taken to be smaller. As T goes to infinity, ε can be taken zero; as a result we can write the following lower bound for the regret

$$\lim_{T \rightarrow +\infty} \inf \frac{R + \sum_{i, \Delta_i > 0} m_i^T \Delta_i}{\ln T} \geq \sum_{i, \Delta_i > 0} \frac{\Delta_i}{KL(\mu_i, \mu^*)}.$$

□

6 Conclusion

Many modern technologies make a sequence of choices in the presence of uncertainty. Multi-armed bandits (MAB) is one of the simplest, yet one of the most powerful settings for optimizing a sequence of choices within an exploration-exploitation framework.

In this dissertation, we studied several important problems from three different perspectives: (1) how MAB framework can improve two important stochastic optimization algorithms in machine learning, (2) how MAB algorithms can negatively affect humans when deployed in recommendation systems, and (3) how human interactions can be studied in an MAB setting.

In Chapters 2 and 3, we focused on reducing the training time of machine-learning algorithms by improving two of the most well-known optimization algorithms: stochastic-gradient descent (SGD) and stochastic-coordinate descent (CD). Optimization algorithms are at the core of machine-learning problems, and improving them is therefore of great interest.

In Chapter 2, we studied accelerating SGD by selecting datapoints from a non-uniform distribution. SGD optimizes a cost function by drawing one of the datapoints uniformly at random and by using the gradient computed only at this datapoint. The main issue of SGD is its high variance. This variance can be reduced by using a non-uniform distribution, where the non-uniform distribution should weigh more the datapoints that are wrongly classified. The challenge lies in finding the appropriate non-uniform sampling distribution with a lightweight mechanism that preserves the computational tractability of SGD. We used MAB framework to design scalable algorithms and we prove that our algorithm asymptotically approximates the minimal variance within a constant factor. We showed that using this datapoint-selection technique results in a significant reduction of the convergence time and the variance of several stochastic optimization algorithms such as SGD and SAGA.

Chapter 6. Conclusion

In Chapter 3, we shifted our attention from accelerating SGD to accelerating CD by selecting coordinates from a non-uniform distribution. CD optimizes a cost function by selecting one of the coordinates uniformly at random and updating this coordinate. Updating the model based on different coordinates yields various improvements. In Chapter 3, we designed an MAB algorithm that selects the coordinate that most improves the model. We showed that this approach significantly reduces the training time.

In Chapter 4, we initiated a formal study of combating polarization in personalization algorithms that learn user behavior. We alleviated polarization by introducing a set of constraints, which ensure that a diverse set of items are displayed to a user. We showed how an existing bandit algorithm can be modified to satisfy these constraints and to rapidly reach the theoretical optimum. For laminar constraints, our experiments on large datasets show that the modified algorithm rapidly converges to the theoretical optimum, and that this optimum solution of constrained setting is close to the optimum solution of an unconstrained setting.

In Chapter 5, we cast a model for social learning as a problem in the bandit framework. Our proposed model enables the problem to be analyzed both in the stochastic and adversarial bandit settings, and we provided algorithms for both cases. Depending on the information provided by neighbors, the regret of our algorithms interpolates between the regret of the classic bandit setting (e.g., when an individual has no neighbors) and the regret of the full information setting (e.g., when the number of neighbors goes to infinity). We showed, both theoretically and empirically, that the proposed algorithms outperform state-of-the-art bandit algorithms that could also be applied to this setting.

Future Research Directions

The approach that we adopted in most of this thesis consists of breaking down the challenges into simple decision-making problems, which we then studied in an MAB framework. However, there are interesting open directions for future work, we discuss here some that we believe are among the most important.

First, we would like to use the underlying structure of the decision-making problems to design better MAB algorithms. For example, in Chapters 2 and 3, we cast the datapoint-selection and coordinate-selection part of SGD and CD as an adversarial and stochastic bandit problem, respectively. This means that no additional assumption is used in the developed algorithms. However, optimization problems have additional structures such as smoothness or strong convexity. Exploiting these additional structures can improve the MAB algorithms for datapoint-selection and coordinate-selection. It could also be of interest to extend the work to other stochastic optimization methods, both by providing theoretical guarantees and by observing their performance in practice.

With regard to combating polarization in Chapter 4, our algorithms require the groups' labels in advance. These labels would either need to be inferred from the data, which could bring with it additional bias associated with this learning algorithm, or would need to be self-reported, which can lead to adversarial manipulation. Additionally, designing the right constraints that can better prevent polarization, not just of the items in the feed but of the beliefs and opinions of those viewing them, is an important direction for future work.

With respect to improvements to the social-learning model in Chapter 5, relaxing the assumption on the algorithms of neighbors would be ideal. We show that removing this assumption would result in strictly weaker regret bounds. Therefore, additional assumptions on the behavior of neighbors are still needed. Lastly, the regret of each individual player is analyzed locally; further analysis is needed to take the topology of the social network into account.

Bibliography

- Allsides media bias ratings. <https://www.allsides.com/media-bias/media-bias-ratings>. [Cited on page 106]
- Webhose news api. <https://webhose.io/data-feeds/news-api/>. [Cited on page 106]
- Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances In Neural Information Processing Systems*, 2011. [Cited on pages 97, 99, 103, 111, 113, and 114]
- E. Accinelli and J. Sánchez-Carrera. Corruption driven by imitative behavior. *Econ. Letters*, 117:84–87, 2012. [Cited on pages 14 and 120]
- G. Adomavicius and Y. Kwon. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Transactions on Knowledge and Data Engineering*, 24(5):896–911, 2012. [Cited on page 97]
- G. Adomavicius, J. Bockstedt, C. Shawn, and J. Zhang. De-biasing user preference ratings in recommender systems. In *Joint Workshop on Interfaces and Human Decision Making for Recommender Systems, Co-located with ACM Conference on Recommender Systems*, 2014. [Cited on page 97]
- S. Agrawal and N. Devanur. Linear contextual bandits with knapsacks. In *Advances In Neural Information Processing Systems*, pages 3450–3458, 2016. [Cited on page 98]
- R. Alghamdi and K. Alfalqi. A survey of topic modeling in text mining. *Int. J. Adv. Comput. Sci. Appl.(IJACSA)*, 6(1), 2015. [Cited on page 92]
- Z. Allen-Zhu and Y. Yuan. Improved svrg for non-strongly-convex or sum-of-non-convex objectives. In *Proceedings of International Conference on Machine Learning*, pages 1080–1089, 2016. [Cited on pages 23 and 44]

Bibliography

- Z. Allen-Zhu, Z. Qu, P. Richtárik, and Y. Yuan. Even faster accelerated coordinate descent using non-uniform sampling. In *International Conference on Machine Learning*, pages 1110–1119, 2016. [Cited on pages 65 and 80]
- N. Alon, N. Cesa-Bianchi, C. Gentile, and Y. Mansour. From bandits to experts: A tale of domination and independence. In *Proceedings of the 26th Conference on Advances in Neural Information Processing Systems (NIPS)*, 2013. [Cited on pages 7 and 124]
- N. Alon, N. Cesa-Bianchi, C. Gentile, S. Mannor, and Y. Mansour. Nonstochastic multi-armed bandits with graph-structured feedback. *Arxiv*, 2014. [Cited on page 133]
- N. Alon, N. Cesa-Bianchi, O. Dekel, and T. Koren. Online learning with feedback graphs: Beyond bandits. In *Conference on Learning Theory (COLT)*, 2015. [Cited on pages 120, 124, 131, and 134]
- K. Amin, S. Kale, G. Tesauro, and D. Turaga. Budgeted prediction with expert advice. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015. [Cited on pages 120, 124, and 134]
- Y. Arjevani and O. Shamir. Dimension-free iteration complexity of finite sum optimization problems. In *Advances in Neural Information Processing Systems*, pages 3540–3548, 2016. [Cited on page 69]
- S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8:121–164, 2012. [Cited on page 7]
- P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 2002a. [Cited on pages 10, 11, 12, and 99]
- P. Auer, N. Cesa-Bianchi, Y. Freund, and R. Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002b. [Cited on pages 7, 8, 20, 24, 29, and 32]
- P. Auer, N. Cesa-Bianchi, and C. Gentile. Adaptive and self-confident on-line learning algorithms. *Journal of Computer and System Sciences*, 64(1):48–75, 2002c. [Cited on pages 130 and 142]
- O. Avner, S. Mannor, and O. Shamir. Decoupling exploration and exploitation in multi-armed bandits. In *International Conference on Machine Learning*, 2012. [Cited on pages 34, 35, and 124]
- D. Baer. The ‘Filter Bubble’ Explains Why Trump Won and You Didn’t See It Coming, November 2016. NY Mag. [Cited on page 92]
- V. Bala and S. Goyal. Learning from neighbours. *Review of Econ. Studies*, 65:595–621, 1998. [Cited on page 123]

- V. Bala and S. Goyal. Conformism and diversity under social learning. *Econ. Theory*, 17:101–120, 2001. [Cited on page 123]
- H. Bauschke and P. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*, volume 408. Springer, 2011. [Cited on page 64]
- S. Boldrini, L. De Nardis, G. Caso, M. Le, J. Fiorina, and M.-G. Di Benedetto. mumab: A multi-armed bandit model for wireless network selection. *Algorithms*, 11(2):13, 2018. [Cited on page 2]
- Z. Borsos, A. Krause, and K. Levy. Online variance reduction for stochastic optimization. In *International Conference on Learning Theory*, 2018. [Cited on page 24]
- L. Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT*, pages 177–186. Springer, 2010. [Cited on page 18]
- D. Bouneffouf and I. Rish. A survey on practical applications of multi-armed and contextual bandits. *arXiv preprint arXiv:1904.10040*, 2019. [Cited on page 2]
- E. Bozdog and J. van den Hoven. Breaking the filter bubble: democracy and design. *Ethics and Information Technology*, 17(4):249–265, Dec 2015. [Cited on page 92]
- S. Bubeck. *Bandits games and clustering foundations*. PhD thesis, Universite Lille, 2010. [Cited on page 10]
- S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012. [Cited on pages 6, 11, 24, 28, 125, 128, and 143]
- S. Bucciapatnam, A. Eryilmaz, and N. B. Shroff. Multi-armed bandits in the presence of side observations in networks. In *Proceedings of the 2014 SIGMETRICS conference*, 2013. [Cited on page 124]
- S. Bucciapatnam, A. Eryilmaz, and N. B. Shroff. Stochastic bandits with side observations on networks. In *Proceedings of the 52nd IEEE Conference on Decision and Control*, 2014. [Cited on page 124]
- S. Caron, B. Kveton, M. Lelarge, and S. Bhagat. Leveraging side observations in stochastic bandits. In *Proceedings of Uncertainty in Artificial Intelligence (UAI)*, 2012. [Cited on page 124]
- L. E. Celis and F. Salehi. Lean from thy neighbor: Stochastic & adversarial bandits in a network. *arXiv preprint arXiv:1704.04470*, 2017. [Cited on page 119]
- L. E. Celis, L. Huang, V. Keswani, and N. K. Vishnoi. Classification with Fairness Constraints: A Meta-Algorithm with Provable Guarantees. *ArXiv e-prints*, June 2018. [Cited on page 94]

Bibliography

- L. E. Celis, S. Kapoor, F. Salehi, and N. Vishnoi. Controlling polarization in personalization: An algorithmic framework. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 160–169. ACM, 2019. [Cited on page 91]
- N. Cesa-Bianchi, C. Gentile, and G. Zappella. A gang of bandits. In *Proceedings of the 26th Conference on Advances in Neural Information Processing Systems (NIPS)*, 2013. [Cited on pages 124 and 138]
- C. Chang and C. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27, 2011. [Cited on pages 47 and 78]
- M. Conover, J. Ratkiewicz, M. R. Francisco, B. Gonçalves, F. Menczer, and A. Flammini. Political polarization on twitter. *ICWSM*, 133:89–96, 2011. [Cited on pages 14 and 92]
- P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 39–46. ACM, 2010. [Cited on page 1]
- D. Csiba and P. Richtárik. Importance sampling for minibatches. *arXiv:1602.02283*, 2016. [Cited on page 22]
- D. Csiba, Z. Qu, and P. Richtárik. Stochastic dual coordinate ascent with adaptive probabilities. In *International Conference on Machine Learning*, 2015. [Cited on pages 39, 65, 66, and 68]
- V. Dani, T. P. Hayes, and S. M. Kakade. Stochastic Linear Optimization under Bandit Feedback. In *Proceedings of the Annual Conference on Learning Theory (COLT)*, 2008. [Cited on pages 97, 98, 99, 103, 111, 113, and 114]
- A. Datta, M. C. Tschantz, and A. Datta. Automated experiments on ad privacy settings. *Proceedings on Privacy Enhancing Technologies*, 2015(1):92–112, 2015. [Cited on page 93]
- A. Defazio, F. Bach, and S. Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Proceedings of Advances in Neural Information Processing Systems*, pages 1646–1654, 2014. [Cited on pages 19, 23, 41, and 44]
- W. Ding, T. Qin, X.-D. Zhang, and T.-Y. Liu. Multi-armed bandit with budget constraint and variable costs. In *AAAI*, 2013. [Cited on page 98]
- C. Dünnér, S. Forte, M. Takáč, and M. Jaggi. Primal-dual rates and certificates. In *International Conference on Machine Learning*, 2016. [Cited on page 81]
- C. Dünnér, T. Parnell, and M. Jaggi. Efficient use of limited-memory accelerators for linear learning on heterogeneous systems. In *Advances in Neural Information Processing Systems*, pages 4261–4270, 2017. [Cited on pages 65, 75, and 76]

- A. Durand, C. Achilleos, D. Iacovides, K. Strati, G. D. Mitsis, and J. Pineau. Contextual bandits for adapting treatment in a mouse model of de novo carcinogenesis. In *Machine Learning for Healthcare Conference*, pages 67–82, 2018. [Cited on page 2]
- G. Ellison and D. Fudenberg. Rules of thumb for social learning. *J. of Political Economy*, 101, 1993. [Cited on page 123]
- R. Epstein and R. E. Robertson. The search engine manipulation effect (SEME) and its possible impact on the outcomes of elections. *Proceedings of the National Academy of Sciences*, 112(33):E4512–E4521, 2015. [Cited on pages 4 and 92]
- A. Farahat and M. C. Bailey. How effective is targeted advertising? In *Proceedings of the 21st international conference on World Wide Web*. ACM, 2012. [Cited on page 92]
- O. Fercoq and P. Richtárik. Accelerated, parallel, and proximal coordinate descent. *SIAM Journal on Optimization*, 2015. [Cited on page 80]
- A. D. Flaxman, A. T. Kalai, and H. B. McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the 16th ACM/SIAM symposium on Discrete algorithms (SODA)*, 2005. [Cited on pages 7 and 8]
- T. Fox-Brewster. Creepy Or Cool? Twitter Is Tracking Where You’ve Been, What You Like And Is Telling Advertisers, May 2017. Forbes Magazine. [Cited on page 92]
- M. J. Frank, B. B. Doll, J. Oas-Terpstra, and F. Moreno. Prefrontal and striatal dopaminergic genes predict individual differences in exploration and exploitation. *Nature neuroscience*, 12(8):1062, 2009. [Cited on page 1]
- Y. Freund and R. 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. [Cited on page 20]
- Y. Freund and R. 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. [Cited on page 7]
- Y. Freund and R. E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1-2):79–103, 1999. [Cited on page 8]
- D. Gale and S. Kariv. Bayesian learning in social networks. *Games and Econ. Behavior*, 45, 2003. [Cited on page 124]
- V. R. K. Garimella and I. Weber. A long-term analysis of polarization on twitter. In *ICWSM*, 2017. [Cited on pages 14 and 92]
- A. Garivier and O. Cappé. The kl-ucb algorithm for bounded stochastic bandits and beyond. In *Proceedings of the Conference on Learning Theory (COLT)*, 2011. [Cited on page 10]

Bibliography

- T. Glasmachers and U. Dogan. Accelerated coordinate descent with adaptive coordinate frequencies. In *Asian Conference on Machine Learning*, pages 72–86, 2013. [Cited on page 62]
- A. Goldfarb and C. Tucker. Online display advertising: Targeting and obtrusiveness. *Marketing Science*, 2011. [Cited on page 91]
- B. Golub and M. O. Jackson. Naive learning in social networks and the wisdom of crowds. *American Econ. Journal: MicroEcon.*, 2, 2010. [Cited on page 124]
- S. Goyal. Learning in networks. In G. Demange and M. Wooders, editors, *Group formation in Econ.: networks, clubs, and coalitions*, chapter 4, pages 122–167. Cambridge University Press, 2005. [Cited on page 123]
- F. M. Harper and J. A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4):19:1–19:19, Dec. 2015. ISSN 2160-6455. doi: 10.1145/2827872. URL <http://doi.acm.org/10.1145/2827872>. [Cited on pages 104 and 107]
- E. Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016. [Cited on pages 7, 8, and 9]
- T. Hofmann, A. Lucchi, S. Lacoste-Julien, and B. McWilliams. Variance reduced stochastic gradient descent with neighbors. In *Advances in Neural Information Processing Systems*, pages 2305–2313, 2015. [Cited on page 47]
- S. Hong and S. H. Kim. Political polarization on twitter: Implications for the use of social media in digital governments. *Government Information Quarterly*, 33(4):777–782, 2016. [Cited on pages 14 and 92]
- X. Huo and F. Fu. Risk-aware multi-armed bandit problem with application to portfolio selection. *Royal Society open science*, 4(11):171377, 2017. [Cited on page 2]
- T. Johnson and C. Guestrin. StingyCD: Safely avoiding wasteful updates in coordinate descent. In *International Conference on Machine Learning*, pages 1752–1760, 2017. [Cited on pages 64 and 81]
- M. Joseph, M. Kearns, J. H. Morgenstern, and A. Roth. Fairness in learning: Classic and contextual bandits. In *Advances in Neural Information Processing Systems*, pages 325–333, 2016. [Cited on page 98]
- E. Katz and P. Lazarsfeld. *Personal Influence*. The Free Press, 1955. [Cited on page 120]
- T. Kern and A. György. Svrg++ with non-uniform sampling. 2016. [Cited on pages 22 and 44]
- T. Kocák, G. Neu, M. Valko, and R. Munos. Efficient learning by implicit exploration in bandit problems with side observations. In *Proceedings of the 27th Conference on Advances in Neural Information Processing Systems (NIPS)*, 2014. [Cited on pages 124 and 134]

- J. Konečný and P. Richtárik. Semi-stochastic gradient descent methods. *Frontiers in Applied Mathematics and Statistics*, 3:9, 2017. [Cited on page 19]
- V. Kuleshov and D. Precup. Algorithms for multi-armed bandit problems. *arXiv preprint arXiv:1402.6028*, 2014. [Cited on page 12]
- S. Lacoste-Julien, M. Schmidt, and F. Bach. A simpler approach to obtaining an $o(1/t)$ convergence rate for the projected stochastic subgradient method. *arXiv:1212.2002*, 2012. [Cited on pages 22, 49, 51, 55, and 58]
- T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6, 1985. [Cited on page 147]
- P. Lazarsfeld, B. Berelson, and H. Gaudet. *The People's Choice*. Columbia University Press, 1948. [Cited on page 120]
- L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010. [Cited on pages 4, 14, 94, and 95]
- S. Li, A. Karatzoglou, and C. Gentile. Collaborative filtering bandits. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 539–548. ACM, 2016. [Cited on page 94]
- J. Liu, P. Dolan, and E. R. Pedersen. Personalized news recommendation based on click behavior. In *Proceedings of the 15th international conference on Intelligent user interfaces*. ACM, 2010. [Cited on page 91]
- O. A. Maillard, R. Munos, and G. Stoltz. A finite-time analysis of multiarmed bandits problems with kullback-leibler divergences. In *Proceedings of the Conference on Learning Theory (COLT)*, 2011. [Cited on page 10]
- S. Mannor and O. Shamir. From bandits to experts: On the value of side-observations. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 684–692, 2011. [Cited on page 124]
- K. Misra, E. M. Schwartz, and J. Abernethy. Dynamic online pricing with incomplete information using multiarmed bandit experiments. *Marketing Science*, 2019. [Cited on page 2]
- H. Namkoong, A. Sinha, S. Yadlowsky, and J. Duchi. Adaptive sampling probabilities for non-smooth optimization. In *International Conference on Machine Learning*, 2017. [Cited on page 24]
- D. Needell, R. Ward, and N. Srebro. Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. In *Proceedings of Advances in Neural Information Processing Systems*, pages 1017–1025, 2014. [Cited on pages 18, 20, 22, 27, 39, and 54]

Bibliography

- J. Nutini, M. Schmidt, I. Laradji, M. Friedlander, and H. Koepke. Coordinate descent converges faster with the gauss-southwell rule than random selection. In *International Conference on Machine Learning*, pages 1632–1641, 2015. [Cited on pages 64, 65, 66, 69, and 76]
- A. Osokin, J. Alayrac, I. Lukasewitz, P. Dokania, and S. Lacoste-Julien. Minding the gaps for block frank-wolfe optimization of structured svms. In *International Conference on Machine Learning*, 2016. [Cited on page 65]
- S. Pandey and C. Olston. Handling advertisements of unknown quality in search advertising. In *Advances in Neural Information Processing Systems*, 2006. [Cited on page 93]
- G. Papa, P. Bianchi, and S. Cl  men  on. Adaptive sampling for incremental optimization using stochastic gradient descent. In *International Conference on Algorithmic Learning Theory*, pages 317–331. Springer, 2015. [Cited on pages 23 and 48]
- E. Pariser. *The filter bubble: What the Internet is hiding from you*. Penguin UK, 2011. [Cited on pages v, viii, 4, 92, 96, and 97]
-   . Payzan-LeNestour and P. Bossaerts. Do not bet on the unknown versus try to find out more: estimation uncertainty and “unexpected uncertainty” both modulate exploration. *Frontiers in neuroscience*, 6:150, 2012. [Cited on page 1]
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. [Cited on page 107]
- D. Perekrestenko, V. Cevher, and M. Jaggi. Faster coordinate descent via adaptive importance sampling. In *International Conference on Artificial Intelligence and Statistics*, 2017. [Cited on pages 62, 65, 68, 73, 75, 76, 80, and 81]
- A. Rakotomamonjy, S. Ko  o, and L. Ralaivola. Greedy methods, randomization approaches, and multiarm bandit algorithms for efficient sparsity-constrained optimization. *IEEE transactions on neural networks and learning systems*, 28(11):2789–2802, 2017. [Cited on page 81]
- F. N. Ribeiro, L. Henrique, F. Benevenuto, A. Chakraborty, J. Kulshrestha, M. Babaei, and K. P. Gummadi. Media bias monitor: Quantifying biases of social media news outlets at scale. In *Proceedings of the 12th International AAAI Conference on Web and Social Media (ICWSM)*, June 2018. [Cited on page 92]
- H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951. [Cited on page 39]

- R. E. Robertson, D. Lazer, and C. Wilson. Auditing the personalization and composition of politically-related search engine results pages. In *Proceedings of the 2018 World Wide Web Conference*, 2018. [Cited on page 92]
- P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 1987. [Cited on page 107]
- P. Sakulkar and B. Krishnamachari. Stochastic contextual bandits with known reward functions. *arXiv preprint arXiv:1605.00176*, 2016. [Cited on page 92]
- F. Salehi, L. E. Celis, and P. Thiran. Stochastic optimization with bandit sampling. *arXiv:1708.02544*, 2017a. [Cited on page 17]
- F. Salehi, P. Thiran, and L. E. Celis. Stochastic dual coordinate descent with bandit sampling. *arXiv:1712.03010*, 2017b. [Cited on pages 34 and 53]
- F. Salehi, P. Thiran, and E. Celis. Coordinate descent with bandit sampling. In *Advances in Neural Information Processing Systems 32*, pages 9247–9257, 2018. [Cited on page 61]
- B. Sanditov. *Essays on Social Learning and Imitation*. PhD thesis, Universitaire Pers Maastricht, 2006. [Cited on pages 4, 14, and 120]
- M. Schmidt, R. Babanezhad, M. Ahmed, A. Defazio, A. Clifton, and A. Sarkar. Non-uniform stochastic average gradient method for training conditional random fields. In *Proceedings of Artificial Intelligence and Statistics*, pages 819–828, 2015a. [Cited on pages 22 and 39]
- M. Schmidt, R. Babanezhad, M. Ahmed, A. Defazio, A. Clifton, and A. Sarkar. Non-uniform stochastic average gradient method for training conditional random fields. In *artificial intelligence and statistics*, pages 819–828, 2015b. [Cited on page 23]
- S. Shalev-Shwartz and A. Tewari. Stochastic methods for l_1 -regularized loss minimization. *Journal of Machine Learning Research*, 12(Jun):1865–1892, 2011. [Cited on pages 64, 66, 80, and 81]
- S. Shalev-Shwartz and T. Zhang. Accelerated mini-batch stochastic dual coordinate ascent. In *Advances in Neural Information Processing Systems*, pages 378–385, 2013a. [Cited on page 62]
- S. Shalev-Shwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14(Feb):567–599, 2013b. [Cited on pages 62, 65, 66, 68, 73, 81, and 86]
- W. Shen, J. Wang, Y.-G. Jiang, and H. Zha. Portfolio choices with orthogonal bandit learning. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015. [Cited on page 2]

Bibliography

- Z. Shen, H. Qian, T. Zhou, and T. Mu. Adaptive variance reducing for stochastic gradient descent. In *IJCAI*, pages 1990–1996, 2016. [Cited on page 23]
- H. Shi, S. Tu, Y. Xu, and W. Yin. A primer on coordinate descent algorithms. *arXiv preprint arXiv:1610.00040*, 2016. [Cited on page 65]
- T. Speicher, M. Ali, G. Venkatadri, F. N. Ribeiro, G. Arvanitakis, F. Benevenuto, K. P. Gummadi, P. Loiseau, and A. Mislove. Potential for discrimination in online targeted advertising. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*. PMLR, 2018. [Cited on page 92]
- K. Sridharan. A gentle introduction to concentration inequalities. 2002. [Cited on page 101]
- S. Stich, A. Raj, and M. Jaggi. Approximate steepest coordinate descent. In *International Conference on Machine Learning*, 2017. [Cited on pages 65 and 81]
- R. S. Sutton, A. G. Barto, et al. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998. [Cited on page 11]
- B. Szorenyi, R. Busa-Fekete, I. Hegedüs, R. Ormándi, M. Jelasity, and B. Kégl. Gossip-based distributed stochastic bandit algorithms. In *30th International Conference on Machine Learning (ICML 2013)*, volume 28, pages 19–27. Acm Press, 2013. [Cited on page 124]
- C. Wang and D. M. Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 448–456. ACM, 2011. [Cited on page 95]
- J. Wasilewski and N. Hurley. Incorporating diversity in a learning to rank recommender system. In *FLAIRS Conference*, pages 572–578, 2016. [Cited on page 97]
- I. Weber, V. R. K. Garimella, and A. Batayneh. Secular vs. islamist polarization in egypt on twitter. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. ACM, 2013. [Cited on pages 14 and 92]
- L. Xiao and T. Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014. [Cited on pages 18, 19, 23, 44, and 47]
- Y. Yan. Mehrotra’s Predictor-Corrector Interior Point Method. <https://github.com/YimingYAN/mpc>. [Cited on page 105]
- D. Yoo. Individual and social learning in bio-technology adoption: The case of gm corn in the u.s. In *Agricultural & Applied Econ. Association’s Annual Meeting (AAEA)*, 2012. [Cited on pages 4, 14, 119, and 120]

- J. Y. Yu and S. Mannor. Piecewise-stationary bandit problems with side observations. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, 2009. [Cited on page 124]
- A. Zhang and Q. Gu. Accelerated stochastic block coordinate descent with optimal sampling. In *International Conference on Knowledge Discovery and Data Mining*, pages 2035–2044. ACM, 2016. [Cited on page 65]
- C. Zhang, H. Kjellstrom, and S. Mandt. Stochastic learning on imbalanced data: Determinantal point processes for mini-batch diversification. *arXiv:1705.00607*, 2017. [Cited on page 22]
- J. Zhang, M. S. Ackerman, and L. Adamic. Expertise networks in online communities: Structures and algorithms. In *Proceedings of the World Wide Web Conference (WWW)*, 2007. [Cited on pages 4, 14, and 120]
- Y. Zhang. Bayesian graphical models for adaptive filtering. In *PhD Thesis*, 2005. [Cited on pages 104 and 107]
- P. Zhao and T. Zhang. Accelerating minibatch stochastic gradient descent using stratified sampling. *arXiv:1405.3080*, 2014. [Cited on page 22]
- P. Zhao and T. Zhang. Stochastic optimization with importance sampling for regularized loss minimization. In *International Conference on Machine Learning*, 2015a. [Cited on pages 18, 19, 22, 42, 45, 58, 62, 65, 68, and 80]
- P. Zhao and T. Zhang. Stochastic optimization with importance sampling for regularized loss minimization. In *Proceedings of International Conference on Machine Learning*, pages 1–9, 2015b. [Cited on pages 20 and 22]

Farnood SALEHI



📍 Avenue du-tir Fédéral 22
1024 Ecublens
Switzerland

💻 farnoodsalehi.me
✉ salehifarnood@gmail.com
☎ +41 (0)78 720 65 62

🌐 github.com/F-Salehi

EXPERTISE: Machine Learning • Probabilistic Modeling • Data Mining • Optimization • Bayesian Inference

EDUCATION

- | | | |
|-------------|---|-----------------------|
| 2014 – 2019 | Swiss Federal Institute of Technology Lausanne (EPFL)
Ph.D. in Computer Science (Machine Learning) <ul style="list-style-type: none">Developed methods to improve speed and accuracy of machine learning algorithms. Research topics include: Optimization, Bayesian Methods, Deep Learning, Online Learning, Knowledge Base Graphs, Probabilistic Models, Time Series. | Lausanne, Switzerland |
| 2010 – 2014 | Sharif University of Technology
B.Sc. in Communication Systems <ul style="list-style-type: none">Bachelor project focuses on making network communications robust to attacks using consensus algorithmGPA 18.93/20, Rank 4/190. | Tehran, Iran |

PROFESSIONAL EXPERIENCE

- | | | |
|-------------|--|----------------------|
| Summer 2018 | Disney Research
Research Intern <ul style="list-style-type: none">Developed a novel scalable hyper-parameter learning algorithm for knowledge graphs.The proposed method improves state-of-the-art results by 2% (UAI 2019, [2]). | Los Angeles, CA, USA |
| Summer 2013 | The Hong Kong University of Technology
Research Intern <ul style="list-style-type: none">Simulated the performance of wireless networks for different loads of mobile users. | Hong Kong |

SKILLS

- | | |
|---------------------|--|
| Data science | Machine Learning, Deep Learning, Optimization, Probabilistic Modelling, Computer Vision. |
| Programming | Python, PyTorch, Pandas, Keras, Spark, Git, Matlab, Bash, SQL, NoSQL, LaTeX. |

FELLOWSHIPS, AWARDS & PRIZES

- | | |
|------|--|
| 2019 | Best paper award at ACM FAT* conference [3] |
| 2014 | EDIC 1-year Fellowship (50,000 CHF), EPFL |
| 2010 | Silver medal in Iranian national Physics Olympiad |
| 2010 | Ranked 30 among 200,000 participants in the nationwide engineering university exam in Iran |

LANGUAGES

English (Fluent), German (A2-B1), Turkish (Conversational), Persian (Native)

SUPERVISED STUDENT PROJECTS

- Nicolas Brandt-Dit-Grieurin, "Grouped SAGA for large models", 2019.
- Ritabrata Ray, "Optimal Regularizer for the Matrix Factorization", 2019.
- Gümüs Orcun, "Network alignment using graph embedding", 2017.
- Delisle Maxime, "Find the best regularizer for Lasso", 2016.

- [1] Learning Hawkes Processes from a Handful of Events,
Farnood Salehi*, William Trouleau*, Matthias Grossglauser and Patrick Thiran,
Conference on Neural Information Processing Systems (NeurIPS), 2019.
- [2] Augmenting and Tuning Knowledge Graph Embeddings,
Robert Bamler*, Farnood Salehi* and Stephan Mandt,
Conference on Uncertainty in Artificial Intelligence (UAI), 2019.
- [3] Controlling Polarization in Personalization: An Algorithmic Framework (**best paper award**),
Elisa Celis, Sayash Kapoor, Farnood Salehi and Nisheeth Vishnoi,
ACM Conference on Fairness, Accountability, and Transparency (FAT*), 2019.
- [4] Making Variance Reduction more Effective for Deep Networks,
Nicolas Brandt-Dit-Grieurin, Farnood Salehi and Patrick Thiran,
NeurIPS Workshop on Beyond First Order Methods in Machine Learning, 2019.
- [5] Coordinate Descent with Bandit Sampling,
Farnood Salehi, Patrick Thiran and Elisa Celis,
Conference on Neural Information Processing Systems (NeurIPS), 2018.
- [6] Dictionary Learning Based on Sparse Distribution Tomography,
Pedram Pad*, Farnood Salehi*, Elisa Celis, Patrick Thiran and Michael Unser,
International Conference on Machine Learning (ICML), 2017.
- [7] Auctions for Online Advertising with Constraints,
Elisa Celis, Farnood Salehi and Salman Salamatian,
Manufacturing & Service Operations Management Society Conference (MSOM), 2017.
- [8] Stochastic Optimization with Bandit Sampling,
Farnood Salehi, Patrick Thiran and Elisa Celis,
arXiv:1708.02544 (currently under review at JMLR).
- [9] Learn from the Neighbor: Stochastic and Adversarial Bandits in a Network,
Farnood Salehi and Elisa Celis
Presented at the International Symposium on Mathematical Programming (ISMP), 2015, arXiv:1704.04470.