LEARNING FROM EXPERT ADVICE FRAMEWORK: ALGORITHMS AND APPLICATIONS

BY

ANH TRUONG

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Industrial and Enterprise Systems Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2018

Urbana, Illinois

Doctoral Committee:

Associate Professor Negar Kiyavash, Chair
Associate Professor Ramavarapu S. Sreenivas
Professor Minh N. Do
Assistant Professor Lavanya Marla
Assistant Professor S. Rasoul Etesami

ABSTRACT

Online recommendation systems have been widely used by retailers, digital marketing, and especially in e-commerce applications. Popular sites such as Netflix and Amazon suggest movies or general merchandise to their clients based on recommendations from peers. At core of recommendation systems resides a prediction algorithm, which based on recommendations received from a set of experts (users), recommends objects to other users. After a user "consumes" an object, his feedback provided to the system is used to assess the performance of experts at that round and adjust the predictions of the recommendation system for the future rounds. This so-called "learning from expert advice" framework has been extensively studied in the literature. In this dissertation, we investigate various settings and applications ranging from partial information, adversarial scenarios, to limited resources. We propose provable algorithms for such systems, along with theoretical and experimental results.

In the first part of the thesis, we focus our attention to a generalized model of learning from expert advice in which experts could abstain from participating at some rounds. Our proposed online algorithm falls into the class of weighted average predictors and uses a time varying multiplicative weight update rule. This update rule changes the weight of an expert based on his relative performance compared to the average performance of available experts at the current round. We prove the convergence of our algorithm to the best expert, defined in terms of both availability and accuracy, in the stochastic setting.

Next, we study the optimal adversarial strategies against the weighted average prediction algorithm. All but one expert are honest and the malicious expert's goal is to sabotage the performance of the algorithm by strategically providing dishonest recommendations. We formulate the problem as a Markov decision process (MDP) and apply policy iteration to solve it. For the logarithmic loss, we prove that the optimal strategy for the adversary is the greedy policy, whereas for the absolute loss, in the 2-experts, discounted cost setting, we prove that the optimal strategy is a threshold policy. We extend the results to the infinite horizon problem and find the exact thresholds for the stationary optimal policy. As an effort to investigate the extended problem, we use a mean field approach in the N-experts setting to find the optimal strategy when the predictions of the honest experts are i.i.d.

In addition to designing an effective weight update rule and investigating optimal strategies of malicious experts, we also consider active learning applications for learning with expert advice framework. In this application, the target is to reduce the number of labeling while still keeping the regret bound as small as possible. We proposed two algorithms, EPSL and EPAL, which are able to efficiently request label for each object. In essence, the idea of two algorithms is to examine the opinion ranges of experts, and decide to acquire labels based on the maximum difference of those opinion using a randomized policy. Both algorithms obtain nearly optimal regret bound up to some constant depending on the characteristics of experts' predictions.

Last but not least, we turn our attention to the generalized "best arm identification" problem in which, at each time, there is a subset of products whose rewards or profits are unknown (but follow some fixed distributions), and the goal is to select the best product to recommend to users after trying on a number of sampling. We propose UCB based (Upper Confidence Bound) algorithms that provide flexible parameter tuning based on the availability of each arm in the collection. We also propose a simple, yet efficient, uniform sampling algorithm for this problem. We proved that, for these algorithms, the error of selecting the incorrect arm decays exponentially over time.

To my wife and my little daughter, for their love and support. To my parents, brother and sister, for their encouragement.

ACKNOWLEDGMENTS

I would like to thank Prof. Negar Kiyavash, my advisor, who has been extremely supportive and encouraging throughout my PhD. Her enthusiastic guidance and interesting ideas guide me through the whole process toward completing this dissertation. This thesis could not have materialized without her support. I also would like to thank Prof. Rasoul Etesami for his enthusiastic and helpful discussions during the projects. I am very grateful with the precious time of the committee members, Prof. Ramavarapu S. Sreenivas, Professor Minh N. Do, Professor Lavanya Marla who provide valuable feedback and suggestions to develop the thesis.

Next, I would like to thank all of my collaborators and friends. Many thanks to Gergely Neu for great discussions on online learning paradigm. I would like to thank Ashutosh Nayyar for his awesome ideas on adversarial setting. I also want to thank Prof. Yuliy Baryshnikov for sharing math skills to deep dive into the solutions. My special thanks to Jalal Etesami, my labmate and a nice friend, who is always joyful and helpful. I also would like to thank other labmates, Chris, Xun, Sachin, YingXiang, Ashish, Taposh, Jiaming, Siva. It has been my pleasure talking with them, taking courses with them, and be friends with them. Many thanks to my colleagues at Capital One, Mark, Vincent and other folks for their encouragement and support.

I would like to thank my friends, Tuan Hoang, Vuong Le, Huy Bui, Trong Nguyen, Phong Le, Chinh Nguyen, just to name a few, for having great time with me, playing sports, games and having a lot of fun with me.

Last but not least, I would like to gratefully thank my wife, Han Le and my little daughter, Fiona Truong for their love and support. They have been always being beside me, bringing me more energy and motivations, enjoying great moments and spending great time with me for the whole process. I owe them a debt of gratitude.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Introduction and motivations

Online recommendation systems have been widely used by retailers, digital marketing, and especially in e-commerce applications. Popular sites such as Netflix and Amazon suggest movies or general merchandise to their clients based on recommendations from peers. At core of recommendation systems resides a prediction algorithm, which based on recommendations received from a set of experts (users), recommends objects to other users. After a user "consumes" an object, his feedback provided to the system is used to assess the performance of experts at that round and adjust the predictions of the recommendation system for the future rounds.

We consider a specific recommendation algorithm that combines weighted opinions of the experts. The system initially assigns uniform weights to experts, and changes the weights from time to time based on the performance of the experts evaluated through user's feedback. This general framework of learning from expert advice was introduced by Littlestone and Warmuth [1] and Vovk [2]. Beside online recommendation systems, this framework has been applied to various other learning problems such as the shortest path problem [3], [4], [5], metrical task system [6], and online paging [7]. In this dissertation, we address the issues of missing information, adversarial behaviors and limited resources of the framework. We aim to answer these questions: (i) how the system deals with the difficulty of missing experts at some time instances; (ii) can we investigate the effect of malicious experts in the system; (iii) how the system reduces the cost of object labeling; (iv) how the system selects the best object given partial information and limited sampling budget. In particular, our motivations are as follows.

Missing expert predictions

In the aforementioned applications, it is often assumed that all experts are present at all rounds of voting. This assumption is reasonable in scenarios where a dedicated set of users, say movie critics, watch and rate majority of movies. However, such an assumption does not hold true for recommending merchandise on a website such as Amazon where the set of users who have rated various objects may not even intersect. We consider the scenario where experts vote in a safe way (intentionally) in order to earn credit (high weight) from the system. In other words, they only vote for the famous items and avoid voting for the difficult ones. If these voting behavior are governed by an adversary, it will degrade the performance of the recommendation systems. In fact, the effect of such predictions in practical applications is even more pronounced as it is described in the following examples:

- **Movies recommendation**: People have been recently living in smart homes where they are recommended to a set of good movies whenever their televisions are turned on. A movie is recommended if it obtains high ratings from those users (experts) who are trustable to the recommendation system. Consider the case when an adversary attempts to drive local residence in a specific area to watch some specific movie in order to increase audience attentions, sell more ads, or for a certain political incentive. This adversary can indirectly influence the recommendations of the experts through social media such as Twitter [8], text review of movie critics [9], or news analysis [10]. The adversary's goal is to manipulate the expert's voting in such a way that they can get high trust from the system on a few objects, then mispredict on the target movies.

- **Commute routes recommendation**: With the fast development of smart car, drivers get updated routes information for their commute using GPS or other application devices in their car (see e.g., [11] for a real-time route recommendation system). Consider an adversary who attempts to cause traffic at a specific area. By manipulating the received signals of a set of designated GPS applications (experts in our setting), the adversary can deceitfully recommend the drivers to commute on the same road at a certain time of the day. Such attack was foreseen from [12] where the authors used the term 'imperfect information'.

- **Byzantine attack on wireless sensor network**: Nowadays, smart buildings (or tree houses) have been equipped with a set of sensors (experts) to collect the temperature of the surrounding environment in order to intelligently adjust it toward comforting their residence. Those sensors send the information back to a centralized system which, after calculating the average temperature, makes a decision on how the temperature must change. Now if an adversary attempts to intrude those sensors, it can significantly change the temperature of the building, hence affecting people (trees) inside the smart building (tree house). One example of this kind of attack is Byzantine attack whose effects have been investigated by [13], [14], [15].

With exception of a few works ([16], [17], [18]), recommendation system with "sleeping experts", a term coined by Freund et al. [19], remains largely understudied in the literature. As recommendation systems are designed to perform not much worse than the best expert, identifying such an expert is crucial. When all experts are present at each voting round, the best expert is simply defined as the one with the smallest loss over the decision horizon. However, it is not clear who the best expert is when "sleeping" is allowed (i.e., an expert does not necessarily vote on all instances). We will present a definition for the best expert in this scenario.

Adversarial scenarios for malicious experts

In the classical setting of the learning-with-expert-advice framework, all experts are presumed to be honest. Very little work is done on analyzing whether the algorithm is robust to adversarial experts who aim to throw of the predictions. In this part of the dissertation, we consider an adversarial setting in which a malicious expert, who wants to sabotage the system, provides strategically dishonest recommendations. Learning with expert advice has been extensively studied in the literature $[1, 2, 20, 21]$, in which the algorithm's goal is to minimize the system's overall regret with respect to all experts.

Here, we address the problem from the perspective of the malicious expert who attempts to maximize the overall loss of the system by playing his best *dynamic* policy. Some real world examples of such adversarial settings are *recommendation* and *sensor fusion* systems:

- **Recommendation Systems** Recommendation systems are vulnerable to the malicious identities who intentionally cast misleading votes to confuse the systems. Those identities can directly act or hack to the system users' accounts and give false recommendations [22]. The longer these malicious identities stay unidentified, the more damage they can cause to the reputation of the recommendation systems. Such behavior surprisingly can even occur without malicious intention. The following two quotes are from two different reviewers for the movie "Interstellar" on IMDB official website (rating range is from 1 to 10, 1 for the worst and 10 for the best):

  – "...I give 1 star to bring balance to the current rating, in reality this movie is of course not that bad."

  – "My honest rating would be 6 for that movie but I rated it 1 to balance the 'emotional' ratings."

  Such experts cast their rates to manipulate the outcome of the system rather than honestly reporting their actual ratings. Understanding the best strategy for such experts, and hence the amount of damage they can do, is the main goal of this paper.

- **Sensor Fusion** In this application, a central decision maker receives reading from a set of sensors, and combines them to make a decision. One or a subset of the sensors in the system could be malicious and attempts to ruin the quality of the central decision making. In scenarios where the reading from the sensors is costly, if the malicious sensor is successful at making the center confused several times, the damage it causes to the system is significant.

Adaptive labeling with expert advice

We consider applications of learning with expert advice framework on active learning, which has drawn much interest recently. In this framework, the challenging problem is that the labeling procedure is expensive or time-consuming, and thus the goal is to find the good examples to query for the true labels. This has a wide range of applications, from medical diagnosis to recommendation systems [23], [24] to natural language processing [25]. We consider applications of learning with expert advice framework where the labels are retrieved with expensive cost or through a time consuming procedure. Our motivation is from the following examples:

- In the moving rating systems, the true opinion or ground truth from a specific user for each of movie is required in order to update the losses, which then update the weights for experts. However, it is very time-consuming to watch the whole movie so that the user can give the exact feedback on that movie.

- For text classification and information retrieval tasks, it is required to get labels of documents (relevant or non-relevant), detailed annotations such as name entities and word relations to update features' weights. Those procedures usually take a lot of time so that users can read through the documents, and sometimes restrict users from uncommon domain knowledge.

The purpose is then to reduce the number of requests for labeling while keeping the regret rate as small as possible.

Simple regret in multiarmed bandit problems

All above settings focuses on full-information scenarios where predictions of all experts are revealed at any time. In this final part, we turn our attention to the partial-information setting and our goal is to minimize a single recommendation error instead of accumulated error. Specifically, we consider the product recommendation problem in which there is a collection of products whose rewards or profits are unknown, and the goal is to select the best product to recommend to users after a number of sampling. This problem has various applications in telecommunications, e-commerce and advertising. As an example, a cellular system needs to select the best wireless channel for a specific customer, an e-commerce website needs to recommend the best product to their customers, and an advertiser tends to show an advertisement piece to the web users to maximize the profit. The problem is widely explored by a large proportion or work in the literature. Most of the work focused on the full setting where all products are available for pick up at all time. In this paper, we consider a more general setting where we allow some products to be unavailable at some time. This brings practical use case for the aforementioned applications: at one time, some communication channels are noisy, then cannot be the good candidate for user; the set of products and advertisements may not be the same every time. Specifically, we assume that at each time, there is a subset of arms available, each of them has a reward that follows from some fixed, but unknown distribution. The ultimate goal is to recommend the best arm in the collection with a limited number of sampling.

1.2 Our Contribution

In Chapter 2, we study the sleeping expert setting. We propose a weighted average recommendation algorithm that changes the weight of an expert based on his relative performance compared to the average performance of the *available* experts at that round. This update rule ensures that informative predictions (ones differing from the average recommendation) are rewarded as opposed to merely accurate predictions. Our algorithm allows continuous value predictions, but the feedback of the user is assumed to be binary. We consider the stochastic setting for this problem where the availability and accuracy of experts are assumed to be *stationary*, and follow some unknown joint distribution. We prove that the proposed algorithm converges to the best expert, defined as the one with the highest average performance based on his availability and accuracy. The experimental results show that our algorithm outperforms other recent algorithms such as Dsybil [26] and SBayes [19] for the absolute loss and binary prediction values in both stochastic and adversarial settings. Moreover, we consider a modified version of our algorithm which assigns a constant loss to sleeping experts in the stochastic setting and show that it also outperforms several existing algorithms for appropriate choices of the constant loss.

In Chapter 3, we study the adversarial setting where there exist some experts who intentionally give dishonest predictions to ruin the system. This work differs from most of the aforementioned literature in the sense that we formulate the adversarial learning system as a more realistic Markov decision process (MDP) [1] rather than a typical min-max regret game between an algorithm and an adversary who attempts to maximize the regret by manipulating the sequence of losses of all the experts. In our setting the adversary plays against the algorithm and the random predictions of other experts. Since the problem we are considering can be cast as an MDP (for single malicious expert) or stochastic game (for multiple malicious experts), there are general methods such as reinforcement learning or policy iteration to analyze it. Such approaches even though may not provide closed form solutions, they still provide tractable analytical tools to approximate the optimal policies. Indeed, this is one of the significant advantages of our model compared to the existing ones in the literature such as [29] whose analysis for more than 3 experts remains open. On the other hand, our results generalize those in [30] which was only given for the case of $N = 2$ experts and the logarithmic loss function.

We formulate the problem as an MDP and find the optimal strategy for the malicious expert for some specific class of loss functions. More specifically, we consider binary predictions and two types of losses: *logarithmic* and *absolute*. For the logarithmic loss, somewhat surprisingly, we prove that the greedy policy is optimal. For the absolute loss and two experts with discounted factor, we prove the optimality of a *threshold* type policy and extend our result to the infinite horizon setting by characterizing the optimal threshold in a closed form. Finally, for large number of experts we propose a mean field approximation approach to find the solutions for the setting where all the honest experts have the same behaviors.

In Chapter 4, we study the efficient labeling in learning with expert advice. We define the regret based on the total number of requests as opposed to the whole time horizon from which the standard regret notion is defined. In fact, this definition is a natural definition in this setting since the algorithm does not suffer loss if it decides not to acquire the label. We proposed an efficient algorithm to determine whether to ask for label of each object. Based on experts' opinion on each round, a random variable, following a Bernoulli distribution whose parameter is the maximal difference of experts' predictions, is drawn to decide whether the labeling is necessary. The main idea is that when most experts roughly agree on one object, it is not needed to ask for its label. On another hand, if experts tend to disagree with each other, then the request for label is significant. We proposed two algorithms, EPSL and EPAL, both of them aim to reduce the number of queries by exploring the characteristic of expert predictions in each round, without the knowledge of the number of queries. However, while EPSL yields the better performance than EPAL, it requires the prior knowledge of

[1] MDP is a stochastic control process in which the decision maker chooses an action at each time based on the current state. That action incurs a current loss and moves the state to the next one. The decision maker's goal is to select a sequence of actions to optimize his total loss. We refer the reader to Bellman [27] and Howard [28] for more details on MDP.

the ranges of experts predictions for the whole horizon. EPAL relaxes that requirement by using a time-varying learning rate, which is updated on the run of the algorithm. We proved that both algorithms obtain the optimal upper bound of the regret up to some constant that depends on the characteristic of experts predictions. While EPAL relax the requirement of the access to the prior information from EPSL, its performance is slightly worse than EPSL, by a constant of $\sqrt{2}$. In the experimental results, we compare EPAL with other algorithms in this setting and show that our algorithm outperforms the others on the regret rate, on both synthetic datasets and various real datasets.

In Chapter 5, we study the simple regret framework where the goal is to identify the best arms in a multiarmed bandit problem with a limited sampling budget. Our main results are the following two folds. We propose UCB based (Upper Confidence Bound) algorithms that can provide different ways to tune the parameters based on the availability of each arm in the collection. We also propose a simple, yet efficient, uniform sampling algorithm for this problem. We proved that all above algorithms end up with recommend the best arm in the sense that the error of selecting the incorrect arm converges exponentially by time. Although there exist some limitations on the parameter tuning, we prove in the experimental results that by applying the approximate algorithms, we still get performance nearly as good as those algorithms without spending too much effort on parameters selection.

1.3 Literature Review

Learning from Expert Advice

Learning from expert advice has a long development history dating back to the sequential predictions, first introduced in the framework of repeated game by Blackwell [31] and Hannan [32]. Later, Warmuth and Littlestone [1] and Vovk [2] formally introduced the framework, notations, and established seminal results with weighted majority algorithm and aggregating forecaster, respectively. Since then, the framework has drawn great attention in the literature. Kivinen [33] developed further the weighted average algorithms. Kivinen and Warmuth proposed the exponential weighted average algorithm [34]. The regret bounds from those algorithms have been improved further using doubling trick and time-varying learning rate by Cesa-Bianchi et al. [20] and later on by Yaroshinsky et al. [35], van Erven et al. [36], Auer et al. [37], and Grünwald [38]. In the same vein, Even-Dar et al. [39], Adamskiy et al. [40], Gofer and Mansour [41], Moroshko and Crammer [42], Adamskiy et al. [40], Moroshko et al. [42], György and Szepesväri [43] proposed different regret-based approaches. Foster [44] conducted the analysis on worst-case scenarios. Herbster and Warmuth investigated the situations where the best experts may change over time [45]. Vovk [46] introduced another type of forecaster called defensive forecaster which was later compared to his first algorithm by Chernov [47]. Chernov and Vovk [48] introduced an algorithm with unknowned number of experts. Gyorgy et al. [49] considered the setting with large number of experts. Chernov and Zhdanov [50] considered the framework with discounted loss. Other online learning algorithms were introduced in [51–61]. Related algorithms for online ranking were mentioned in [62–67]. Enthusiastic readers can refer to Cesa-Bianchi and Lugosi [21] who provided an excellent source for this framework, summarized most of above results and proposed a perspective applying potential functions for regret analysis on such system. The usage of potential function was also introduced by Hart and Mas-Colell [68].

Since first introduced, learning with expert advice has been adopted to a wide range of applications ranging back from information theory (Cover [29], Ziv [69]), data compression (Ziv and Lempel [70], Ziv [71]), data sequences (Merhav and Feder [72]) to competitive analysis (Borodin and El-Yaniv [73], Vovk [46]), Kozat and Singer [74]. Recently, this framework has been applied to various other learning problems such as multitask learning [75], stock prediction [76], sport games and market prediction [77], the shortest path problem [3], [4], [5], metrical task system [6], online paging [7], calendar scheduling [78] and text classification [79].

Sleeping experts setting

Cesa-Bianchi et al. [20] showed that a weighted average prediction algorithm which is originally designed to guarantee sublinear regret for adversarial (non-stochastic) experts can asymptotically perform as good as the best expert in the hindsight. Recently, there have been several works for both adversarial setting [80], and stochastic setting [81], or the combination of two [82]. However, all the above works have not dealt with the sleeping expert scenario where some experts might abstain from giving predictions at some time instances. Sleeping experts were not considered until recently with the presence of the two following research directions in the literature.

In the adversarial setting, Freund et al. [19] considered predictions of available experts at each time and combined them using an exponentially weighted averaging rule. In their algorithm, while the weight of an available expert is updated by his performance, the weight of a sleeping expert remains unchanged. Blum and Mansour [16] presented a time selection function to indicate the availability of experts. Their proposed external regret of one expert, defined by the difference of algorithm's loss and the loss of that expert, is calculated on the rounds that expert was available. Our algorithm with the constant step size is somewhat similar to the "multilinear forecaster" proposed by Bianchi and Lugosi [21]. However, their algorithm does not use the time-varying step size as in ours, and their algorithm does not apply to the case of sleeping experts, neither it does incorporate the informativeness of a prediction in the weight update rule. Moreover, in our proof of the main results, we use the stochastic approximation approach which is, to the best of our knowledge, first introduced in this framework and potentially extensible for stochastic settings. Interested readers can refer to Robbins [83], Chung [84], Polyak and Juditsky [85] for more details on stochastic approximation.

On the other hand, in the stochastic setting, Kleinberg et al. [17] proposed a so-called "Follow the Awake Leader" strategy in which, the algorithm chooses at one round, the best expert among available ones to follow. At each round, the best expert is defined by the one obtaining the best average performance over his votes until that round. They obtained a nearly optimal bound up to a logarithmic factor. Compared to our algorithm, theirs does not directly address the adversarial settings mentioned in the introduction. Truong et al. proved that the algorithm in [19] asymptotically converges to the best expert (if there exists only one such expert) defined by product of his accuracy and availability [86]. However, the algorithm in [86] assumes symmetric availability for the experts, which may not hold true in some practical applications. Kanade et al. [18] proposed an exponential weighted algorithm (EWSA) for the full-information setting, and Bandit Sleeping Follow the Perturbed Leader (BSFPL) algorithm for the bandit setting when availability of the experts is stochastic but their predictions are adversarial. Their algorithm obtains an upper bound on regret comparable to [19]. However, the setting in their work is differently defined from ours.

Recently, Yu et al. proposed a multiplicative update rule using constant multipliers for available experts [26]. They considered an adversarial scenario and imposed strong assumptions on the proportion of good objects and the number of experts with the same taste as the user in order for their algorithm to converge. In our setting, that assumption is no longer needed, and we also allow negative voting as opposed to [26]. Moreover, under the same research thrust, [30] and [87] studied the structure of optimal strategies for malicious experts aiming to degrade the performance of a recommendation system.

Adversarial strategies in learning with expert advice

In this part of the dissertation, we consider an attacking model against the weighted average algorithm introduced by Littlestone and Warmuth [1] and Freund and Schapire [80]. The attacking model considered here falls into the causative attack from the taxonomy of adversarial machine learning [88–90], where the attacker can modify the data in the training set in order to degrade the performance of machine learning algorithms. The attack against recommendation systems that we mentioned in the example above is Sybil attack [22] where the adversary forges multiple identities to subvert these systems. The effect of this attack has been investigated recently on other systems: online social networks [91], rating systems [92], and mobile adhoc networks [93]. While there have been some works to diminish such effects, especially on recommendation systems [26, 94], they mostly need strong assumptions on the learning system such as ordering of voting or percentage of good movies. We refer the readers to [95], and [96] for other examples of adversarial attacks in signature generation system and email spam system, respectively. The readers can also refer to security risk related to adversarial machine learning in [97–108]. Beside machine learning systems, other systems are also vulnerable to attacks: multimedia [109, 110], network scheduling [111–115], fingerprinting [116–119], message encryption and recovery [120], information leak in covert channels [121] or time channel [122–125], traffic analysis [126], secure network cloud [127], attacks on telephone network [128], network flow [129–133], user privacy [134], covert channel [135–137], website attacks [138, 139].

Perhaps, the most related works to ours are the ones by Cover [29] and Gravin et al. [140]. Cover studied the adversarial sequential prediction of binary sequences in the 2-experts setting and found the optimal strategy for the adversary [29]. A related adversarial setting was recently introduced by Abernethy et al. [141] and Gravin et al. [140]. Abernethy et al. [141] proposed optimal strategies for both adversary and algorithm for the Gambler-Casino game in which the Gambler has some budgeted loss constraints and aims to minimize the accumulated loss on his bets. Gravin et al. [140] also investigated the same adversarial setting but without constraints. They attempted to find the optimal strategy for an adversary who controls the sequence of experts' losses, for all the N experts. They were able to find the optimal strategy for the adversary when $N = 2, 3$, but were not able to extend their results for general N.

In our work, we applied policy iteration to find the optimal solutions for our problem formalized as an MDP. The readers can refer to [142], [143], [144] and [145], [146] for general dynamic programming approaches to solve an MDP. Policy iteration has been used to solve an MDP given the predictable structures of optimal value functions Lin and Kumar [147], Walrand [148], Koole [149], Larsen [150], Puterman and Shin [151], vanNunen [152]. However, in their settings, the cost functions are either in linear or quadratic forms which provides strong support for their analysis.

In one of our main results, mean-field approach is used to reduce the complexity of the experts system. We refer the readers to Lasry and Lions [153], Guéant et al. [154], Kadanoff [155] for more details on this method.

Selective labeling

Active learning has been extensively studied recently [156], [157], [158]. Settles [159] introduced an excellent literature survey of framework overview and practical applications. Two approaches have been researched in this framework. In the first direction, the focus is on exploitation of decision boundary, for example uncertainty sampling [160], minimization error reduction [161] and variance reduction [162]. Recently, in the other direction, Baram et al. [163], Osugi et al. [164], and Bouneffouf [165] proposed the random exploration method in order to discover potentially good data points for querying. Their setting is different from ours in the sense that they attempt to select which examples for labeling from a pool of options while we tackle the online active learning problem where all examples are not given at the decision time. For more details on online active learning, we refer the readers to Sculley [166], Dasgupta et al. [167], Helmbold and Panizza [168], Freund et al. [169], Olsson [170]. Moreover, our main concentration is to efficiently label examples on the framework of learning from expert advice.

Recently, there has been a large amount of work in limited information setting for this framework. Auer et al. [171] proposed the so-called ‘partial information setting’ where only prediction of the selected expert is revealed in each round. Kale [172] and Seldin et al. [173] considered the limited experts advice in the multiarmed bandit setting. Lugosi considered the setting with limited feedback [174]. In this part of the dissertation, we consider the problem of label efficient, first termed by Cesa-Bianchi et al. [175], where the number of labeled examples is limited. Perhaps, the most relevant work for this setting is [175] and the work of Zhao et al. [176]. Cesa-Bianchi et al. [175] proposed a randomized seleting mechanism to select an object for labelling with a budget limit on the number of queries. Specifically, they did a simple flip-a-coin algorithm based on the limited query rate and obtained the upper bound of the regret depending on that rate. However, their algorithm depends on the number of queries which must be known in advance as a parameter. On another hand, Zhao et al. [176] proposed a so-called confidence condition to check when an object should be labeled. In particular, given a threshold, a sample is selected if the maximal difference of experts’ predictions is beyond the threshold, meaning that the disagreement between experts is large enough to make a query on that object. However, the choice of threshold in their setting is not obvious and the proposed regret bound of the performance is between the loss of algorithm over the requested time with the loss of the best expert over the whole horizon, which is not widely applicable for this setting. Moreover, their regret upper bound increases when the number of queries increases which is intuitively unexpected. In our setting, we use the maximal difference of experts’ predictions as the parameter in each round to decide if the query is necessary. Moreover, we also derive an upper bound for the expected regret defined by the difference of the algorithm’s loss and that of the best expert on the same horizon.

Simple regret in sleeping multiarmed bandit

Multiarmed bandit problem has been widely studied in the literature, [177–182]. The problem of selecting one object among a set, also known as trial design, was first mentioned by Paulson [183] and Bechhofer [184]. Robbins [185] and Gittins [186] later considered the renowned multiarmed bandit settings for this problem. Recently, the "best arm selection" problem was formally introduced in Bubeck et al. [187] with the so-called "pure exploration" framework. In this work, they proposed many variants of UCB based algoithms (which chooses the arm with highest index defined by the summation of emperical mean of the arm and the confidence interval) and uniform algorithms, and prove that the recommendation errors decay to zero when time is very large. The UCB algorithms had been proposed by Auer et al. [188], and later on Kleinberg et al. [17], but their purpose is to minimize the accumulated loss of the whole procedure. On another the hand, the work in [187] attempts to minimize the simple regret defined by how good the algorithm can recommend an arm at the end of the process. Audibert et al. [189] improved the error rate in [187] by using an appropriate choice of the parameter. Those algorithms concentrate on one of the settings of the best arm selection problem where the number of samplings is limited. In another setting, Gabillon et al. [190], Maron and Moore [191], Mnih et al. [192] proposed algorithms for the fixed confidence setting where the purpose is to minimize the number of samplings given a certain error rate. Jamieson et al. [193] later applied a stopping time algorithm to avoid the union bound in the error encountered by most of the previous work. There have been other works on this setting including successive elimination algorithms Audibert et al. [189], Mannor and Tsitsiklis [194], Even-Dar et al. [195] and selecting m-best arms Bubeck et al. [196], Kalyanakrishnan and Stone [197], Kalyanakrishnan [198]. Thus far, there has not been work addressing the situations when there is only a subset of arms available at a time. We will focus on this setting.

1.4 Problem notations and definitions

We introduce herein definitions and notations that will come handy later in the analysis.

1.4.1 Experts setting

Let $E = \{1, 2, ..., N\}$ be the set of all experts. We denote the set of available experts at round t by E_t, where $E_t \subseteq E$. Note that, round, time instance, object are interchangeably used in this thesis. For example, when we say "given an object", we mean the round at which the object occurs. At round t, expert i's weight is p_t^i. Often used later in this thesis are the notations of normalized weight, $\tilde{p}_t^i = \frac{p_t^i}{\sum\limits_{i \in E} p_t^i}$, and weight vector, $\vec{p}_t = (p_t^1, p_t^2, ..., p_t^N)$. This weight vector are updated through an update rule, based on how well experts have performed. The higher the weight of an expert, the more influence he can affect on the prediction of the algorithm.

Definition 1. *The true outcome (or outcome, ground truth) of an object is the true feedback from a specific user to an object.*

The outcome is sometimes referred as the label, and is denoted by $y_t \in \mathcal{Y}$. For example, the outcome of a movie in the binary setting is either *Good* or *Bad*. Examples of an object include a movie, a story or a book.

Definition 2. *The prediction value of a system (expert) is the value that the system (expert) predicts on a given object.*

The prediction of the system and expert i at time t is denoted by $\hat{y}_t \in \mathcal{Y}$ and $x_t^i \in \mathcal{X}$, respectively. After all experts provide their predictions on an object, the algorithm computes the averaged prediction on that object. Upon receiving the outcome for that object, the algorithm updates the losses of the experts and the algorithm.

Definition 3. *The loss function is a function that measures the difference between the prediction value and the outcome, i.e.,* $l(.,.) : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}^+$.

The loss of the algorithm and expert i is denoted as $l(\hat{y}_t, y_t)$ and $l(x_t^i, y_t)$, respectively. In Chapter 3, we focus on two kinds of losses:

- Logarithmic Loss: $l(y_t, \hat{y}_t) := -\mathbb{I}\{y_t = 1\} \ln(\hat{y}_t) - \mathbb{I}\{y_t = 0\} \ln(1 - \hat{y}_t)$.
- Absolute Loss: $l(y_t, \hat{y}_t) = |y_t - \hat{y}_t|$.

Above, $\mathbb{I}\{\}$ is the indicator function.

Definition 4. *The best expert over a time horizon T is the one who incurs the least loss over horizon T, i.e.,*

$$i* = \arg\min_{i \in E} \sum_{t=1}^{T} l(x_t^i, y_t).$$

In the learning with expert advice framework, we would like to see how close the performance of the algorithm to that of the best expert. Regret is a commonly used term in this setting.

Definition 5. *Regret of the algorithm, with respect to the best expert, is the difference of total loss of the algorithm and the total loss of the best expert over a horizon T.*

$$R_T = \sum_{t=1}^{T} l(\hat{y}_t, y_t) - \min_{i \in E} \sum_{t=1}^{T} l(x_t^i, y_t)$$

Similarly, regret of the algorithm, with respect to the expert i, is the difference of total loss of the algorithm and the total loss of the best expert i over a horizon T.

$$R_T^i = \sum_{t=1}^{T} l(\hat{y}_t, y_t) - \sum_{t=1}^{T} l(x_t^i, y_t)$$

1.4.2 Multiarmed bandit setting

In this setting, we denote $\mathbb{S} = \{1, 2, ..., K\}$ as the set of K arms, and $\mathbb{S}_t \subseteq \mathbb{S}$ as the set of available arms at time t. For the stochastic setting, we have the following assumptions.

Assumption 1. *The set of available arms, $\mathbb{S}_t$, is drawn from a fixed, but unknown distribution. The reward of each arm i is drawn from a fixed, but unknown, distribution with the mean μ_i.*

For simplicity, we assume that all rewards are bounded in $[0, 1]$. Without loss of generality, we assume that $\mu_1 \geq \mu_2 \geq ... \geq \mu_K$, i.e., the set of arms have been already sorted in the descending order of their mean. At time t, denote $\mu_t^* = \max_{i \in \mathbb{S}_t} \mu_i$ as the current best arm.

Definition 6. *Mean gap of two arms i and j is the difference of mean values between the two arms,*

$$\Delta_{i,j} := \mu_i - \mu_j.$$

This term takes a crucial role in conducting our simple regret analysis in the sequel.
To simplify the analysis later, we introduce the following notations. Denote $S^i = \{S : i \in S \text{ and } i \leq j \; \forall j \in S\}$ as the collection of subsets which have i as the best arm, and $\mathbb{T}^i = \{t : i \in S_t \text{ and } S_t \in S^i\}$ as the collection of times that arm i is the best available arm. We also denote t_i, t_{ij} as the final time in $\mathbb{T}^i$ and the final time within $\mathbb{T}^i$ that arm j is chosen instead of i, respectively. Define K_i as the total number of available arms in the set $\mathbb{T}^i$. We note that $|\mathbb{T}^i| = Tq_i$, where q_i is the probability that arm i is the leading arm of any subset. We abuse the notation a bit by denoting $T_j^i(t)$ as total number of times the arm j is chosen up to time t whenever the arm i is the best available arm.

CHAPTER 2

LEARNING FROM SLEEPING EXPERTS

We consider a generalized model of learning from expert advice in which experts could abstain from participating at some rounds. Our proposed online algorithm falls into the class of weighted average predictors and uses a time varying multiplicative weight update rule. This update rule changes the weight of an expert based on his relative performance compared to the average performance of available experts at the current round. We prove the convergence of our algorithm to the best expert, defined in terms of both availability and accuracy, in the stochastic setting, and justify by experimental results the out-performance of our proposed algorithms compared to the existing ones in the literature.

2.1 Preliminaries and Proposed Algorithm

In this section, we introduce the problem setup and notations which will be used in subsequent sections. Let $E = \{1, 2, ..., N\}$ be the set of all experts. We denote the set of available experts at round $t = 0, 1, 2, \ldots$ by E_t, where $E_t \subseteq E$. At round t, expert i's weight is $p_t^i \in [0, 1]$, and his prediction on a given object is $x_t^i \in [0, 1]$. The true outcome, or user's feedback, of the given object is denoted as y_t, which is an adversarial binary $\{0, 1\}$ sequence.

Our proposed algorithm to aggregate experts' opinion is given in Algorithm 1. It computes a weighted average of the predictions of the available experts at each round, as shown in (2.1). Once the outcome is revealed, weights of experts are updated as in (2.2). This update rule can be rewritten as:

$$p_t^i = p_{t-1}^i + a(t)p_{t-1}^i \left[I\{i \in E_t\}(r_t^i - 1/2) - \sum_{j \in E} I\{j \in E_t\} p_{t-1}^j (r_t^j - 1/2) \right], \tag{2.4}$$

where r_t^i is defined by (2.3), and $I\{i \in E_t\}$ is the indicator function for availability of expert i, i.e., $I\{i \in E_t\} = 1$ if $i \in E_t$, and $I\{i \in E_t\} = 0$, otherwise. r_t^i may be interpreted as the accuracy of expert i in the sense that, a high value of r_t^i corresponds to an accurate prediction, i.e., one that is close to the outcome, for expert i. $a(t)$ is a decreasing step

Algorithm 1

```
  **Input:** Set of expert $E = \{1, ..., N\}$
  **Initialize:** $p_0^i = 1/N$ for i = 1,...,N.
  **for** each round $t = 1, 2, ...$ **do**
    Nature chooses an object.
    **Prediction:**
    Each expert $i$ predicts $x_t^i$, for $i \in E_t$.
    Algorithm predicts $\hat{y}_t$,
        $$\hat{y}_t = \frac{\sum_{i \in E_t} p_{t-1}^i x_t^i}{\sum_{i \in E_t} p_{t-1}^i}. \qquad (2.1)$$
    Nature reveals the outcome $y_t$.
    **Update:**
    Algorithm updates weights of all experts. Each weight is updated by
        $$p_t^i = \begin{cases} p_{t-1}^i + a(t) p_{t-1}^i \left[ (r_t^i - 1/2) - \sum_{j \in E_t} p_{t-1}^j (r_t^j - 1/2) \right] & \text{if } i \in E_t, \\ p_{t-1}^i - a(t) p_{t-1}^i \sum_{j \in E_t} p_{t-1}^j (r_t^j - 1/2) & \text{if } i \notin E_t, \end{cases} \qquad (2.2)$$
    where $r_t^i$ is defined by
        $$r_t^i := I\{y_t = 1\} x_t^i + I\{y_t = 0\}(1 - x_t^i). \qquad (2.3)$$
  **end for**
```

size such that $\sum_{t=0}^{\infty} a(t) = \infty$, and $\sum_{t=0}^{\infty} a^2(t) < \infty$ (more details in Section 2.2.1), e.g., $a(t) = \frac{1}{1+t}$. We denote the term between brackets of (2.4),

$$I\{i \in E_t\}(r_t^i - 1/2) - \sum_{j \in E} I\{j \in E_t\} p_{t-1}^j (r_t^j - 1/2)$$

as the information innovation. It captures the informative value of expert i's prediction at the current time. Therefore, the update rule of (2.4) rewards not only an accurate prediction (high value of r_t^i) but also the information value of such a prediction in terms of its deviation from the average prediction of available experts at each time. This captures the fact that if an instance is hard to predict, a correct expert must be rewarded more than when the instant is easy to predict (as everyone in an easy instance may predict correctly).

2.2 Convergence Analysis of the Algorithm

Let us define the availability and the accuracy of expert i at instant t by $I\{i \in E_t\}$ and r_t^i, respectively. To have a precise definition of best expert, we consider the following assumption throughout the paper.

Assumption 2. *We assume that the process $\{I\{i \in E_t\}(r_t^i - \frac{1}{2}), t = 0, 1, \ldots\}$ is weakly stationary for each expert i meaning that $\mathbb{E}[I\{i \in E_t\}(r_t^i - 1/2)]$ does not depend on time t.*

Intuitively, Assumption 2 implies that the expected chance that an expert votes on an instance and predicts correctly is a constant. Based on this definition, we now define the best expert as follows:

Definition 7. *The best expert is defined as*

$$i^* = \arg\max_{i \in E} \mathbb{E}[I\{i \in E_t\}(r_t^i - 1/2)], \tag{2.5}$$

where the expectation is taken over the randomization of experts' accuracy and availability.

Essentially, Definition 7 states that the best expert is the one achieving the highest expected performance in terms of both availability and accuracy over the decision horizon. Note that Algorithm 1 penalizes reliable experts who do not vote frequently in order to prevent them from earning high weights by employing a safe voting strategy (i.e., voting only on easy instances or the instances which already have enough votes to determine their quality). For instance, in the case of movie recommendation, a critic should not be rewarded with a high weight just because he voted favorably for well-known excellent movies or he voted against all-time flops. Also note that the algorithm does not solely applaud the experts aiming to be present but with very low accuracy (at the level of random guess). One of our immediate goals is to show that Algorithm 1 converges to the best expert defined as in Definition 7.

2.2.1 Convergence Analysis

Herein, we address the question of whether the algorithm can asymptotically recognize the best expert and follow him. To answer this, let us examine the evolution of weights of all experts to see if the best expert's weight indeed dominates the other weights in the long run.

Let us denote $\vec{p}_{t-1}$ as the vector of weights for all experts at time $t-1$, i.e., $\vec{p}_{t-1} = \{p_{t-1}^i\}_{i=1}^N$, and $\vec{\xi}_t$ as $\vec{\xi}_t = \{I\{i \in E_t\}r_t^i\}_{i=1}^N$, which is a collection of the products of availability and accuracy of the experts. Rewrite the weight update rule of (2.4) as

$$p_t^i = p_{t-1}^i + a(t) f_i(\vec{p}_{t-1}, \vec{\xi_t}), \tag{2.6}$$

where

$$f_i(\vec{p}_{t-1}, \vec{\xi_t}) := p_{t-1}^i \left[I\{i \in E_t\}(r_t^i - 1/2) - \sum_{j \in E} I\{j \in E_t\} p_{t-1}^j (r_t^j - 1/2) \right].$$

Let F_{t-1} denote the history of predictions and presence of all experts up to time $t-1$, i.e.,

$$F_{t-1} := \{\{x_\tau^i\}_{i=1}^N, \{I\{i \in E_\tau\}\}_{i=1}^N, \text{ for } \tau = 1, \ldots, t-1\}.$$

Define $h_i(\vec{p}_{t-1})$ as,

$$h_i(\vec{p}_{t-1}) := \mathbb{E}[f_i(\vec{p}_{t-1}, \vec{\xi_t}) | F_{t-1}],$$

where $\mathbb{E}[\cdot | F_{t-1}]$ is the conditional expectation given the past history. Note that by Assumption 2, $\mathbb{E}[f_i(\vec{p}_{t-1}, \vec{\xi_t}) | F_{t-1}]$ is only a function of $\vec{p}_{t-1}$. We also define M_t^i as

$$M_t^i := f_i(\vec{p}_{t-1}, \vec{\xi_t}) - \mathbb{E}[f_i(\vec{p}_{t-1}, \vec{\xi_t}) | F_{t-1}].$$

Then we can rewrite equation (2.6) as

$$p_t^i = p_{t-1}^i + a(t)[h_i(\vec{p}_{t-1}) + M_t^i], \tag{2.7}$$

where $\{M_t^i\}$ is a martingale difference sequence. In particular, since it is uncorrelated with the history of predictions and availabilities of experts, we can consider it as a noise. Stacking all the equations of (2.7) for $i = 1, \ldots, N$ in a vector form, we get

$$\vec{p}_t = \vec{p}_{t-1} + a(t)[h(\vec{p}_{t-1}) + M_t], \ t = 0, 1, \ldots \tag{2.8}$$

where $h(\cdot) = (h_1(\cdot), \ldots, h_N(\cdot))$, and $M_t = (M_t^1, \ldots, M_t^N)$. Equation (2.8) is commonly used to define the state update in a dynamical system. In this formulation, the state is incremented by a function of past states and an exogenous noise multiplied by a decreasing step size. It is shown in [199, Theorem 2] that under appropriate conditions, the solution to the difference equation of (2.8) asymptotically approaches the solution to an ordinary differential equation (ODE) given by $\dot{\rho}(s) = h(\rho(s)), s \in \mathbb{R}^N$, with identical initial condition $\rho(0) = \vec{p}_0$. The required conditions are:

- **(A1)** Function $h(\cdot)$ is Lipschitz, i.e., there exists a positive constant L such that

$\|h(\vec{p}) - h(\vec{p}')\| \leq L\|\vec{p} - \vec{p}'\|$, for any $\vec{p}$ and $\vec{p}'$.

- **(A2)** Step size $a(t)$ is not summable but squared summable, i.e.,

$$\sum_{t=0}^{\infty} a(t) = \infty \text{ and} \sum_{t=0}^{\infty} a(t)^2 < \infty.$$

- **(A3)** $\{M_t\}$ is a martingale difference sequence[1] such that $\mathbb{E}\left[M_t^2|F_{t-1}\right] \leq K(1+\|\vec{p}_{t-1}\|^2)$ for some positive constant K.

- **(A4)** $\sup_t \|\vec{p}_t\| < \infty$, almost surely.

Theorem 1. *[199, Theorem 2] Almost surely, the sequence $\{\vec{p}_t\}$ generated by $\vec{p}_{t+1} = \vec{p}_t + a(t)[h(\vec{p}_t) + M_{t+1}]$ converges to a compact connected internally chain transitive invariant set of $\dot{\rho}(s) = h(\rho(s))$, where $\rho(0) = \vec{p}_0$.*[2]

The key idea in establishing Theorem 1 is the fact that the discretization error and the effect of noise tend to be zero asymptotically. Specifically, since the step size $a(t)$ tends to zero when t goes to infinity, the discretization error is negligible. Also, the effect of noise is asymptotically reduced since $\{M_t\}$ is bounded and the series $\sum\limits_{t=0}^{n} a(t)M_t$ converges. Applying Theorem 1, the following lemma is immediate.

Lemma 1. *For $i = 1, \ldots, N$, almost surely the sequence p_t^i given by (2.7) tracks the trajectory of the following ODE:*

$$\dot{\rho}^i(s) = h_i(\rho(s)) = \rho^i(s)\left(c_i - \sum_{j\in E} c_j \rho^j(s)\right), \quad \rho(0) = \vec{p}_0, \tag{2.9}$$

where $\rho(s) = (\rho^1(s), \ldots, \rho^N(s))$, and $c_i := \mathbb{E}[I\{i \in E_t\}(r_t^i - 1/2)]$, $i = 1, \ldots, N$.

Proof. See Appendix A.1.1. □

We now are ready to state our main convergence result.

Theorem 2. *If there exists only one best expert defined by Definition 7, then Algorithm 1 will converge to him. If there is more than one expert satisfying Definition 7, then Algorithm 1 will alternate between them.*

Proof. See Appendix A.1.2. □

[1] That is $\mathbb{E}[M_t|F_{t-1}] = 0$, a.s., for $t \geq 0$.

[2] A closed set A is said internally chain transitive if for any pair x and y in A, there exist a set of points in A such that the trajectory given by the solution to the ODE starts from x, passes through those points to y after some certain amount of time.

2.3 Alternative algorithm in sleeping-expert setting

In this section we consider a slight variant of Algorithm 1 for the sleeping expert problem. So far, an expert incurs some loss when he is not available. In this section, we assign a constant loss for a sleeping expert and see how it changes the performance of our algorithm under the stochastic setting assumptions given in Section 2.2. Since in the adversarial settings such as the ones mentioned in Section 2.1, experts might be intentionally absent from voting, we penalize non-voting experts at one round by assigning them a constant vote and hence a loss. Specifically, when an expert i does not vote in one round, we assume that his vote was a constant value $c \in [0, 1]$, i.e.,

$$z_t^i = \begin{cases} x_t^i & \text{if } i \in E_t, \\ c & \text{if } i \notin E_t. \end{cases} \tag{2.10}$$

Now we can compare all experts based on their expected losses since they have recommendations at each round regardless of their presence. Denote $l(.)$ as a bounded loss function. From (2.10), the loss of expert i at time t is given by

$$l(z_t^i) = I\{i \in E_t\} l(x_t^i) + I\{i \notin E_t\} l(c).$$

The expected loss of expert i is then computed by

$$\mathbb{E}(l(z_t^i)) = \mathbb{E}\left[I\{i \in E_t\} l(x_t^i) + I\{i \notin E_t\} l(c) \right]. \tag{2.11}$$

Note that the expectation is taken over the randomization of availability and accuracy of experts.

Definition 8. *The best expert is defined as the one who has the least expected loss over all experts, i.e.,*

$$\begin{aligned} i^* &= \arg\min_{i \in E} \mathbb{E}(l(z_t^i)), \\ &= \arg\min_{i \in E} \mathbb{E}\left[I\{i \in E_t\} l(x_t^i) + I\{i \notin E_t\} l(c) \right]. \end{aligned} \tag{2.12}$$

Algorithm 2 describes a prediction framework where a missing vote is treated as in (2.10).

Note that Algorithm 2 essentially differs from Algorithm 1 only in the weight update rule given by (2.13) and (2.14). In the following, we show that for the absolute loss function, the Definition 7 of the best expert coincides with that of Definition 8 (which is a more natural definition under this ‘all-awake-experts’ setting). To see that, define the absolute loss of

Algorithm 2

```
Input: Set of expert $E = \{1, ..., N\}$
Initialize: $p_0^i = 1/N$ for i = 1,...,N.
for each round $t = 1, 2, ...$ do
    Nature chooses an object.
    Prediction:
    Each expert $i$ predicts $x_t^i$, for $i \in E_t$.
    Algorithm predicts $\hat{y}_t$,
        $$\hat{y}_t = \frac{\sum_{i \in E} p_{t-1}^i x_t^i}{\sum_{i \in E} p_{t-1}^i}.$$
    Nature reveals the outcome $y_t$.
    Update:
    Algorithm updates weights of all experts. Each weight is updated by
        $$p_t^i = p_{t-1}^i + a(t) p_{t-1}^i \left[ u_t^i - \sum_{j \in E} p_{t-1}^j u_t^j \right], \quad (2.13)$$
    where $u_t^i$ is defined by
        $$u_t^i = \begin{cases} I\{y_t = 1\} x_t^i + I\{y_t = 0\}(1 - x_t^i) & \text{if } i \in E_t, \\ I\{y_t = 1\} c + I\{y_t = 0\}(1 - c) & \text{if } i \notin E_t. \end{cases} \quad (2.14)$$
end for
```

expert i at time t as $l(x_t^i) = |y_t - x_t^i|$. By the definition of u_t^i in (2.14), we observe that

$$u_t^i = \begin{cases} 1 - l(x_t^i) & \text{if } i \in E_t, \\ 1 - l(c) & \text{if } i \notin E_t. \end{cases} \tag{2.15}$$

Therefore, we will have the following corollary:

Corollary 1. *Algorithm 2 converges to the best expert defined by Definition 8.*

Proof. See Appendix A.1.3. □

The value of constant c is chosen based on the degree that the algorithm wants to penalize the absent experts. For example, for a "non-strict" algorithm, c is chosen to minimize the expected losses of the experts. As we shall see soon through experimental results, with some appropriate choice of c, the algorithm can obtain high performance compared to the other existing algorithms for expert advice problem.

Table 2.1: Availability and accuracy of experts.

Expert	Availability	Accuracy
1	0.95	0.95
2	0.9	0.9
3	0.8	0.8
4	0.7	0.7
5	0.6	0.7
6	0.5	0.9
7	0.4	0.6
8	0.3	0.5
9	0.2	0.6
10	0.1	1

2.4 Experimental Results

In this experiment, we compare the performance of our proposed algorithms to other algorithms in this sleeping expert setting. We run algorithms on both synthetic dataset and real dataset (Netflix) to prove that our algorithms not only works in the stochastic setting but also in a more general case without any stochastic assumption. First, we consider a synthetic data set consisting of recommendations for objects from 10 experts in 1000 rounds, during which some experts could abstain from voting. The predictions of experts take values in $[0, 1]$, while the outcomes (objects) are binary $\{0, 1\}$ generated from a Bernoulli distribution with parameter 0.5. Each expert votes only when he is present, frequency of which depends on his availability. The prediction of an available expert depends on his accuracy. We simulate over the set of availability and accuracy given in table 2.1.

To define the accuracy of an expert, we define a tolerance ρ as follows. Expert i is considered to have a correct prediction if his prediction lies within a distance ρ from the outcome, i.e., $|x_t^i - y_t| \leq \rho$. The accuracy of expert i is then defined by the percentage of time that his recommendations are correct, and is denoted by μ^i. For example, if an expert i votes in a system with $\rho = 0.3$ for 100 rounds and his accuracy $\mu^i = 80\%$, it implies that 80 of his reccommendations satisfy $|x_t^i - y_t| \leq 0.3$. The value of ρ was chosen to be 0.3 in this simulation.

Let $A1$ and $A2$ represent our proposed Algorithm 1 and Algorithm 2, respectively, with decreasing step size, $\frac{1}{1+t}$. We also investigated performance of Algorithm 1 for a fixed step size. Specifically, $A1\_001$ and $A1\_05$ are two other versions of Algorithm 1 when the step size is set equal to 0.01 and 0.5, respectively.

We first compare the loss of the proposed algorithms with those of other algorithms:

Dsybil [26] and *SBayes* [19].
In *Dsybil*, there are two classes of objects: overwhelming and non-overwhelming. An object is identified as overwhelming if sum of the weights of experts voting for it exceeds a threshold th, otherwise, the object is non-overwhelming. Experts only vote for an object if they believe it is good. If expert i's prediction for a non-overwhelming object is correct, his weight is increased by $w_t^i = w_{t-1}^i \alpha$, where $\alpha > 1$. Correct votes for an overwhelming object do not result in weight change since the votes are not important. Whenever an expert i votes for a bad object, his weight is decreased by $w_t^i = w_{t-1}^i \beta$, where $\beta < 1$. In this simulation, we chose $\alpha = 5, \beta = 0.1, th = 11$ to optimize $Dsybil's$ performance.
SBayes [19] is the weight update rule which keeps the weights of sleeping experts unchanged. One difficulty in comparing these algorithms is that the loss definition of each algorithm differs from the others. Therefore, we use a unified common definition of loss which is defined as the total number of mistakes the algorithm makes. In other words, the prediction of the algorithm is quantized to a binary value and is compared with the outcome.

Since *SBayes* uses the logarithmic loss function, for a more fair comparison, we also add another algorithm, *SBayes_abs* which is an adaptation of *SBayes* when the loss is an absolute function.

Figure 2.1 depicts losses incurred by the above algorithms when the constant c of $A2$ is chosen as 0.2. It is shown that $A1$ and $A1\_001$ suffers the least loss while *Dsybil* incurs the most loss and *SBayes*, *SBayes_abs* are in between. In our simulations, we assumed that a given object to be rated is equally likely to be good or bad, i.e., outcomes 0 and 1 are equiprobable. Since in *Dsybil*, good experts only vote for good objects, this algorithm must rely on experts that have not performed well when a bad object is considered. Consequently, it might suffer much loss due to these non-performing experts' predictions. This is an inherent flaw of the algorithm *Dsybil*.

Compared to *SBayes* and *SBayes_abs*, $A1$ converges slower to the best expert since it uses a decreasing step size in the update rule. This slowness in convergence is deliberate. In *SBayes* and *SBayes_abs*, the quick convergence to the best expert means that weights of other experts are decreased quickly. Therefore, when the best expert is not available to vote (is sleeping), the algorithm has to choose among experts that all have small weights. The same is not true for $A1$. When the best expert goes to sleep alternative "nearly best experts", which have higher weights than the corresponding $SBayes'$ experts, are available to vote and help the performance of the algorithm.

The fixed step size versions of Algorithm 1 act differently depending on the value of step size. As expected, these algorithms converge faster than $A1$ which uses a decreasing step size. In fact, for the experimental setup of table 2.1, $A1\_001$ behaves approximately the same

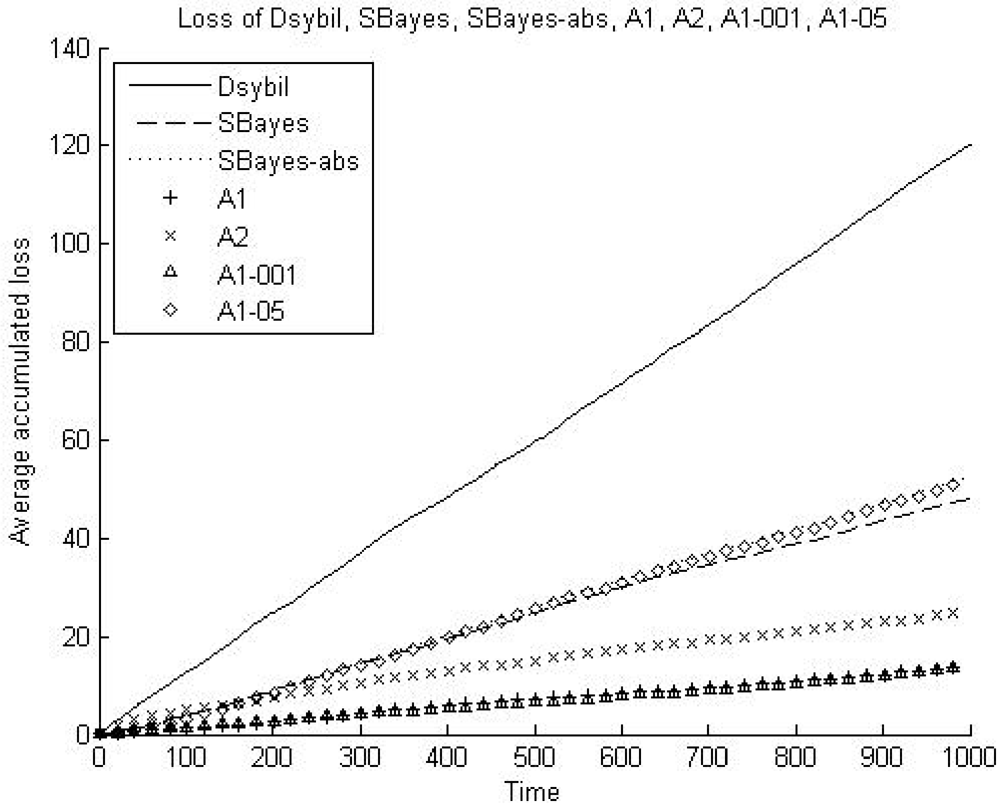

Figure 2.1: Comparison of loss of *Dsybil*, *SBayes*, *SBayes_abs*, $A1$, $A2$, $A1\_001$, and $A1\_05$ when $c = 0.2$.

as $A1$ when the constant step size is set small ,0.01, while $A1\_05$ behaves approximately the same as *SBayes_abs* and *SBayes* when the step size is increased to 0.5. The performance of $A2$ varies with the choice of value c. In particular, when c is chosen to be 0.2, its performance is not comparable to $A1$, as showed in Figure 2.1. We change the constant c to find the value at which $A2$ can improve its performance. Figure 2.2 illustrates such a case when $c = 0.5$, in which $A2$ outperforms all other algorithms.

Also in this setting, the weights evolutions of algorithms are investigated. Figure 2.3 shows the weights evolutions of all experts. Since *Dsybil* use multiplier $\alpha = 5$ and threshold $th = 11$, an expert's weight of this algorithm could go up to 55 if that expert has been rewarded from the weight roughly the threshold. Therefore, we normalize weights of *Dsybil* to obtain the fair comparison with other algorithms. It can be observed that while weights of *Dsybil* still fluctuate after long time, weights of *SBayes* converge much faster. As mentioned above, $A1$ converges more slowly than *SBayes*. However, the fixed step size algorithms can increase the convergence rate. Specifically, A_05 converges faster than $A1\_001$ which is faster

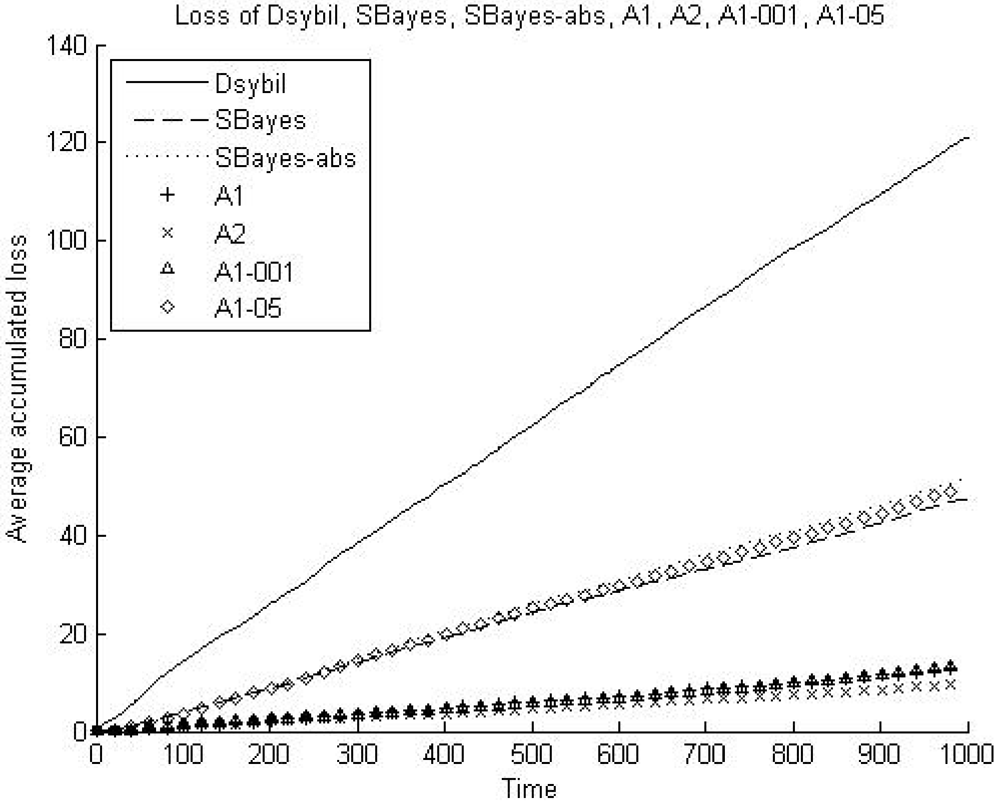

Figure 2.2: Comparison of loss of *Dsybil*, *SBayes*, *SBayes_abs*, $A1$, $A2$, $A1\_001$, and $A1\_05$ when $c = 0.5$.

than $A1$.

In the second part of the simulation, we compare the performance of algorithms on the Netflix dataset. For the purpose of comparing the algorithms' performance and reduce the running time, we only use a subset of this dataset, including 3153 experts, 14 movies and the voting period is within 2180 days. The predictions of experts are given in the normalized five-star scale $\{0.2, 0.4, 0.6, 0.8, 1\}$. The outcomes are obtained from feedback of an experienced movie consumer. Figure 2.4 shows the loss comparisons when the constant c of $A2$ is set equal to 0.2. In this figure, *Dsybil* again gets the poor performance while $A1$ and $A1\_001$ still outperform the rest ($A1$ is slightly better than $A1\_001$). In the experiment with this dataset, *SBayes* and *SBayes_abs* algorithms perform slightly worse than $A1$ and $A1\_001$ but still better than $A1\_05$. Note that the number of movies noticeably increases in the time period 1100 to 1500. Therefore, it is more likely that the best expert misses on voting for some of the movies during this time. Algorithms do not solely rely on the best expert in that case such that our algorithms do more favorably in this scenario due to the higher weights

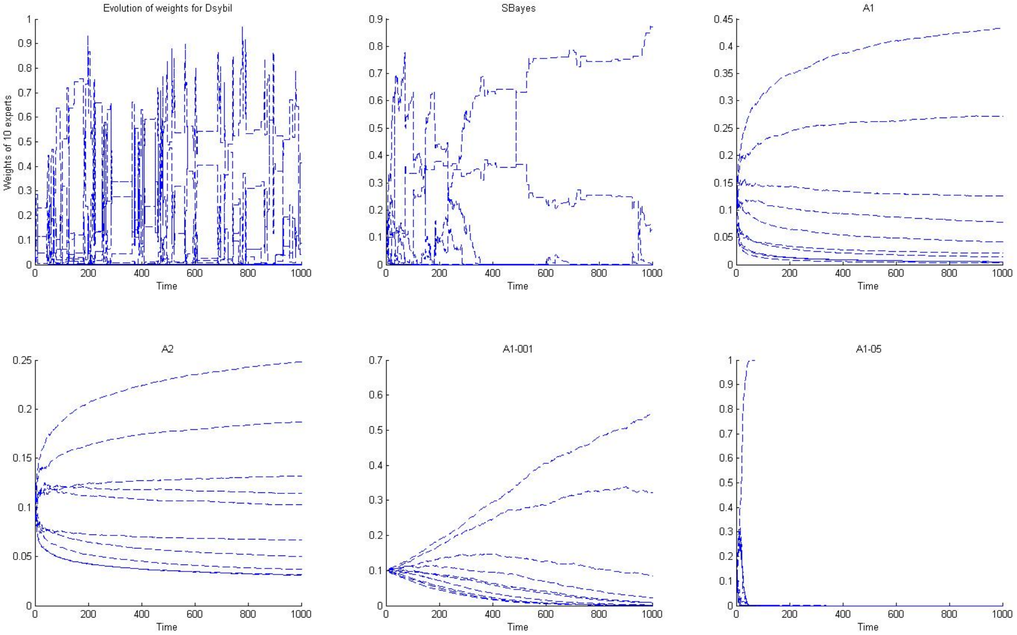

Figure 2.3: Experts' weights evolution of algorithms when $c = 0.5$.

for other good experts. Since for this dataset, there is no obvious way (at least after some runs with different values of c) to choose c the performance of $A2$ is not good as opposed to its performance on the synthetic data. It is slightly better than $Dsybil$, but not better than $A1\_05$ even when c is changed to a better chosen value, e.g., 0.8, as illustrated in Figure 2.5. It has been observed that while $A2$ achieves superior performance in stochastic settings with an appropriate choice of constant c, it does not practically seem to guarantee such a good performance in an adversarial setting, e.g., in Netflix dataset. In two cases, $A1$ and $A1\_001$ always outperform others.

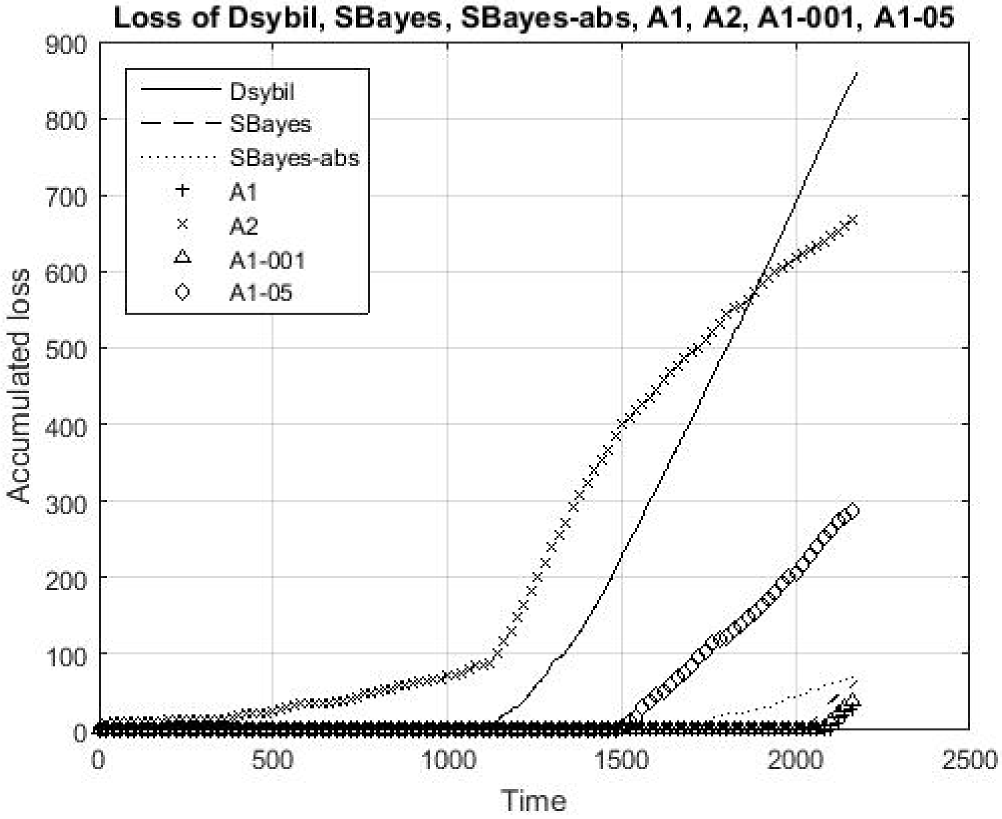

Figure 2.4: Comparison of loss of $Dsybil$, $SBayes$, $SBayes\_abs$, $A1$, $A2$, $A1\_001$, and $A1\_05$ when $c = 0.2$ and Netflix dataset is in used.

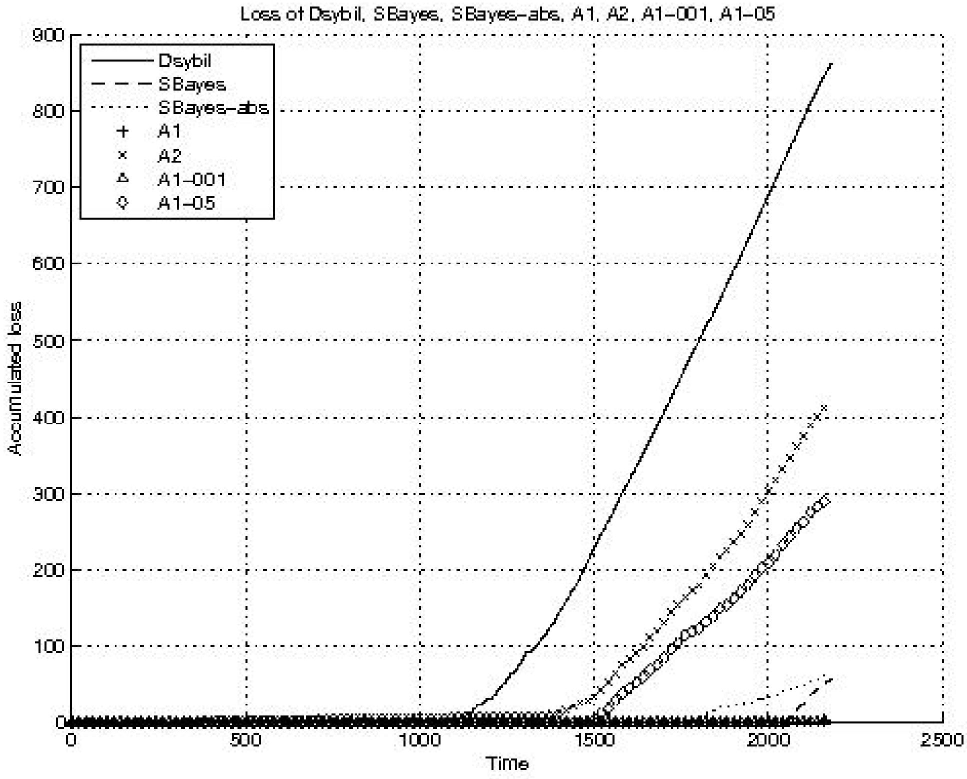

Figure 2.5: Comparison of loss of *Dsybil*, *SBayes*, *SBayes_abs*, $A1$, $A2$, $A1\_001$, and $A1\_05$ when $c = 0.8$ and Netflix dataset is in used.

CHAPTER 3

ADVERSARIAL ATTACKING STRATEGIES

In this chapter, we analyze optimal adversarial strategies against the weighted average prediction algorithm in the learning with expert advice framework. All but one expert are honest and the malicious expert's goal is to sabotage the performance of the algorithm by strategically providing dishonest recommendations. We formulate the problem as a Markov decision process (MDP) and analyze it under various settings with two kinds of losses: logarithmic loss and absolute loss.

3.1 Notations and Problem Formulation

Let $E = \{1, 2, ..., N\}$ be the set of experts. At round k, each expert i has a weight $p_{k-1}^i \in [0, 1]$. The prediction of expert i is denoted by $x_k^i \in \{0, 1\}$. Upon receiving the expert predictions, the algorithm calculates a weighted average prediction, $\hat{y}_k$. After the prediction is made, the outcome, denoted by y_k, is revealed. We assume the outcome is in $\{0, 1\}$. After the outcome is revealed, the algorithm incurs a loss $l(\hat{y}_k, y_k)$ and expert i incurs a loss $l(x_k^i, y_k)$. The algorithm updates the weights of experts based on the losses they incurred using a multiplicative update rule. The learning process is summarized by Algorithm 3.

In this paper, we only focus on two kinds of losses:

- **Logarithmic Loss**:
  $l(y_k, \hat{y}_k) := -\mathbb{I}\{y_k = 1\} \ln(\hat{y}_k) - \mathbb{I}\{y_k = 0\} \ln(1 - \hat{y}_k)$.

- **Absolute Loss**: $l(y_k, \hat{y}_k) = |y_k - \hat{y}_k|$.

Note that we can rewrite the logarithmic loss as $l(y_k, \hat{y}_k) = -\ln(1-|y_k-\hat{y}_k|)$, that will be convenient for later use. In this case, to avoid the loss function going to infinity, we slightly modify the binary predictions to $\{\epsilon, 1-\epsilon\}$, where ϵ is a small number. We let $\vec{p}_k = (p_k^1, p_k^2, ..., p_k^N)$ be the *state* or *weight vector* of all experts at round k, and $\vec{\tilde{p}}_k = (\tilde{p}_k^1, \ldots, \tilde{p}_k^N)$ be the corre-

Algorithm 3 The weighted average learning algorithm

```
**Initialize:** $p_0^i = 1$ for i = 1,2,...,N.
**for** each round $k = 1, 2, ...$ **do**
    Nature chooses an outcome.
    **Prediction:**
    Each expert $i$ predicts $x_k^i \in \{0, 1\}$. Algorithm predicts $\hat{y}_k$,
        $$\hat{y}_k = \frac{\sum_{i \in E} p_{k-1}^i x_k^i}{\sum_{i \in E} p_{k-1}^i}. \quad (3.1)$$
    Nature reveals the outcome $y_k \in \{0, 1\}$.
    **Update:**
    Each expert's weight is updated as:
        $$p_k^i = p_{k-1}^i e^{-l(x_k^i, y_k)}. \quad (3.2)$$
**end for**
```

sponding normalized weight vector where

$$\tilde{p}_k^i = \frac{p_k^i}{\sum_{i \in E} p_k^i}, \ i = 1, 2, \ldots, N \tag{3.3}$$

is the normalized weight of expert i. Note that we always have $\sum_{i \in E} \tilde{p}_k^i = 1, \forall k$.

Throughout this paper, we assume that expert i ($i \neq 1$) makes a correct prediction, i.e., one that agrees with the outcome, with probability μ_i (the accuracy of expert i). That is,

$$x_k^i = \begin{cases} y_k & \text{w.p} \quad \mu_i, \\ 1 - y_k & \text{w.p} \quad 1 - \mu_i. \end{cases} \tag{3.4}$$

Without loss of generality assume that the malicious expert is expert 1 and recall that all the other experts are honest. We assume expert 1 knows the prediction distribution of each expert i (this can be learned empirically from the history of predictions, for example). Furthermore, at round k, expert 1 knows the true outcome y_k and the whole history of predictions up to round $k-1$. Thus, at round k, the information set given to expert 1 is $\{y_k, y_\ell, x_\ell^i, \vec{\tilde{p}}_\ell, \ell = 1, ..., k-1, i \in E\}$. Based on this information set, this expert selects an action (prediction) $x_k^1 \in \{T, L\}$, standing for "truth" or "lie", where $T := y_k$, and $L := 1 - y_k$. After the predictions of all the experts (honest and malicious) are revealed, their weights will be updated according to (3.2). The malicious expert's program is cast as an MDP[1], in

[1] Here, the malicious expert's action at time k is x_k^1 which incurs the loss $l(\hat{y}_k, y_k)$, and change the state $\vec{p}_k = (p_k^1, p_k^2, ..., p_k^N)$ to the next state $\vec{p}_{k+1} = (p_{k+1}^1, p_{k+1}^2, ..., p_{k+1}^N)$.

which, he aims to maximize the expected accumulated loss of the algorithm over the horizon K, i.e.,

$$\max_{x_1^1,\ldots,x_K^1} \sum_{k=1}^{K} \mathbb{E}_{x_k^2,\ldots,x_k^N}(l(\hat{y}_k, y_k)), \tag{3.5}$$

where the expectation is taken over the randomization of $x_k^2, ..., x_k^N$, i.e., predictions of honest experts.

Algorithm 4 summarizes the adversary's optimal policy for the problem defined by (3.5). In this algorithm, $c_{x_k^1}(y_k, \vec{p}_{k-1})$ denotes the current cost that the adversary can impose on the system by taking action x_k^1 and is defined as the expected loss of the algorithm at round k with respect to actions of the honest experts:

Algorithm 4 Adversary's optimal strategy (DP)

```
**Initialize:** $V_K(.) = c_K(.) = 0$
**for** each step $k = K-1$ downto 1 **do**
    Find the optimal action,
        $u_k^*(y_k, \vec{p}_{k-1}) = \arg\max_{x_k^1} \left[c_{x_k^1}(y_k, \vec{p}_{k-1}) + \mathbb{E}V_{k+1}^*(y_{k+1}, \phi_{x_k^1}(\vec{p}_{k-1}))\right],$
    and the corresponding value function,
        $V_k^*(y_k, \vec{p}_{k-1}) = \max_{x_k^1} \left[c_{x_k^1}(y_k, \vec{p}_{k-1}) + \mathbb{E}V_{k+1}^*(y_{k+1}, \phi_{x_k^1}(\vec{p}_{k-1}))\right].$    (3.6)
**end for**
**Output:** sequence $u_{K-1}^*(.), V_{K-1}^*(.), ..., u_1^*(.), V_1^*(.)$.
```

$$c_{x_k^1}(y_k, \vec{p}_{k-1}) = \mathbb{E}(l(\hat{y}_k, y_k)). \tag{3.7}$$

For further analysis, we denote the value function at stage k by $V_k(\cdot)$,

$$V_k(y_k, \vec{p}_{k-1}, x_k^1) = c_{x_k^1}(\vec{p}_{k-1}, y_k) + \mathbb{E}V_{k+1}^*(y_{k+1}, \phi_{x_k^1}(\vec{p}_{k-1})), \tag{3.8}$$

where $V_{k+1}^*(\cdot)$ denotes the optimal value function, i.e., the optimally accumulated loss from time step $k+1$ onward and $\phi_{x_k^1}(\vec{p}_k)$ denotes the state transition associated with this MDP, i.e.,

$$\phi_{x_k^1}(\vec{p}_k) = \left(p_{k+1}^1, \ldots, p_{k+1}^N\right), \tag{3.9}$$

where $p_{k+1}^i = p_k^i e^{-l(x_k^i, y_k)}$. For simplicity, we denote $\phi_T(\vec{p}_k)$ and $\phi_L(\vec{p}_k)$ as the next state from

$\vec{p}_k$ when $x_k^1 = T$ and $x_k^1 = L$, respectively.

3.2 Preliminary Results

In this section we review some salient properties of the learning algorithm given in Algorithm 3 and establish some relevant results for later use. With a slight abuse of notation, from now on, we use the notation $\mathbb{E}[\cdot]$ to denote the expectation of an event with respect to its ambient space.

Next, we state a useful lemma, which allows us to remove the dependency of the value function and the optimal policy from the actual values of $y_k, k = 1, \dots, K$.

Lemma 2. *For any loss function of the form $l(\hat{y}, y) := Q(|\hat{y} - y|)$, where $Q(\cdot) : [0,1] \to \mathbb{R}$ is an arbitrary function, the expected loss given in* (3.5) *is fully determined by the weight vector $\vec{p}_k$, the horizon length K, and the adversary's policy $\pi := (x_1^1, \dots, x_K^1) \in \{T, L\}^K$.*

Proof. See Appendix A.2.1. □

Therefore, using the above lemma and from now on we remove the dependency of the current costs and value functions from the actual values of y_k. In particular, for a policy π of the adversary we simply define $V_K^\pi(\vec{p}) := \sum_{k=1}^K \mathbb{E}_{x_k^2, \dots x_k^N}[l(\hat{y}_k, y_k)]$, where $\vec{p}$ denotes the weight vector and K is the total number of stages. For simplicity of notation, we may suppress the dependency on the policy π whenever there is no ambiguity, and we simply write $V_K(\vec{p})$.

Since calculating the value functions is in general a difficult task and somehow intractable for exponentially many states, we attempt to find the structural properties of the optimal actions. In the next section, we derive key properties of the current costs and value functions for the two types of loss functions that we consider in this paper.

3.2.1 Current costs and value functions

Logarithmic loss From the definition and relations (3.1), (3.4), and (3.9), we can write the current cost given in (3.7) for two different choices of the adversary's action in $\{L, T\}$ at some generic time by

$$\begin{aligned} c_L(\vec{p}) &= -\mathbb{E}_R\left[\ln\Big(\frac{\epsilon p^1 + (1-\epsilon)\sum_{i\in R} p^i + \epsilon \sum_{j\in R^c} p^j}{\vec{p}\mathbf{1}}\Big)\right], \\ c_T(\vec{p}) &= -\mathbb{E}_R\left[\ln\Big(\frac{(1-\epsilon)p^1 + (1-\epsilon)\sum_{i\in R} p^i + \epsilon \sum_{j\in R^c} p^j}{\vec{p}\mathbf{1}}\Big)\right], \end{aligned} \tag{3.10}$$

where R and R^c denote the (random) set of honest experts which are correct and incorrect at that generic time, respectively, and $\mathbf{1}$ denotes a column vector of all ones of proper dimension.

An immediate consequence of the above relations is the following two properties of the current cost,

- **(P1)**: $c_L(\vec{q}) < c_L(\vec{p})$, if $q^1 < p^1$, and $q^i = p^i$ for $i \neq 1$.
- **(P2)**: $c_L(\vec{p}) \geq c_T(\vec{p}), \forall \vec{p}$.

Absolute loss The absolute loss is defined as $l(\hat{y}_k, y_k) = |y_k - \hat{y}_k|$. Similar to the logarithmic loss, one can see that

$$\begin{aligned} c_L(\vec{p}) &= \mathbb{E}_R\Big[\frac{p^1 + \sum_{j\in R^c} p^j}{\vec{p}\mathbf{1}}\Big], \\ c_T(\vec{p}) &= \mathbb{E}_R\Big[\frac{\sum_{j\in R^c} p^j}{\vec{p}\mathbf{1}}\Big]. \end{aligned} \tag{3.11}$$

Again we note that for absolute loss, the current costs satisfy properties **(P1)** and **(P2)**.

Finally, in the following proposition we state one of the properties of the value function defined by (3.8), namely, monotonicity for both logarithmic and absolute loss function.

Proposition 1. *Given two weight vectors $\vec{p}_{k-1}$ and $\vec{q}_{k-1}$ with $q^1_{k-1} \leq p^1_{k-1}$ and $q^i_{k-1} = p^i_{k-1}$ for $i \neq 1$, $V^*_k(\vec{q}_{k-1}) \leq V^*_k(\vec{p}_{k-1})$ for both logarithmic and absolute losses.*

Proof. See Appendix A.2.2. □

Proposition 1 states that given an arbitrary but fixed vector of weights for honest experts, the value function is a nonincreasing function of the adversary's weight.

3.3 Finite Horizon-Logarithmic Loss

In this section, we describe the optimal strategy for the malicious expert in the general N-expert setting when the loss function is logarithmic. Based on the evolution of normalized weights given in (3.3), it is not hard to see that the normalized weight of the adversary, i.e., $\tilde{p}^1_k$ will not decrease when he tells the truth. Thus, property **(P2)** of current costs and Proposition 1 imply a trade-off between the current costs and the value function. More precisely, while adversary (expert 1) can cause the system to incur a higher current cost by telling a lie, his weight would decrease at the next round as does his value function (Proposition 1). This suggests that perhaps the optimal strategy might be to tell the truth until "enough" weight is gained and only begin to lie after that. We will see in the following that surprisingly this intuition is not true for the logarithmic loss.

Theorem 3. *For the logarithmic loss function in the setting of Algorithm 3, the optimal policy for the malicious expert is the greedy policy of telling a lie at every step.*

Proof. See Appendix A.2.3. □

Note that here our goal is to characterize the optimal policy for the malicious expert rather than to evaluate the value of the maximum loss on the system. The structure of the optimal policy of course may be used to compute or approximate the maximum expected loss of the system. To provide a concrete example, consider the case of $N = 2$ experts with identical initial weights $\vec{p}_0 = (1, 1)$, and the logarithmic loss function. Based on Theorem 3, the optimal policy for the malicious expert is to lie at all stages. This allows us to compute the maximum expected loss of the system as

$$V_K^* = K(1-\mu)\ln(\frac{1}{\epsilon}) + \sum_{j=0}^{K} \mathbb{P}(Z > j)\ln(1 + e^j),$$

where $Z \sim Bin(K, \mu)$ is a binomial distribution with mean μ, and ϵ is the small constant in the definition of the logarithmic loss function. To see why this relation holds, let us fix the adversary's strategy to the false policy, and we look at all the possible sample paths which can be realized by predictions of the honest expert. Any sample path in which the honest expert predicts correctly k times and makes mistakes $K - k$ times will occur with the probability of $\mu^k(1-\mu)^{K-k}$. There are exactly $\binom{K}{k}$ such sample paths, and for any of them, independent of what instances the honest agent predicts correctly or wrongly, the loss incurred with respect to the adversary's false policy equals to $(K-k)\ln(\frac{1}{\epsilon}) + \sum_{j=0}^{k-1}\ln(1+e^j)$. This is because for any of $K - k$ false predictions of the honest agent on the sample path the system incurs a loss of $\ln(\frac{1}{\epsilon})$, and for the remaining k correct predictions, independent of the order of them, the system incurs a loss of $\sum_{j=0}^{k-1}\ln(1+e^j)$. Therefore, by taking expectation over all possible sample paths we get

$$
\begin{aligned}
V_K^* &= \sum_{k=0}^{K} \binom{K}{k} \mu^k (1-\mu)^{K-k} \Big((K-k)\ln(\frac{1}{\epsilon}) + \sum_{j=0}^{k-1} \ln(1+e^j) \Big) \\
&= K(1-\mu)\ln(\frac{1}{\epsilon}) + \sum_{k=0}^{K}\sum_{j=0}^{k-1} \binom{K}{k} \mu^k (1-\mu)^{K-k} \ln(1+e^j) \\
&= K(1-\mu)\ln(\frac{1}{\epsilon}) + \sum_{j=0}^{K-1} \left(\sum_{k=j+1}^{K} \binom{K}{k} \mu^k (1-\mu)^{K-k} \right) \ln(1+e^j) \\
&= K(1-\mu)\ln(\frac{1}{\epsilon}) + \sum_{j=0}^{K} \mathbb{P}(Z > j) \ln(1+e^j),
\end{aligned}
$$

where in the last equality we used the fact that $Z \sim Bin(K, \mu)$ and $\mathbb{P}(Z > K) = 0$.

In general, the structure of optimal policy heavily depends on the choice of the loss function. In the remainder of the paper our goal is to characterize such optimal policy for absolute loss function.

3.4 Optimal Policy for the Absolute Loss with Discounted Factor

In this section we turn our attention to the problem of adversary's optimal policy for the case of absolute loss function. Unlike the logarithmic loss function, the structure of optimal policy for the absolute loss function in finite horizon even for the case of two experts could be very chaotic. This is because the absolute loss function grows much faster than the logarithmic loss, resulting in strong coupling trade off between the growth of value function and the instantaneous costs, which in turn makes the analysis of the absolute loss function in finite horizon much more complicated. Therefore, in this section we focus on the finite and infinite horizon discounted problem when there are only two experts. Although some of our analysis can be extended to the case where there are more than two exerts, however, as we will see, even for the case of two experts finding the optimal policy is a nontrivial and challenging task.

To begin, we note that for the case of two experts (one adversary and one honest expert), knowing the relative weight of the adversary $\tilde{p}^1_{k-1}$ at step k suffices to make a decision. This is because using $\tilde{p}^1_{k-1} + \tilde{p}^2_{k-1} = 1$, the adversary can always infer the relative weight of the honest expert from his own relative weight. Therefore, in this section we find it easier to work with the adversary's relative weight $\tilde{p}^1_k$ as the state rather than the actual weights p^1_k, p^2_k.

Specializing the general law of the relative weight given in (3.3) for the case of two experts, one can easily see that the relative weight of adversary can be written as:

$$\phi_{x_k^1}(\tilde{p}_{k-1}^1) = \begin{cases} \frac{1}{1+\left(\frac{1}{\tilde{p}_{k-1}^1}-1\right)e} & \text{if } x_k^1 = L, x_k^2 = T, \\ \frac{1}{1+\left(\frac{1}{\tilde{p}_{k-1}^1}-1\right)e^{-1}} & \text{if } x_k^1 = T, x_k^2 = L, \\ \tilde{p}_{k-1}^1 & \text{if } x_k^1 = x_k^2. \end{cases} \tag{3.12}$$

It is clear from (3.12) that when two experts make the same prediction at a time, their next (updated) normalized weights do not change. On the other hand, adversary's normalized weight increases if he makes the right recommendation while the honest expert makes a wrong one, and his normalized weight decreases if the opposite is true. Finally, using (3.11) specialized for the case of 2-experts, and for a relative weight of the adversary $\tilde{p}$, one can write the current costs explicitly as

$$\begin{aligned} c_L(\tilde{p}) &= \mu_2\tilde{p} + (1-\mu_2), \\ c_T(\tilde{p}) &= (1-\mu_2)(1-\tilde{p}). \end{aligned} \tag{3.13}$$

Before stating our main results for the absolute loss function, we first provide some simulation results in order to illustrate some of the optimal patterns for the adversary. This will be very helpful to establish our main results later.

3.4.1 Experimental Results for Finite Horizon Problem

We run the dynamic programming (Algorithm 4) for the setting of 2 experts, absolute loss. The honest expert (expert 2) predicts correctly with probabilities $\mu_2 = 0.7$. Figure 3.1 illustrates the optimal actions of the malicious expert (expert 1) as a function of its normalized weight $\tilde{p}^1$ at each time $k \in \{1, 2, ..., 18\}$.

Blue colored points encode the weights $\tilde{p}^1$ at which the optimal action is to tell the truth while red points indicate the weights $\tilde{p}^1$ at which lying is optimal. Figure 3.1 clearly shows that a threshold policy is optimal. It can be observed from these figures that the threshold value of the first expert decreases as time passes by. Motivated by this numerical result, we prove that the threshold policy is optimal under the discounted problem.

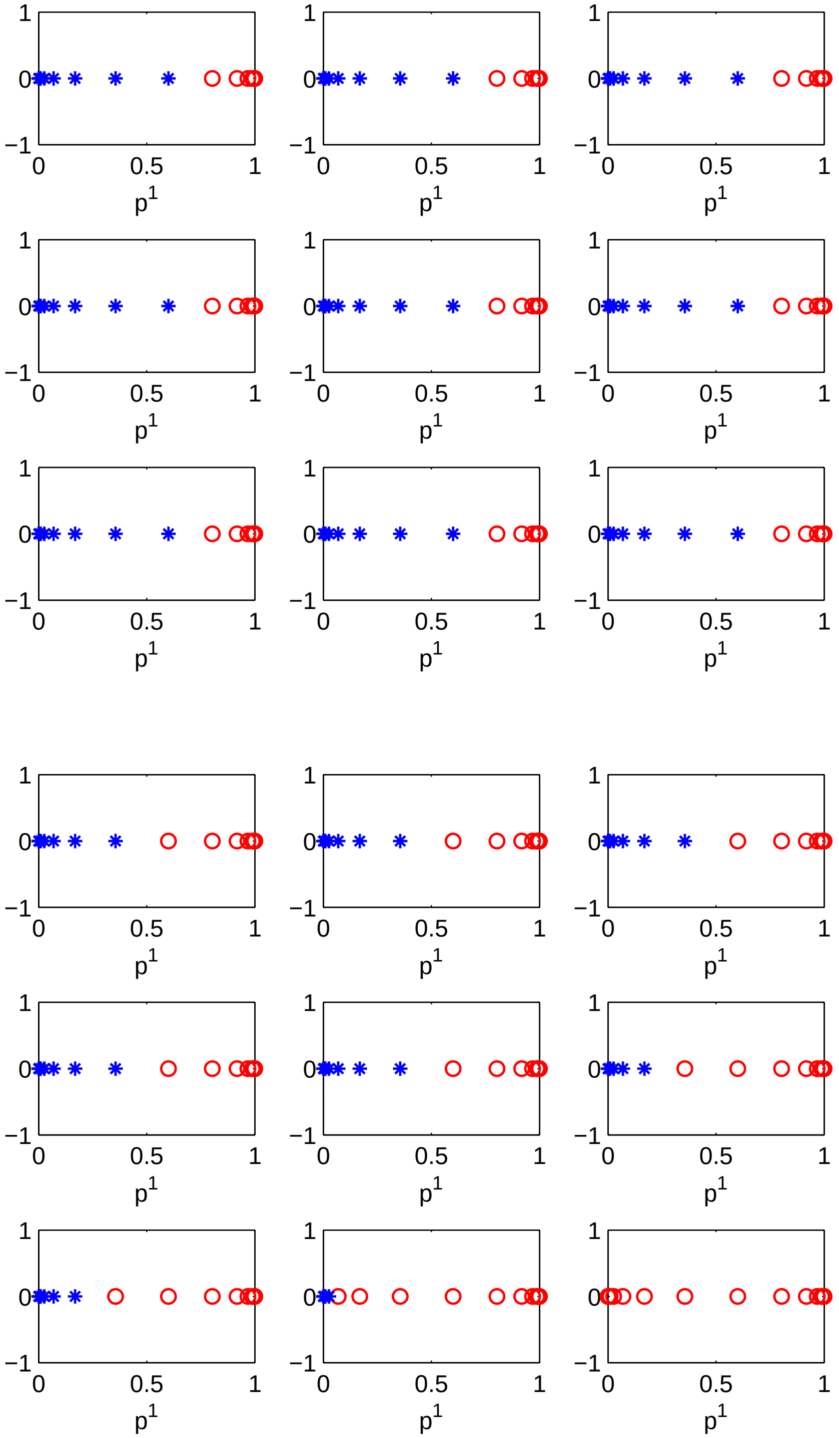

Figure 3.1: Optimal action of expert 1 in the 2-experts, absolute loss setting with a horizon of length 18. Stages are numbered left-to-right, top-to-bottom, i.e., the first stage is depicted in the top-left plot. p^1 is the normalized weight of expert 1. '*' represents the point at which expert 1 tells the truth, and 'o' represents the point at which expert 1 lies.

3.4.2 Finite Horizon Discounted Problem

In this section, we consider the discounted problem for the absolute loss function in finite horizon. For this problem, the adversary's goal is to take optimal actions at each stage in order to maximize the expected discounted loss function given by

$$\max_{x_1^1,\ldots,x_K^1} \sum_{k=1}^{K} \beta^k c_{x_k^1}(\tilde{p}_{k-1}). \tag{3.14}$$

Note that from Lemma 2, the value of y_k is dropped from the notation of the cost function. Next we consider the following definition:

Definition 9. *A policy for the adversary is a threshold policy if there exists a threshold τ such that the adversary tells the truth whenever his relative weight is below τ and lies as soon as his relative weight passes the threshold.*

Based on this definition, in the following theorem we show that for the finite horizon discounted problem, the optimal policy is a threshold one. That is, to impose the most loss on the system, the adversary must start lying once his relative weight exceeds a certain threshold.

Theorem 4. *For the absolute loss function, the optimal policy for the adversary in the case of 2-experts finite horizon problem with the discounted factor $\beta < \frac{1}{e}$ is a threshold policy.*

Proof. See Appendix A.2.4. □

As we mentioned earlier, when the adversary lies, it inflicts a loss on the system at the cost of loosing its relative weight. When $\beta < \frac{1}{e}$, as shown in Theorem 4, the adversary should tell the truth up to some stage and then keep lying. When $\beta > \frac{1}{e}$, the weight v.s. loss trade-off becomes more complicated and the backward induction of Theorem 4 is not sufficient for analyzing the problem.

Finally, we mention here that although we have established our results for a fixed learning rate of 1 (learning rate, in this framework, is a parameter used to adjust the update rate of experts' weights, in order to optimize the regret of the algorithm), they can be naturally extended to any fixed learning rate η. See Appendix A.2.6 for more details.

3.4.3 Infinite Horizon Discounted Problem

Next we consider the infinite horizon discounted factor problem in the case of 2-experts with absolute loss function defined by

$$\max_{x_1^1, x_2^1, \cdots} \sum_{k=1}^{\infty} \beta^k c_{x_k^1}(\tilde{p}_{k-1}). \tag{3.15}$$

In fact, the optimal threshold policy established in Theorem 4 can be extended naturally to the infinite horizon case using the one stage deviation principle[2], defined in Definition 10, provided that the expected value function satisfies the following continuity assumption:

Assumption 3. *Given two sequences of actions of expert* 1*,* $s := \{u_t^1\}_{t=1}^{\infty}$ *and* $s' := \{v_t^1\}_{t=1}^{\infty}$*, let us define the expected value function corresponding to the sequence* s *with initial weight* $\vec{p}_0$ *as follows,*

$$V_s(\vec{p}_0) := \sum_{t=1}^{\infty} \beta^t c_{u_t^1}(\tilde{p}_{t-1}).$$

The continuity assumption states that for any $\epsilon > 0$*, there exists a number* K_ϵ *such that* $\forall k \geq K_\epsilon$*, and when two sequences* s *and* s' *share the first* k *actions, i.e.,* $u_t^1 = v_t^1, \forall t \leq k$*, then* $|V_s(\vec{p}_0) - V_{s'}(\vec{p}_0)| < \epsilon$*.*

Remark 1. *One can easily check that the infinite horizon problem* (3.15) *satisfies the above assumption because of the bounded current costs and discounted factor* $\beta < 1$*.*

Definition 10. *One stage deviation from a strategy is another strategy that differs from that strategy at only one stage. One stage deviation principle states that a strategy is optimal if there is no better one-stage-deviation strategy from that strategy.*

Using Theorem 4 and the continuity assumption given above, a standard application of one stage deviation principle shows that the optimal policy for the infinite horizon discounted problem is also a threshold policy. This has been stated in the following proposition:

Proposition 2. *The optimal policy for the adversary for the infinite horizon discounted problem with* $\beta < \frac{1}{e}$ *is a threshold policy.*

Proof. See Appendix A.2.5. □

An important feature of the infinite horizon problem is that one can explicitly characterize the threshold function at each stage based on the parameters of the problem. More specifically, due to the symmetry of the problem, the threshold for the infinite horizon problem

[2] One stage deviation was originally introduced by Blackwell [200]

denoted by τ is unique and does not change from one stage to the other. This is because the optimal cost from stage k onward is exactly β^k times of that when we start from the initial stage with the same adversary's relative weight. Therefore, the optimal threshold at the initial stage must be the same as the optimal threshold for the kth stage, which implies that the optimal threshold τ is independent of the stage for the infinite horizon problem. Next, in the following theorem, we characterize the optimal threshold for the infinite horizon problem based on the parameters of the problem.

Theorem 5. *The adversary's optimal threshold for the discounted infinite horizon problem and absolute loss function is given by*

$$\tau := \frac{1}{2}\left(1+\theta-\sqrt{(1+\theta)^2-4\frac{(1+e^2)\theta-e}{(1-e)^2}}\right),$$

where $\theta = \frac{\beta\mu_2(1-\mu_2)}{1-\beta(\mu_2^2+(1-\mu_2)^2)}$.

Proof. See Appendix A.2.5. □

3.5 Mean-Field Approach

In this section, we investigate the general case of absolute loss with N experts in finite horizon K by assuming that all experts except the malicious one have the same prediction accuracy, i.e., $\mu_2 = \mu_3 = ... = \mu_N$, and their predictions are independent. To do so, we approximate this system with a system of two experts, one malicious and one honest expert, and first show that the best strategy for the malicious expert in the approximated system is to lie at each stage. Next, we showed that the performance of the optimal strategy in the approximated system converges to that of the optimal strategy in the original system when the number of experts goes to infinity.

For simplicity, we denote μ as the accuracy of prediction of those experts. We also let $y_t = 1$ in our analysis. The analysis for the case $y_t = 0$ is conducted similarly.

Let us rewrite the updated weight (3.3) of an expert i at step k as

$$\begin{aligned}\tilde{p}_k^i &= \frac{\tilde{p}_{k-1}^i e^{x_k^i-1}}{\tilde{p}_{k-1}^1 e^{x_k^i-1}+\bar{p}_{k-1}\sum\limits_{j\neq 1} q_{k-1}^j e^{x_k^j-1}},\\ &= \frac{\tilde{p}_{k-1}^i e^{x_k^i}}{\tilde{p}_{k-1}^1 e^{x_k^i}+\bar{p}_{k-1}\bar{x}_k}.\end{aligned} \tag{3.16}$$

where $\bar{p}_{k-1} = \sum_{j\neq 1} \tilde{p}^j_{k-1}$, and $q^j_{k-1} = \frac{\tilde{p}^j_{k-1}}{\bar{p}_{k-1}}$ and $\bar{x}_k = \sum_{j\neq 1} q^j_{k-1} e^{x^j_k}$. The system now can be viewed as the one with two experts, one of which is malicious and another one is virtual expert whose weight is $\bar{p}$ and prediction is $\bar{x}$. We will approximate $\bar{x}_k$ as follows

$$\begin{aligned}\bar{x}_k &\approx \sum_{j\neq 1} q^j_{k-1} \mathbb{E}(e^{x^j_k}),\\ &= \mu e + (1-\mu).\end{aligned}$$

Denote $\hat{\phi}_T(p), \hat{\phi}_L(p)$ as the approximated versions of $\phi_T(p), \phi_L(p)$ when we apply $\bar{x}$ into the transitions at the state $\tilde{p}^1 = p$. In particular, we have

$$\begin{aligned}\hat{\phi}_T(p) &= \frac{pe}{pe + (1-p)(\mu e + (1-\mu))}\\ &= \frac{1}{1 + (1/p - 1)(\mu + (1-\mu)e^{-1})},\end{aligned} \tag{3.17}$$

and

$$\begin{aligned}\hat{\phi}_L(p) &= \frac{p}{p + (1-p)(\mu e + (1-\mu))}\\ &= \frac{1}{1 + (1/p - 1)(\mu e + (1-\mu))}.\end{aligned} \tag{3.18}$$

Note that, derived from the definition of absolute loss, when $y_t = 1$, the current costs are given as

$$c_L(p) = p + (1-p)(1 - \sum_{i\neq 1} q^i_{t-1} x^i_t),$$

and

$$c_T(p) = (1-p)(1 - \sum_{i\neq 1} q^i_{t-1} x^i_t)).$$

We approximate these costs, using $\sum_{i\neq 1} q^i_{t-1} x^i_t \approx \mu$ as

$$\hat{c}_L(p) = p + (1-p)(1-\mu), \tag{3.19}$$

$$\hat{c}_T(p) = (1-p)(1-\mu). \tag{3.20}$$

The proof of next results are given in Appendix A.2.7.

Lemma 3. *For any relative weight p of expert 1,*

$$p - \hat{\phi}_L(p) - \mu(\hat{\phi}_T(p) - \hat{\phi}_L(p)) > 0.$$

Lemma 4. *The approximated transition satisfies transitive property, i.e., $\hat{\phi}_L(\hat{\phi}_{T^{(k)}}(p)) = \hat{\phi}_{T^{(k)}}(\hat{\phi}_L(p))$ $\forall k \in \mathbb{N}$.*

Instead of considering the adversarial setting with one malicious expert and $N-1$ experts whose accuracy are the same, we consider the setting with 2 experts in which the approximated state transitions are given in (3.17) and (3.18), and the approximated current cost functions are given in (3.19) and (3.20). We call this setting "approximated setting".

Theorem 6. *For the adversarial approximated setting, it is optimal to always tell a lie, i.e., $D\hat{V}_k(\hat{p}) > 0$ $\forall \hat{p}, \forall k$, where $D\hat{V}_k(\hat{p}) = \hat{V}_k(\hat{p}, L) - \hat{V}_k(\hat{p}, T)$.*

Proof. See Appendix A.2.7. □

Now, we prove that the optimal strategy for approximated setting is nearly optimal in the sense that it gains the performance close to that of the optimal strategy in the original setting. The following lemma is crucial for the proof.

Lemma 5. *At any stage $k \leq K$, $\forall \epsilon_k > 0$, there exists $\delta_k > 0$ such that, if $|p - \hat{p}| < \delta_k$, we have $|V_k(p) - \hat{V}_k(\hat{p})| < \epsilon_k$ with probability $1 - \xi_k$, where $\xi_k = exp(-c_k N)$, and the constant c_k depends on δ_k and ϵ_k.*

Proof. See Appendix A.2.7. □

Theorem 7. *For the setting of N experts, absolute loss, finite horizon K, the optimal strategy for the approximated setting incurs asymptotically (when $N \to \infty$) the total loss of the optimal strategy for the original setting, given that the two algorithms start from the same initial weight of the malicious expert.*

Proof. See Appendix A.2.7. □

Remark 2. *One can expect that the malicious expert affects the system less and less when the number of honest experts increases. This is illustrated in Figure 3.2, in which the total losses inflicted on the system by the malicious expert are compared using two policies. Each algorithm is run with the number of honest experts varying from 2 to 20, all with horizon 20. The 'lying' policy outperforms the random policy where the malicious expert just simply picks up a random prediction at any time. The difference between the losses of the two algorithm decreases when the number of honest experts increases. In this experiment, we assigned the values $\mu = 0.5$ for all the honest experts. Then, since the horizon is 20, the loss of the algorithms will converge to 10 eventually.*

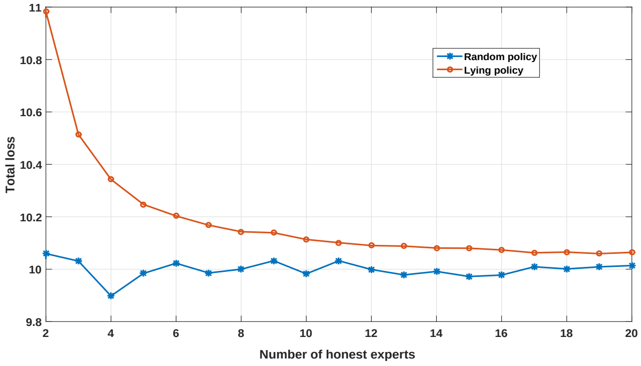

Figure 3.2: Loss comparison of lying policy and random policy. Number of experts varies from 2 to 20, all with horizon 20. Accuracy of all honest experts is $\mu = 0.5$.

CHAPTER 4

ADAPTIVE LABELING WITH EXPERT ADVICE

Active learning addresses the problems where the labels are either costly or obtained through a time-consuming procedure. In this chapter, we consider an application of active learning on learning from expert advice framework where the target is to reduce the number of label requests while still keep the regret bound (regret is the difference of the total loss of the algorithm and total loss of the best expert) as small as possible. We proposed two efficient algorithms, Experts-Predictions-based-Selective-Labeling (EPSL) and Experts-Predictions-based-Adaptive-Labeling (EPAL), to determine, for each example, whether it is necessary to require its label. Both algorithms obtain nearly optimal regret bound up to some constant depending on the characteristics of experts' predictions. Experimental results show that our algorithms outperform the others in this setting.

4.1 Problem Setting and Notations

Denote $E = \{1, 2, ..., N\}$ as the set of experts. At each time t, each expert i provides a prediction $x_t^i \in [0, 1]$ for the given object. Our first proposed algorithm, EPSL, is depicted as in Algorithm 5.

In Algorithm 5, the weighted average value $\hat{y}_t$ is predicted by (4.1), where we denote p_t^i as the weight of expert i at time t. At each time t, given experts predictions, the algorithm calculates the prediction range, γ_t, defined by the maximum difference of predictions of any pair of experts. The algorithm then decides whether to request for labeling by drawing a Bernoulli random variable with the prediction range, γ_t, as the parameter. Intuitively, if experts tend to agree on a certain item, i.e., x_t^i are more or less similar and close to each other, that item seems easy (or at least very popular) to predict. It implies that it is not necessary to require label for that object. If the outcome of Bernoulli random variable is 1, the algorithm requests the label for that object. Denote $l(x_t^i, y_t)$ as the loss of expert i at time t given the true label y_t, where we assume $l(x_t^i, y_t) \in [0, 1]$. The weights of experts are then updated by (4.3), where $\hat{l}(x_t^i, y_t)$ is the estimated loss, defined by (4.4).

Algorithm 5 EPSL - Experts Predictions based Selective Labeling

```
**Input:** Set of experts $E = \{1, ..., N\}$, learning rate $\eta$
**Initialize:** $p_0^i = 1$ for i = 1,...,N.
**for** each round $t = 1, 2, ..., T$ **do**
    Each expert gives his prediction $x_t^i$
    Algorithm calculates the maximal difference of experts' predictions $\max\limits_{i,j} |x_t^i - x_t^j|$
    **Prediction:**
    Algorithm predicts the value of the object based on the weighted average,
        $\hat{y}_t = \frac{\sum\limits_{i \in E} p_t^i x_t^i}{\sum\limits_{i \in E} p_t^i}, \quad (4.1)$
    **Selection:**
    Algorithm draws a Bernoulli random variable with parameter $\gamma_t := \max\limits_{i,j} |x_t^i - x_t^j|$. If its value is 1, request the label $y_t$.
    **Update:**
    Algorithm updates weights of all experts. Each weight is updated by
        $p_t^i = p_{t-1}^i e^{-\eta \hat{l}(x_t^i, y_t)}, \quad (4.2)$
    where
        $\hat{l}(x_t^i, y_t) = \begin{cases} l(x_t^i, y_t)/\gamma_t & \text{w.p } \gamma_t, \\ 0 & \text{otherwise }, \end{cases} \quad (4.3)$
**end for**
```

We denote the expected accumulated regret of the algorithm up to time T as R_T,

$$R_T = \mathbb{E}\left(\sum_{t=1}^{T} l(\hat{y}_t, y_t)\right) - \mathbb{E}\left(\min_{i \in E} \sum_{t=1}^{T} l(x_t^i, y_t)\right), \tag{4.4}$$

where the expectation is taken over the randomization of selecting sample to be labeled. In the next sections, we derive the upper bound for the regret for our two proposed algorithms.

4.2 Selective labeling based on the experts predictions

In this section, we introduce our first algorithm, EPSL. This algorithm sets a constant learning rate which is calculated using a prior information from the ranges of experts predictions during the horizon. The whole procedure is given in Algorithm 5.

Theorem 8. *The regret of EPSL (Algorithm 5) satisfies*

$$R_T \leq 2\sqrt{\ln N \sum_{t=1}^{T} \frac{1}{2\gamma_t}} \text{ with } \eta = \sqrt{\frac{\ln N}{\sum_{t=1}^{T} 1/(2\gamma_t)}}, \gamma_t = \max_{i,j} |x_t^i - x_t^j|.$$

Proof. From the weight update rule (4.2) and the inequality $e^{-x} \leq 1 - x + x^2/2, x \geq 0$,

$$p_t^i \leq p_{t-1}^i \Big(1 - \eta \hat{l}(x_t^i, y_t) + \frac{\eta^2}{2} \hat{l}(x_t^i, y_t)^2\Big).$$

It follows that

$$\frac{\sum_{i \in E} p_t^i}{\sum_{i \in E} p_{t-1}^i} \leq 1 - \sum_{i \in E} \eta \tilde{p}_t^i \hat{l}(x_t^i, y_t) + \sum_{i \in E} \frac{\eta^2}{2} \tilde{p}_t^i \hat{l}(x_t^i, y_t)^2,$$

where $\tilde{p}_t^i$ is the normalized weight of expert i, given by $\tilde{p}_t^i = \frac{p_t^i}{\sum_{i \in E} p_t^i}$. Taking ln of two sides, applying the inequality $\ln(1 + x) \leq x$, and summing over t=1,...,T, we obtain

$$\ln \sum_{i \in E} p_T^i - \ln N \leq - \sum_{t=1}^{T} \sum_{i \in E} \eta \tilde{p}_t^i \hat{l}(x_t^i, y_t) + \sum_{t=1}^{T} \sum_{i \in E} \frac{\eta^2}{2} \tilde{p}_t^i \hat{l}(x_t^i, y_t)^2. \tag{4.5}$$

Denote $\mathbb{E}_t(.) := \mathbb{E}(.|F_{t-1})$ as the conditional expectation on the previous predictions and label selections, where $F_{t-1} =< \hat{y}_s, u_s >, s = 1, ..., t-1$, where u_s is the outcome of the Bernoulli random variable at step s. From (4.3), it is observed that $\mathbb{E}_t(\hat{l}(x_t^i)) := \mathbb{E}(\hat{l}(x_t^i)|F_{t-1}) = l(x_t^i)$. Also note that

$$\ln \sum_{i \in E} p_T^i \geq \ln p_T^i = \ln \Big(exp(\sum_{t=1}^{T} -\eta \hat{l}(x_t^i, y_t)) \Big) = \sum_{t=1}^{T} -\eta \hat{l}(x_t^i, y_t).$$

Plug this into (4.5) and take the expectation on two side, we get

$$\mathbb{E}\left(\sum_{t=1}^{T} \sum_{i \in E} \tilde{p}_t^i l(x_t^i, y_t)\right) - \mathbb{E}\left(\sum_{t=1}^{T} \hat{l}(x_t^i, y_t)\right) \leq \frac{\ln N}{\eta} + \eta \mathbb{E} \sum_{t=1}^{T} \sum_{i \in E} \frac{\tilde{p}_t^i}{2} \hat{l}(x_t^i, y_t)^2. \tag{4.6}$$

Let $A = \mathbb{E}\sum_{t=1}^{T}\sum_{i\in E}\frac{\tilde{p}_t^i}{2}\hat{l}(x_t^i, y_t)^2$. Since $\hat{l}(x_t^i, y_t) \leq \frac{1}{\gamma_t}$, we have

$$\begin{aligned} A &\leq \mathbb{E}\sum_{t=1}^{T}\sum_{i\in E}\frac{\tilde{p}_t^i}{2\gamma_t}\hat{l}(x_t^i, y_t), \\ &= \mathbb{E}\sum_{t=1}^{T}\frac{1}{2\gamma_t}\sum_{i\in E}\tilde{p}_t^i l(x_t^i, y_t), \\ &\leq \mathbb{E}\sum_{t=1}^{T}\frac{1}{2\gamma_t}\sum_{i\in E}\tilde{p}_t^i, \\ &= \mathbb{E}\sum_{t=1}^{T}\frac{1}{2\gamma_t}, \end{aligned}$$

where we have used the fact that $l(x_t^i, y_t) \leq 1$ in the last inequality and $\sum_{i\in E}\tilde{p}_t^i = 1$. Thus, from (4.6), we obtain

$$\mathbb{E}\left(\sum_{t=1}^{T}\sum_{i\in E}\tilde{p}_t^i l(x_t^i, y_t)\right) - \mathbb{E}\left(\sum_{t=1}^{T}\hat{l}(x_t^i, y_t)\right) \leq \frac{\ln N}{\eta} + \eta\sum_{t=1}^{T}\frac{1}{2\gamma_t}.$$

Finally, since $\mathbb{E}\left(\sum_{t=1}^{T}\hat{l}(x_t^i, y_t)\right) = \mathbb{E}\left(\mathbb{E}_t\left(\sum_{t=1}^{T}\hat{l}(x_t^i, y_t)\right)\right) = \mathbb{E}\left(\sum_{t=1}^{T}l(x_t^i, y_t)\right)$,

$$\mathbb{E}\left(\sum_{t=1}^{T}\sum_{i\in E}\tilde{p}_t^i l(x_t^i, y_t)\right) - \mathbb{E}\left(\sum_{t=1}^{T}l(x_t^i, y_t)\right) \leq \frac{\ln N}{\eta} + \eta\sum_{t=1}^{T}\frac{1}{2\gamma_t}.$$

Applying the convexity of the loss function, $\sum_{t=1}^{T}l(\hat{y}_t) \leq \sum_{t=1}^{T}\sum_{i\in E}\tilde{p}_t^i l(x_t^i, y_t)$ and choosing $\eta = \sqrt{\frac{\ln N}{\sum_{t=1}^{T}1/(2\gamma_t)}}$, the result follows. □

Note that from Algorithm 5, the expected number of request is $\sum_{t=1}^{T}\gamma_t$ which depends on the characteristic of the sequence of experts' predictions. It follows that the number of requests tends to be large if the experts predictions differ on most rounds (in this case, we can get good regret bound in the expense of much labeling cost). On the other hand, if experts agree on a large proportion of objects, the number of requests is significantly reduced. We will see that more in the experimental results.

4.2.1 Remarks on the regret bound

In this subsection, we would like to make some comparisons of our regret bound with the bound in [175] through several scenarios. For convenience, the regret bound in [175] is shown here as

$$R'_T \leq n\sqrt{\frac{2\ln N}{m}},$$

where m, n is the number of queries and the horizon, respectively. Denote α as the request ratio, i.e, $\frac{m}{n}$, their regret bound can be written as

$$R'_T \leq \frac{1}{\alpha}\sqrt{2m\ln N}. \tag{4.7}$$

1. In the first example, we assume that the prediction ranges of experts always exceed a constant, i.e., $\gamma_t \geq \gamma \forall\ t$. This implies for our regret that

   $$R_T \leq \frac{1}{\sqrt{\gamma}}\sqrt{2m\ln N}.$$

   Comparing to (4.7), one can draw the following two observations:

   - If the two algorithms use the same request rate, our algorithm obtains the better regret bound of $\sqrt{\gamma}$.
   - Our algorithm needs $\sqrt{\gamma}n$ number of queries to obtain the same regret bound as in (4.7) while [175] needs more queries, γn.

2. Assume that predictions of experts follow some Bernoulli distributions and they are all independent of each other. In particular, prediction of expert i is given by

   $$x_t^i = \begin{cases} 1 & \text{w.p } q_i, \\ 0 & \text{w.p } 1 - q_i, \end{cases}$$

   Above, q_i can be represented as the accuracy of expert i when the outcome $y_t = 1 \forall\ t$. Denote $z_{ij} = |x^i - x^j|$, we obtain

   $$z_{ij} = \begin{cases} 1 & \text{w.p } q_i(1-q_j) + q_j(1-q_i), \\ 0 & \text{w.p } q_iq_j + (1-q_i)(1-q_j), \end{cases}$$

Since $\gamma_t = \max_{i,j} |z_{ij}|$, it is obvious that

$$\gamma_t = \begin{cases} 1 & \text{w.p } q', \\ 0 & \text{w.p } 1 - q', \end{cases} \tag{4.8}$$

where q' is some certain probability. Now, let us rewrite our regret bound as

$$R_T \leq \sqrt{2m \ln N (\frac{1}{m} \sum_{t=1}^{m} \frac{1}{\gamma_t})}.$$

We will approximate the term $\frac{1}{m} \sum_{t=1}^{m} \frac{1}{\gamma_t}$ as $\mathbb{E}(\frac{1}{\gamma_t})$. Since labels are always requested when $\gamma_t \neq 0$,

$$R_T \leq \sqrt{2m \ln N}.$$

In this case, one can see that our bound is better than that in [175] by a factor of $\frac{1}{\alpha}$.

3. Assume that predictions of expert i follow a uniform distribution on $[0, 1]$ denoted by $U(0, 1)$. As above, we aim to find the upper bound of $\mathbb{E}(\frac{1}{\gamma_t})$. To that end, let us start finding the distribution of $z_{ij} = |x^i - x^j|$.

$$P(|x^i - x^j| \leq u) = 1 - 2(\frac{1}{2}(1 - u^2)) = 2u - u^2,$$

where the first equality follows from the fact that the probability is equal to area in between two lines $y = u + x$ and $y = -u + x$ in a box $[0, 1]^2$. It implies that the pdf of Z_{ij} is given by $f_{Z_{ij}}(u) = 2 - 2u$. As in the example above, let us denote π^k as a matching of $\lfloor \frac{N}{2} \rfloor$ values of z_{ij} such that $\forall (i, j)$ and (i', j'), we have $i \neq i',\ j \neq j'$. It is obvious that $\gamma_t \geq \pi^k$, and therefore, $\mathbb{E}(\frac{1}{\gamma_t}) \leq \mathbb{E}(\frac{1}{\pi^k})$.
The distribution of π^k is derived from

$$P(\pi^k \leq u) = \prod_{(i,j) \in \pi^k} P(z_{ij} \leq u) = \prod_{(i,j) \in \pi^k} (2u - u^2) = (2u - u^2)^{\lfloor \frac{N}{2} \rfloor},$$

and then

$$f_{\pi^k}(u) = \lfloor \frac{N}{2} \rfloor (2 - 2u)(2u - u^2)^{\lfloor \frac{N}{2} \rfloor - 1}.$$

Expectation of $\frac{1}{\pi^k}$ is calculated as

$$\begin{aligned}\mathbb{E}(\frac{1}{\gamma_t}) &= \int_0^1 \frac{1}{u}\lfloor\frac{N}{2}\rfloor(2-2u)(2u-u^2)^{\lfloor\frac{N}{2}\rfloor-1}du\\ &= \lfloor\frac{N}{2}\rfloor\int_0^1(2-2u)(2-u)(2u-u^2)^{\lfloor\frac{N}{2}\rfloor-2}du\end{aligned}$$

Using integration by part and the fact that $(1-u)^2 \geq 0$,

$$\begin{aligned}\mathbb{E}(\frac{1}{\pi^k}) &= \frac{\lfloor\frac{N}{2}\rfloor}{\lfloor\frac{N}{2}\rfloor-1} + \frac{\lfloor\frac{N}{2}\rfloor}{\lfloor\frac{N}{2}\rfloor-1}\int_0^1(2u-u^2)^{\lfloor\frac{N}{2}\rfloor-1}du\\ &\leq \frac{\lfloor\frac{N}{2}\rfloor}{\lfloor\frac{N}{2}\rfloor-1} + \frac{\lfloor\frac{N}{2}\rfloor}{\lfloor\frac{N}{2}\rfloor-1}\\ &\leq 2\frac{\lfloor\frac{N}{2}\rfloor}{\lfloor\frac{N}{2}\rfloor-1}.\end{aligned}$$

It implies that our regret bound is approximated by $\sqrt{2m\ln N\frac{2\lfloor\frac{N}{2}\rfloor}{\lfloor\frac{N}{2}\rfloor-1}}$ which is less than that of [175] if $\alpha < \sqrt{\frac{\lfloor\frac{N}{2}\rfloor-1}{2\lfloor\frac{N}{2}\rfloor}}$. When the number of experts is large enough and the budget for labeling is limited so that $\alpha < 0.7$, our regret bound is better than that of [175].

4.3 Adaptive labeling using time-varying learning parameter

Although the bound in Theorem 8 guarantees the vanishing regret, it has been implied that the learner knows the experts' predictions in advance or at least knows some prior information to choose parameter η appropriately. We will overcome this now by proposing an algorithm that chooses η properly on the fly without the access to those information. Algorithm 6 follows the same procedure as Algorithm 5 except that the constant learning rate is replaced by a time-varying rate, which is updated on the run based on the predictions of experts. The following lemma is important for the proof of the main results.

Lemma 6.

$$\mathbb{E}_t(\tilde{p}_t^i\hat{l}(x_t^i,y_t)) = w_t^i l(x_t^i,y_t),$$

where $w_t^i = \frac{p_{t-1}^i e^{-\eta_t l(x_t^i,y_t)/\gamma_t}}{\sum\limits_{i\in E} p_{t-1}^i e^{-\eta_t l(x_t^i,y_t)/\gamma_t}}$.

Algorithm 6 EPAL - Experts Predictions based Adaptive Labeling

```
Input: Set of experts $E = \{1, ..., N\}$.
Initialize: $p_0^i = 1$ for i = 1,...,N.
for each round $t = 1, 2, ..., T$ do
    Each expert gives his prediction $x_t^i$
    Algorithm calculates the maximal difference of experts' predictions $\gamma_t := \max_{i,j} |x_t^i - x_t^j|$
    Prediction:
    Algorithm predicts the value of the object based on the weighted average,
        $$\hat{y}_t = \frac{\sum_{i \in E} p_t^i x_t^i}{\sum_{i \in E} p_t^i}, \quad (4.9)$$
    Selection:
    Algorithm draws a Bernoulli random variable with parameter $\gamma_t$. If its value is 1, request the label $y_t$.
    Update:
    Algorithm updates the learning rate: $\eta_t = \sqrt{\frac{\ln N}{\sum_{s=1}^{t} 1/\gamma_s}}$
    Algorithm updates weights of all experts. Each weight is updated by
        $$p_t^i = p_{t-1}^i e^{-\eta_t \hat{l}(x_t^i, y_t)}, \quad (4.10)$$
    where
        $$\hat{l}(x_t^i, y_t) = \begin{cases} l(x_t^i, y_t)/\gamma_t & \text{w.p } \gamma_t, \\ 0 & \text{otherwise} , \end{cases} \quad (4.11)$$
end for
```

Proof. From the definition of $\mathbb{E}_t(.)$, $\tilde{p}_t^i$ and weight update rule (4.10), we have

$$\begin{aligned} \mathbb{E}_t(\tilde{p}_t^i \hat{l}(x_t^i, y_t)) &= \mathbb{E}_t \left(\frac{p_{t-1}^i e^{-\eta_t \hat{l}(x_t^i, y_t)}}{\sum_{i \in E} p_{t-1}^i e^{-\eta_t \hat{l}(x_t^i, y_t)}} \hat{l}(x_t^i, y_t) \right), \\ &= \gamma_t \left(\frac{p_{t-1}^i e^{-\eta_t l(x_t^i, y_t)/\gamma_t}}{\sum_{i \in E} p_{t-1}^i e^{-\eta_t l(x_t^i, y_t)/\gamma_t}} l(x_t^i, y_t)/\gamma_t \right), \\ &= w_t^i l(x_t^i, y_t). \end{aligned}$$

□

Theorem 9. *The regret of EPAL (Algorithm 6) with time-varying parameter* $\eta_t = \sqrt{\frac{\ln N}{\sum_{s=1}^{t} 1/\gamma_s}}$ *has the following upper bound*

$$R_T \leq 2\sqrt{\sum_{t=1}^{T} \frac{1}{\gamma_t} \ln N}.$$

It is worthy noting here that, by choosing a non-increasing sequence of η_t, we still get the nearly optimal bound as obtained in Theorem 8, up to a constant factor of $\sqrt{2}$.

Proof. We use the following result (which is analyzed in [172], [173], [201]) that, given a non-increasing sequence $(\eta_t)_t$,

$$\sum_{t=1}^{T} \sum_{i \in E} \tilde{p}_t^i \hat{l}(x_t^i, y_t) - \min_{i \in E} \Big(\sum_{t=1}^{T} \hat{l}(x_t^i, y_t) \Big) \leq \sum_{t=1}^{T} \frac{\eta_t}{2} \sum_{i \in E} \tilde{p}_t^i (\hat{l}(x_t^i, y_t))^2 + \frac{\ln N}{\eta_T}. \tag{4.12}$$

Using the same argument as in Theorem 8,

$$\mathbb{E}_t(\hat{l}(x_t^i, y_t)) = l(x_t^i, y_t),$$

Also, from Lemma 6, we obtain

$$\mathbb{E}_t(\tilde{p}_t^i \hat{l}(x_t^i, y_t)) = w_t^i l(x_t^i, y_t).$$

Therefore,

$$\mathbb{E}_t\Big(\sum_{i \in E} \tilde{p}_t^i (\hat{l}(x_t^i, y_t))^2 \Big) \leq \frac{1}{\gamma_t} \sum_{i \in E} \mathbb{E}_t(\tilde{p}_t^i \hat{l}(x_t^i, y_t)) = \frac{1}{\gamma_t} \sum_{i \in E} w_t^i l(x_t^i, y_t) \leq \frac{1}{\gamma_t} \sum_{i \in E} w_t^i = \frac{1}{\gamma_t},$$

where the last inequality follows from the fact that $l(x_t^i, y_t) \leq 1$. Taking the expectation of two sides of (4.12), we obtain

$$\mathbb{E}\Big(\sum_{t=1}^{T} \sum_{i \in E} \tilde{p}_t^i l(x_t^i, y_t) \Big) - \min_{i \in E} \mathbb{E}\Big(\sum_{t=1}^{T} l(x_t^i, y_t) \Big) \leq \sum_{t=1}^{T} \frac{\eta_t}{2\gamma_t} + \frac{\ln N}{\eta_T}. \tag{4.13}$$

Choosing $\eta_t = \sqrt{\frac{lnN}{\sum_{s=1}^{t} 1/\gamma_s}}$ and applying the inequality $\sum_{t=1}^{T} \frac{1/\gamma_t}{\sqrt{\sum_{s=1}^{t} 1/\gamma_s}} \leq 2\sqrt{\sum_{s=1}^{T} 1/\gamma_t}$ from

[173], [37],

$$\frac{\ln N}{\eta_T} = \sqrt{\ln N \sum_{t=1}^{T} 1/\gamma_t},$$

$$\sum_{t=1}^{T} \frac{\eta_t}{2\gamma_t} \leq \sqrt{\ln N \sum_{t=1}^{T} 1/\gamma_t},$$

the proof is completed. □

4.4 Experimental Results

In this section, we run the experiments on both synthetic and real datasets. For each dataset, three algorithms are run to compare the performance: the random algorithm in [175], denoted by R, Adaptive Exponentially Weighted Average in [176], denoted by AEWA, and our Experts Predictions based Adaptive Labeling, denoted by EPAL. For performance comparison, three criteria are measured: accumulated regret, number of requests and regret rate. Accumulated regret is the total regret over all times that the label is required, and the regret rate is the ratio of accumulated regret and the number of requests. These three values together indicate the performance of each algorithm. We consider 5 experts for each dataset. Each expert is a learning algorithm which, based on the features of each example, gives the prediction for that example. The followings are algorithms used as experts in this experiment: Linear Discriminant Analysis (LDA), Random Forest, AdaBoost, Quadratic Discriminant Analysis (QDA) and Naive Bayes.

On the first part of the experiment, we show the results on the synthetic datasets. In order to provide a wide-range comparison of the algorithms, we generate datasets with different sizes ranging from 5000 to 50000. Figure 4.1, Figure 4.2, and Figure 4.3 show the accumulated regret, number of requests, and regret rate of the three algorithms, respectively.

From Figure 4.1, one can observe that the randomized algorithm picks up examples without caring whether that might be good or not. As a consequence, if the request rate is high, its accumulated regret is high. AEWA, on another hand, can obtain different request rates by controlling the threshold value. In this experiment, we assign this value as 0.5 since we assume no prior information for each dataset. Figure 4.2 shows that AEWA obtains the best request rate comparing to the others. However, its accumulated regret is worse than EPAL. This can be seen, first, from the fact that EPAL uses the adaptive learning rate which is updated on the run to keep track of the trends of experts predictions. Secondly, since EPAL also uses random selection based on a criteria, it explores some potential ex-

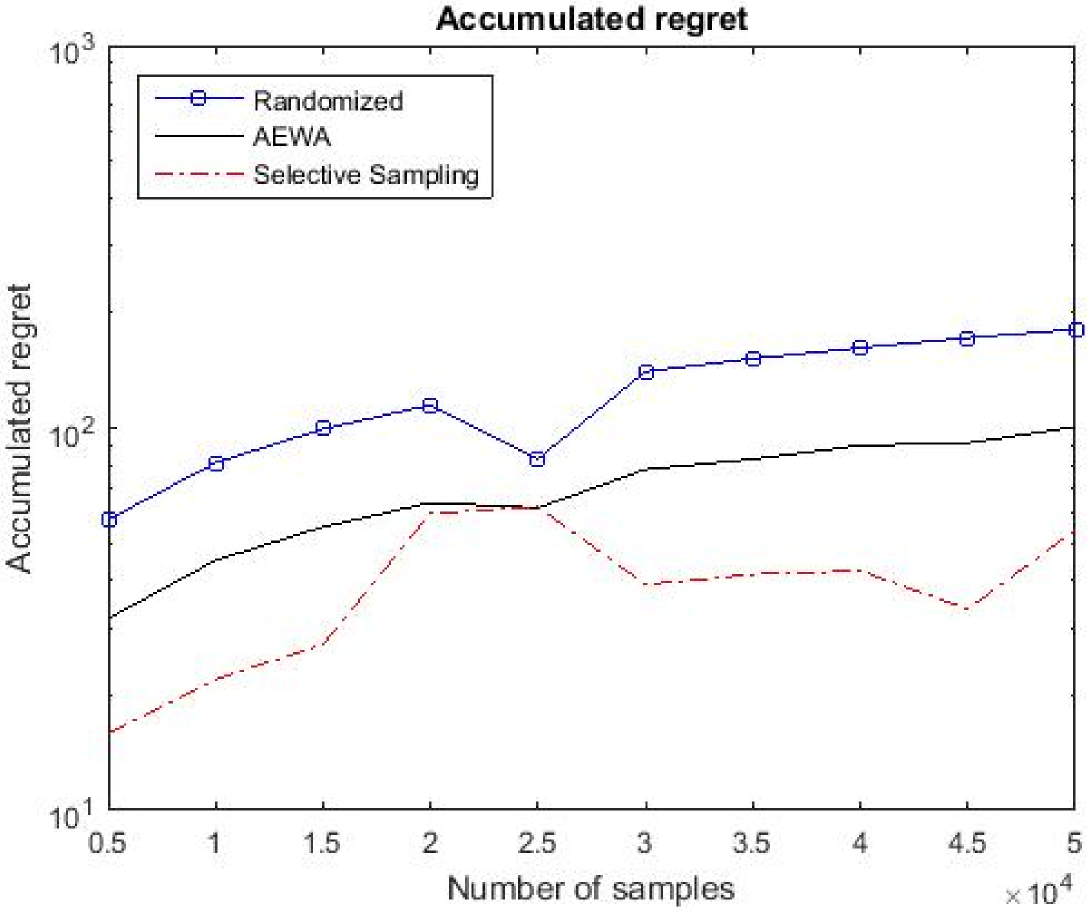

Figure 4.1: Accumulated regrets of R, AEWA, and EPAL on synthetic datasets.

amples that experts might make mistake. As the result, EPAL gets better regret rate than AEWA. Comparing to R, EPAL also gets better regret rate since, even R learns from more examples than EPAL, it cannot take advantage from learning easy examples while most of the time, EPAL only learns from the hard examples.

On the second part of the experiment, we run the algorithms on different real datasets obtained from the UCI datasets (https://archive.ics.uci.edu/ml/datasets.html). Those datasets names and sizes are shown on the first comlumn of Table 4.1. Note that the size displayed is the size of the test set of each original dataset. Using the same logic with the synthetic simulation, we chooses the datasets with different size and the results reflect roughly the same with the synthetic results. There are only two datasets that R and AEWA can perform better than EPAL, but the regret rate is slightly different between the algorithms.

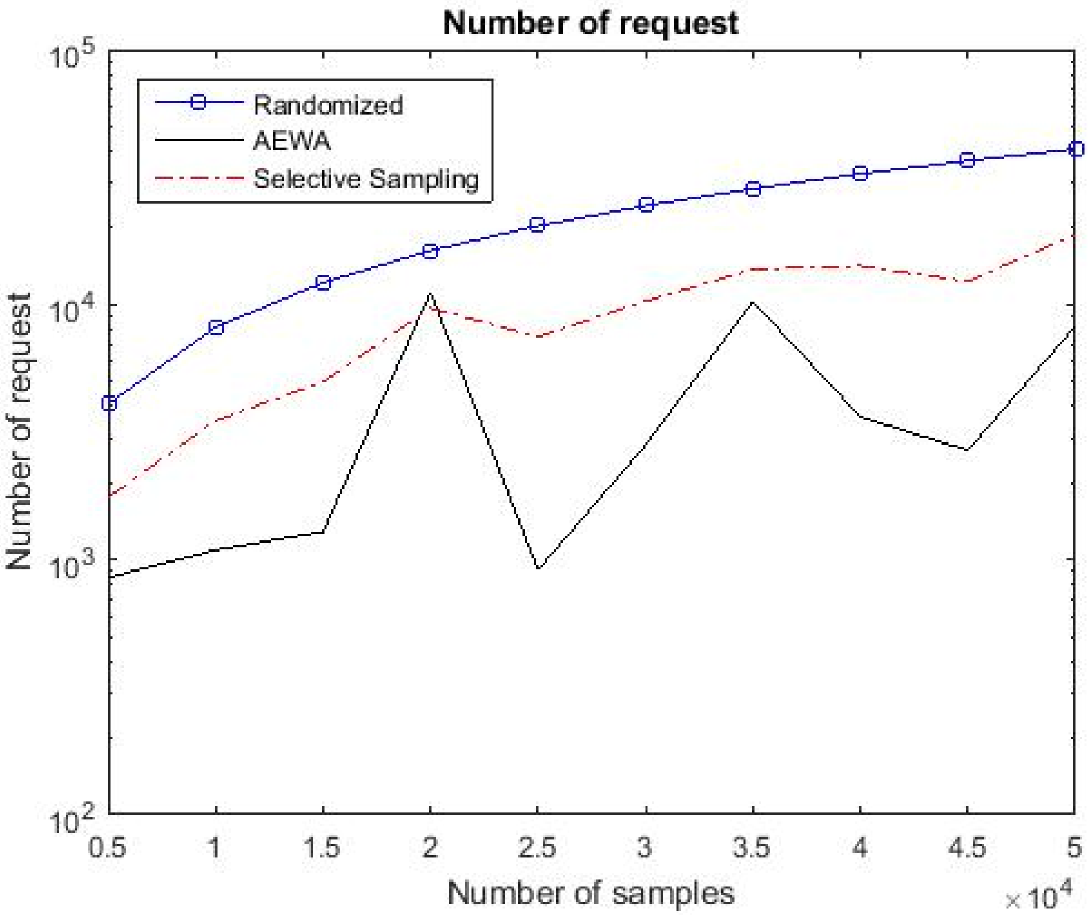

Figure 4.2: Number of requests of R, AEWA, and EPAL on synthetic datasets.

Table 4.1: Comparison of R, AEWA and EPAL on real datasets.

Dataset	Total Regret			Number of Requests			Regret Rate		
	R	AEWA	EPAL	R	AEWA	EPAL	R	AEWA	EPAL
eighthr-2027	36.8539	20.7603	38.3192	1653	585	1075	0.0223	0.0355	0.0357
EEG Eye State-3809	50.5502	27.9945	7.025	3105	1454	2162	0.0163	0.0193	0.0032
mushroom-6499	65.4576	36.5184	18.1883	5304	757	2661	0.0123	0.0482	0.0068
room occupancy-7802	71.6991	39.6892	23.354	6358	305	3346	0.0113	0.1301	0.007
skin segmentation-11831	88.2592	46.878	33.7681	9646	462	5706	0.0092	0.1015	0.0059
magic04-15216	100.109	56.4695	41.458	12410	4480	8332	0.0081	0.0126	0.005
skin segmentation-22774	122.4099	68.2069	46.8976	18554	1851	11331	0.0066	0.0368	0.0041
adult-26048	131.5644	73.1807	135.874	21236	20950	21172	0.0062	0.0035	0.0064

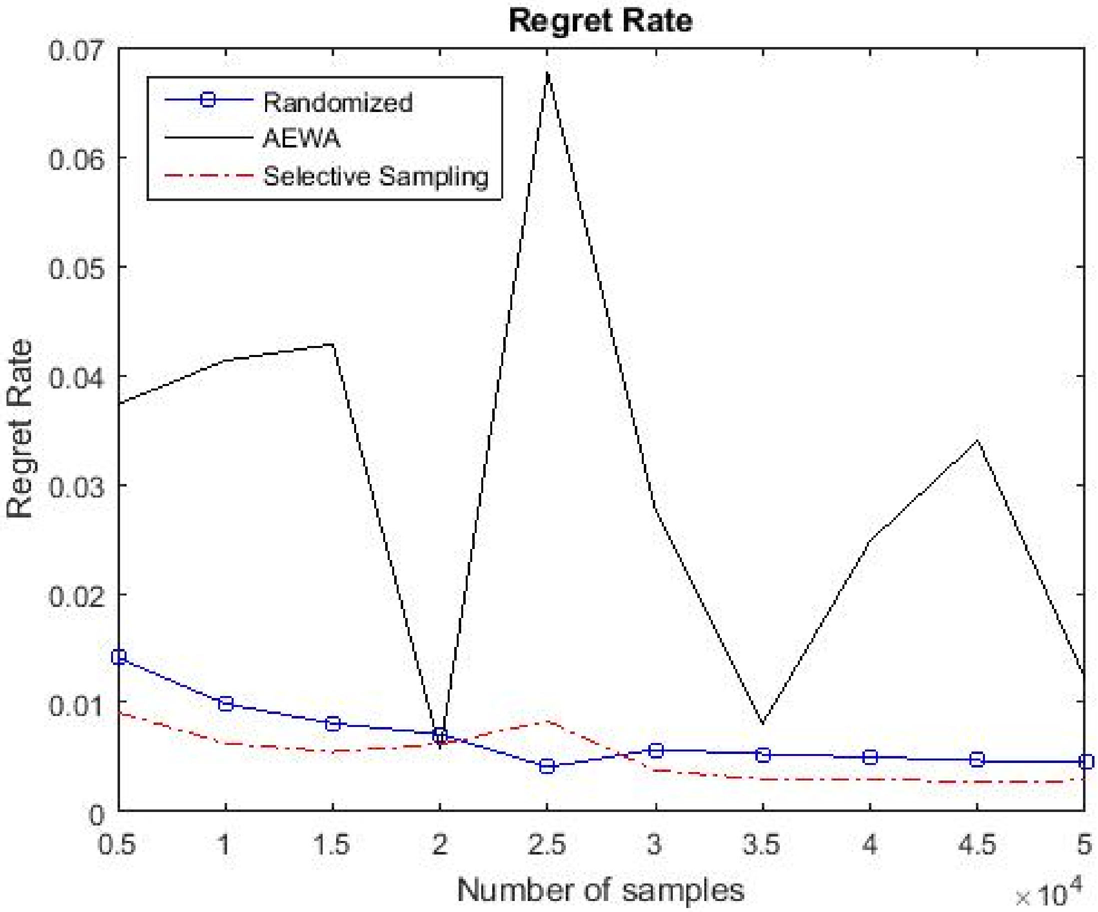

Figure 4.3: Regret rate of R, AEWA, and EPAL on synthetic datasets.

CHAPTER 5

SIMPLE REGRET IN SLEEPING MULTIARMED BANDIT

In this chapter, we consider the product recommendation problem in which there is a collection of products whose rewards or profits are unknown, and the goal is to select the best product to recommend to users after a number of sampling. We investigate a general setting where we assume that at each time, there is a subset of arms available, each of them has a reward that follows from some fixed, but unknown distribution. We propose UCB based (Upper Confidence Bound) algorithms that can provide different ways to tune the parameters based on the availability of each arm in the collection. We also propose a simple, yet efficient, uniform sampling algorithm for this problem. We proved that all above algorithms end up with recommend the best arm in the sense that the error of selecting the incorrect arm converges exponentially by time.

5.1 Notations and Problem Formulation

Denote $\mathbb{S} = \{1, 2, ..., K\}$ as the set of K arms, and $\mathbb{S}_t \subseteq \mathbb{S}$ as the set of available arms at time t. In this paper, we consider the stochastic availability, i.e., we assume that $\mathbb{S}_t$ is drawn from a fixed, but unknown distribution. Assume also that the reward of each arm i is drawn from a fixed, but unknown, distribution with the mean μ_i. For simplicity, we assume that all rewards are bounded in $[0, 1]$. Without loss of generality, we assume that $\mu_1 \geq \mu_2 \geq ... \geq \mu_K$, i.e., the set of arms have been already sorted in the descending order of their mean. At time t, denote $\mu_t^* = \max_{i \in \mathbb{S}_t} \mu_i$ as the current best arm. Define the advantage (or difference) of arm i over arm j as $\Delta_{i,j} := \mu_i - \mu_j$. This term takes an important role in deriving our simple regret in the sequel.

To simplify further analysis, we introduce the following notations. Denote $S^i = \{S : i \in S \text{ and } i \leq j \ \forall j \in S\}$ as the collection of subsets which have i as the best arm, and $\mathbb{T}^i = \{t : i \in S_t \text{ and } S_t \in S^i\}$ as the collection of times that arm i is the best available arm. We also denote t_i, t_{ij} as the final time in $\mathbb{T}^i$ and the final time within $\mathbb{T}^i$ that arm j is chosen instead of i, respectively. Define K_i as the total number of available arms in the set $\mathbb{T}^i$. We note that $|\mathbb{T}^i| = Tq_i$, where q_i is the probability that arm i is the leading arm of any subset.

We abuse the notation a little bit by denoting $T_j^i(t)$ as total number of times the arm j is chosen up to time t whenever the arm i is the best available arm.
The arm recommendation procedure is conducted as follows. First, the adversary chooses a set $\mathbb{S}_t$ of available arms based on the distribution mentioned above. Upon updating the indices of arms from previous rounds, the algorithm selects an arm and observes the corresponding reward. Note that the rewards of other arms are not given in this multiarm bandit setting. Finally, the algorithm recommends an arm (or product) to the users. In our problem, we aim to minimize the regret of the recommendation of the algorithm with the best possible ordering of arms.

5.2 UCB Algorithms

In this section, we introduce two types of algorithms adapted from UCB. Those algorithms utilize different ways to set the parameters in upper confidence bound.

5.2.1 Available set based parameter selection

The algorithm is given in Algorithm 7.

This is a generalized version of UCB-E algorithm in [189] where there is a subset $\mathbb{S}_t \subseteq \mathbb{S}$ of arms available for choosing. The algorithm chooses an arm with the highest index at any time. If an arm has never been played before the algorithm selects that arm first. After an arm is chosen, its reward is observed by the algorithm, which then is applied to update the emperical mean for that arm. At the end of the procedure, the algorithm recommends the best emperical arm which is available. In this algorithm, the selection of constant $b_i^{S_t}$ relies on the knowledge of available arms distribution. Specifically, denote p_S as the probability that the subset S is available, the constants are chosen such that $b_i^S \le \frac{p_S(T-K/p_S)}{\frac{(1+c)^2}{(1-c)^2}H_i^S}$, where $H_S^i = \sum_{j\in S}\frac{1}{\Delta_{i,j}^2}$.

Theorem 10. *SRSB-AS algorithm provides the upper bound of probability of error at:*
$\sum_{i=1}^{K}\sum_{S\in S^i} p(S)2(Tp_S)|S|exp\left(-2c^2\frac{Tp_S-K}{\frac{(1+c)^2}{(1-c)^2}H}\right)$

Proof. Define the event $\xi = \{\forall i \in [K], \forall u \in [T], \forall S \in S^i, \forall j \in S, |\hat{\mu}_{u,j} - \mu_j| < c\sqrt{\frac{b_j^S}{u}}\}$
We need to prove that from the event ξ, we are able to select the best arm after T. Equiva-

Algorithm 7 SRSB-AS: Simple-Regret minimization in Sleeping Bandits - Available Set based

```
Input: Set of arms $\mathbb{S} = \{1, ..., K\}$.
Parameters: Exploration constant $b_i^{S_t}$.
Initialize: $B_{i,0} = +\infty$ and $T_i(0) = 0$ for i = 1,...,K.
for each round $t = 1, 2, ..., T$ do
    Adversary draws an available subset $S_t$ from a certain distribution
    Algorithm calculates the indices of all arms $i \in \mathbb{S}_t$,
        $$B_{i,t} = \begin{cases} \hat{\mu}_{i,T_i(t-1)} + \sqrt{\frac{b_i^{S_t}}{T_i(t-1)}} & \text{if } T_i(t-1) > 0, \\ +\infty & \text{otherwise ,} \end{cases} \quad (5.1)$$
    Select the arm: $I_t = \arg\max_{i \in \mathbb{S}_t} B_{i,t}$.
    Update
        $$\begin{cases} T_{I_t}(t) = T_{I_t}(t-1) + 1, \\ \hat{\mu}_{I_t,T_{I_t}(t)} = \frac{1}{T_{I_t}(t)} \sum_{s=1}^{T_{I_t}(t)} X_{I_t,s}. \end{cases} \quad (5.2)$$
end for
Recommend an arm $J_T = \arg\max_{i \in \mathbb{S}_T} \hat{\mu}_{i,T_i(T)}$
```

lently, our goal is to prove

$$T_j(T) \geq 4b_j^S c^2 \frac{1}{\Delta_{i,j}^2} \forall i \in [K], \forall S \in S^i, \forall j \in S.$$

At time t, we consider a subset $S \in S^i$, and $\forall j \in S$. Consider two following cases:

- If the suboptimal arm is chosen, i.e., $\mathbb{I}_t = j$. This implies that $\hat{\mu}_{j,T_j(t-1)} + \sqrt{\frac{b_j^S}{T_j(t-1)}} \geq \hat{\mu}_{i,T_i(t-1)} + \sqrt{\frac{b_i^S}{T_i(t-1)}}$. Since the event ξ holds true, we have $\mu_j + (1+c)\sqrt{\frac{b_j^S}{T_j(t-1)}} \geq \mu_i + (1-c)\sqrt{\frac{b_i^S}{T_i(t-1)}}$, then $(1+c)\sqrt{\frac{b_j^S}{T_j(t-1)}} \geq \Delta_{i,j} + (1-c)\sqrt{\frac{b_i^S}{T_i(t-1)}} \geq \Delta_{i,j}$.
  Since j is chosen, we obtain
  $$T_j(t) \leq (1+c)^2 \frac{b_j^S}{\Delta_{i,j}^2} + 1. \quad (5.3)$$

- If the best arm i is chosen in S, it implies $\mu_i + (1+c)\sqrt{\frac{b_i^S}{T_i(t-1)}} \geq \mu_j + (1-c)\sqrt{\frac{b_j^S}{T_j(t-1)}}$,

It follows that

$$T_j(t-1) \geq \frac{b_j^S(1-c)^2}{4} min\{\frac{1}{\Delta_{i,j}^2}, \frac{T_i(t)-1}{b_i^S(1+c)^2}\}. \tag{5.4}$$

Note that for the subset $S \in S^i$, the total number of arm selections is Tp_S, we obtain

$$\begin{aligned} T_i^S(T) - 1 &= Tp_S - 1 - \sum_{j \in S, j \neq i} \sum_{t \in \mathbb{T}^i} \{I_t = j\}, \\ &\geq Tp_S - 1 - \sum_{j \in S, j \neq i} T_j(t_{ij}), \\ &\geq Tp_S - 1 - (|S| - 1) - (1+c)^2 \sum_{j \in S, j \neq i} \frac{b_j^S}{\Delta_{i,j}^2}, \\ &\geq Tp_S - |S| - \frac{p_S(T - K/p_S)}{H_S^i} (\sum_{j \in S} \frac{1}{\Delta_{i,j}^2} - \frac{1}{(\Delta_i^*)^2}), \\ &\geq Tp_S - |S| - \frac{p_S(T - K/p_S)}{H_S^i} (H_S^i - \frac{1}{(\Delta_i^*)^2}), \\ &= \frac{p_S(T - K/p_S)}{H_S^i (\Delta_i^*)^2}, \\ &\geq \frac{b_i^S(1+c)^2}{(\Delta_i^*)^2} \forall S \in S^i, \end{aligned} \tag{5.5}$$

where in the first inequality, we use the definition of $T_j(t_{ij})$, the second inequality follows from (5.3), the next inequalities follow from the choice of constant b_j^S. Now, denote t_i^S as the final time that the arm i is available as the best arm in S and arm i is chosen at that time. From (5.4),

$$\begin{aligned} T_j(t_i^S - 1) &\geq \frac{b_j^S(1-c)^2}{4} min\{\frac{1}{\Delta_{i,j}^2}, \frac{T_i(t_i^S)-1}{b_i^S(1+c)^2}\}, \\ &\geq \frac{b_j^S(1-c)^2}{4} min\{\frac{1}{\Delta_{i,j}^2}, \frac{T_i^S(T)-1}{b_i^S(1+c)^2}\}, \\ &\geq \frac{b_j^S(1-c)^2}{4} \frac{1}{\Delta_{i,j}^2}, \\ &\geq \frac{4b_j^S c^2}{\Delta_{i,j}^2}, \end{aligned}$$

where the second last inequality follows from (5.5). It follows that $T_j(T) \geq \frac{4b_j^S c^2}{\Delta_{i,j}^2} \forall S \in$

$S^i, \forall j \in S$. The probability of error is then derived as

$$\begin{aligned} P(error) &= \sum_{i=1}^{K} \sum_{S \in S^i} p(S) P(\mu_{\pi_S} \neq \mu_S^*), \\ &\leq \sum_{i=1}^{K} \sum_{S \in S^i} p(S) 2(Tp_S)|S| exp\left(-2c^2 \frac{Tp_S - K}{\frac{(1+c)^2}{(1-c)^2} H}\right). \end{aligned}$$

□

5.2.2 Leading arm based parameter selection

From the previous subsection, one can observe from the error bound analysis that by increasing the constant b_i^S, we can to obtain a better upper bound for the UCB algorithm. Motivated by this, we change the constant as follows: for a subset S, and i is the best arm in S, $b_j^S \leq \frac{Tq_i - |S^i|}{\frac{(1+c)^2}{(1-c)^2} H_{S^i}}$ $\forall j \in S$, where $H_{S^i} = \sum_{j \in S^i} \frac{1}{\Delta_{i,j}^2}$. We name the algorithms using this constant as SRSB-LA.

Theorem 11. *SRSB-LA algorithm provides the upper bound of probability of error at:* $\sum_{i=1}^{K} \sum_{S \in S^i} p(S) 2(Tp_S)|S| exp\left(-2c^2 \frac{Tq_i - |S^i|}{\frac{(1+c)^2}{(1-c)^2} H_{S^i}}\right)$

Proof. Using the same derivation as in the proof of Theorem 13, we obtain the followings for every arm i and every subset $S \in S^i$,

$$\begin{aligned} T_i^i(T) - 1 &\geq Tq_i - 1 - \sum_{j \in S^i, j \neq i} T_j(t_{ij}), \\ &\geq Tq_i - 1 - (|S^i| - 1) - (1+c)^2 \sum_{j \in S^i, j \neq i} \frac{b_j^S}{\Delta_{i,j}^2}, \\ &\geq Tq_i - |S^i| - \sum_{j \in S^i, j \neq i} \frac{Tq_i - |S^i|}{H_{S^i}} \frac{1}{\Delta_{i,j}^2}, \\ &= Tq_i - |S^i| - \frac{Tq_i - |S^i|}{H_{S^i}} (H_{S^i} - \frac{1}{(\Delta_S^*)^2}), \\ &\geq \frac{Tq_i - |S^i|}{H_{S^i}} \frac{1}{(\Delta_S^*)^2}, \\ &\geq \frac{(1+c)^2 b_i^S}{(\Delta_S^*)^2}, \text{ forall } S \in S^i. \end{aligned} \tag{5.6}$$

Now, denote t_i as the final time that the arm i is available as the best arm and arm i is chosen at that time. From (5.4),

$$\begin{aligned}
T_j(t_i - 1) &\geq \frac{b_j^S(1-c)^2}{4} min\{\frac{1}{\Delta_{i,j}^2}, \frac{T_i(t_i) - 1}{b_i^S(1+c)^2}\}, \\
&\geq \frac{b_j^S(1-c)^2}{4} min\{\frac{1}{\Delta_{i,j}^2}, \frac{T_i^i(T) - 1}{b_i^S(1+c)^2}\}, \\
&\geq \frac{b_j^S(1-c)^2}{4} \frac{1}{\Delta_{i,j}^2}, \\
&\geq \frac{4b_j^S c^2}{\Delta_{i,j}^2}.
\end{aligned}$$

It follows that $T_j(T) \geq \frac{4b_j^S c^2}{\Delta_{i,j}^2} \forall S \in S^i, \forall j \in S$. Upper bound of probability of error:

$$\begin{aligned}
P(error) &= \sum_{i=1}^{K} \sum_{S \in S^i} p(S) P(\mu_{\pi_S} \neq \mu_S^*), \\
&\leq \sum_{i=1}^{K} \sum_{S \in S^i} p(S) 2(Tp_S)|S| exp\left(-2c^2 \frac{Tq_i - |S^i|}{\frac{(1+c)^2}{(1-c)^2} H_{S^i}}\right).
\end{aligned}$$

□

5.3 Uniform sampling

In this section, we consider a simple algorithm which chooses arms evenly within every subset. Specifically, for an available subset S, the algorithm selects each arm in S equal number of times. The algorithm is given in Algorithm 8.

Theorem 12. *The SRSB-U obtain the following probability of error:*
$$\sum_{i=1}^{K} \sum_{S \in S^i} p(S) \sum_{j \neq i} exp\big(-\frac{\Delta_{i,j}^2}{2} \sum_{S: i \in S} \left\lfloor \frac{Tp_S}{|S|} \right\rfloor \big) + exp\big(-\frac{\Delta_{i,j}^2}{2} \sum_{S: j \in S} \left\lfloor \frac{Tp_S}{|S|} \right\rfloor \big)$$

Algorithm 8 SRSB-U : SRSB - Uniform Sampking

```
Input: Set of arms $\mathbb{S} = \{1, ..., K\}$.
Initialize: $T_i(0) = 0$ for i = 1,...,K, and $T_S(0) = 0$ for $S \in \{1, ..., K\}$.
for each round $t = 1, 2, ..., T$ do
    Adversary draws in available subset $S$
    Algorithm selects the arm $I_t = (T_S(t) mod |S|)$
    Update
        $\begin{cases} T_{I_t}(t) = T_{I_t}(t-1) + 1, \\ T_S(t) = T_S(t-1) + 1, \\ \hat{\mu}_{I_t, T_{I_t}(t)} = \frac{1}{T_{I_t}(t)} \sum_{s=1}^{T_{I_t}(t)} X_{I_t, s}. \end{cases} \quad (5.7)$
end for
Recommend an arm $J_T = \arg\max_{i \in \mathbb{S}_T} \hat{\mu}_{i, T_i(T)}$
```

Proof. The error of this algorithm is derived as follows,

$$
\begin{aligned}
P(err) &= \sum_{i=1}^{K} \sum_{S \in S^i} p(S) \sum_{j \neq i} P(I_T = j), \\
&= \sum_{i=1}^{K} \sum_{S \in S^i} p(S) \sum_{j \neq i} P(\hat{\mu}_j \geq \hat{\mu}_i), \\
&= \sum_{i=1}^{K} \sum_{S \in S^i} p(S) \sum_{j \neq i} P\Big(\frac{1}{T_j(T)}\Big(\sum_{s=1}^{T_j(T)} X_{j,s} - T_j(T)\mu_j\Big) - \frac{1}{T_i(T)}\Big(\sum_{s=1}^{T_i(T)} X_{i,s} - T_i(T)\mu_i\Big) \geq \Delta_{i,j}\Big), \\
&\leq \sum_{i=1}^{K} \sum_{S \in S^i} p(S) \sum_{j \neq i} P\Big(\frac{1}{T_j(T)}\Big(\sum_{s=1}^{T_j(T)} X_{j,s} - T_j(T)\mu_j\Big) + \frac{1}{T_i(T)}\Big(\sum_{s=1}^{T_i(T)} X_{i,s} - T_i(T)\mu_i\Big) \geq \Delta_{i,j}\Big), \\
&\leq \sum_{i=1}^{K} \sum_{S \in S^i} p(S) \sum_{j \neq i} P\Big(\frac{1}{T_j(T)}\Big(\sum_{s=1}^{T_j(T)} X_{j,s} - T_j(T)\mu_j\Big) \geq \frac{\Delta_{i,j}}{2}\Big) \\
&+ P\Big(\frac{1}{T_i(T)}\Big(\sum_{s=1}^{T_i(T)} X_{i,s} - T_i(T)\mu_i\Big) \geq \frac{\Delta_{i,j}}{2}\Big), \\
&\leq \sum_{i=1}^{K} \sum_{S \in S^i} p(S) \sum_{j \neq i} exp\Big(-\frac{T_j(T)\Delta_{i,j}^2}{2}\Big) + exp\Big(-\frac{T_i(T)\Delta_{i,j}^2}{2}\Big), \\
&\leq \sum_{i=1}^{K} \sum_{S \in S^i} p(S) \sum_{j \neq i} exp\Big(-\frac{\Delta_{i,j}^2}{2} \sum_{S : i \in S} \Big\lfloor \frac{Tp_S}{|S|} \Big\rfloor\Big) + exp\Big(-\frac{\Delta_{i,j}^2}{2} \sum_{S : j \in S} \Big\lfloor \frac{Tp_S}{|S|} \Big\rfloor\Big).
\end{aligned}
$$

where the second last inequality follows from the Hoeffding's inequality, and the last inequality follows from the uniform sampling algorithm. □

5.4 Adversarial availability

In this section, we relax the stochastic assumption on the availability of the arms. To define the regret for this setting, let us denote $\mathbb{I}_{i,j}$ as the indicator function of choosing the arm j when some arm $1, .., i$ could have been selected. The simple regret is defined as follow:

$$r_n = \mathbb{E}\left[\sum_{j=2}^{K}\sum_{i=1}^{j-1}(\mathbb{I}_{i,j} - \mathbb{I}_{i-1,j})\Delta_{i,j}\right].$$

By regrouping terms inside the sum, we get,

$$\begin{aligned} r_n &= \mathbb{E}\left[\sum_{j=2}^{K}\sum_{i=1}^{j-1}\mathbb{I}_{i,j}(\Delta_{i,j} - \Delta_{i+1,j})\right], \\ &= \mathbb{E}\left[\sum_{j=2}^{K}\sum_{i=1}^{j-1}\mathbb{I}_{i,j}\Delta_{i,i+1}\right]. \end{aligned} \tag{5.8}$$

The following lemma is important in our proof of the main theorem.

Lemma 7. *For any distribution* $(a_1, a_2, ..., a_K)$ *over* K *arms, and any* $l > 0$,

$$P(T_i(t-1) \geq l, I_t = i) \leq 2te^{-2b},$$

where $l \geq \frac{4b}{\Delta^2_{i-1,i}}$.

Proof. The proof is a slight modification from [?]. Since the arm i is chosen at t only if its index is better than the current best arm,

$$\begin{aligned} P(T_i(t-1) \geq l, \mathbb{I}_t = i) &\leq P\left(\hat{\mu}_{i,t} + \sqrt{\frac{b}{T_i(t-1)}} \geq \hat{\mu}_{i^*_t,t} + \sqrt{\frac{b}{T_{i^*_t}(t-1)}}, T_i(t-1) \geq l\right), \\ &\leq P\left(\max_{l \leq u \leq t}\hat{\mu}_{i,u} + \sqrt{\frac{b}{u}} \geq \min_{1 \leq s \leq t}\hat{\mu}_{i^*_s,s} + \sqrt{\frac{b}{s}}\right). \end{aligned} \tag{5.9}$$

We observe that $\hat{\mu}_{i,u} + \sqrt{\frac{b}{u}} \geq \hat{\mu}_{i^*_s,s} + \sqrt{\frac{b}{s}}$ when one of these events happens,
$\hat{\mu}_{i,u} \geq \mu_i + \sqrt{\frac{b}{u}}$,
$\mu_i \geq \mu_{i^*_s} - 2\sqrt{\frac{b}{u}}$,

$\mu_{i_s^*} \geq \hat{\mu}_{i_s^*,s} + \sqrt{\frac{b}{s}}$.

From the choice of l, $u > \frac{4b}{\Delta^2_{i_s^*,i}}$. It follows that $\sqrt{\frac{b}{u}} < \frac{1}{2}\Delta_{i_s^*,i}$, and then $P(\mu_i \geq \mu_{i_s^*} - 2\sqrt{\frac{b}{u}}) = 0$. Therefore, from (5.9) and apply the Chernoff-Hoeffding bound, we obtain

$$\begin{aligned} P(T_i(t-1) \geq l, \mathbb{I}_t = i) &\leq \sum_{u=l}^{t} P\left(\hat{\mu}_{i,u} \geq \mu_i + \sqrt{\frac{b}{u}}\right) + \sum_{s=1}^{t} P\left(\mu_{i_s^*} \geq \hat{\mu}_{i_s^*,s} + \sqrt{\frac{b}{s}}\right), \\ &\leq 2te^{-2b}. \end{aligned}$$

□

Theorem 13. *For any distribution $(a_1, a_2, ..., a_K)$ over K arms,*

$$r_n \leq Kn^2(\sum_{j=2}^{K} \Delta_{1,j})e^{-2b},$$

where n is large enough such that $\lceil a_i n \rceil - 1 \geq \frac{4b}{\Delta^2_{i-1,i}}$.

Proof. We start the proof with an observation that if an arm is recommended at n, there must be at least one arm i satisfying $T_i(n) \geq a_i n$. We will bound the probability of recommending arm j using this fact.

$$\begin{aligned} P_{i,j} &\leq \sum_{i=1}^{K} P(T_i(n) \geq a_i n), \\ &\leq \sum_{i=1}^{K} P(T_i(n) \geq \lceil a_i n \rceil), \\ &\leq \sum_{i=1}^{K} \sum_{t=\lceil a_i(n) \rceil}^{n} P(T_i(t-1) \geq \lceil a_i n \rceil - 1, I_t = i), \qquad (5.10) \\ &\leq \sum_{i=1}^{K} \sum_{t=\lceil a_i(n) \rceil}^{n} 2te^{-2b}, \qquad (5.11) \\ &\leq \sum_{i=1}^{K} \int_{t=\lceil a_i(n) \rceil}^{n} 2te^{-2b} dt, \\ &\leq Kn^2 e^{-2b}. \qquad (5.12) \end{aligned}$$

where (5.10) follows from the union bound, (5.11) follows from Lemma 7. From (5.8) and

(5.12), we obtain

$$\begin{aligned} r_n &\le \sum_{j=2}^{K}\sum_{i=1}^{j-1} Kn^2 e^{-2b}\Delta_{i,i+1}, \\ &= Kn^2\big(\sum_{j=2}^{K}\Delta_{1,j}\big)e^{-2b}. \end{aligned}$$

□

5.5 Experimental Results

In this section, we conduct a simulation running different algorithms in the general setting of sleeping bandit. A synthetic dataset is generated which includes 7 arms whose frequency of availability is 0.5, from that, we generate the distribution of available set of arms at each time. Six following algorithms are run to compare the performance.

- SRSB-AS: UCB algorithm with the parameter selection based on the available set.
- SRSB-AS-apprx: SRSB-AS algorithm, but relaxes the knowledge of the sampling budget by choosing the constant $b_j^S \le \frac{tp_S}{\frac{(1+c)^2}{(1-c)^2}H_{S^i}}$ $\forall j \in S$
- SRSB-LA: UCB algorithm with the parameter selection based on the leading arm
- SRSB-AA: UCB algorithm with the parameter selection based on the available arm. In this algorithm, the constant is selected such that $b_j^S \le \frac{Tp_i - |S^i|}{\frac{(1+c)^2}{(1-c)^2}H_{S^i}}$ $\forall j \in S$. Note that with this algorithm, the selection of parameter does not depends on the knowledge of p_S.
- SRSB-AA-apprx: SRSB-AA algorithm, but relaxes the knowledge of the sampling budget by choosing $b_j^S \le \frac{tp_i}{\frac{(1+c)^2}{(1-c)^2}H_{S^i}}$ $\forall j \in S$
- SRSB-U: Uniform algorithm

We run the algorithms in a range of T, from 50000 to 54500, where the mean of arms are given as $[0.7, 0.63, 0.55, 0.5, 0.45, 0.39, 0.3]$. Figure 5.1 shows the comparison of error among the algorithms.

One can observe from Figure 5.1 that SRSB-AA, SRSB-AS, and SRSB-U outperform the others. This is expected since these algorithms incorporate full information from the

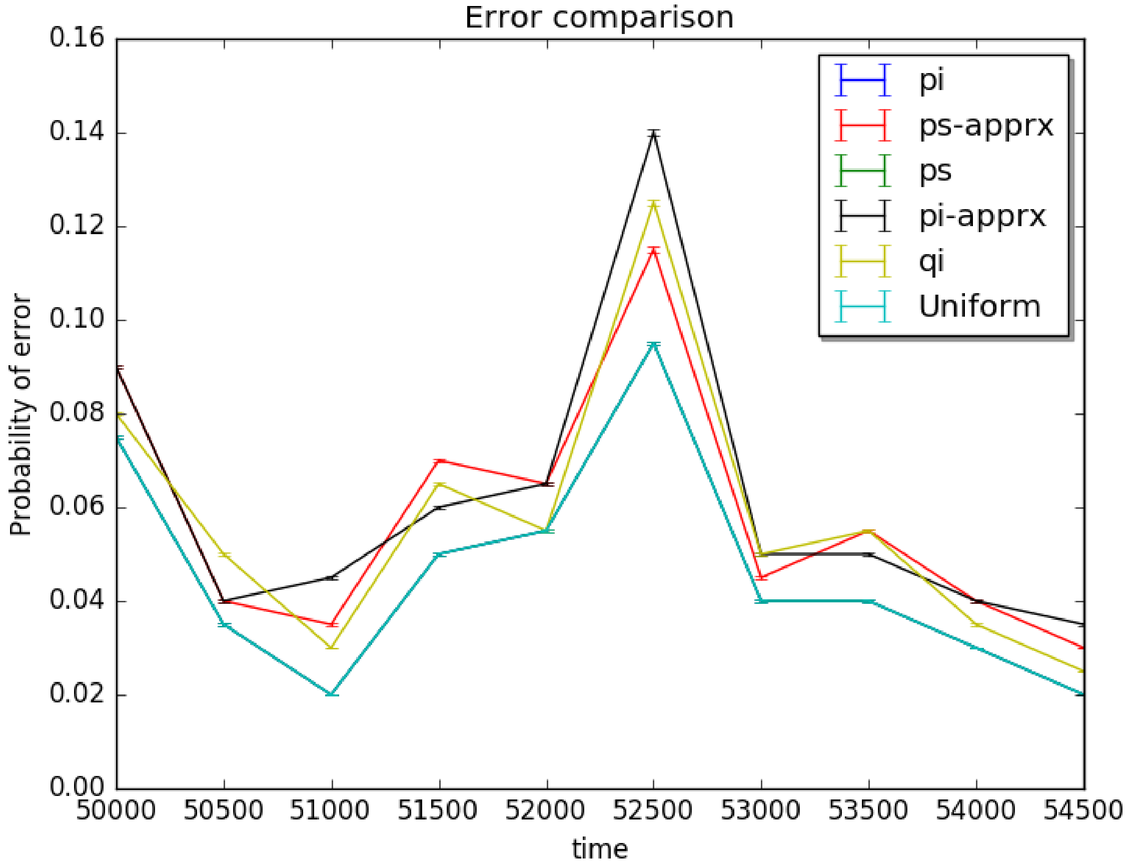

Figure 5.1: Comparison of error among algorithms

sampling budget into the learning procedure. On the other hand, SRSB-AA-apprx, SRSB-AS-apprx, and SRSB-LA perform slight worse than the above algorithms, but they are very flexible in terms of relaxing the required knowledge about the sampling budget. They can be widely used for the applications that needs the so-call “any time” algorithms.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

We proposed algorithms for the framework of learning with expert advice under various settings and applications. In the scenarios when not all experts are available at all time, we provided weight update rules, from that experts are not only rewarded by correct predictions but also encouraged to vote frequently. The proposed algorithms are proved to converge to the best experts defined in terms of their accuracy and availability. In the adversarial settings where malicious experts might intentionally ruin the system, we found the optimal attacking strategies for those experts with two kinds of losses, absolute loss and logarithmic loss, under finite discounted and infinite horizon settings. We also extend the results with more experts using mean-field theory. In the active learning application using expert advice, we proposed two algorithms to efficiently select objects to be labeled based on the ranges of experts' predictions, and found nearly optimal regret bounds for those algorithms. In the last setting, we consider the simple recommendation scenario where after a number of trials, a product is suggested to a user. We proposed two algorithms, UCB-extended algorithms and uniform algorithms, for the general setting where not all products are available to pick up at all time. Those algorithms are proved to decay exponentially over the number of samplings. In the next section, we propose some potential furure directions to develop further our results.

6.2 Future Directions

- Sleeping experts setting: We believe that using the stochastic approximation is one of the interesting directions for the analysis of learning with expert advice framework, especially for the stochastic setting. Further usage of this approach in other settings should be our next steps.

- Adversarial setting: As a future direction, one could generalize our result to the N-

expert setting for the absolute loss, with or without discounting factor. For example, relaxing the condition on the discounted factor in Theorem 4, establishing results in the presence of a time-varying learning rate of the exponentially weighted averaging algorithm in light of what has been discussed in [82], and applying the neural network to reduce the complexity and improve the convergence rate of the policy iteration are other interesting problems. Another potential direction is to consider the setting as a game-theoretic problem where players are algorithms and malicious experts (may collude with each other). Finding solutions for such a problem with multiple experts is still an open problem.

- Simple regret setting: There are still opportunities to improve the performance of recommendation algorithms, either by tuning the parameters for UCB-typed algorithms, or by applying different set of algorithms, e.g., the ones given in [190].

APPENDIX A

PROOFS OF THEOREMS

A.1 Proofs of Chapter 2

A.1.1 Proof of Lemma 1

Proof. Based on Theorem 1, we only need to show that all the conditions **(A1-4)** are satisfied for dynamics (2.8). For an appropriately selected step size, e.g., $a(t) = \frac{1}{1+t}$, assumption **(A2)** clearly holds. Moreover, **(A4)** also holds as each weight $p_t^i, i = 1, \ldots, N$ is nonnegative and never exceeds 1. To check Lipschitz condition **(A1)**, we show a sufficient condition by showing that $\sup_{\vec{p} \in \mathbb{R}^n} \|\nabla_{\vec{p}} h(\vec{p})\| < L$, for some constant L. From the definition of $h(\cdot)$, we have $h_i(\vec{p}) = p_i(c_i - \sum\limits_{j \in E} c_j p_j)$. Since for any $i, k \in \{1, \ldots, N\}$, we have

$$\left|\frac{\partial h_i}{\partial p_k}\right| = \begin{cases} |c_i - \sum\limits_j c_j p_j - c_i p_i| & \text{if } i = k, \\ |c_k p_i| & \text{if } i \neq k, \end{cases}$$

we get $\left|\frac{\partial h_i}{\partial p_k}\right| \leq N + 2$ (note that we always have $r_t^i, p_t^i \in [0,1], \forall t, i$, and so $c_i \in [0,1], \forall i$). Now, we can write

$$\|\nabla_{\vec{p}} h(\vec{p})\| \leq \sum_{i=1}^{N} \|\nabla_{\vec{p}} h_i(\vec{p})\| \leq \sum_{i=1}^{N} \sum_{k=1}^{N} \left|\frac{\partial h_i}{\partial p_k}\right| \leq N^2(N+2).$$

This shows that $h(\cdot)$ is Lipschitz with a constant $L \leq N^2(N+2)$.

Finally to check **(A3)**, one can show that

$$\begin{aligned}
E\left[(M_t^i)^2|F_{t-1}\right] &= (p_{t-1}^i)^2\left[E((I\{i\in E_t\}(r_t^i-1/2))^2)+\sum_j E((r_t^j-1/2)^2)(p_{t-1}^j)^2\right] \\
&+ (p_{t-1}^i)^2\left[2\sum_{j\neq k} E[I\{j\in E_t\}(r_t^j-1/2)]E(I\{k\in E_t\}(r_t^k-1/2))p_{t-1}^j p_{t-1}^k\right] \\
&- (p_{t-1}^i)^2\left[\left(\sum_j E[I\{j\in E_t\}(r_t^j-1/2)]p_{t-1}^j\right)^2\right] \\
&- (p_{t-1}^i)^2\left[\big(E[I\{i\in E_t\}(r_t^i-1/2)]\big)^2\right] \\
&\leq (p_{t-1}^i)^2\left(1+N+N(N-1)+N^2+1\right),
\end{aligned}$$

where the last inequality again uses the unity bound on r_t^i, and p_{t-1}^i. Assumption **(A3)** is satisfied immediately following this inequality. □

A.1.2 Proof of Theorem 2

Proof. First, note that if we initially set the weights as $1/N$ for all experts, then $\sum\limits_{i\in E} p_t^i = 1$ for all t. To see this, let us expand the sum as follows,

$$\begin{aligned}
\sum_{i\in E} p_t^i &= \sum_{i\in E} p_{t-1}^i + a(t)\sum_{i\in E_t} p_{t-1}^i\left[(r_t^i-1/2)-\sum_{j\in E_t} p_{t-1}^j(r_t^j-1/2)\right] \\
&\quad + a(t)\sum_{i\notin E_t} p_{t-1}^i\left[-\sum_{j\in E_t} p_{t-1}^j(r_t^j-1/2)\right], \\
&= \sum_{i\in E} p_{t-1}^i + a(t)\left[\sum_{j\in E_t} p_{t-1}^j(r_t^j-1/2)\right]\left[1-\sum_{i\in E} p_{t-1}^i\right] \\
&= \sum_{i\in E} p_{t-1}^i.
\end{aligned}$$

Using induction on t, the result is immediate. Next, we consider the two following cases:

1. There is only one best expert which we denote it by $i = \arg\max\limits_{j\in E} c_j$ where we recall that $c_j := \mathbb{E}[I\{j\in E_t\}(r_t^j - 1/2)]$. In this case, since $\sum\limits_{i\in E} \rho^i(s) = 1$, we must have $\sum\limits_{i\in E} \dot{\rho}^i(s) = 0$. Moreover, initially we have $\dot{\rho}^i(0) > 0$. This is because for initial weights $\rho(0) =$

$(\frac{1}{N}, \ldots, \frac{1}{N})$, the right-hand side of equation (2.9) given by $\rho^i(s)\left(c_i - \sum\limits_{j\in E} c_j\rho^j(s)\right)$ is strictly positive. Therefore, as time goes by, it always remains positive, i.e., $\dot{\rho}^i(s) > 0$. This in turn implies that $\sum\limits_{j\neq i} \dot{\rho}^j(s) < 0$. In other words, $\rho^i(s)$ increases to a constant $K \leq 1$, while $\sum\limits_{j\neq i} \rho^j(s)$ decreases to $1 - K$. To reach a contradiction, let us assume that $K \neq 1$. Then, the right-hand side of (2.9) is always positive. Consequently, $\dot{\rho}^i(s) > 0$ for all subsequent time which implies $\rho^i(s)$ will be unbounded, which is a contradiction. This shows that $\lim_{s\to\infty} \rho^i(s) = 1$ a.s. In other words, the system asymptotically follows only the best expert i.

2. If the best expert in Definition 7 is not unique, say at least two best experts i and k exist such that $c_i = c_k > c_j, j \neq i, k$, then by the same argument as above, these two experts have weights satisfying: $\rho^i(s) + \rho^k(s) \to 1$ while $\rho^j(s) \to 0$ for all $j \neq i, k$. Thus, the system alternates between these two experts (probability of the case where both two weights are the same is zero).

□

A.1.3 Proof of Corollary 1

Proof. By the same argument as in the proof Theorem 2, one can show that Algorithm 2 converges to the following expert:

$$i^* = \arg\max_{i\in E} \mathbb{E}(u_t^i).$$

Moreover, from (2.15) we have $\mathbb{E}(u_t^i) = \mathbb{E}\left[I\{i \in E_t\}(1 - l(x_t^i)) + I\{i \notin E_t\}(1 - l(c))\right]$. It follows that

$$\begin{aligned} i^* &= \arg\max_{i\in E} \mathbb{E}\left[1 - I\{i \in E_t\}l(x_t^i) - I\{i \notin E_t\}l(c)\right], \\ &= \arg\min_{i\in E} \mathbb{E}\left[I\{i \in E_t\}l(x_t^i) + I\{i \notin E_t\}l(c)\right]. \end{aligned}$$

□

A.2 Proofs of Chapter 3

A.2.1 Proof of Lemma 2

Proof. By induction on k we show that the expected loss function defined in (3.5) is only a function of weight vector $\vec{p}$, the horizon length K, and the strategy of the adversary $\pi := (x_1^1, \ldots, x_K^1) \in \{T, L\}^K$. For $k = K$, the statement becomes trivial. Let us assume that at the stage $s+1$ the statement is correct and denote the expected loss of the system for a policy π of the adversary by $V_{s+1}^{\pi}(x) := \sum_{k=s+1}^{K} \mathbb{E}_{x_k^2,\ldots,x_k^N}[l(\hat{y}_k, y_k)]$. Now depending on whether the adversary lies or tells the truth in the first stage we can write

$$\begin{aligned}&\sum_{k=s}^{K} \mathbb{E}[l(\hat{y}_k, y_k)] = \\ &= \begin{cases} \mathbb{E}[l(\hat{y}_1, y_1)|x_1^1 = L] + \mathbb{E}[V_{s+1}^{\pi}(\phi_L(\vec{p}))] & \text{if } \; x_1^1 = L, \\ \mathbb{E}[l(\hat{y}_1, y_1)|x_1^1 = T] + \mathbb{E}[V_{s+1}^{\pi}(\phi_T(\vec{p}))] & \text{if, } \; x_1^1 = T. \end{cases}\end{aligned} \tag{A.1}$$

We consider two cases:

- **Case I**: $x_1^1 = L$. In this case if we have $y_1 = 1$, then it implies that $x_1^1 = 0$. Moreover, since $x_1^i = y_1$ with probability μ_i, we have $x_1^i = Ber(\mu_i), i = 2, \ldots, N$. Thus

$$\begin{aligned}&\mathbb{E}[l(\hat{y}_1, y_1)|x_1^1 = L] = \mathbb{E}[l(\hat{y}_1, 1)|x_1^1 = 0] \\ &= \mathbb{E}[Q(|\hat{y}_1 - 1|)|x_1^1 = 0] \\ &= \mathbb{E}_{x_1^i}\left[Q\left(\left|\frac{p_1^1 \times 0 + p_1^2 x_1^2 + \ldots + p_1^N x_1^N}{p_1^1 + p_1^2 + \ldots + p_1^N} - 1\right|\right)\right] \\ &= \mathbb{E}_{x_1^i}\left[Q\left(\left|\frac{p_1^1 + p_1^2(1 - x_1^2) + \ldots + p_1^N(1 - x_1^N)}{p_1^1 + p_1^2 + \ldots + p_1^N}\right|\right)\right] \\ &:= f(\vec{p}),\end{aligned}$$

  where $f(\vec{p})$ is some function which only depends on $\vec{p}$. On the other hand, if $y_1 = 0$, then $x_1^1 = 1$. In this case denoting the prediction of honest experts by $z_1^2, \ldots, z_1^N$, we have $z_1^i = Ber(1 - \mu_i), i = 2, \ldots, N$ due to the fact that $z_1^i = y_1$ with probability μ_i.

Now we can write

$$\begin{aligned}
&\mathbb{E}[l(\hat{y}_1, y_1)|x_1^1 = L] = \mathbb{E}[l(\hat{y}_1, 0)|x_1^1 = 1] \\
&= \mathbb{E}[Q(|\hat{y}_1 - 0|)|x_1^1 = 1] \\
&= \mathbb{E}_{z_1^i}[Q(|\frac{p_1^1 \times 1 + p_1^2 z_1^2 + \ldots + p_1^N z_1^N}{p_1^1 + p_1^2 + \ldots + p_1^N} - 0|)] \\
&= \mathbb{E}_{x_1^i}[Q(|\frac{p_1^1 + p_1^2(1 - x_1^2) + \ldots + p_1^N(1 - x_1^N)}{p_1^1 + p_1^2 + \ldots + p_1^N}|)] \\
&= f(\vec{p}),
\end{aligned}$$

where the second last equality is due to the fact that $z_1^i = 1 - x_1^i, i = 2, \ldots, N$.

- **Case II**: $x_1^1 = T$. In this case, similar to the Case I by considering two possibilities for $y_1 = 0$ or $y_1 = 1$, one can show that $\mathbb{E}[l(\hat{y}_1, y_1)|x_1^1 = T] = g(\vec{p})$ for some function $g(\cdot)$.

Therefore, independent of the actual value of y_1 and using (A.1) we can write

$$\sum_{k=s}^{K} \mathbb{E}_{x_k^2}[l(\hat{y}_k, y_k)] = \begin{cases} f(\vec{p}) + \mathbb{E}[V_{s+1}^{\pi}(\phi_L(\vec{p}))] & \text{if } \ x_1^1 = L, \\ g(\vec{p}) + \mathbb{E}[V_{s+1}^{\pi}(\phi_T(\vec{p}))] & \text{if, } \ x_1^1 = T \end{cases}$$

The above relation shows that indeed $\sum_{k=s}^{K} \mathbb{E}_{x_k^2}[l(\hat{y}_k, y_k)]$ is independent of actual values of $y_k, k \in [K]$, and is only a function of $\vec{p}$, π and K. □

A.2.2 Proof of Proposition 1

Proof. We prove the nontrivial case ($q_{k-1}^1 < p_{k-1}^1$) by induction. The base case: at the final step, i.e., $k = K - 1$, the result follows by properties **(P1)** and **(P2)**, and the fact that the terminal cost is zero. Next, we show if the result holds for step $k + 1$, it also holds for step k.

First, note that when the adversary takes the same action (lies or tells the truth) at both states $\vec{p}_{k-1}$ and $\vec{q}_{k-1}$ at time k, the updated weights (Algorithm 3) at time $k + 1$ will still satisfy $q_k^1 \leq p_k^1$. In this case, property **(P1)** and the induction hypothesis imply $V_k(\vec{q}_{k-1}, L) \leq V_k(\vec{p}_{k-1}, L) \leq \max\{V_k(\vec{p}_{k-1}, L), V_k(\vec{p}_{k-1}, T)\}$. It remains to show $V_k(y_k, \vec{q}_{k-1}, T) \leq \max\{V_k(y_k, \vec{p}_{k-1}, L), V_k(y_k, \vec{p}_{k-1}, T)\}$. Note that, from the weight update rule in Algorithm 3, if $q_{k-1}^1 < p_{k-1}^1$, we must have $q_{k-1}^1 \leq p_{k-1}^1 \epsilon$ for the case of logarithmic loss, and $q_{k-1}^1 \leq p_{k-1}^1 e^{-1}$ for the case of absolute loss. This observation along with properties **(P1)** and **(P2)**, and the induction hypothesis suffices to show $V_k(\vec{q}_{k-1}, T) \leq V_k(\vec{p}_{k-1}, L)$. □

A.2.3 Proof of Theorem 3

Proof. We will show by induction that telling lie at each stage is optimal. For the base case, i.e., when $t = K - 1$ the claim trivially holds due to the fact that the terminal cost is zero and the current cost satisfies property **(P1)**. Now suppose that telling lie is the optimal strategy for every $\vec{p}$ at step $t \geq k+1$. In particular, using the dynamic program recursion (3.8) we must have

$$V^*_{k+1}(\vec{p}) = c_L(\vec{p}) + \mathbb{E}V^*_{k+2}(\phi_L(\vec{p})),\ \forall \vec{p}. \tag{A.2}$$

We want to show that telling lie at stage $t = k$ is also optimal for any $\vec{p}$, i.e., $V_k(\vec{p}, L) > V_k(\vec{p}, T),\ \forall \vec{p}$. Using (3.8), this is equivalent to show that $\Delta V^*_{k+1}(\vec{p}) < \Delta c(\vec{p}), \forall \vec{p}$, where

$$\begin{aligned}\Delta c(\vec{p}) &:= c_L(\vec{p}) - c_T(\vec{p}),\\ \Delta V^*_{k+1}(\vec{p}) &:= \mathbb{E}V^*_{k+1}(\phi_T(\vec{p})) - \mathbb{E}V^*_{k+1}(\phi_L(\vec{p})).\end{aligned} \tag{A.3}$$

Now let R and R' be two random subsets of $\{2, 3, \dots, n\}$ denoting the set of indices of honest experts which are correct at instances k and $k+1$, respectively. Starting from (A.3) we can write

$$\begin{aligned}\Delta V^*_{k+1}(\vec{p}) &= \mathbb{E}\Big[V^*_{k+1}(\phi_T(\vec{p})) - V^*_{k+1}(\phi_L(\vec{p}))\Big]\\ &\overset{(a)}{=} \mathbb{E}\Big[c_L(\phi_T(\vec{p})) + \mathbb{E}V^*_{k+2}(\phi_{LT}(\vec{p})) - c_L(\phi_L(\vec{p})) - \mathbb{E}V^*_{k+2}(\phi_{LL}(\vec{p}))\Big]\\ &\overset{(b)}{=} \mathbb{E}\Big[c_L(\phi_T(\vec{p})) - c_L(\phi_L(\vec{p})) + \Delta V^*_{k+2}(\phi_L(\vec{p}))\Big]\\ &\overset{(c)}{<} \mathbb{E}\Big[c_L(\phi_T(\vec{p})) - c_L(\phi_L(\vec{p})) + \Delta c(\phi_L(\vec{p}))\Big]\\ &\overset{(d)}{=} \mathbb{E}\Big[c_L(\phi_T(\vec{p})) - c_T(\phi_L(\vec{p}))\Big]\\ &\overset{(e)}{=} \mathbb{E}_{R'}\Big[c_L\big((1-\epsilon)p^1, \vec{p}_{R'}\big) - c_T\big(\epsilon p^1, \vec{p}_{R'}\big)\Big],\end{aligned} \tag{A.4}$$

where (a) follows from the induction hypothesis given in (A.2), in which $\phi_{LT}(\vec{p})$ is the random weight vector (state) when the adversary first tells the truth at stage k, and lies after that in stage $k+1$. (b) is obtained using the definition (A.3) for stage $k+2$ and the observation that the weight of expert 1 in $\phi_{LT}(\vec{p})$ are the same as in $\phi_{TL}(\vec{p})$ and the other weights of honest experts are kept the same in two cases, (c) holds by the induction hypothesis, and (d) is valid by replacing the definition of $\Delta c(\phi_L(\vec{p}))$ given in (A.3). Finally, (e) is obtained by using the state update formula given in (3.9) over random subset realization of correct experts R', where we denote $\vec{p}_{R'} = (p^2_{R'}, \dots, p^N_{R'})$ is a random vector of size $N-1$ associated with the honest experts such that $p^i_{R'} = (1-\epsilon)p^i$ if $i \in R'$, and $p^i_{R'} = \epsilon p^i$, otherwise. Continuing with

(A.4) we can write

$$
\begin{aligned}
\Delta V^*_{k+1}(\vec{p}) &< \mathbb{E}_{R'}\Big[c_L\big((1-\epsilon)p^1, \vec{p}_{R'}\big) - c_T\big(\epsilon p^1, \vec{p}_{R'}\big)\Big] \\
&\overset{(g)}{=} -\mathbb{E}_{R',R}\left[\ln\left(\frac{\epsilon(1-\epsilon)p^1+(1-\epsilon)\sum\limits_{i\in R} p^i_{R'}+\epsilon\sum\limits_{j\in R^c} p^j_{R'}}{\epsilon p^1+\vec{p}_{R'}\mathbf{1}}\right)\right] \\
&\quad + \mathbb{E}_{R',R}\left[\ln\left(\frac{\epsilon(1-\epsilon)p^1+(1-\epsilon)\sum\limits_{i\in R} p^i_{R'}+\epsilon\sum\limits_{j\in R^c} p^j_{R'}}{(1-\epsilon)p^1+\vec{p}_{R'}\mathbf{1}}\right)\right] \\
&\overset{(h)}{=} \mathbb{E}_{R',R}\left[\ln\left(\frac{\epsilon p^1+\vec{p}_{R'}\mathbf{1}}{(1-\epsilon)p^1+\vec{p}_{R'}\mathbf{1}}\right)\right] \\
&\overset{(i)}{=} \mathbb{E}_{R'}\left[\ln\left(\frac{\epsilon p^1+\vec{p}_{R'}\mathbf{1}}{(1-\epsilon)p^1+\vec{p}_{R'}\mathbf{1}}\right)\right] \\
&\overset{(j)}{=} c_L(\vec{p}) - c_T(\vec{p}) = \Delta c(\vec{p}),
\end{aligned}
\tag{A.5}
$$

where (g) is obtained using (3.10), (h) follows by linearity of expectation and the fact that both the logarithmic terms have the same numerator, and (i) holds since the expression inside the expectation does not depend on the random set R. Finally, (j) is obtained by substituting the expressions of $c_T(\cdot)$ and $c_L(\cdot)$ given in (3.10). This completes the induction proof. □

A.2.4 Proof of Theorem 4

We first prove the following lemma which is important for Theorem 4.

Lemma 8. *For the discounted factor $\beta < \frac{1}{e}$, and any adversary's relative weight p we have*

$$c_L(p) - c_T(p) + \beta\mathbb{E}[c_T(\phi_L(p)) - c_L(\phi_T(p))] > 0.$$

Proof. Let $x = \frac{1}{1+(\frac{1}{p}-1)e}$ and $y = \frac{1}{1+(\frac{1}{p}-1)e^{-1}}$. Note that in particular $x < p < y$. Starting from the left hand side we can write

$$\begin{aligned}
LHS &= p + \beta\Big(\mu_2 c_T(x) + (1-\mu_2)c_T(p)\Big)\\
&\quad - \beta\Big(\mu_2 c_L(p) + (1-\mu_2)c_L(y)\Big)\\
&\geq p + \beta\Big(\mu_2 c_T(p) + (1-\mu_2)c_T(p)\Big)\\
&\quad - \beta\Big(\mu_2 c_L(y) + (1-\mu_2)c_L(y)\Big)\\
&= p - \beta\Big(c_L(y) - c_T(p)\Big)\\
&= (1-\beta+\beta\mu_2)p - \beta\mu_2 y,\\
&> (1-\beta)(1-\mu_2)p > 0,
\end{aligned}$$

where in the first inequality we have used the fact that $c_T(x) > c_T(p)$ and $c_L(y) > c_L(p)$. Finally, the second last inequality follows from the fact that $y < ep < \frac{p}{\beta}$. □

Proof. First, let us introduce some notations, which will be handy in our proof. Let p_m denote a realization of $\tilde{p}^1_{k-1}$, and let $\mathbb{A} = \{..., p_{m-1}, p_m, p_{m+1}, ...\}$ denote the state space of normalized weights of expert 1 in ascending order, i.e., for instance $p_{m-1} < p_m < p_{m+1}$. Define $DV^*_k(p_m) := V_k(p_m, L) - V_k(p_m, T)$, where we have denoted $V_k(p_m, L)$ and $V_k(p_m, T)$ as the value function that the adversary imposes on the system at the weight p_m and stage k, provided that he lies or tells the truth at that stage, respectively. We will show that for any p_m and $k \in \mathbb{N}$ there exists a positive constant $\alpha_k > 0$ (note that α_k in general can depend on p_m and k) such that

$$DV^*_k(p_{m+1}) \geq \alpha_k DV^*_k(p_m). \tag{A.6}$$

To see why (A.6) is sufficient to establish the threshold policy at stage k, we note that if $DV^*_k(p_{th}) \geq 0$ for some p_{th}, by repeatedly using (A.6) one can see that $DV^*_k(p_m) \geq 0, \forall p_m \geq p_{th}$. In other words, if at stage k, the optimal policy is to lie at the weight p_{th}, then for any $p_m \geq p_{th}$ the optimal policy is to lie as well. Similarly, if $DV^*_k(p_{th}) < 0$ for some p_{th}, using (A.6) one can see that $DV^*_k(p_m) < 0, \forall p_m \leq p_{th}$, which means that if telling the truth at stage k with relative weight p_{th} is optimal, then for any $p_m \leq p_{th}$ telling the truth will be optimal as well.

Next, we proceed to establish (A.6) using induction on k. For $k = K$, and using (3.13) we have $DV^*_1(p_{m+1}) = p_{m+1} \geq p_m = DV^*_1(p_m)$. Therefore, in this case we can easily set $\alpha_K = 1$. Now let us assume that (A.6) holds for all stages t when $t \geq k+1$, and denote the threshold weight at stage $k+1$ by p_{th} that is at stage $k+1$ for any $p < p_{th}$ the best strategy for the adversary is telling the truth. We will show that (A.6) holds at stage k. We consider two cases for p_m at stage k:

Case I: $p_m < p_{th}$

$$\begin{aligned}
&DV_k^*(p_{m+1}) - \alpha_k DV_k^*(p_m) \\
&= c_L(p_{m+1}) - c_T(p_{m+1}) - \alpha_k[c_L(p_m) - c_T(p_m)] \\
&\qquad + \beta\Big(\mathbb{E}V_{k+1}(\phi_L(p_{m+1})) - \mathbb{E}V_{k+1}(\phi_T(p_{m+1}))\Big) \\
&\qquad - \alpha_k\beta\Big(\mathbb{E}V_{k+1}(\phi_L(p_m)) - \mathbb{E}V_{k+1}(\phi_T(p_m))\Big) \\
&= c_L(p_{m+1}) - c_T(p_{m+1}) - \alpha_k[c_L(p_m) - c_T(p_m)] \\
&\qquad + \beta\mathbb{E}\Big[c_T(\phi_L(p_{m+1})) - c_T(\phi_T(p_{m+1}))\Big] \\
&\qquad - \alpha_k\beta\mathbb{E}\Big[c_T(\phi_L(p_m)) - c_T(\phi_T(p_m))\Big] \\
&\qquad + \beta^2\mathbb{E}\Big[V_{k+2}(\phi_{TL}(p_{m+1})) - V_{k+2}(\phi_{TT}(p_{m+1}))\Big] \\
&\qquad - \alpha_k\beta^2\mathbb{E}\Big[V_{k+2}(\phi_{TL}(p_m)) - V_{k+1}(\phi_{TT}(p_m)))\Big].
\end{aligned}$$

Herein, we used the induction hypothesis and the fact that p_m is less than the threshold p_{th}. By continuing the same procedure, we obtain

$$\begin{aligned}
&DV_k^*(p_{m+1}) - \alpha_k DV_k^*(p_m) \\
&= c_L(p_{m+1}) - c_T(p_{m+1}) - \alpha_k[c_L(p_m) - c_T(p_m)] \\
&\qquad + \beta\mathbb{E}\Big[c_T(\phi_L(p_{m+1})) - c_T(\phi_T(p_{m+1}))\Big] \\
&\qquad - \alpha_k\beta\mathbb{E}\Big[c_T(\phi_L(p_m)) - c_T(\phi_T(p_m))\Big] \\
&\qquad + \beta^2\mathbb{E}\Big[c_T(\phi_{TL}(p_{m+1})) - c_T(\phi_{TT}(p_{m+1}))\Big] \\
&\qquad - \alpha_k\beta^2\mathbb{E}\Big[c_T(\phi_{TL}(p_m)) - c_T(\phi_{TT}(p_m))\Big] + ... \\
&\qquad + \beta^{J-1}\mathbb{E}\Big[c_T\Big(\phi_{T^{(J-2)}L}(p_{m+1})\Big) - c_L\Big(\phi_{T^{(J-1)}}(p_{m+1})\Big)\Big] \\
&\qquad - \alpha_k\beta^{J-1}\mathbb{E}\Big[V_{k+J-1}\Big(\phi_{T^{(J-2)}L}(p_m)\Big) - V_{k+J-1}\Big(\phi_{T^{(J-1)}}(p_m)\Big)\Big] \\
&:= A(\alpha_k) + \alpha_k B, \qquad\qquad \text{(A.7)}
\end{aligned}$$

where J is the number of times that adversary tells the truth before he lies. Finally, in the last relation, we have defined

$$B := -\beta^{J-1}\mathbb{E}\Big[V_{k+J-1}\Big(\phi_{T^{(J-2)}L}(p_m)\Big) - V_{k+J-1}\Big(\phi_{T^{(J-1)}}(p_m)\Big)\Big]$$

and $A(\alpha_k)$ to be the remaining terms. Using Proposition 1, we know that $B > 0$ (note that $\phi_{T^{(J-2)}L}(p_m) \le \phi_{T^{(J-1)}}(p_m)$) and by letting

$$\alpha_k = \min_{\ell=0,\dots,J-2} \frac{\beta\mathbb{E}[c_T(\phi_{T^\ell L}(p_{m+1})) - c_T(\phi_{T^{\ell+1}}(p_{m+1}))]}{\mathbb{E}[c_L(\phi_{T^{\ell-1}L}(p_m)) - c_T(\phi_{T^\ell}(p_m))]},$$

one can see that not only $\alpha_k > 0$ (by Lemma 8, in the appendix), but also $A(\alpha_k) > 0$. Overall, by the above choice of α_k we have $A(\alpha_k) + \alpha_k B > 0$, which completes the induction proof for Case I. Note that in the definition of α_k, we assumed $\phi_{T^{-1}L}(p_m) = \phi_{T^0}(p_m) = p_m$.
Case II If $p_m \geq p_{th}$. In this case we can write

$$\begin{aligned}
&DV_k^*(p_{m+1}) - \alpha_k DV_k^*(p_m) = \\
&= c_L(p_{m+1}) - c_T(p_{m+1}) - \alpha_k \left(c_L(p_m) - c_T(p_m)\right) \\
&\quad + \beta\mathbb{E}[V_{k+1}(\phi_L(p_{m+1})) - V_{k+1}(\phi_T(p_{m+1}))] \\
&\quad - \alpha_k \beta\mathbb{E}[V_{k+1}(\phi_L(p_m)) - V_{k+1}(\phi_T(p_m))] \geq \\
&p_{m+1} - \alpha_k p_m + \beta\mathbb{E}[V_{k+1}(\phi_L(p_{m+1})) - V_{k+1}(\phi_T(p_{m+1}))] := G - \alpha_k H,
\end{aligned}$$

where the first inequality follows from Proposition 1, and we have defined

$$\begin{aligned}
&G := p_{m+1} + \beta\mathbb{E}[V_{k+1}(\phi_L(p_{m+1})) - V_{k+1}(\phi_T(p_{m+1}))], \\
&H := p_m.
\end{aligned}$$

In particular, by expanding the terms in G we can write

$$\begin{aligned}
G &= p_{m+1} - \beta\mathbb{E}[c_L(\phi_T(p_{m+1})) - c_L(\phi_L(p_{m+1}))] \\
&\quad - \beta^2\mathbb{E}[V_{k+2}(\phi_{LT}(p_{m+1})) - V_{k+2}(\phi_{LL}(p_{m+1}))] \\
&= p_{m+1} - \beta\mathbb{E}[c_L(\phi_T(p_{m+1})) - c_L(\phi_L(p_{m+1}))] \\
&\quad - \beta^2\mathbb{E}[V_{k+2}(\phi_T(\phi_L(p_{m+1})) - V_{k+2}(\phi_L(\phi_L(p_{m+1}))] \\
&\geq p_{m+1} - \mathbb{E}\Big[\beta(c_L(\phi_T(p_{m+1})) - c_L(\phi_L(p_{m+1}))) \\
&\qquad\qquad + \beta(c_L(\phi_L(p_{m+1})) - c_T(\phi_L(p_{m+1})))\Big] \\
&= p_{m+1} - \beta\mathbb{E}\left(c_L(\phi_T(p_{m+1})) - c_T(\phi_L(p_{m+1}))\right) \\
&> p_{m+1} - \beta\mathbb{E}_{R'}\left(c_L(p^1, p^2_{R'}) - c_T(p^1 e^{-1}, p^2_{R'})\right) \\
&= p_{m+1} - \beta\mu_2\left(\frac{p^1}{p^1 + p^2}\right) - \beta(1 - \mu_2)\left(\frac{p^1 + p^2}{p^1 + p^2} - \frac{p^2}{p^1 e^{-1} + p^2}\right) \\
&> p_{m+1} - \beta\mathbb{E}\left(\frac{p^1}{p^1 + p^2}\right) \\
&= p_{m+1} - \beta p_{m+1} > 0,
\end{aligned}$$

where the first inequality follows from the induction hypothesis. In the second inequality, we denote p^1, p^2 as weights of 2 experts such that $\tilde{p}^1 = p_{m+1}$, and update weights of two experts over the randomization of R' such that $p^2_{R'} = p^2$ if expert 2 tells the truth and $p^2_{R'} = p^2 e^{-1}$, otherwise. The next equality follows from the definition of the current cost function. Then, the proof is complete with the choice of $\alpha_k < G/H$. □

A.2.5 Proof for Infinite Horizon Discounted Problem

Proof of Proposition 2. Let us formulate the dynamic programming solutions as a decision tree whose each node represents a state (here, relative weight of the adversary), and each path is a sequence of nodes visited by following a sequence of the adversary's actions. Two paths are said to diverge at a node if they go through the same nodes up to that node and diverge afterwards.

Consider a strategy π and another strategy π' such that π' deviates from π at only one stage. The strategy π is called "unimprovable" if there is no such other strategy π' such that $V_{s'}^{\pi'}(\vec{p_0}) > V_s^{\pi}(\vec{p_0})$, where $\vec{p_0}$ is the initial weight vector, s and s' are sequences of actions obtained from the policy π and π', respectively. From the previous section, we have proved that the threshold policy is an improvable strategy using induction argument. Using one stage deviation principle (Tirole an Fudenberg [202], Osborne and Rubinstein [203]), and due to the fact that the infinite horizon discounted problem satisfies the continuity assumption (Assumption 3), we conclude that the threshold policy is an optimal policy for the infinite horizon problem. □

Proof of Theorem 5. Given an adversary's relative weight $p \in (0, 1)$, let us define

$$g(p) := \frac{1}{1 + \left(\frac{1}{p} - 1\right)e}, \quad f(p) = \frac{1}{1 + \left(\frac{1}{p} - 1\right)e^{-1}}.$$

Note that $g(\cdot)$ and $f(\cdot)$ are inverse of each other, i.e., $g(f(p)) = p$. Now suppose that the adversary's relative weight is right at $f(\tau)$, where τ is the optimal threshold which we know its existence by Proposition 2. Since $f(\tau) > \tau$, based on the threshold optimal policy the optimal action is to lie at state $f(\tau)$. On the other hand, since lying can only change the relative weight of the adversary to either τ or $f(\tau)$ with probabilities μ_2 and $1 - \mu_2$, respectively, we can write

$$\begin{aligned} V^*(f(\tau)) = &\Big(\mu_2 f(\tau) + \mu_2 \beta V^*(\tau)\Big) \\ &+ \Big((1 - \mu_2) + (1 - \mu_2)\beta V^*(f(\tau))\Big). \end{aligned}$$

This implies

$$V^*(f(\tau)) = \frac{1 - \mu_2 + \mu_2 f(\tau)}{1 - (1 - \mu_2)\beta} + \frac{\mu_2 \beta V^*(\tau)}{1 - (1 - \mu_2)\beta} \tag{A.8}$$

Similarly, given that the adversary's relative weight is right at $g(\tau)$, and since $g(\tau) < \tau$, the optimal action for the adversary is to tell the truth at state $g(\tau)$. Therefore, we have

$$\begin{aligned} V^*(g(\tau)) = &\Big((1-\mu_2)(1-g(\tau)) + (1-\mu_2)\beta V^*(\tau)\Big) \\ &+ \Big(\mu_2 \times 0 + \mu_2\beta V^*(g(\tau))\Big), \end{aligned}$$

which implies

$$V^*(g(\tau)) = \frac{(1-\mu_2)(1-g(\tau))}{1-\mu_2\beta} + \frac{(1-\mu_2)\beta V^*(\tau)}{1-\mu_2\beta}. \tag{A.9}$$

Furthermore, starting the adversary's relative weight right at the threshold τ will make the adversary indifferent between lying and telling the truth. Using similar derivations as in (A.8) and (A.9), we get

$$\begin{aligned} V^*(\tau) &= \frac{1-\mu_2+\mu_2\tau}{1-(1-\mu_2)\beta} + \frac{\mu_2\beta V^*(g(\tau))}{1-(1-\mu_2)\beta} \\ &= \frac{(1-\mu_2)(1-\tau)}{1-\mu_2\beta} + \frac{(1-\mu_2)\beta V^*(f(\tau))}{1-\mu_2\beta}. \end{aligned} \tag{A.10}$$

Solving (A.8), (A.9), and (A.10) together (note that there are exactly four equations and four unknowns $V^*(f(\tau)), V^*(g(\tau))$, $V^*(\tau)$, and τ, which can be uniquely determined) we obtain the threshold τ as the solution of the following equation

$$\tau = \frac{\beta\mu_2(1-\mu_2)}{1-\beta(\mu_2^2+(1-\mu_2)^2)}(f(\tau)+g(\tau)).$$

Finally, substituting the expressions of $f(\tau)$ and $g(\tau)$ into the above relation and solving for τ, we get

$$\tau = \frac{1}{2}\left(1+\theta-\sqrt{(1+\theta)^2-4\frac{(1+e^2)\theta-e}{(1-e)^2}}\right),$$

where $\theta := \frac{\beta\mu_2(1-\mu_2)}{1-\beta(\mu_2^2+(1-\mu_2)^2)}$. □

A.2.6 Extended results of optimal policy with learning rate η

Logarithmic loss

By some modifications on the proof of Theorem 3, we will prove that the optimal policy of the malicious expert when the algorithm uses a learning rate $\eta > 0$ is the same as in Theorem 3. Indeed, by using a learning rate η, the weight update rule in (3.2) is replaced by

$$p_k^i = p_{k-1}^i e^{-\eta l(x_k^i, y_k)}. \tag{A.11}$$

Note that the current cost functions of logarithmic loss, defined in (3.10), do not change and so does Proposition 1. The proof of Theorem 3 is modified to adapt with the learning rate η as follows. In (A.4), the updated weight of the malicious expert is changed from $(1-\epsilon)p^1$ and ϵp^1 to $(1-\epsilon)^\eta p^1$ and $\epsilon^\eta p^1$, respectively. Expression on the right-hand-side of (g) in equation (A.5) becomes

$$-\mathbb{E}_{R',R}\left[\ln\left(\frac{\epsilon(1-\epsilon)^\eta p^1 + (1-\epsilon)\sum\limits_{i\in R} p^i_{R'} + \epsilon\sum\limits_{j\in R^c} p^j_{R'}}{\epsilon p^1 + \vec{p}_{R'}\mathbf{1}}\right)\right] + \mathbb{E}_{R',R}\left[\ln\left(\frac{\epsilon^\eta(1-\epsilon)p^1 + (1-\epsilon)\sum\limits_{i\in R} p^i_{R'} + \epsilon\sum\limits_{j\in R^c} p^j_{R'}}{(1-\epsilon)p^1 + \vec{p}_{R'}\mathbf{1}}\right)\right].$$

Due to the monotonicity of the log function, (h) is changed to the inequality and the proof follows.

Absolute loss

As noted above, the current cost functions of absolute loss, defined in (3.11) and Proposition 1 remain unchanged while the weight update rule is changed as given in (A.11) and consequently, the weight transition in (3.12) becomes

$$\phi_{x_k^1}(\tilde{p}_{k-1}^1) = \begin{cases} \frac{1}{1+\left(\frac{1}{\tilde{p}_{k-1}^1}-1\right)e^{\eta}} & \text{if } x_k^1 = L, x_k^2 = T, \\ \frac{1}{1+\left(\frac{1}{\tilde{p}_{k-1}^1}-1\right)e^{-\eta}} & \text{if } x_k^1 = T, x_k^2 = L, \\ \tilde{p}_{k-1}^1 & \text{if } x_k^1 = x_k^2. \end{cases} \tag{A.12}$$

In this setting, it is straightforward to check that the condition in Lemma 8, and hence in Theorem 4 will change to $\beta < \frac{1}{e^\eta}$.

A.2.7 Proof of mean-field results

Proof of Lemma 3. $p - \hat{\phi}_L(p) - \mu(\hat{\phi}_T(p) - \hat{\phi}_L(p))$

$$\begin{aligned} &= p - \frac{1}{1+(1/p-1)(\mu e+(1-\mu))}, \\ &- \mu\frac{(1/p-1)(\mu e+(1-\mu))(1-e^{-1})}{\big(1+(1/p-1)(\mu+(1-\mu)e^{-1})\big)\big(1+(1/p-1)(\mu e+(1-\mu))\big)}, \\ &= \frac{\mu(1-\mu)(1-p)(e-1)(1-e^{-1})}{\big(1+(1/p-1)(\mu+(1-\mu)e^{-1})\big)\big(1+(1/p-1)(\mu e+(1-\mu))\big)}, \\ &> 0. \end{aligned}$$

□

Proof of Lemma 4. We prove this using induction. For the base case when k = 1, the claim holds true since

$$\begin{aligned}\hat{\phi}_L(\hat{\phi}_T(p)) &= \frac{1}{1+(1/\hat{\phi}_T(p)-1)(\mu+(1-\mu))}\\ &= \frac{1}{1+(1/p-1)(\mu+(1-\mu)e^{-1})^2 e},\end{aligned} \tag{A.13}$$

and

$$\begin{aligned}\hat{\phi}_T(\hat{\phi}_L(p)) &= \frac{1}{1+(1/\hat{\phi}_L(p)-1)(\mu+(1-\mu))e^{-1}},\\ &= \frac{1}{1+(1/p-1)(\mu+(1-\mu)e^{-1})^2 e}.\end{aligned} \tag{A.14}$$

Assume now that the claim holds at k, we prove it is true for $k+1$. Indeed,

$$\begin{aligned}\hat{\phi}_L(\hat{\phi}_{T^{(k+1)}}(p)) &= \hat{\phi}_L(\hat{\phi}_{T^{(k)}}(\hat{\phi}_T(p))),\\ &= \hat{\phi}_{T^{(k)}}(\hat{\phi}_L(\hat{\phi}_T(p))),\\ &= \hat{\phi}_{T^{(k)}}(\hat{\phi}_T(\hat{\phi}_L(p))),\\ &= \hat{\phi}_{T^{(k+1)}}(\hat{\phi}_L(p)),\end{aligned}$$

where the second equality follows from the induction hypothesis, and the third one follows from the base case. □

Proof of Theorem 6. We again apply the induction technique. At the final step, the claim is immediate from the fact that $\Delta\hat{c}_K(\hat{p}) = \hat{c}_L(\hat{p}) - \hat{c}_T(\hat{p}) = \hat{p} > 0$.
Induction step: suppose the claim holds true at the step $k+1$, we consider the step k,

$$\begin{aligned}D\hat{V}_k(\hat{p}) &= \hat{c}_L(\hat{p}) - \hat{c}_T(\hat{p}) + \hat{V}_{k+1}(\hat{\phi}_L(\hat{p}), L) - \hat{V}_{k+1}(\hat{\phi}_T(\hat{p}), L),\\ &> \hat{c}_L(\hat{p}) - \hat{c}_T(\hat{p}) + \hat{V}_{k+1}(\hat{\phi}_L(\hat{p}), T) - \hat{V}_{k+1}(\hat{\phi}_T(\hat{p}), L),\\ &> \hat{c}_L(\hat{p}) - \hat{c}_T(\hat{p}) + \hat{c}_T(\hat{\phi}_L(\hat{p})) - \hat{c}_L(\hat{\phi}_T(\hat{p})),\\ &= \hat{p} - \hat{\phi}_T(\hat{p}) + (1-\mu)(\hat{\phi}_T(\hat{p}) - \hat{\phi}_L(\hat{p})),\\ &= \hat{p} - \hat{\phi}_L(\hat{p}) - \mu(\hat{\phi}_T(\hat{p}) - \hat{\phi}_L(\hat{p})),\\ &> 0.\end{aligned}$$

where the first inequality follows from the induction hypothesis, the second inequality follows from Lemma 4, and the last inequality follows from Lemma 3. □

Proof of Lemma 5. The idea of the proof is to bound the difference of the optimal and approximate value functions based on the dynamic programming algorithm. To do this, at each step, we compare the true value functions with the approximate value functions when the malicious expert tells a lie and the truth, respectively. We prove this by induction.

At step K,

$$|c_L(p) - \hat{c}_L(\hat{p})| \leq |c_L(p) - \hat{c}_L(p)| + |\hat{c}_L(p) - \hat{c}_L(\hat{p})|,$$
$$= (1-\bar{p})\left|\sum_{i\neq 1} q^i_{K-1}x^i_K - \mu\right| + \mu|p-\hat{p}|.$$

Define the function

$$f(x^2_K, ..., x^N_K) = (1-\bar{p}_{K-1})\sum_{i\neq 1} q^i_{K-1}x^i_K,$$

we observe that

$$\left|f(x^2_K, ..., x^i_K, ..., x^N_K) - f(x^2_K, ..., (x^i_K)', ..., x^N_K)\right|$$
$$= |p^i_K x^i_K - (p^i_K)'(x^i_K)'| \leq p^i_K + (p^i_K)'.$$

Note that since the updated weight of an expert is increased if that expert tells the truth,

$$p^i_K \leq \frac{p^i_{K-1}e}{p^i_{K-1}e + (1-p^i_{K-1})(\mu e + (1-\mu))} \leq p^i_{K-1}e.$$

Therefore, we obtain $p^i_K \leq p^i_0 e^K = \frac{e^K}{N}$, and thus the function $f(.)$ has the bounded difference of $\frac{2e^K}{N}$. Using McDiarmird's inequality, $\forall\ \nu_K > 0$

$$P\left((1-\bar{p}_{K-1})\left|\sum_{i\neq 1} q^i_{K-1}x^i_K - \mu\right| \geq \nu_K\right)$$
$$\leq exp\left(\frac{-2\nu_K^2}{\sum\limits_{i\neq 1}\left(\frac{2e^K}{N}\right)^2}\right) = exp\left(\frac{-\nu_K^2}{2e^{2K}}\frac{N^2}{N-1}\right).$$

Thus,

$$|V_k(p) - \hat{V}_k(\hat{p})| = |c_L(p) - \hat{c}_L(\hat{p})| \leq \nu_K + \mu\delta_K := \epsilon_K,$$

with probability at least $1 - exp\left(\frac{-(\epsilon_K-\mu\delta_K)^2}{2e^{2K}}\frac{N^2}{N-1}\right)$.
Suppose that the claim holds true at step $k+1$, we consider at step k two states p and $\hat{p}$ such that $|p-\hat{p}| < \delta_k$. Similar to the proof for the base case, we obtain the difference of current costs

$$|c_L(p) - \hat{c}_L(\hat{p})| \leq \tilde{\nu}_k + \mu\delta_k,$$

with probability at least $1 - exp\left(\frac{-(\tilde{\nu}_k)^2}{2e^{2k}}\frac{N^2}{N-1}\right)$.
Next, we bound the difference of two value functions after telling a lie at the step k. We first observe that,

$$
\begin{aligned}
&|\phi_L(p) - \hat{\phi}_L(\hat{p})| \\
&\qquad = \Big|\frac{p}{p + (1-p)\sum\limits_{i\neq 1} q^i_{k-1} e^{x^i_k}} - \frac{\hat{p}}{\hat{p} + (1-\hat{p})(\mu e + (1-\mu))}\Big|, \\
&\qquad = \left|\frac{p(1-\hat{p})(\mu e + (1-\mu)) - \hat{p}(1-p)(\sum\limits_{i\neq 1} q^i_{k-1} e^{x^i_k})}{(p + (1-p)\sum\limits_{i\neq 1} q^i_{k-1} e^{x^i_k})(\hat{p} + (1-\hat{p})(\mu e + (1-\mu)))}\right|, \\
&\qquad = \left|\frac{\hat{p}(1-p)(\mu e + (1-\mu) - \sum\limits_{i\neq 1} q^i_{k-1} e^{x^i_k}) + (p-\hat{p})(\mu e + (1-\mu))}{(p + (1-p)\sum\limits_{i\neq 1} q^i_{k-1} e^{x^i_k})(\hat{p} + (1-\hat{p})(\mu e + (1-\mu)))}\right|, \\
&\qquad \leq \Big|\hat{p}(1-p)(\mu e + 1 - \mu - \sum_{i\neq 1} q^i_{k-1} e^{x^i_k})\Big| + |p-\hat{p}|(\mu e + 1 - \mu), \\
&\qquad \leq a_k + \delta_k e, \qquad\qquad (A.15)
\end{aligned}
$$

with probability at least $1 - exp\left(\frac{-a_k^2}{2e^{2(k+1)}}\frac{N^2}{N-1}\right)$, where $a_k = \Big|\hat{p}(1-p)(\mu e + 1 - \mu - \sum\limits_{i\neq 1} q^i_{k-1} e^{x^i_k})\Big|$ which is bounded since $x^i_k \in \{0,1\}$ and $\sum\limits_{i\neq 1} q^i_{k-1} = 1$. The second last inequality follows from the triangle inequality and the fact that the denominator is greater than 1, and the last expression follows from McDiarmird's inequality.

Using induction hypothesis, $\forall\ \epsilon_{k+1} > 0$, if $|c_L(p) - \hat{c}_L(\hat{p})| < \delta_k$, we have $|V_{k+1}(\phi_L(p)) - \hat{V}_{k+1}(\hat{\phi}_L(\hat{p}))| \leq \epsilon_{k+1}$ with probability $\xi_{k+1} = exp(-c_{k+1}N)$. Now, from (A.15) and the union bound, we infer

$$|V_{k+1}(\phi_L(p)) - \hat{V}_{k+1}(\hat{\phi}_L(\hat{p}))| \leq \epsilon_{k+1},$$

with probability at least $1 - exp\left(\frac{-a_k^2}{2e^{2(k+1)}}\frac{N^2}{N-1}\right) - \xi_{k+1}$.

Then, using the triangle inequality, we obtain

$$
\begin{aligned}
&|V_k(p, L) - \hat{V}_k(\hat{p}, L)| \\
&\qquad \leq |c_L(p) - \hat{c}_L(\hat{p})| + |V_{k+1}(\phi_L(p)) - \hat{V}_{k+1}(\hat{\phi}_L(\hat{p}))|, \\
&\qquad \leq \epsilon_k := \tilde{\nu}_k + \mu\delta_k + \epsilon_{k+1},
\end{aligned}
$$

with probability $1 - exp\left(\frac{-a_k^2}{2e^{2(k+1)}}\frac{N^2}{N-1}\right) - \xi_{k+1} - exp\left(\frac{-(\tilde{\nu}_k)^2}{2e^k}\frac{N^2}{N-1}\right)$.

We apply exactly the same technique to bound the difference of $|V_k(p,T) - \hat{V}_k(\hat{p},T)|$:

$$|c_T(p) - \hat{c}_T(\hat{p})| \leq \tilde{\nu}_k + (1-\mu)\delta_k$$

with probability at least $1 - exp\left(\frac{-(\tilde{\nu}_k)^2}{2e^{2K}}\frac{N^2}{N-1}\right)$. Next, we bound the difference of two value functions after telling the truth at the step k. We first observe that

$$\begin{aligned}
&|\phi_T(p) - \hat{\phi}_T(\hat{p})| \\
&= \left| \frac{pe}{pe + (1-p)\displaystyle\sum_{i \neq 1} q^i_{k-1} e^{x^i_k}} - \frac{\hat{p}e}{\hat{p}e + (1-\hat{p})(\mu e + (1-\mu))} \right|, \\
&\leq \left| \hat{p}e(1-p)(\mu e + (1-\mu) - \sum_{i \neq 1} q^i_{k-1} e^{x^i_k}) \right| + e|p - \hat{p}|(\mu e + (1-\mu)), \\
&\leq ea_k + \delta_k e^2, \qquad \text{(A.16)}
\end{aligned}$$

with probability at least $1 - exp\left(\frac{-a_k^2}{2e^{2(k+1)}} \frac{N^2}{N-1}\right)$.
From the induction hypothesis and (A.16), we infer

$$|V_{k+1}(\phi_T(p)) - \hat{V}_{k+1}(\hat{\phi}_T(\hat{p}))| \leq \epsilon_{k+1},$$

with probability at least $1 - exp\left(\frac{-a_k^2}{2e^{2(k+1)}} \frac{N^2}{N-1}\right) - \xi_{k+1}$.
Then, using the triangle inequality, we obtain

$$\begin{aligned}
&|V_k(p,T) - \hat{V}_k(\hat{p},T)| \\
&\leq |c_T(p) - \hat{c}_T(\hat{p})| + |V_{k+1}(\phi_T(p)) - \hat{V}_{k+1}(\hat{\phi}_T(\hat{p}))|, \\
&\leq \epsilon_k := \tilde{\nu}_k + (1-\mu)\delta_k + \epsilon_{k+1},
\end{aligned}$$

with probability $1 - exp\left(\frac{-a_k^2}{2e^{2(k+1)}} \frac{N^2}{N-1}\right) - \xi_{k+1} - exp\left(\frac{-(\tilde{\nu}_k)^2}{2e^k} \frac{N^2}{N-1}\right)$.
From the optimality principle, the claim holds true with probability of at least $1 - \xi_k$ where $\xi_k = exp\left(\frac{-a_k^2}{2e^{2(k+1)}} \frac{N^2}{N-1}\right) + \xi_{k+1} + exp\left(\frac{-(\tilde{\nu}_k)^2}{2e^k} \frac{N^2}{N-1}\right) = exp(-c_k N)$. □

Proof of Theorem 7. From the above lemma, if the optimal strategy of the approximated setting and the optimal setting of the original setting start from the same initial weight of the malicious expert, i.e., $\tilde{p}^1_0 = \hat{p}^1_0$, their value functions are close, which implies that the updated weight $\tilde{p}^1_1, \hat{p}^1_1$ are close to each other as well. To see this, let us consider the step 1. From theorem 6, $\hat{V}_0(\hat{p}, L) > \hat{V}_0(\hat{p}, T)$. Based on the proof of Lemma 5, $V_0(p, L) > V_0(p, T)$ with high probability. This implies that $\tilde{p}^1_1 = \phi_L(\tilde{p}^1_0)$ and $\hat{p}^1_1 = \phi_L(\hat{p}^1_0)$. From the analysis similar to (A.15), we can see that the difference of states at step 1 is small with high probability. The process is conducted similarly for the next steps. □

REFERENCES

[1] N. Littlestone and M. K. Warmuth, "The weighted majority algorithm," in *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, NC, Feb. 1989, pp. 212–261.

[2] V. G. Vovk, "Aggregating strategies," in *Proceedings of the third annual workshop on Computational learning theory (COLT '90)*, San Francisco, CA, USA, 1990, pp. 371–386.

[3] A. György and G. Ottucsák, "Adaptive routing using expert advice," *Computer Journal*, vol. 49, pp. 180–189, Mar. 2006.

[4] B. Awerbuch and R. Kleinberg, "Adaptive routing with end-to-end feedback: distributed learning and geometric approaches," in *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, New York, NY, USA, 2004, pp. 45–53.

[5] A. Kalai and S. Vempala, "Efficient algorithms for online decision problems," *Journal of Computer and System Sciences*, vol. 71, pp. 291–307, Oct. 2005.

[6] A. Blum and C. Burch, "On-line learning and the metrical task system problem," in *Proceedings of the tenth annual conference on Computational learning theory*, New York, NY, USA, 1997, pp. 45–53.

[7] A. Blum, C. Burch, and A. Kalai, "Finely-competitive paging," in *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, Washington, DC, USA, 1999, pp. 450–458.

[8] S. Asur and B. Huberman, "Predicting the future with social media," in *Proceedings of 2010 IEEE/WIC/ACM International Conference on Web Intelligence*, Toronto, Canada, 2010, pp. 492–499.

[9] M. Joshi, D. Das, K. Gimpel, and N. Smith, "Movie reviews and revenues: An experiment in text regression," in *Proceedings of the North American Chapter of the Association for Computational Linguistics Human Language Technologies Conference*, Los Angeles, CA, USA, 2010.

[10] W. Zhang and S. Skiena, "Improving movie gross prediction through news analysis," in *Proceedings of 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, 2009, pp. 301–304.

[11] L. Liu, J. Xu, S. Liao, and H. Chen, "A real-time personalized route recommendation system for self-drive tourists based on vehicle to vehicle communication," *Expert Systems with Applications*, vol. 41, pp. 3409–3417, 2014.

[12] R. Arnott, A. de Palma, and R. Lindsey, "Does providing information to drivers reduce traffic congestion?" *Transportation Research Part A: General*, vol. 25, pp. 309–318, 1991.

[13] K. Agrawal, A. Vempaty, H. Chen, and P. K. Varshney, "Target localization in wireless sensor networks with quantized data in the presence of byzantine attacks," in *Proceedings of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, 2011, pp. 1669–1673.

[14] A. Vempaty, O. Ozdemir, K. Agrawal, H. Chen, and P. K. Varshney, "Localization in wireless sensor networks: Byzantines and mitigation techniques," *IEEE Transactions on Signal Processing*, vol. 61, pp. 1495–1508, 2013.

[15] A. Vempaty, O. Ozdemir, and P. K. Varshney, "Mitigation of byzantine attacks for target location estimation in wireless sensor networks," in *Proceedings of 46th Annual Conference on Information Sciences and Systems (CISS)*, 2012, pp. 1–6.

[16] A. Blum and Y. Mansour, "From external to internal regret," *The Journal of Machine Learning Research*, vol. 8, pp. 1307–1324, Dec. 2007.

[17] R. D. Kleinberg, A. Niculescu-Mizil, and Y. Sharma, "Regret bounds for sleeping experts and bandits," in *Proceedings of the 21st Annual Conference on Learning Theory - COLT 2008*, Helsinki, Finland, 2008, pp. 425–436.

[18] V. Kanade, B. McMahan, and B. Bryan, "Sleeping experts and bandits with stochastic action availability and adversarial rewards," in *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, Florida, USA, 2009, pp. 272–279.

[19] Y. Freund, R. E. Schapire, Y. Singer, and M. K. Warmuth, "Using and combining predictors that specialize," in *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing*, New York, NY, USA, 1997, pp. 334–343.

[20] N. Cesa-Bianchi, Y. Freund, D. Helmbold, D. Haussler, R. E. Schapire, and M. K. Warmuth, "How to use expert advice," in *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing (STOC '93)*, New York, NY, USA, May 1993, pp. 427–485.

[21] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games.* Oakland: Cambridge University Press, 2006.

[22] J. R. Douceur, "The sybil attack," in *International Workshop on Peer-to-Peer Systems*, London, UK, 2002, pp. 251–260.

[23] N. Rubens, D. Kaplan, and M. Sugiyama, *Active Learning in Recommender Systems.* Springer US, 2011.

[24] M. Elahi, F. Ricci, and N. Rubens, “A survey of active learning in collaborative filtering recommender systems,” *Computer Science Review*, vol. 20, pp. 29–50, 2016.

[25] F. Olsson, “A literature survey of active machine learning in the context of natural language processing,” Swedish Institute of Computer Science, Tech. Rep., 2009.

[26] H. Yu, C. Shi, M. Kaminsky, P. B. Gibbons, and F. Xiao, “Dsybil: Optimal sybil-resistance for recommendation systems,” in *Proceedings of the 2009 30th IEEE Symposium on Security and Privacy*, Washington, DC, USA, 2009, pp. 283–298.

[27] R. Bellman, “A markovian decision process,” *Journal of Mathematics and Mechanics*, 1957.

[28] A. R. Howard, *Dynamic Programming and Markov Processes.* The MIT Press, 1960.

[29] T. M. Cover, “Behavior of sequential predictors of binary sequences,” in *Proceedings of the 4th Prague Conference on Information Theory, Statistical Decision Functions, Random Processes*, Prague, 1965, pp. 263–272.

[30] A. Truong and N. Kiyavash, “Optimal adversarial strategies in learning with expert advice,” in *Proceedings of the 52th IEEE Conference on Decision and Control*, Florence, Italia, Dec. 2013.

[31] D. Blackwell, “An analog of the minimax theorem for vector payoffs,” *Pacific Journal of Mathematics*, vol. 6, pp. 1–8, 1956.

[32] J. Hannan, “Approximation to bayes risk in repeated play,” *Contributions to the Theory of Games*, vol. 3, p. 97139, 1957.

[33] J. Kivinen and M. K. Warmuth, “Averaging expert predictions,” in *Proceedings of the 4th European Conference on Computational Learning Theory*, London, UK, 1999, pp. 153–167.

[34] J. Kivinen and M. K. Warmuth, “Exponentiated gradient versus gradient descent for linear predictors,” *Information and Computation*, vol. 132, pp. 1–63, Jan. 1997.

[35] R. Yaroshinsky, R. El-Yaniv, and S. Seiden, “How to better use expert advice,” *Machine Learning*, vol. 55, pp. 271–309, 2004.

[36] T. van Erven, P. Grunwald, W. M. Koolen, and S. de Rooij, “Adaptive hedge,” in *Proceedings of the 24th International Conference on Neural Information Processing Systems*, USA, 2011, pp. 1656–1664.

[37] P. Auer, N. Cesa-Bianchi, and C. Gentile, “Adaptive and self-confident on-line learning algorithms,” *Journal of Computer and System Sciences*, vol. 64, pp. 48–75, 2002.

[38] P. Grünwald, “The safe bayesian: Learning the learning rate via the mixability gap,” in *Proceedings of the 23rd International Conference on Algorithmic Learning Theory (ALT 12)*, Lyon, France, 2012, pp. 169–183.

[39] E. Even-Dar, M. Kearns, Y. Mansour, and J. Wortman, "Regret to the best vs. regret to the average," *Machine Learning*, vol. 72, pp. 21–37, Aug. 2008.

[40] D. Adamskiy, W. M. Koolen, A. Chernov, and V. Vovk, "A closer look at adaptive regret," *The Journal of Machine Learning Research*, vol. 17, pp. 706–726, Jan. 2016.

[41] E. Gofer and Y. Mansour, "Lower bounds on individual sequence regret," in *Proceedings of the 23rd International Conference on Algorithmic Learning Theory*, 2012, pp. 275–289.

[42] E. Moroshko and K. Crammer, "Weighted last-step min-max algorithm with improved sub-logarithmic regret," in *Proceedings of the 23rd International Conference on Algorithmic Learning Theory*, 2012, pp. 245–259.

[43] A. György and C. Szepesvári, "Shifting regret, mirror descent, and matrices," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, 2016, pp. 2943–2951.

[44] D. Foster, "Prediction in the worst-case," *Annals of Statistics*, vol. 19, pp. 1084–1090, 1991.

[45] M. Herbster and M. K. Warmuth, "Tracking the best expert," *Machine Learning - Special issue on context sensitivity and concept drift*, vol. 32, pp. 151–178, Aug. 1998.

[46] V. Vovk, "Competitive on-line statistics," *International Statistical Review*, vol. 69, pp. 213–248, 2001.

[47] A. Chernov, Y. Kalnishkan, F. Zhdanov, and V. Vovk, "Supermartingales in prediction with expert advice," *Journal of Theoretical Computer Science*, vol. 411, pp. 2647–2669, June 2010.

[48] A. Chernov and V. Vovk, "Prediction with advice of unknown number of experts," in *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, 2010, pp. 117–125.

[49] A. Gyorgy, T. Linder, and j. . I. v. . . m. . n. y. . . p. . . G. Lugosi, title = Efficient Tracking of Large Classes of Experts.

[50] A. Chernov and F. Zhdanov, "Prediction with expert advice under discounted loss," in *Proceedings of the 21st International Conference on Algorithmic Learning Theory*, 2010, pp. 255–269.

[51] O. Maillard and R. Munos, "Online learning in adversarial lipschitz environments," in *Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases: Part II*, 2010, pp. 305–320.

[52] L. Jie, F. Orabona, and B. Caputo, "An online framework for learning novel concepts over multiple cues," in *Proceedings of the 9th Asian Conference on Computer Vision - Volume Part I*, 2010, pp. 269–280.

[53] P. Bartlett, E. Hazan, and A. Rakhlin, "Adaptive online gradient descent," in *Proceedings of the 20th International Conference on Neural Information Processing Systems*, 2007, pp. 65–72.

[54] V. Mendonça, F. Melo, L. Coheur, and A. Sardinha, "A conversational agent powered by online learning," in *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, 2017, pp. 1637–1639.

[55] K. Chaudhuri, Y. Freund, and D. Hsu, "An online learning-based framework for tracking," in *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, 2010, pp. 101–108.

[56] A. György, G. Lugosi, and G. Ottucsàk, "On-line sequential bin packing," *The Journal of Machine Learning Research*, vol. 11, pp. 89–109, Mar. 2010.

[57] M. Raginsky, R. Marcia, J. Silva, and R. M. Willett, "Sequential probability assignment via online convex programming using exponential families," in *Proceedings of the 2009 IEEE International Conference on Symposium on Information Theory - Volume 2*, 2009, pp. 1338–1342.

[58] B. Kveton, J. Y. Yu, G. Theocharous, and S. Mannor, "Online learning with expert advice and finite-horizon constraints," in *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 1*, 2008, pp. 331–336.

[59] P. Zhao and S. Hoi, "Otl: A framework of online transfer learning," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, 2010, pp. 1231–1238.

[60] F. Orabona and K. Crammer, "New adaptive algorithms for online classification," in *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 2*, 2010, pp. 1840–1848.

[61] S. Yasutake, K. Hatano, S. Kijima, E. Takimoto, and M. Takeda, "Online linear optimization over permutations," in *Proceedings of the 22Nd International Conference on Algorithms and Computation*, 2011, pp. 534–543.

[62] E. Delage, "Regret-based online ranking for a growing digital library," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 229–238.

[63] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, "Learning to rank using gradient descent," in *Proceedings of the 22Nd International Conference on Machine Learning*, 2005, pp. 89–96.

[64] K. Crammer and Y. Singer, "Loss bounds for online category ranking," in *Proceedings of the 18th Annual Conference on Learning Theory*, 2005, pp. 48–62.

[65] F. Radlinski and T. Joachims, "Active exploration for learning rankings from click-through data," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2007, pp. 570–579.

[66] E. Agichtein, E. Brill, and S. Dumais, "Improving web search ranking by incorporating user behavior information," in *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2006, pp. 19–26.

[67] R. Busa-Fekete, B. Kégl, T. Éltető, and G. Szarvas, "A robust ranking methodology based on diverse calibration of adaboost," in *Proceedings of the 2011th European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part I*, 2011, pp. 263–279.

[68] S. Hart and A. Mas-Colell, "A general class of adaptive strategies," *Journal of Economic Theory*, vol. 98, pp. 26–54, 2001.

[69] J. Ziv, "Distortion-rate theory for individual sequences," *IEEE Transactions on Information Theory*, vol. 26, pp. 137–143, 1980.

[70] J. Ziv and A. Lempel, "A universal algorithm for sequential data-compression," *IEEE Transactions on Information Theory*, vol. 23, pp. 337–343, 1977.

[71] J. Ziv, "Coding theorems for individual sequences," *IEEE Transactions on Information Theory*, vol. 24, pp. 405–412, 1978.

[72] M. Merhav and M. Feder, "Universal schemes for sequential decision from individual data sequences," *IEEE Transactions on Information Theory*, vol. 39, pp. 1280–1292, 1993.

[73] A. Borodin and R. El-Yaniv, *Online Computation and Competitive Analysis.* New York, NY, USA: Cambridge University Press, 1998.

[74] S. Kozat and A. Singer, "Competitive prediction under additive noise," *IEEE Transactions on Signal Processing*, vol. 57, no. 9, pp. 3698–3703, Sep. 2009.

[75] J. Abernethy, P. Bartlett, and A. Rakhlin, "Multitask learning with expert advice," in *Proceedings of the 20th Annual Conference on Learning Theory*, 2007, pp. 484–498.

[76] A. Borodin, R. El-Yaniv, and V. Gogan, "Can we learn to beat the best stock," *Journal of Artificial Intelligence Research*, vol. 21, pp. 579–594, May 2004.

[77] V. Dani, O. Madani, D. Pennock, S. Sanghai, and B. Galebach, "An empirical comparison of algorithms for aggregating expert predictions," in *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence (UAI 2006)*, Cambridge, MA, USA, 2006.

[78] A. Blum, "Empirical support for winnow and weighted-majority algorithms: Results on a calendar scheduling domain," *Machine Learning*, vol. 26, pp. 5–23, Jan. 1997.

[79] W. W. Cohen and Y. Singer, “Context-sensitive learning methods for text categorization,” *ACM Transactions on Information Systems (TOIS)*, vol. 17, pp. 141–173, Apr. 1999.

[80] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” in *EuroCOLT ’95 Proceedings of the Second European Conference on Computational Learning Theory*, London, UK, 1995, pp. 23–37.

[81] A. Kalai and S. Vempala, “Efficient algorithms for online decision problems,” *Journal of Computer and System Sciences*, pp. 291–307, 2005.

[82] S. D. Rooij, T. V. Erven, P. D. Grünwald, and W. M. Koolen, “Follow the leader if you can, hedge if you must,” *Journal of Machine Learning Research*, vol. 15, pp. 1281–1316, 2014.

[83] H. Robbins and S. Monro, “A stochastic approximation method,” *The Annals of Mathematical Statistics*, vol. 22, pp. 400–407, 1951.

[84] K. L. Chung, “On a stochastic approximation method,” *The Annals of Mathematical Statistics*, vol. 25, pp. 463–483, 1954.

[85] B. T. Polyak and A. B. Juditsky, “Acceleration of stochastic approximation by averaging,” *SIAM Journal on Control and Optimization*, vol. 30, pp. 838–855, July 1992.

[86] A. Truong, N. Kiyavash, and V. Borkar, “Convergence analysis for an online recommendation system,” in *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, Orlando, Florida, US, Dec. 2011, pp. 3889 – 3894.

[87] A. Truong, S. R. Etesami, J. Etesami, and N. Kiyavash, “Optimal attack strategies against predictors - learning from expert advice,” *IEEE Transactions on Information Forensics and Security*, vol. 13, pp. 6–19, 2017.

[88] J. Newsome, B. Karp, and D. Song, “Paragraph: Thwarting signature learning by training maliciously,” in *Proceedings of the 9th International Conference on Recent Advances in Intrusion Detection*, Berlin, Germany, 2006, pp. 81–105.

[89] S. P. Chung and A. K. Mok, “Allergy attack against automatic signature generation,” in *Proceedings of the 9th International Conference on Recent Advances in Intrusion Detection*, Berlin, Germany, 2006, pp. 61–80.

[90] L. Huang, A. D. Joseph, B. Nelson, B. I. P. Rubinstein, and J. D. Tygar, “Adversarial machine learning,” in *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence*, New York, NY, USA, 2011, pp. 43–58.

[91] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro, “Aiding the detection of fake accounts in large scale social online services,” in *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, Berkeley, CA, USA, 2012.

[92] N. Tran, B. Min, J. Li, and L. Subramanian, "Sybil-resilient online content voting," in *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation*, Berkeley, CA, USA, 2009, pp. 15–28.

[93] K. Govindan and P. Mohapatra, "Trust computations and trust dynamics in mobile adhoc networks: A survey," *IEEE Communications Surveys Tutorials*, vol. 14, pp. 279–298, 2012.

[94] P. Resnick and R. Sami, "The information cost of manipulation-resistance in recommender systems," in *Proceedings of the 2008 ACM Conference on Recommender Systems*, New York, NY, USA, 2008, pp. 147–154.

[95] S. Venkataraman, A. Blum, and D. Song, "Limits of learning-based signature generation with adversaries," in *Proceedings of the Network and Distributed System Security Symposium, NDSS 2008*, San Diego, California, USA, 2008.

[96] B. Nelson, M. Barreno, F. J. Chi, A. D. Joseph, B. I. P. Rubinstein, U. Saini, C. Sutton, J. D. Tygar, and K. Xia, "Exploiting machine learning to subvert your spam filter," in *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, Berkeley, CA, USA, 2008, pp. 71–79.

[97] M. Barreno, B. Nelson, R. Sears, A. Joseph, and J. D. Tygar, "Can machine learning be secure?" in *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security*, 2006, pp. 16–25.

[98] M. Kantarcioglu, B. Xi, and C. Clifton, "Classifier evaluation and attribute selection against active adversaries," *Data Mining and Knowledge Discovery*, vol. 22, pp. 291–335, Jan. 2011.

[99] B. Nelson, B. Biggio, and P. Laskov, "Understanding the risk factors of learning in adversarial environments," in *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence*, 2011, pp. 87–92.

[100] J. Sakuma and H. Arai, "Online prediction with privacy," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, 2010, pp. 935–942.

[101] O. Maillard and R. Munos, "Online learning in adversarial lipschitz environments," in *Proceedings of the 2010th European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part II*, 2010, pp. 305–320.

[102] Y. Abbasi-Yadkori, P. L. Bartlett, V. Kanade, Y. Seldin, and C. Szepesvári, "Online learning in markov decision processes with adversarially chosen transition probability distributions," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, 2013, pp. 2508–2516.

[103] O. Anava, E. Hazan, and S. Mannor, "Online learning for adversaries with memory: Price of past mistakes," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, 2015, pp. 784–792.

[104] M. Kantarcioglu and B. Xi, "Adversarial data mining: Big data meets cyber security," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 1866–1867.

[105] B. Nelson, B. Rubinstein, L. Huang, A. Joseph, and J. D. Tygar, "Classifier evasion: Models and open problems," in *Proceedings of the International ECML/PKDD Conference on Privacy and Security Issues in Data Mining and Machine Learning*, 2011, pp. 92–98.

[106] D. Lowd and C. Meek, "Adversarial learning," in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, 2005, pp. 641–647.

[107] B. Biggio, G. Fumera, and F. Roli, "Multiple classifier systems under attack," in *Proceedings of the 9th International Conference on Multiple Classifier Systems*, 2010, pp. 74–83.

[108] B. Li, "Secure learning and mining in adversarial environments," in *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, Nov. 2015, pp. 1538–1539.

[109] N. Kiyavash, "Information theoretic limits for secure multimedia and magnetic recording," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2006.

[110] P. Moulin and N. Kiyavash, "Performance of random fingerprinting codes under arbitrary nonlinear attacks," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, vol. 2, 2007.

[111] S. Kadloor and N. Kiyavash, *Exploiting Timing Side Channel in Secure Cloud Scheduling*. New York, NY: Springer New York, 2014, pp. 147–168.

[112] S. Kadloor and N. Kiyavash, "Delay-privacy tradeoff in the design of scheduling policies," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2557–2573, 2015.

[113] S. Kadloor, X. Gong, N. Kiyavash, and P. Venkitasubramaniam, "Designing router scheduling policies: A privacy perspective," *IEEE Transactions on Signal Processing*, vol. 60, no. 4, pp. 2001–2012, 2012.

[114] S. Kadloor, N. Kiyavash, and P. Venkitasubramaniam, "Mitigating timing based information leakage in shared schedulers," in *2012 Proceedings IEEE INFOCOM*, 2012, pp. 1044–1052.

[115] S. Kadloor and N. Kiyavash, "Delay optimal policies offer very little privacy," in *2013 Proceedings IEEE INFOCOM*, 2013, pp. 2454–2462.

[116] J. Etesami and N. Kiyavash, "On the vulnerability of digital fingerprinting systems to finite alphabet collusion attacks," *CoRR*, 2016.

[117] J. Etesami and N. Kiyavash, "A novel collusion attack on finite alphabet digital fingerprinting systems," in *2014 IEEE International Symposium on Information Theory*, 2014, pp. 2237–2241.

[118] N. Kiyavash and P. Moulin, “On optimal collusion strategies for fingerprinting,” in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 5, 2006.

[119] N. Kiyavash and P. Moulin, “A framework for optimizing nonlinear collusion attacks on fingerprinting systems,” in *2006 40th Annual Conference on Information Sciences and Systems*, 2006, pp. 1170–1175.

[120] A. Ghassami, D. Cullina, and N. Kiyavash, “Message partitioning and limited auxiliary randomness: Alternatives to honey encryption,” in *2016 IEEE International Symposium on Information Theory (ISIT)*, 2016, pp. 1371–1375.

[121] A. Ghassami and N. Kiyavash, “A covert queueing channel in fcfs schedulers,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 6, pp. 1551–1563, 2018.

[122] S. Kadloor, N. Kiyavash, and P. Venkitasubramaniam, “Mitigating timing side channel in shared schedulers,” *IEEE/ACM Transaction on Networking*, vol. 24, no. 3, pp. 1562–1573, June 2016.

[123] X. Gong, N. Kiyavash, and P. Venkitasubramaniam, “Information theoretic analysis of side channel information leakage in fcfs schedulers,” in *2011 IEEE International Symposium on Information Theory Proceedings*, 2011, pp. 1255–1259.

[124] N. Kiyavash, F. Koushanfar, T. P. Coleman, and M. Rodrigues, “A timing channel spyware for the csma/ca protocol,” *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 3, pp. 477–487, 2013.

[125] S. Kadloor, X. Gong, N. Kiyavash, T. Tezcan, and N. Borisov, “Low-cost side channel remote traffic analysis attack in packet networks,” in *2010 IEEE International Conference on Communications*, 2010, pp. 1–5.

[126] X. Gong and N. Kiyavash, “Timing side channels for traffic analysis,” in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 8697–8701.

[127] R. Tahir, M. T. Khan, X. Gong, A. Ahmed, A. Ghassami, H. Kazmi, M. Caesar, F. Zaffar, and N. Kiyavash, “Sneak-peek: High speed covert channels in data center networks,” in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, 2016, pp. 1–9.

[128] S. Heuser, B. Reaves, P. K. Pendyala, H. Carter, A. Dmitrienko, W. Enck, N. Kiyavash, A. Sadeghi, and P. Traynor, “Phonion: Practical protection of metadata in telephony networks,” in *Proceedings on Privacy Enhancing Technologies, 2017*, 2017, pp. 170–187.

[129] X. Gong, M. Rodrigues, and N. Kiyavash, “Invisible flow watermarks for channels with dependent substitution, deletion, and bursty insertion errors,” *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 11, pp. 1850–1859, 2013.

[130] A. Houmansadr, N. Kiyavash, and N. Borisov, “Non-blind watermarking of network flows,” *IEEE/ACM Transaction on Networking*, vol. 22, no. 4, pp. 1232–1244, Aug. 2014.

[131] A. Houmansadr, N. Kiyavash, and N. Borisov, “Multi-flow attack resistant watermarks for network flows,” in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 1497–1500.

[132] N. Kiyavash, A. Houmansadr, and N. Borisov, “Multi-flow attacks against network flow watermarking schemes,” in *Proceedings of the 17th Conference on Security Symposium*, 2008, pp. 307–320.

[133] A. Houmansadr, N. Kiyavash, and N. Borisov, “Rainbow: A robust and invisible non-blind watermark for network flows,” in *Proceedings of the Network and Distributed System Security Symposium (NDSS 2009)*, 2009.

[134] D. Cullina and N. Kiyavash, “Improved achievability and converse bounds for erdos-renyi graph matching,” in *Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science*, 2016, pp. 63–72.

[135] T. P. Coleman and N. Kiyavash, “Practical codes for queueing channels: An algebraic, state-space, message-passing approach,” in *2008 IEEE Information Theory Workshop*, 2008, pp. 318–322.

[136] N. Kiyavash and T. Coleman, “Covert timing channels codes for communication over interactive traffic,” in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 1485–1488.

[137] S. K. Gorantla, S. Kadloor, N. Kiyavash, T. P. Coleman, I. S. Moskowitz, and M. H. Kang, “Characterizing the efficacy of the nrl network pump in mitigating covert timing channels,” *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 1, pp. 64–75, 2012.

[138] X. Gong, N. Kiyavash, and N. Borisov, “Fingerprinting websites using remote traffic analysis,” in *Proceedings of the 17th ACM Conference on Computer and Communications Security*, 2010, pp. 684–686.

[139] X. Gong, N. Borisov, N. Kiyavash, and N. Schear, “Website detection using remote traffic analysis,” in *Privacy Enhancing Technologies*, Berlin, Heidelberg, 2012, pp. 58–78.

[140] N. Gravin, Y. Peres, and B. Sivan, “Towards optimal algorithms for prediction with expert advice,” in *Proceedings of the Twenty-seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, Philadelphia, PA, USA, 2016, pp. 528–547.

[141] J. Abernethy, M. K. Warmuth, and J. Yellin, “When random play is optimal against an adversary,” in *Proceedings of the 21th annual workshop on Computational learning theory (COLT 2008)*, Helsinki, Findland, 2008.

[142] R. Bellman, *Dynamic Programming.* Princeton University Press, 2003.

[143] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 2nd ed. Athena Scientific, 2000.

[144] A. Feinberg and A. Shwartz, *Handbook of Markov decision processes: Methods and applications.* Kluwer, 2002.

[145] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 1st ed. New York, NY, USA: John Wiley & Sons, Inc., 1994.

[146] S. M. Ross, *Introduction to Stochastic Dynamic Programming: Probability and Mathematical.* Academic Press, Inc., 1983.

[147] W. Lin and P. Kumar, "Optimal control of a queueing system with two heterogeneous servers," *IEEE Transactions on Automatic Control*, vol. 29, pp. 696–703, Aug. 1984.

[148] J. Walrand, "A note on optimal control of a queuing system with two heterogeneous servers," *Systems and Control Letters*, vol. 4, pp. 131 – 134, 1984.

[149] G. Koole, "A simple proof of the optimality of a threshold policy in a two-server queueing system," *Systems and Control Letters*, vol. 26, pp. 301–303, Dec. 1995.

[150] R. Larsen, "Control of multiple exponential servers with application to computer systems," Ph.D. dissertation, University of Maryland at College Park, College Park, MD, USA, 1981.

[151] M. L. Puterman and M. C. Shin, "Modified policy iteration algorithms for discounted markov decision problems," *Management Science*, vol. 24, pp. 1127–1137, July 1978.

[152] J. A. E. E. van Nunen, "A set of successive approximation methods for discounted markovian decision problems," *Zeitschrift für Operations Research*, vol. 20, pp. 203–208, Oct. 1976.

[153] J.-M. Lasry and P.-L. Lions, "Mean field games," *Japanese Journal of Mathematics*, vol. 2, pp. 229–260, Mar. 2007.

[154] O. Guéant, J.-M. Lasry, and P.-L. Lions, *Mean Field Games and Applications.* Springer Berlin Heidelberg, 2011.

[155] L. P. Kadanoff, "More is the same; phase transitions and mean field theories," *Journal of Statistical Physics*, vol. 137, pp. 777–797, Sep. 2009.

[156] S. Dasgupta and J. Langford, "Tutorial summary: Active learning," in *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009*, Montreal, Quebec, Canada, 2009.

[157] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *Journal of Machine Learning Research*, vol. 2, pp. 45–66, 2002.

[158] N. Roy and A. McCallum, "Toward optimal active learning through sampling estimation of error reduction," in *Proceedings of the Eighteenth International Conference on Machine Learning*, San Francisco, CA, USA, 2001, pp. 441–448.

[159] B. Settles, "Active learning literature survey," University of Wisconsin, Madison, Tech. Rep., 07 2010.

[160] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, Dublin, Ireland, July 1994, pp. 3–12.

[161] X. Zhu, J. Lafferty, and Z. Ghahramani, "Combining active learning and semi-supervised learning using gaussian fields and harmonic functions," in *ICML 2003 workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, 2003, pp. 58–65.

[162] T. Zhang and F. J. Oles, "A probability analysis on the value of unlabeled data for classification problems," in *Proceedings of the 17th International Conference on Machine Learning*, 2000, pp. 1191–1198.

[163] Y. Baram, R. El-Yaniv, and K. Luz, "Online choice of active learning algorithms," *The Journal of Machine Learning Research*, vol. 5, pp. 255–291, Dec. 2004.

[164] T. Osugi, D. Kim, and S. Scott, "Balancing exploration and exploitation: a new algorithm for active machine learning," in *Proceedings of Fifth IEEE International Conference on Data Mining, (ICDM)*, Washington, DC, USA, Nov. 2005.

[165] D. Bouneffouf, "Exponentiated gradient exploration for active learning," *Computers*, 2016.

[166] D. Sculley, "Online active learning methods for fast label-efficient spam filtering," in *Proceedings of the Fourth Conference on Email and Anti-Spam*, Berlin, Germany, 2007.

[167] S. Dasgupta, A. T. Kalai, and C. Monteleoni, "Analysis of perceptron-based active learning," *The Journal of Machine Learning Research*, vol. 10, pp. 281–299, June 2009.

[168] D. Helmbold and S. Panizza, "Some label efficient learning results," in *Proceedings of the Tenth Annual Conference on Computational Learning Theory*, Nashville, Tennessee, USA, July 1997, pp. 218–230.

[169] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby, "Selective sampling using the query by committee algorithm," *Machine Learning*, vol. 28, pp. 133–168, 1997.

[170] F. Olsson, "A literature survey of active machine learning in the context of natural language processing," Swedish Institute of Computer Science, Tech. Rep., 2009.

[171] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, pp. 235–256, May 2002.

[172] S. Kale, "Multiarmed bandits with limited expert advice," *CoRR*, 2013.

[173] Y. Seldin, P. Bartlett, K. Crammer, and Y. Abbasi-yadkori, "Prediction with limited advice and multiarmed bandits with paid observations," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, Beijing, China, 2014, pp. 280–287.

[174] G. Lugosi, "Sequential prediction under incomplete feedback," in *Proceedings of the 2007 Conference on Artificial Intelligence Research and Development*, Amsterdam, The Netherlands, The Netherlands, 2007, pp. 3–5.

[175] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni, "Worst-case analysis of selective sampling for linear-threshold algorithms," in *Proceedings of Advances in Neural Information Processing Systems 17 (NIPS 2005)*, Cambridge, MA, USA, 2005, p. 241248.

[176] P. Zhao, S. C. H. Hoi, and J. Zhuang, "Active learning with expert advice," in *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence (UAI2013)*, Bellevue, Washington, USA, July 2013.

[177] J. Belluz, M. Gaudesi, G. Squillero, and A. Tonda, "Operator selection using improved dynamic multi-armed bandit," in *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, 2015, pp. 1311–1317.

[178] H. Valizadegan, R. Jin, and S. Wang, "Learning to trade off between exploration and exploitation in multiclass bandit prediction," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2011, pp. 204–212.

[179] S. M. Kakade, S. Shalev-Shwartz, and A. Tewari, "Efficient bandit algorithms for online multiclass prediction," in *Proceedings of the 25th International Conference on Machine Learning*, 2008, pp. 440–447.

[180] W. Li, X. Wang, R. Zhang, Y. Cui, J. Mao, and R. Jin, "Exploitation and exploration in a performance based contextual advertising system," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2010, pp. 27–36.

[181] B. Szörényi, R. Busa-Fekete, I. Hegedüs, R. Ormándi, M. Jelasity, and B. Kégl, "Gossip-based distributed stochastic bandit algorithms," in *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, 2013.

[182] N. Cesa-Bianchi and G. Lugosi, "Combinatorial bandits," *Journal of Computer System Science*, vol. 78, no. 5, pp. 1404–1422, Sep. 2012.

[183] E. Paulson, "A sequential procedure for selecting the population with the largest mean from k normal populations," *The Annals of Mathematical Statistics*, vol. 35, pp. 174–180, 1964.

[184] R. E. Bechhofer, "A sequential multiple-decision procedure for selecting the best one of several normal populations with a common unknown variance, and its use with various experimental designs," *Biometrics*, vol. 14, pp. 408–429, 1958.

[185] H. Robbins, "Some aspects of the sequential design of experiments," *Bulletin of the American Mathematics Society*, vol. 58, pp. 527–535, 1952.

[186] J. Gittins, *Multi-Armed Bandit Allocation Indices.* Wiley-Interscience series in Systems and Optimization, 1989.

[187] S. Bubeck, R. Munos, and G. Stoltz, "Pure exploration in finitely-armed and continuous-armed bandits," *Theoretical Computer Science*, vol. 412, pp. 1832–1852, 2011.

[188] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM Journal of Computing*, vol. 32, pp. 48–77, Jan. 2003.

[189] J. Y. Audibert, S. Bubeck, and R. Munos, "Best arm identification in multi-armed bandits," in *Proceedings of 23th Conference on Learning Theory (COLT 2010)*, Haifa, Israel, 2010.

[190] V. Gabillon, M. Ghavamzadeh, and A. Lazaric, "Best arm identification: A unified approach to fixed budget and fixed confidence," in *Proceedings of Advances in Neural Information Processing Systems 25 (NIPS 2012)*, Nevada, USA, 2012.

[191] O. Maron and A. Moore, "Hoeffding races: Accelerating model selection search for classification and function approximation," in *Proceedings of Advances in Neural Information Processing Systems 6*, Denver, Colorado, USA, 1993, pp. 59–66.

[192] V. Mnih, C. Szepesvári, and J.-Y. Audibert, "Empirical bernstein stopping," in *Proceedings of the Twenty-Fifth International Conference on Machine Learning*, Helsinki, Findland, 2008, p. 672679.

[193] K. Jamieson, M. Malloy, R. Nowak, and S. Bubeck, "lil' ucb : An optimal exploration algorithm for multi-armed bandits," in *Proceedings of 27th Conference on Learning Theory (COLT 2014)*, Bacerlona, Spain, 2014.

[194] S. Mannor and J. N. Tsitsiklis, "The sample complexity of exploration in the multiarmed bandit problem," *The Journal of Machine Learning Research*, vol. 5, p. 623648, 2004.

[195] E. Even-Dar, S. Mannor, and Y. Mansour, "Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems," *The Journal of Machine Learning Research*, vol. 7, p. 10791105, 2006.

[196] S. Bubeck, T. Wang, and N. Viswanathan, "Multiple identifications in multi-armed bandits," in *Proceedings of the 30th International Conference on International Conference on Machine Learning*, Atlanta, GA, USA, 2013, pp. 258–265.

[197] S. Kalyanakrishnan and P. Stone, "Efficient selection of multiple bandit arms: Theory and practice," in *Proceedings of the Twenty-Seventh International Conference on Machine Learning*, Haifa, Israel, 2010, pp. 511–518.

[198] S. Kalyanakrishnan, A. Tewari, P. Auer, and P. Stone, "Pac subset selection in stochastic multiarmed bandits," in *Proceedings of the 29th International Coference on International Conference on Machine Learning*, Edinburgh, Scotland, 2012, pp. 227–234.

[199] V. Borkar, *Stochastic Approximation : A Dynamical Systems Viewpoint.* Cambridge University Press, 2008.

[200] D. Blackwell, "Discounted dynamic programming," *The Annals of Mathematical Statistics*, vol. 36, pp. 226–235, 1965.

[201] S. Bubeck, "Bandits games and clustering foundations," Ph.D. dissertation, Université Lille, 2010.

[202] D. Fudenberg and J. Tirole, *Game Theory.* Chambridge: The MIT Press, 1991.

[203] M. J. Osborne and A. Rubinstein, *A Course in Game Theory.* The MIT Press, 1994.