New Directions in Online Learning: Boosting, Partial Information, and Non-Stationarity

by

Young Hun Jung

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Statistics)
in the University of Michigan
2020

Doctoral Committee:

Associate Professor Ambuj Tewari, Chair
Associate Professor Long Nguyen
Professor Clayton Scott
Professor Ji Zhu

Young Hun Jung
yhjung@umich.edu
ORCID iD: 0000-0003-1625-4526

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

LIST OF APPENDICES

APPENDIX

ABSTRACT

Online learning, where a learning algorithm fits a model on-the-fly with streaming data, has become an important research area in machine learning. Batch learning, where the entire data set has to be available to the learning algorithm, is not always a suitable paradigm for the big data era. It is increasingly common in many practical situations, such as online ads prediction or control of self-driving cars, that data instances naturally arrive in a sequential manner. In these situations, researchers want to update their model in an online fashion. This dissertation pursues several topics at the frontier of online learning research.

In Chapter 2 and Chapter 3, the journey starts with *online boosting*. Online boosting studies how to combine multiple online weak learners to get a stronger learner. Chapter 2 considers online *multi-class classification* problems. Chapter 3 focuses on the more challenging *multi-label ranking* problem where there are multiple correct labels and the learner outputs a ranking of labels based on their relevance. In both chapters, an optimal algorithm and an adaptive algorithm are proposed. The optimal algorithms require a minimal number of weak learners to attain the desired accuracy. The adaptive algorithms are practically more useful since they do not require a priori knowledge about the strength of weak learners and are more computationally efficient. The adaptive algorithms are not statistically optimal but they still come with reasonable performance guarantees. The empirical results on real data sets support the theoretical findings and the proposed boosting algorithms outperformed existing competitors on benchmark data sets.

Chapter 4 considers the partial information setting, where the learner does not receive the true labels. Partial feedback is common in practice as obtaining complete feedback can be costly.

The chapter revisits the boosting algorithms that are presented in Chapter 2 and Chapter 3 and extends them to work with partial information feedback. Despite the learner receiving much less information, comparable performance guarantees can be made.

Later in Chapter 5 and Chapter 6, we move on to another interesting area in online learning called *restless bandit problems*. Unlike the classical (stochastic) multi-armed bandit problems where the reward distributions are unknown but stationary, in restless bandit problems the distributions can change over time. This extra layer of complexity allows us to study more complicated models, but the analysis becomes even more difficult. In restless bandit problems, it is assumed that each arm has a state that evolves according to an unknown Markov process, and the reward distribution depends on the arm's current state. This setting can be thought of as a sub-class of *reinforcement learning* and the partial observability inherent in this problem makes the analysis very challenging. The well known *Thompson Sampling* algorithm is analyzed and a Bayesian regret bound for it is derived. Chapter 5 considers the episodic case where the system periodically resets. Chapter 6 extends the analysis to the more challenging non-episodic (i.e., infinite time horizon) case. In both settings, Thompson Sampling algorithms (with slight modifications) enjoy sub-linear regret bounds, and the empirical results on simulated data support this fact. The experiments also suggest the possibility that the algorithm can be used in the frequentist setting even though the theoretical bounds are only shown for the Bayesian regret.

CHAPTER 1

Introduction

Online Learning is a well-developed branch of machine learning that studies how to dynamically update models as new data instances arrive. This field is distinguished from classical *batch learning* where there is a training set upon which the model is fully optimized. As the model keeps changing along with the data, theoretically providing a performance guarantee in this setting can be challenging. There are two main reasons why online learning gets so much of researchers' attention nowadays. First, the enormous size of the data that we have makes it almost impossible to load them on a memory. Since a single computer cannot process the entire training set simultaneously, the batch learning becomes no longer an option, and the scientists must split the training set and update the model dynamically. Second, in many applications, data naturally arrive in a sequential manner. For example, in ads click prediction [Cheng et al., 2012], a new user comes to the platform and gives feedback by either clicking or ignoring the ads that are selected by an online model. In this scenario, even the i.i.d. assumption can easily break, and the researchers are looking for performance guarantees without such an assumption. Throughout this thesis, I will discuss several topics in online learning and propose new directions.

Chapter 2 and Chapter 3 discuss *online boosting*. Boosting studies how to combine weak learners to obtain a stronger learner, and online boosting aggregates multiple online learners. Recall that the classical boosting adds an additional weak learner in each round of iteration while the training set is fixed. In contrast, as data arrive sequentially in online learning, this paradigm is no longer feasible. Instead, online boosting algorithms start with a fixed number of online weak learners and update the weights in an online fashion (with each weak learner keeps updating its internal parameters as well). In this manuscript, multiple prediction problems are considered. Chapter 2 studies multi-class classification problems, and Chapter 3 considers multi-label ranking (MLR) problems. In MLR problems, there are multiple correct answers, and the learner predicts a ranking of labels based on their predicted relevance. In both settings, one optimal algorithm and one adaptive algorithm are proposed with theoretical guarantees. The optimal algorithm requires the minimal number of weak

learners to attain the desired accuracy (with a proven lower bound), while the adaptive algorithm is computationally more feasible. The experimental results show that the proposed algorithms beat the state-of-art results, and adaptive algorithms demonstrate competitive performance despite their loose theoretical bounds.

Chapter 4 extends the boosting algorithms in the preceding chapters to the partial information setting. In many scenarios, such as too many candidate labels or complex combinatorial structures, obtaining true labels can cost too much time or money. For example, in online recommendation problems, users only show their preferences by clicking relevant items among the list presented by a learner. In this case, if a user does not click any items, there is no way that the learner can infer what would be the relevant items for the user. Designing a boosting algorithm with partial feedback is very difficult because there are multiple weak learners. Since the correct label is not available to the boosting algorithm, some weak learners can only get very limited feedback (even weaker than the feedback that the boosting algorithm receives). The multi-class classification with bandit feedback and the MLR with top-k feedback are considered in this chapter. Quite surprisingly, the asymptotic accuracy remains the same as the full information algorithms, and the partial information only increases the sample complexity.

In Chapter 5 and Chapter 6, we move on to the non-stationary world with partial feedback. The stochastic *multi-armed bandits* assume the reward distributions are stationary. That is to say, the distribution remains the same over time. This stationary assumption often fails in practices. For example, Meshram et al. [2017] consider a recommendation system where a user's preference depends on his current state. To tackle this problem, researchers have studied *restless bandits*, where each arm has a state which evolves according to some Markov process and the reward distribution is a function of the current state. This extra layer of flexibility allows restless bandits to solve more complicated modeling problems but at the same time makes the analysis of learning algorithms much more challenging. The famous *Thompson Sampling* algorithms with slight modifications are analyzed in this setting. Chapter 5 assumes the episodic case where the system periodically resets, which makes the setting simpler, and Chapter 6 extends this to the non-episodic case. In both settings, Bayesian regret bounds are proven, and the experimental results even suggest that the proposed algorithms can be used to optimize the frequentist regret as well.

1.1 List of Completed Projects

The following list consists of completed projects during my Ph.D. (sorted chronologically):

1. Online Multiclass Boosting, NIPS 2017 (joint work with Jack Goetz), [Jung et al., 2017]

2. Online Boosting Algorithms for Multi-label Ranking, AISTATS 2018 , [Jung and Tewari, 2018]

3. Online Multiclass Boosting with Bandit Feedback, AISTATS 2019 (joint work with Daniel Zhang), [Zhang et al., 2019]

4. Regret Bounds for Thompson Sampling in Episodic Restless Bandit Problems, NeurIPS 2019, [Jung and Tewari, 2019]

5. Online Learning via the Differential Privacy Lens, NeurIPS 2019 (joint work with Jacob Abernethy, Chansoo Lee, and Audra McMillan), [Abernethy et al., 2019]

6. Thompson Sampling in Non-Episodic Restless Bandits, arXiv preprint 2019 (joint work with Marc Abeille), [Jung et al., 2019]

7. Online Boosting for Multilabel Ranking with Top-k Feedback, arXiv preprint 2019 (joint work with Daniel Zhang), [Zhang et al., 2019]

This manuscript consists of a subset of these projects to which I solely (or primarily) contributed. The only exceptions are project 3 and 7 that are briefly summarized in Chapter 4. I was a second author of these two papers but contributed to complete the theoretical aspects (which are the summarized portion here). Chapter 2 corresponds to paper 1; Chapter 3 to paper 2; Chapter 5 to paper 4; and Chapter 6 to paper 6.

The only paper that is missed in this manuscript is paper 5. This work is also very interesting in that it bridges two fairly different fields: online learning and differential privacy. We propose a condition, called differential stability, inspired by differential privacy, and if an online learning algorithm satisfies this condition, then we provide a methodology to prove its regret bound. This framework turns out to be very general in that we could provide unifying proofs for existing online learning algorithms in different settings. Additionally, it can be used to design new online algorithms as well. This work is omitted in this manuscript because the topic is not fully aligned with the main theme of the dissertation. Interested readers can refer to the complete paper.

CHAPTER 2

Online Multiclass Boosting

Boosting methods are ensemble learning methods that aggregate several (not necessarily) weak learners to build a stronger learner [1]. When used to aggregate reasonably strong learners, boosting has been shown to produce results competitive with other state-of-the-art methods (e.g., Korytkowski et al. [2016], Zhang and Wang [2014]). Until recently theoretical development in this area has been focused on batch binary settings where the learner can observe the entire training set at once, and the labels are restricted to be binary (cf. Schapire and Freund [2012]). In the past few years, progress has been made to extend the theory and algorithms to more general settings.

Dealing with *multiclass classification* turned out to be more subtle than initially expected. Mukherjee and Schapire [2013] unify several different proposals made earlier in the literature and provide a general framework for multiclass boosting. They state their weak learning conditions in terms of *cost matrices* that have to satisfy certain restrictions: for example, labeling with the ground truth should have less cost than labeling with some other labels. A weak learning condition, just like the binary condition, states that the performance of a learner, now judged using a cost matrix, should be better than a random guessing baseline. One particular condition they call the *edge-over-random* condition, proves to be sufficient for boostability. The edge-over-random condition will also figure prominently in this chapter. They also consider a necessary and sufficient condition for boostability but it turns out to be computationally intractable to be used in practice.

A recent trend in modern machine learning is to train learners in an *online setting* where the instances come sequentially and the learner has to make predictions instantly. Oza [2005] initially proposed an online boosting algorithm that has accuracy comparable with the batch version, but it took several years to design an algorithm with theoretical justification (Chen et al. [2012]). Beygelzimer et al. [2015] achieved a breakthrough by proposing an optimal algorithm in online binary settings and an adaptive algorithm that works quite well in practice. These theories in online

[1] This chapter is based on the paper with the same title that appeared in NeurIPS 2017. My great colleague, Jack Goetz, performed a significant portion of the experiments.

binary boosting have led to several extensions. For example, Chen et al. [2014] combine one vs all method with binary boosting algorithms to tackle online multiclass problems with bandit feedback, and Hu et al. [2017] build a theory of boosting in regression setting.

In this work, we combine the insights and techniques of Mukherjee and Schapire [2013] and Beygelzimer et al. [2015] to provide a framework for online multiclass boosting. The cost matrix framework from the former work is adopted to propose an online weak learning condition that defines how well a learner can perform over a random guess (Definition 2.1). We show this condition is naturally derived from its batch setting counterpart. From this weak learning condition, a boosting algorithm (Algorithm 2.1) is proposed which is theoretically optimal in that it requires the minimal number of learners and sample complexity to attain a specified level of accuracy. We also develop an adaptive algorithm (Algorithm 2.2) which allows learners to have variable strengths. This algorithm is theoretically less efficient than the optimal one, but the experimental results show that it is quite comparable and sometimes even better due to its adaptive property. Both algorithms not only possess theoretical proofs of mistake bounds, but also demonstrate superior performance over preexisting methods.

2.1 Preliminaries

We first describe the basic setup for online boosting. While in the batch setting, an additional weak learner is trained at every iteration, in the online setting, the algorithm starts with a fixed count of N *weak learners* and a *booster* which manages the weak learners. There are k possible labels $[k] := \{1, \cdots, k\}$ and k is known to the learners. At each iteration $t = 1, \cdots, T$, an *adversary* picks a labeled example $(\mathbf{x}_t, y_t) \in \mathcal{X} \times [k]$, where $\mathcal{X}$ is some domain, and reveals $\mathbf{x}_t$ to the booster. Once the booster observes the unlabeled data $\mathbf{x}_t$, it gathers the weak learners' predictions and makes a final prediction. Throughout this paper, index i takes values from 1 to N; t from 1 to T; and l from 1 to k.

We utilize the *cost matrix framework*, first proposed by Mukherjee and Schapire [2013], to develop multiclass boosting algorithms. This is a key ingredient in the multiclass extension as it enables different penalization for each pair of correct label and prediction, and we further develop this framework to suit the online setting. The booster sequentially computes *cost matrices* $\{\mathbf{C}_t^i \in \mathbb{R}^{k \times k} \mid i = 1, \cdots, N\}$, sends $(\mathbf{x}_t, \mathbf{C}_t^i)$ to the i^{th} weak learner WL^i, and gets its prediction $l_t^i \in [k]$. Here the cost matrix $\mathbf{C}_t^i$ plays a role of loss function in that WL^i tries to minimize the cumulative cost $\sum_t \mathbf{C}_t^i[y_t, l_t^i]$. As the booster wants each learner to predict the correct label, it wants to set the diagonal entries of $\mathbf{C}_t^i$ to be minimal among its row. At this stage, the true label y_t is not

revealed yet, but the previous weak learners' predictions can affect the computation of the cost matrix for the next learner. Given a matrix $\mathbf{C}$, the $(i,j)^{th}$ entry will be denoted by $\mathbf{C}[i,j]$, and i^{th} row vector by $\mathbf{C}[i]$.

Once all the learners make predictions, the booster makes the final prediction $\hat{y}_t$ by majority votes. The booster can either take simple majority votes or weighted ones. In fact for the adaptive algorithm, we will allow weighted votes so that the booster can assign more weights on well-performing learners. The weight for WL^i at iteration t will be denoted by α_t^i. After observing the booster's final decision, the adversary reveals the true label y_t, and the booster suffers 0-1 loss $\mathbb{1}(\hat{y}_t \neq y_t)$. The booster also shares the true label to the weak learners so that they can train on this data point.

Two main issues have to be resolved to design a good boosting algorithm. First, we need to design the booster's strategy for producing cost matrices. Second, we need to quantify weak learner's ability to reduce the cumulative cost $\sum_{t=1}^{T} \mathbf{C}_t^i[y_t, l_t^i]$. The first issue will be resolved by introducing potential functions, which will be thoroughly discussed in Section 2.2.1. For the second issue, we introduce our online weak learning condition, a generalization of the weak learning assumption in Beygelzimer et al. [2015], stating that for any adaptively given sequence of cost matrices, weak learners can produce predictions whose cumulative cost is less than that incurred by random guessing. The online weak learning condition will be discussed in the following section. For the analysis of the adaptive algorithm, we use empirical edges instead of the online weak learning condition.

2.1.1 Online weak learning condition

We propose an online weak learning condition that states the weak learners are better than a random guess. We first define a baseline condition that is better than a random guess. Let $\Delta[k]$ denote a family of distributions over $[k]$ and $\mathbf{u}_\gamma^l \in \Delta[k]$ be a uniform distribution that puts γ more weight on the label l. For example, $\mathbf{u}_\gamma^1 = (\frac{1-\gamma}{k} + \gamma, \frac{1-\gamma}{k}, \cdots, \frac{1-\gamma}{k})$. For a given sequence of examples $\{(\mathbf{x}_t, y_t) \mid t = 1, \cdots, T\}$, $\mathbf{U}_\gamma \in \mathbb{R}^{T \times k}$ consists of rows $\mathbf{u}_\gamma^{y_t}$. Then we restrict the booster's choice of cost matrices to

$$\mathcal{C}_1^{eor} := \{\mathbf{C} \in \mathbb{R}^{k \times k} \mid \forall l, r \in [k],\ \mathbf{C}[l,l] = 0, \mathbf{C}[l,r] \geq 0,\ \text{and}\ ||\mathbf{C}[l]||_1 = 1\}.$$

Note that diagonal entries are minimal among the row, and $\mathcal{C}_1^{eor}$ also has a normalization constraint. A broader choice of cost matrices is allowed if one can assign importance weights on observations, which is possible for various learners. Even if the learner does not take the importance weight as an

input, we can achieve a similar effect by sending to the learner an instance with probability that is proportional to its weight. Interested readers can refer Beygelzimer et al. [2015, Lemma 1]. From now on, we will assume that our weak learners can take weight w_t as an input.

We are ready to present our online weak learning condition. This condition is in fact naturally derived from the batch setting counterpart that is well studied by Mukherjee and Schapire [2013]. The link is thoroughly discussed in Appendix A.1. For the scaling issue, we assume the weights w_t lie in $[0, 1]$.

Definition 2.1. (Online multiclass weak learning condition) *For parameters $\gamma, \delta \in (0, 1)$, and $S > 0$, a pair of online learner and an adversary is said to satisfy online weak learning condition with parameters δ, γ, and S if for any sample length T, any adaptive sequence of labeled examples, and for any adaptively chosen series of pairs of weight and cost matrix $\{(w_t, \boldsymbol{C}_t) \in [0, 1] \times \mathcal{C}_1^{eor} \mid t = 1, \cdots, T\}$, the learner can generate predictions $\hat{y}_t$ such that with probability at least $1 - \delta$,*

$$\sum_{t=1}^{T} w_t \boldsymbol{C}_t[y_t, \hat{y}_t] \leq \boldsymbol{C} \bullet \boldsymbol{U}'_\gamma + S = \frac{1-\gamma}{k} ||\boldsymbol{w}||_1 + S, \tag{2.1}$$

where $\boldsymbol{C} \in \mathbb{R}^{T \times k}$ consists of rows of $w_t \boldsymbol{C}_t[y_t]$ and $\boldsymbol{A} \bullet \boldsymbol{B}'$ denotes the Frobenius inner product $Tr(\boldsymbol{A}\boldsymbol{B}')$. $\boldsymbol{w} = (w_1, \cdots, w_T)$ and the last equality holds due to the normalized condition on $\mathcal{C}_1^{eor}$. γ is called an edge, and S an excess loss.

Remark. *Notice that this condition is imposed on a pair of learner and adversary instead of solely on a learner. This is because no learner can satisfy this condition if the adversary draws samples in a completely adaptive manner. The probabilistic statement is necessary because many online algorithms' predictions are not deterministic. The excess loss requirement is needed since an online learner cannot produce meaningful predictions before observing a sufficient number of examples.*

2.2 Optimal algorithm

We describe the booster's optimal strategy for designing cost matrices. We first introduce a general theory without specifying the loss, and later investigate the asymptotic behavior of cumulative loss suffered by our algorithm under the specific 0-1 loss. We adopt the potential function framework from Mukherjee and Schapire [2013] and extend it to the online setting. Potential functions help both in designing cost matrices and in proving the mistake bound of the algorithm.

2.2.1 A general online multiclass boost-by-majority (OnlineMBBM) algorithm

We will keep track of the weighted cumulative votes of the first i weak learners for the sample $\mathbf{x}_t$ by $\mathbf{s}_t^i := \sum_{j=1}^{i} \alpha_t^j \mathbf{e}_{l_t^j}$, where α_t^i is the weight of WL^i, l_t^i is its prediction and $\mathbf{e}_j$ is the j^{th} standard basis vector. For the optimal algorithm, we assume that $\alpha_t^i = 1,\ \forall i, t$. In other words, the booster makes the final decision by simple majority votes. Given a cumulative vote $\mathbf{s} \in \mathbb{R}^k$, suppose we have a loss function $L^r(\mathbf{s})$ where r denotes the correct label. We call a loss function *proper*, if it is a decreasing function of $\mathbf{s}[r]$ and an increasing function of other coordinates (we alert the reader that "proper loss" has at least one other meaning in the literature). From now on, we will assume that our loss function is proper. A good example of proper loss is multiclass 0-1 loss:

$$L^r(\mathbf{s}) := \mathbb{1}(\max_{l \neq r} \mathbf{s}[l] \geq \mathbf{s}[r]). \tag{2.2}$$

The purpose of the potential function $\phi_i^r(\mathbf{s})$ is to estimate the booster's loss when there remain i learners until the final decision and the current cumulative vote is $\mathbf{s}$. More precisely, we want potential functions to satisfy the following conditions:

$$\begin{aligned} \phi_0^r(\mathbf{s}) &= L^r(\mathbf{s}), \\ \phi_{i+1}^r(\mathbf{s}) &= \mathbb{E}_{l \sim \mathbf{u}_\gamma^r} \phi_i^r(\mathbf{s} + \mathbf{e}_l). \end{aligned} \tag{2.3}$$

Readers should note that $\phi_i^r(\mathbf{s})$ also inherits the proper property of the loss function, which can be shown by induction. The condition (2.3) can be loosened by replacing both equalities by inequalities "$\geq$", but in practice we usually use equalities.

Now we describe the booster's strategy for designing cost matrices. After observing $\mathbf{x}_t$, the booster sequentially sets a cost matrix $\mathbf{C}_t^i$ for WL^i, gets the weak learner's prediction l_t^i and uses this in the computation of the next cost matrix $\mathbf{C}_t^{i+1}$. Ultimately, booster wants to set

$$\mathbf{C}_t^i[r, l] = \phi_{N-i}^r(\mathbf{s}_t^{i-1} + \mathbf{e}_l). \tag{2.4}$$

However, this cost matrix does not satisfy the condition of $\mathcal{C}_1^{eor}$, and thus should be modified in order to utilize the weak learning condition. First to make the cost for the true label equal to 0, we subtract $\mathbf{C}_t^i[r, r]$ from every element of $\mathbf{C}_t^i[r]$. Since the potential function is proper, our new cost matrix still has non-negative elements after the subtraction. We then normalize the row so that each

Algorithm 2.1 Online Multiclass Boost-by-Majority (OnlineMBBM)

for $t = 1, \cdots, T$ **do**
  Receive example $\mathbf{x}_t$
  Set $\mathbf{s}_t^0 = \mathbf{0} \in \mathbb{R}^k$
  for $i = 1, \cdots, N$ **do**
    Set the normalized cost matrix $\mathbf{D}_t^i$ according to (2.5) and pass it to WL^i
    Get weak predictions $l_t^i = WL^i(\mathbf{x}_t)$ and update $\mathbf{s}_t^i = \mathbf{s}_t^{i-1} + \mathbf{e}_{l_t^i}$
  end for
  Predict $\hat{y}_t := \operatorname{argmax}_l \mathbf{s}_t^N[l]$ and receive true label y_t
  for $i = 1, \cdots, N$ **do**
    Set $\mathbf{w}^i[t] = \sum_{l=1}^k [\phi_{N-i}^{y_t}(\mathbf{s}_t^{i-1} + \mathbf{e}_l) - \phi_{N-i}^{y_t}(\mathbf{s}_t^{i-1} + \mathbf{e}_{y_t})]$
    Pass training example with weight $(\mathbf{x}_t, y_t, \mathbf{w}^i[t])$ to WL^i
  end for
end for

row has ℓ_1 norm equal to 1. In other words, we get new normalized cost matrix

$$\mathbf{D}_t^i[r, l] = \frac{\phi_{N-i}^r(\mathbf{s}_t^{i-1} + \mathbf{e}_l) - \phi_{N-i}^r(\mathbf{s}_t^{i-1} + \mathbf{e}_r)}{\mathbf{w}^i[t]}, \tag{2.5}$$

where $\mathbf{w}^i[t] := \sum_{l=1}^k \phi_{N-i}^r(\mathbf{s}_t^{i-1} + \mathbf{e}_l) - \phi_{N-i}^r(\mathbf{s}_t^{i-1} + \mathbf{e}_r)$ plays the role of weight. It is still possible that a row vector $\mathbf{C}_t^i[r]$ is a zero vector so that normalization is impossible. In this case, we just leave it as a zero vector. Our weak learning condition (2.1) still works with cost matrices some of whose row vectors are zeros because however the learner predicts, it incurs no cost.

After defining cost matrices, the rest of the algorithm is straightforward except we have to estimate $||\mathbf{w}^i||_\infty$ to normalize the weight. This is necessary because the weak learning condition assumes the weights lying in $[0, 1]$. We cannot compute the exact value of $||\mathbf{w}^i||_\infty$ until the last instance is revealed, which is fine as we need this value only in proving the mistake bound. The estimate w^{i*} for $||\mathbf{w}^i||_\infty$ requires to specify the loss, and we postpone the technical parts to Appendix A.2.2. Interested readers may directly refer Lemma A.5 before proceeding. Once the learners generate predictions after observing cost matrices, the final decision is made by simple majority votes. After the true label is revealed, the booster updates the weight and sends the labeled instance with weight to the weak learners. The pseudocode for the entire algorithm is depicted in Algorithm 2.1. The algorithm is named after Beygelzimer et al. [2015, OnlineBBM], which is in fact OnlineMBBM with binary labels.

We are ready to present the mistake bound of general OnlineMBBM. The proof appears in Appendix A.2.1 where the main idea is adopted from Beygelzimer et al. [2015, Lemma 3].

Theorem 2.2. (Cumulative loss bound for OnlineMBBM) *Suppose weak learners and an adversary satisfy the online weak learning condition (2.1) with parameters $\delta, \gamma,$ and S. For any T and N satisfying $\delta \ll \frac{1}{N}$, and any adaptive sequence of labeled examples generated by the adversary, the final loss suffered by OnlineMBBM satisfies the following inequality with probability $1 - N\delta$:*

$$\sum_{t=1}^{T} L^{y_t}(\boldsymbol{s}_t^N) \leq \phi_N^1(\boldsymbol{0})T + S\sum_{i=1}^{N} w^{i*}. \tag{2.6}$$

Here $\phi_N^1(\mathbf{0})$ plays a role of asymptotic error rate and the second term determines the sample complexity. We will investigate the behavior of those terms under the 0-1 loss in the following section.

2.2.2 Mistake bound under 0-1 loss and its optimality

From now on, we will specify the loss to be multiclass 0-1 loss defined in (2.2), which might be the most relevant measure in multiclass problems. To present a specific mistake bound, two terms in the RHS of (2.6) should be bounded. This requires an approximation of potentials, which is technical and postponed to Appendix A.2.2. Lemma A.4 and A.5 provide the bounds for those terms. We also mention another bound for the weight in the remark after Lemma A.5 so that one can use whichever tighter. Combining the above lemmas with Theorem 2.2 gives the following corollary. The additional constraint on γ comes from Lemma A.5.

Corollary 2.3. (0-1 loss bound of OnlineMBBM) *Suppose weak learners and an adversary satisfy the online weak learning condition (2.1) with parameters $\delta, \gamma,$ and S, where $\gamma < \frac{1}{2}$. For any T and N satisfying $\delta \ll \frac{1}{N}$ and any adaptive sequence of labeled examples generated by the adversary, OnlineMBBM can generate predictions $\hat{y}_t$ that satisfy the following inequality with probability $1 - N\delta$:*

$$\sum_{t=1}^{T} \mathbb{1}(y_t \neq \hat{y}_t) \leq (k-1)e^{-\frac{\gamma^2 N}{2}}T + \tilde{O}(k^{5/2}\sqrt{N}S). \tag{2.7}$$

Therefore in order to achieve error rate ϵ, it suffices to use $N = \Theta(\frac{1}{\gamma^2}\ln\frac{k}{\epsilon})$ weak learners, which gives an excess loss bound of $\tilde{\Theta}(\frac{k^{5/2}}{\gamma}S)$.

Remark. *Note that the above excess loss bound gives a sample complexity bound of $\tilde{\Theta}(\frac{k^{5/2}}{\epsilon\gamma}S)$. If we use alternative weight bound to get kNS as an upper bound for the second term in (2.6), we end up having $\tilde{O}(kNS)$. This will give an excess loss bound of $\tilde{\Theta}(\frac{k}{\gamma^2}S)$.*

We now provide lower bounds on the number of learners and sample complexity for arbitrary online boosting algorithms to evaluate the optimality of OnlineMBBM under 0-1 loss. In particular, we construct weak learners that satisfy the online weak learning condition (2.1) and have almost matching asymptotic error rate and excess loss compared to those of OnlineMBBM as in (2.7). Indeed we can prove that the number of learners and sample complexity of OnlineMBBM is optimal up to logarithmic factors, ignoring the influence of the number of classes k. Our bounds are possibly suboptimal up to polynomial factors in k, and the problem to fill the gap remains open. The detailed proof and a discussion of the gap can be found in Appendix A.2.3. Our lower bound is a multiclass version of Beygelzimer et al. [2015, Theorem 3].

Theorem 2.4. (Lower bounds for N and T) *For any $\gamma \in (0, \frac{1}{4})$, $\delta, \epsilon \in (0, 1)$, and $S \geq \frac{k \ln(\frac{1}{\delta})}{\gamma}$, there exists an adversary with a family of learners satisfying the online weak learning condition (2.1) with parameters δ, γ, and S, such that to achieve asymptotic error rate ϵ, an online boosting algorithm requires at least $\Omega(\frac{1}{k^2\gamma^2} \ln \frac{1}{\epsilon})$ learners and a sample complexity of $\Omega(\frac{k}{\epsilon\gamma} S)$.*

2.3 Adaptive algorithm

The online weak learning condition imposes minimal assumptions on the asymptotic accuracy of learners, and obviously it leads to a solid theory of online boosting. However, it has two main practical limitations. The first is the difficulty of estimating the edge γ. Given a learner and an adversary, it is by no means a simple task to find the maximum edge that satisfies (2.1). The second issue is that different learners may have different edges. Some learners may in fact be quite strong with significant edges, while others are just slightly better than a random guess. In this case, OnlineMBBM has to pick the minimum edge as it assumes common γ for all weak learners. It is obviously inefficient in that the booster underestimates the strong learners' accuracy.

Our adaptive algorithm will discard the online weak learning condition to provide a more practical method. Empirical edges $\gamma_1, \cdots, \gamma_N$ (see Section 2.3.2 for the definition) are measured for the weak learners and are used to bound the number of mistakes made by the boosting algorithm.

2.3.1 Choice of loss function

Adaboost, proposed by Freund et al. [1999], is arguably the most popular boosting algorithm in practice. It aims to minimize the exponential loss, and has many variants which use some other surrogate loss. The main reason of using a surrogate loss is ease of optimization; while 0-1 loss is not even continuous, most surrogate losses are convex. We adopt the use of a surrogate loss for the

same reason, and throughout this section will discuss our choice of surrogate loss for the adaptive algorithm.

Exponential loss is a very strong candidate in that it provides a closed form for computing potential functions, which are used to design cost matrices (cf. Mukherjee and Schapire [2013, Theorem 13]). One property of online setting, however, makes it unfavorable. Like OnlineMBBM, each data point will have a different weight depending on weak learners' performance, and if the algorithm uses exponential loss, this weight will be an exponential function of difference in weighted cumulative votes. With this exponentially varying weights among samples, the algorithm might end up depending on very small portion of observed samples. This is undesirable because it is easier for the adversary to manipulate the sample sequence to perturb the learner.

To overcome exponentially varying weights, Beygelzimer et al. [2015] use logistic loss in their adaptive algorithm. Logistic loss is more desirable in that its derivative is bounded and thus weights will be relatively smooth. For this reason, we will also use multiclass version of logistic loss:

$$L^r(\mathbf{s}) =: \sum_{l \neq r} \log(1 + \exp(\mathbf{s}[r] - \mathbf{s}[r])). \tag{2.8}$$

We still need to compute potential functions from logistic loss in order to calculate cost matrices. Unfortunately, Mukherjee and Schapire [2013] use a unique property of exponential loss to get a closed form for potential functions, which cannot be adopted to logistic loss. However, the optimal cost matrix induced from exponential loss has a very close connection with the gradient of the loss (cf. Mukherjee and Schapire [2013, Lemma 22]). From this, we will design our cost matrices as following:

$$\mathbf{C}_t^i[r, l] := \begin{cases} \frac{1}{1+\exp(\mathbf{s}_t^{i-1}[r] - \mathbf{s}_t^{i-1}[l])} & , \text{if } l \neq r \\ -\sum_{j \neq r} \frac{1}{1+\exp(\mathbf{s}_t^{i-1}[r] - \mathbf{s}_t^{i-1}[j])} & , \text{if } l = r. \end{cases} \tag{2.9}$$

Readers should note that the row vector $\mathbf{C}_t^i[r]$ is simply the gradient of $L^r(\mathbf{s}_t^{i-1})$. Also note that this matrix does not belong to $\mathcal{C}_1^{eor}$, but it does guarantee that the correct prediction gets the minimal cost.

The choice of logistic loss over exponential loss is somewhat subjective. The undesirable property of exponential loss does not necessarily mean that we cannot build an adaptive algorithm using this loss. In fact, we can slightly modify Algorithm 2.2 to develop algorithms using different surrogates (exponential loss and square hinge loss). However, their theoretical bounds are inferior to the one with logistic loss. Interested readers can refer Appendix A.4, but it assumes understanding

of Algorithm 2.2.

2.3.2 Adaboost.OLM

Our work is a generalization of Adaboost.OL by Beygelzimer et al. [2015], from which the name Adaboost.OLM comes with M standing for multiclass. We introduce a new concept of an *expert*. From N weak learners, we can produce N experts where expert i makes its prediction by weighted majority votes among the first i learners. Unlike OnlineMBBM, we allow varying weights α_t^i over the learners. As we are working with logistic loss, we want to minimize $\sum_t L^{y_t}(\mathbf{s}_t^i)$ for each i, where the loss is given in (2.8). We want to alert the readers to note that even though the algorithm tries to minimize the cumulative surrogate loss, its performance is still evaluated by 0-1 loss. The surrogate loss only plays a role of a bridge that makes the algorithm adaptive.

We do not impose the online weak learning condition on weak learners, but instead just measure the performance of WL^i by $\gamma_i := \frac{\sum_t \mathbf{C}_t^i[y_t, l_t^i]}{\sum_t \mathbf{C}_t^i[y_t, y_t]}$. This *empirical edge* will be used to bound the number of mistakes made by Adaboost.OLM. By definition of cost matrix, we can check

$$\mathbf{C}_t^i[y_t, y_t] \leq \mathbf{C}_t^i[y_t, l] \leq -\mathbf{C}_t^i[y_t, y_t], \ \forall l \in [k],$$

from which we can prove $-1 \leq \gamma_i \leq 1, \ \forall i$. If the online weak learning condition is met with edge γ, then one can show that $\gamma_i \geq \gamma$ with high probability when the sample size is sufficiently large.

Unlike the optimal algorithm, we cannot show the last expert that utilizes all the learners has the best accuracy. However, we can show at least one expert has a good predicting power. Therefore we will use classical *Hedge algorithm* (Littlestone and Warmuth [1989] and Freund and Schapire [1995]) to randomly choose an expert at each iteration with adaptive probability weight depending on each expert's prediction history.

Finally we need to address how to set the weight α_t^i for each weak learner. As our algorithm tries to minimize the cumulative logistic loss, we want to set α_t^i to minimize $\sum_t L^{y_t}(\mathbf{s}_t^{i-1} + \alpha_t^i \mathbf{e}_{l_t^i})$. This is again a classical topic in online learning, and we will use *online gradient descent*, proposed by Zinkevich [2003]. By letting, $f_t^i(\alpha) := L^{y_t}(\mathbf{s}_t^{i-1} + \alpha \mathbf{e}_{l_t^i})$, we need an online algorithm ensuring $\sum_t f_t^i(\alpha_t^i) \leq \min_{\alpha \in F} \sum_t f_t^i(\alpha) + R^i(T)$ where F is a feasible set to be specified later, and $R^i(T)$ is a regret that is sublinear in T. To apply Zinkevich [2003, Theorem 1], we need f_t^i to be convex and F to be compact. The first assumption is met by our choice of logistic loss, and for the second assumption, we will set $F = [-2, 2]$. There is no harm to restrict the choice of α_t^i by F because we can always scale the weights without affecting the result of weighted majority votes.

Algorithm 2.2 Adaboost.OLM

Initialize: $\forall i, v_1^i = 1, \alpha_1^i = 0$
for $t = 1, \cdots, T$ **do**
  Receive example $\mathbf{x}_t$
  Set $\mathbf{s}_t^0 = \mathbf{0} \in \mathbb{R}^k$
  for $i = 1, \cdots, N$ **do**
    Compute $\mathbf{C}_t^i$ according to (2.9) and pass it to WL^i
    Set $l_t^i = WL^i(\mathbf{x}_t)$ and $\mathbf{s}_t^i = \mathbf{s}_t^{i-1} + \alpha_t^i \mathbf{e}_{l_t^i}$
    Set $\hat{y}_t^i = \text{argmax}_l \, \mathbf{s}_t^i[l]$, the prediction of expert i
  end for
  Randomly draw i_t with $\mathbb{P}(i_t = i) \propto v_t^i$
  Predict $\hat{y}_t = \hat{y}_t^{i_t}$ and receive the true label y_t
  for $i = 1, \cdots, N$ **do**
    Set $\alpha_{t+1}^i = \Pi(\alpha_t^i - \eta_t f_t^{i\prime}(\alpha_t^i))$ using (2.10) and $\eta_t = \frac{2\sqrt{2}}{(k-1)\sqrt{t}}$
    Set $\mathbf{w}^i[t] = -\frac{\mathbf{C}_t^i[y_t, y_t]}{k-1}$ and pass $(\mathbf{x}_t, y_t, \mathbf{w}^i[t])$ to WL^i
    Set $v_{t+1}^i = v_t^i \cdot \exp(-\mathbb{1}(y_t \neq \hat{y}_t^i))$
  end for
end for

By taking derivatives, we get

$$f_t^{i\prime}(\alpha) = \begin{cases} \frac{1}{1+\exp(\mathbf{s}_t^{i-1}[y_t] - \mathbf{s}_t^{i-1}[l_t^i] - \alpha)} & \text{, if } l_t^i \neq y_t \\ -\sum_{j \neq y_t} \frac{1}{1+\exp(\mathbf{s}_t^{i-1}[j] + \alpha - \mathbf{s}_t^{i-1}[y_t])} & \text{, if } l_t^i = y_t. \end{cases} \tag{2.10}$$

This provides $|f_t^{i\prime}(\alpha)| \leq k - 1$. Now let $\Pi(\cdot)$ represent a projection onto F:

$$\Pi(\cdot) := \max\{-2, \min\{2, \cdot\}\}.$$

By setting $\alpha_{t+1}^i = \Pi(\alpha_t^i - \eta_t f_t^{i\prime}(\alpha_t^i))$ where $\eta_t = \frac{2\sqrt{2}}{(k-1)\sqrt{t}}$, we get $R^i(T) \leq 4\sqrt{2}(k-1)\sqrt{T}$. Readers should note that any learning rate of the form $\eta_t = \frac{c}{\sqrt{t}}$ would work, but our choice is optimized to ensure the minimal regret.

The pseudocode for Adaboost.OLM is presented in Algorithm 2.2. In fact, if we put $k = 2$, Adaboost.OLM has the same structure with Adaboost.OL. As in OnlineMBBM, the booster also needs to pass the weight along with labeled instance. According to (2.9), it can be inferred that the weight is proportional to $-\mathbf{C}_t^i[y_t, y_t]$.

2.3.3 Mistake bound and comparison to the optimal algorithm

Now we present the second main result that provides a mistake bound of Adaboost.OLM. The main structure of the proof is adopted from Beygelzimer et al. [2015, Theorem 4] but in a generalized cost matrix framework. The proof appears in Appendix A.3.

Theorem 2.5. (Mistake bound of Adaboost.OLM) *For any T and N, with probability $1-\delta$, the number of mistakes made by Adaboost.OLM satisfies the following inequality:*

$$\sum_{t=1}^{T} \mathbb{1}(y_t \neq \hat{y}_t) \leq \frac{8(k-1)}{\sum_{i=1}^{N} \gamma_i^2} T + \tilde{O}\left(\frac{kN^2}{\sum_{i=1}^{N} \gamma_i^2}\right),$$

where $\tilde{O}$ notation suppresses dependence on $\log \frac{1}{\delta}$.

Remark. *Note that this theorem naturally implies Beygelzimer et al. [2015, Theorem 4]. The difference in coefficients is due to different scaling of γ_i. In fact, their γ_i ranges from $[-\frac{1}{2}, \frac{1}{2}]$.*

Now that we have established a mistake bound, it is worthwhile to compare the bound with the optimal boosting algorithm. Suppose the weak learners satisfy the weak learning condition (2.1) with edge γ. For simplicity, we will ignore the excess loss S. As we have $\gamma_i = \frac{\sum_t \mathbf{C}_t^i[y_t, l_t^i]}{\sum_t \mathbf{C}_t^i[y_t, y_t]} \geq \gamma$ with high probability, the mistake bound becomes $\frac{8(k-1)}{\gamma^2 N} T + \tilde{O}(\frac{kN}{\gamma^2})$. In order to achieve error rate ϵ, Adaboost.OLM requires $N \geq \frac{8(k-1)}{\epsilon\gamma^2}$ learners and $T = \tilde{\Omega}(\frac{k^2}{\epsilon^2\gamma^4})$ sample size. Note that OnlineMBBM requires $N = \Omega(\frac{1}{\gamma^2} \ln \frac{k}{\epsilon})$ and $T = \min\{\tilde{\Omega}(\frac{k^{5/2}}{\epsilon\gamma}), \tilde{\Omega}(\frac{k}{\epsilon\gamma^2})\}$. Adaboost.OLM is obviously suboptimal, but due to its adaptive feature, its performance on real data is quite comparable to that by OnlineMBBM.

2.4 Experiments

We compare the new algorithms to existing ones for online boosting on several UCI data sets, each with k classes[2]. Table 2.1 contains some highlights, with additional results and experimental details in the Appendix A.5. Here we show both the average accuracy on the final 20% of each data set, as well as the average run time for each algorithm. Best decision tree gives the performance of the best of 100 online decision trees fit using the VFDT algorithm in Domingos and Hulten [2000], which were used as the weak learners in all other algorithms, and Online Boosting is an algorithm taken from Oza [2005]. Both provide a baseline for comparison with the new Adaboost.OLM and

[2]Codes are available at https://github.com/yhjung88/OnlineBoostingWithVFDT

OnlineMBBM algorithms. Best MBBM takes the best result from running the OnlineMBBM with five different values of the edge parameter γ.

Despite being theoretically weaker, Adaboost.OLM often demonstrates similar accuracy and sometimes outperforms Best MBBM, which exemplifies the power of adaptivity in practice. This power comes from the ability to use diverse learners efficiently, instead of being limited by the strength of the weakest learner. OnlineMBBM suffers from high computational cost, as well as the difficulty of choosing the correct value of γ, which in general is unknown, but when the correct value of γ is used it peforms very well. Finally in all cases Adaboost.OLM and OnlineMBBM algorithms outperform both the best tree and the preexisting Online Boosting algorithm, while also enjoying theoretical accuracy bounds.

Table 2.1: Comparison of algorithm accuracy on final 20% of data set and run time in seconds. Best accuracy on a data set reported in **bold**.

Data sets	k	Best decision tree		Online Boosting		Adaboost.OLM		Best MBBM	
Balance	3	0.768	8	0.772	19	0.754	20	**0.821**	42
Mice	8	0.608	105	0.399	263	0.561	416	**0.695**	2173
Cars	4	0.924	39	0.914	27	**0.930**	59	0.914	56
Mushroom	2	0.999	241	**1.000**	169	**1.000**	355	**1.000**	325
Nursery	4	0.953	526	0.941	302	0.966	735	**0.969**	1510
ISOLET	26	0.515	470	0.149	1497	0.521	2422	**0.635**	64707
Movement	5	0.915	1960	0.870	3437	0.962	5072	**0.988**	18676

CHAPTER 3

Online Boosting Algorithms for Multi-label Ranking

Multi-label learning has important practical applications (e.g., Schapire and Singer [2000]), and its theoretical properties continue to be studied (e.g., Koyejo et al. [2015]) [1]. In contrast to standard multi-class classifications, multi-label learning problems allow multiple correct answers. In other words, we have a fixed set of basic labels, and the actual label is a *subset* of the basic labels. Since the number of subsets increases exponentially as the number of basic labels grows, thinking of each subset as a different class leads to intractability.

It is quite common in applications for the multi-label learner to output a *ranking* of the labels on a new test instance. For example, the popular MULAN library designed by Tsoumakas et al. [2011] allows the output of multi-label learning to be a multi-label ranker. In this chapter, we focus on the multi-label ranking (MLR) setting. That is to say, the learner produces a *score vector* such that a label with a higher score will be ranked above a label with a lower score. We are particularly interested in *online* MLR settings where the data arrive sequentially. The online framework is designed to handle a large volume of data that accumulates rapidly. In contrast to classical *batch learners*, which observe the entire training set, online learners do not require the storage of a large amount of data in memory and can also adapt to non-stationarity in the data by updating the internal state as new instances arrive.

Boosting, first proposed by Freund and Schapire [1997], aggregates mildly powerful learners into a strong learner. It has been used to produce state-of-the-art results in a wide range of fields (e.g., Korytkowski et al. [2016] and Zhang and Wang [2014]). Boosting algorithms take weighted majority votes among weak learners' predictions, and the cumulative votes can be interpreted as a score vector. This feature makes boosting very well suited to MLR problems.

The theory of boosting has emerged in batch binary settings and became arguably complete (cf. Schapire and Freund [2012]), but its extension to an online setting is relatively new. To our knowledge, Chen et al. [2012] first introduced an online boosting algorithm with theoretical

[1] This chapter is based on the paper with the same title that appeared in AISTATS 2018.

justifications, and Beygelzimer et al. [2015] pushed the state-of-the-art in online binary settings further by proposing two online algorithms and proving optimality of one. Recent work has extended the theory to multi-class settings (cf. Chapter 2), but their scope remained limited to single-label problems.

In this chapter, we present the first online MLR boosting algorithms along with their theoretical justifications. The main contribution is to allow general forms of weak predictions whereas the previous online boosting algorithms only considered homogeneous prediction formats. By introducing a general way to encode weak predictions, our algorithms can combine binary, single-label, and MLR predictions.

After introducing the problem setting, we define an *edge* of an online learner over a random learner (Definition 3.1). Under the assumption that every weak learner has a known positive edge, we design an optimal way to combine their predictions (Section 3.2.1). In order to deal with practical settings where such an assumption is untenable, we present an adaptive algorithm that can aggregate learners with arbitrary edges (Section 3.2.2). In Section 3.3, we test our two algorithms on real data sets, and find that their performance is often comparable with, and sometimes better than, that of existing batch boosting algorithms for MLR.

3.1 Preliminaries

The number of candidate labels is fixed to be k, which is known to the learner. Without loss of generality, we may write the labels using integers in $[k] := \{1, \cdots, k\}$. We are allowing multiple correct answers, and the label Y_t is a subset of $[k]$. The labels in Y_t is called *relevant*, and those in Y_t^c, *irrelevant*. At time $t = 1, \cdots, T$, an *adversary* sequentially chooses a labeled example $(\mathbf{x}_t, Y_t) \in \mathcal{X} \times 2^{[k]}$, where $\mathcal{X}$ is some domain. Only the instance $\mathbf{x}_t$ is shown to the learner, and the label Y_t is revealed once the learner makes a prediction $\hat{\mathbf{y}}_t$. As we are interested in MLR settings, $\hat{\mathbf{y}}_t$ is a k dimensional score vector. The learner suffers a loss $L^{Y_t}(\hat{\mathbf{y}}_t)$ where the loss function will be specified later in Section 3.2.1.

In our boosting framework, we assume that the learner consists of a *booster* and N *weak learners*, where N is fixed before the training starts. This resembles a *manager-worker* framework in that booster distributes tasks by specifying losses, and each learner makes a prediction to minimize the loss. Booster makes the final decision by aggregating weak predictions. Once the true label is revealed, the booster shares this information so that weak learners can update their parameters for the next example.

3.1.1 Online weak learners and cost vector

We keep the form of weak predictions $\mathbf{h}_t$ general in that we only assume it is a distribution over $[k]$. This can in fact represent various types of predictions. For example, a *single-label prediction*, $l \in [k]$, can be encoded as a standard basis vector $\mathbf{e}_l$, or a *multi-label prediction* $\{l_1, \cdots, l_n\}$ by $\frac{1}{n}\sum_{i=1}^{n} \mathbf{e}_{l_i}$. Due to this general format, our boosting algorithm can even combine weak predictions of different formats. This implies that if a researcher has a strong family of binary learners, she can simply boost them without transforming them into multi-class learners through well known techniques such as *one-vs-all* or *one-vs-one* [Allwein et al., 2000].

We extend the *cost matrix framework*, first proposed by Mukherjee and Schapire [2013], as a means of communication between booster and weak learners. At round t, booster computes a cost vector $\mathbf{c}_t^i$ for the i^{th} weak learner WL^i, whose prediction $\mathbf{h}_t^i$ suffers the cost $\mathbf{c}_t^i \cdot \mathbf{h}_t^i$. The cost vector is unknown to WL^i until it produces $\mathbf{h}_t^i$, which is usual in online settings. Otherwise, WL^i can trivially minimize the cost.

A binary weak learning condition states a learner can attain over 50% accuracy however the sample weights are assigned. In our setting, cost vectors play the role of sample weights, and we will define the edge of a learner in similar manner.

Finally, we assume that weak learners can take an importance weight as an input, which is possible for many online algorithms.

3.1.2 General online boosting schema

We introduce a general algorithm schema shared by our algorithms. We denote the weight of WL^i at iteration t by α_t^i. We keep track of weighted cumulative votes through $\mathbf{s}_t^j := \sum_{i=1}^{j} \alpha_t^i \mathbf{h}_t^i$. That is to say, we can give more credits to well performing learners by setting larger weights. Furthermore, allowing negative weights, we can avoid poor learner's predictions. We call $\mathbf{s}_t^j$ a prediction made by *expert* j. In the end, the booster makes the final decision by following one of these experts.

The schema is summarized in Algorithm 3.1. We want to emphasize that the true label Y_t is only available once the final prediction $\hat{\mathbf{y}}_t$ is made. Computation of weights and cost vectors requires the knowledge of Y_t, and thus it happens after the final decision is made. To keep our theory general, the schema does not specify which weak learners to use (line 4 and 12). The specific ways to calculate other variables such as α_t^i, $\mathbf{c}_t^i$, and i_t depend on algorithms, which will be introduced in the next section.

Algorithm 3.1 Online boosting schema

Initialize: α_1^i for $i \in [N]$
for $t = 1, \cdots, T$ **do**
  Receive example $\mathbf{x}_t$
  Gather weak predictions $\mathbf{h}_t^i = WL^i(\mathbf{x}_t),\ \forall i$
  Record expert predictions $\mathbf{s}_t^j := \sum_{i=1}^{j} \alpha_t^i \mathbf{h}_t^i$
  Choose an index $i_t \in [N]$
  Make a final decision $\hat{\mathbf{y}}_t = \mathbf{s}_t^{i_t}$
  Get the true label Y_t
  Compute weights $\alpha_{t+1}^i,\ \forall i$
  Compute cost vectors $\mathbf{c}_t^i,\ \forall i$
  Weak learners suffer the loss $\mathbf{c}_t^i \cdot \mathbf{h}_t^i$
  Weak learners update the internal parameters
  Update booster's parameters, if any
end for

3.2 Algorithms with theoretical loss bounds

An essential factor in the performance of boosting algorithms is the predictive power of the individual weak learners. For example, if weak learners make completely random predictions, they cannot produce meaningful outcomes according to the booster's intention. We deal with this matter in two different ways. One way is to define an *edge* of a learner over a completely random learner and assume all weak learners have positive edges. Another way is to measure each learner's *empirical edge* and manipulate the weight α_t^i to maximize the accuracy of the final prediction. Even a learner that is worse than random guessing can contribute positively if we allow negative weights. The first method leads to OnlineBMR (Section 3.2.1), and the second to Ada.OLMR (Section 3.2.2).

3.2.1 Optimal algorithm

We first define the edge of a learner. Recall that weak learners suffer losses determined by cost vectors. Given the true label Y, the booster chooses a cost vector from

$$\mathcal{C}_0^{eor} := \{\mathbf{c} \in [0,1]^k \mid \max_{l \in Y} \mathbf{c}[l] \leq \min_{r \notin Y} \mathbf{c}[r], \\ \min_l \mathbf{c}[l] = 0 \text{ and } \max_l \mathbf{c}[l] = 1\},$$

where the name $\mathcal{C}_0^{eor}$ also appears in Chapter 2 and "eor" stands for *edge-over-random*. Since the booster wants weak learners to put higher scores at the relevant labels, costs at the relevant labels

should be less than those at the irrelevant ones. Restriction to $[0,1]^k$ makes sure that the learner's cost is bounded. Along with cost vectors, the booster passes the importance weights $w_t \in [0,1]$ so that the learner's cost becomes $w_t \mathbf{c}_t \cdot \mathbf{h}_t$.

We also construct a *baseline* learner that has edge γ. Its prediction $\mathbf{u}_\gamma^Y$ is also a distribution over $[k]$ that puts γ more probability for the relevant labels. That is to say, we can write

$$\mathbf{u}_\gamma^Y[l] = \begin{cases} a+\gamma & \text{if } l \in Y \\ a & \text{if } l \notin Y, \end{cases}$$

where the value of a depends on the number of relevant labels, $|Y|$.

Now we state our online weak learning condition.

Definition 3.1. (OnlineWLC) *For parameters $\gamma, \delta \in (0,1)$, and $S > 0$, a pair of an online learner and an adversary is said to satisfy OnlineWLC (δ, γ, S) if for any T, with probability at least $1-\delta$, the learner can generate predictions that satisfy*

$$\sum_{t=1}^{T} w_t \boldsymbol{c}_t \cdot \boldsymbol{h}_t \leq \sum_{t=1}^{T} w_t \boldsymbol{c}_t \cdot \boldsymbol{u}_\gamma^{Y_t} + S.$$

γ is called an edge, and S an excess loss.

This extends the condition in Definition 2.1. The probabilistic statement is needed as many online learners produce randomized predictions. The excess loss can be interpreted as a *warm-up period*. Throughout this section, we assume our learners satisfy OnlineWLC (δ, γ, S) with a fixed adversary.

Cost vectors The optimal design of a cost vector depends on the choice of loss. We will use $L^Y(\mathbf{s})$ to denote the loss without specifying it where $\mathbf{s}$ is the predicted score vector. The only constraint that we impose on our loss is that it is *proper*, which implies that it is decreasing in $\mathbf{s}[l]$ for $l \in Y$, and increasing in $\mathbf{s}[r]$ for $r \notin Y$ (readers should note that "proper loss" has at least one other meaning in the literature).

Then we introduce *potential function*, a well known concept in game theory which is first introduced to boosting by Schapire [2001]:

$$\begin{aligned} \phi_t^0(\mathbf{s}) &:= L^{Y_t}(\mathbf{s}) \\ \phi_t^i(\mathbf{s}) &:= \mathbb{E}_{l \sim \mathbf{u}_\gamma^{Y_t}} \phi_t^{i-1}(\mathbf{s} + \mathbf{e}_l). \end{aligned} \tag{3.1}$$

The potential $\phi_t^i(\mathbf{s})$ aims to estimate booster's final loss when i more weak learners are left until the final prediction and $\mathbf{s}$ is the current state. It can be easily shown by induction that many attributes of L are inherited by potentials. Being proper or convex are good examples.

Essentially, we want to set

$$\mathbf{c}_t^i[l] := \phi_t^{N-i}(\mathbf{s}_t^{i-1} + \mathbf{e}_l), \tag{3.2}$$

where $\mathbf{s}_t^{i-1}$ is the prediction of expert $i-1$. The proper property inherited by potentials ensures the relevant labels have less costs than the irrelevant. To satisfy the boundedness condition of $\mathcal{C}_0^{eor}$, we normalize (3.2) to get

$$\mathbf{d}_t^i[l] := \frac{\mathbf{c}_t^i[l] - \min_r \mathbf{c}_t^i[r]}{\mathbf{w}^i[t]}, \tag{3.3}$$

where $\mathbf{w}^i[t] := \max_r \mathbf{c}_t^i[r] - \min_r \mathbf{c}_t^i[r]$. Since Definition 3.1 assumes that $w_t \in [0,1]$, we have to further normalize $\mathbf{w}^i[t]$. This requires the knowledge of $w^{i*} := \max_t \mathbf{w}^i[t]$. This is unavailable until we observe all the instances, which is fine because we only need this value in proving the loss bound.

Algorithm details The algorithm is named by OnlineBMR (Online Boost-by-majority for Multi-label Ranking) as its potential function based design has roots in the classical *boost-by-majority* algorithm (Schapire [2001]). In OnlineBMR, we simply set $\alpha_t^i = 1$, or in other words, the booster takes simple cumulative votes. Cost vectors are computed using (3.2), and the booster always follows the last expert N, or $i_t = N$. These datails are summarized in Algorithm 3.2.

Algorithm 3.2 OnlineBMR details

1: **Initialize:** $\alpha_1^i = 1$ for $i \in [N]$
6: Set $i_t = N$
9: Set the weights $\alpha_{t+1}^i = 1,\ \forall i \in [N]$
10: Set $\mathbf{c}_t^i[l] = \phi_t^{N-i}(\mathbf{s}_t^{i-1} + \mathbf{e}_l),\ \forall l \in [k],\ \forall i \in [N]$
13: No extra parameters to be updated

The following theorem holds either if weak learners are single-label learners or if the loss L is convex.

Theorem 3.2. (BMR, general loss bound) *For any T and $N \ll \frac{1}{\delta}$, the final loss suffered by OnlineBMR satisfies the following inequality with probability $1 - N\delta$:*

$$\sum_{t=1}^{T} L^{Y_t}(\hat{\mathbf{y}}_t) \leq \sum_{t=1}^{T} \phi_t^N(\boldsymbol{0}) + S \sum_{i=1}^{N} w^{i*}. \tag{3.4}$$

Proof. From (3.1) and (3.2), we can write

$$\begin{aligned}\phi_t^{N-i+1}(\mathbf{s}_t^{i-1}) &= \mathbb{E}_{l\sim\mathbf{u}_\gamma^{Y_t}}\phi_t^{N-i}(\mathbf{s}_t^{i-1}+\mathbf{e}_l)\\ &= \mathbf{c}_t^i\cdot\mathbf{u}_\gamma^{Y_t}\\ &= \mathbf{c}_t^i\cdot(\mathbf{u}_\gamma^{Y_t}-\mathbf{h}_t^i)+\mathbf{c}_t^i\cdot\mathbf{h}_t^i\\ &\geq \mathbf{c}_t^i\cdot(\mathbf{u}_\gamma^{Y_t}-\mathbf{h}_t^i)+\phi_t^{N-i}(\mathbf{s}_t^i),\end{aligned}$$

where the last inequality is in fact equality if weak learners are single-label learners, or holds by Jensen's inequality if the loss is convex (which implies the convexity of potentials). Also note that $\mathbf{s}_t^i = \mathbf{s}_t^{i-1}+\mathbf{h}_t^i$. Since both $\mathbf{u}_\gamma^{Y_t}$ and $\mathbf{h}_t^i$ have ℓ_1 norm 1, we can subtract common numbers from every entry of $\mathbf{c}_t^i$ without changing the value of $\mathbf{c}_t^i\cdot(\mathbf{u}_\gamma^{Y_t}-\mathbf{h}_t^i)$. This implies we can plug in $\mathbf{w}^i[t]\mathbf{d}_t^i$ at the place of $\mathbf{c}_t^i$. Then we have

$$\begin{aligned}&\phi_t^{N-i+1}(\mathbf{s}_t^{i-1})-\phi_t^{N-i}(\mathbf{s}_t^i)\\ &\qquad\geq \mathbf{w}^i[t]\mathbf{d}_t^i\cdot\mathbf{u}_\gamma^{Y_t}-\mathbf{w}^i[t]\mathbf{d}_t^i\cdot\mathbf{h}_t^i.\end{aligned}$$

By summing this over t, we have

$$\begin{aligned}&\sum_{t=1}^T\phi_t^{N-i+1}(\mathbf{s}_t^{i-1})-\sum_{t=1}^T\phi_t^{N-i}(\mathbf{s}_t^i)\\ &\qquad\geq \sum_{t=1}^T\mathbf{w}^i[t]\mathbf{d}_t^i\cdot\mathbf{u}_\gamma^{Y_t}-\sum_{t=1}^T\mathbf{w}^i[t]\mathbf{d}_t^i\cdot\mathbf{h}_t^i.\end{aligned}\tag{3.5}$$

OnlineWLC (δ,γ,S) provides, with probability $1-\delta$,

$$\sum_{t=1}^T\frac{\mathbf{w}^i[t]}{w^{i*}}\mathbf{d}_t\cdot\mathbf{h}_t\leq\frac{1}{w^{i*}}\sum_{t=1}^T\mathbf{w}^i[t]\mathbf{d}_t\cdot\mathbf{u}_\gamma^{Y_t}+S.$$

Plugging this in (3.5), we get

$$\sum_{t=1}^T\phi_t^{N-i+1}(\mathbf{s}_t^{i-1})-\sum_{t=1}^T\phi_t^{N-i}(\mathbf{s}_t^i)\geq -Sw^{i*}.$$

Now summing this over i, we get with probability $1 - N\delta$ (due to union bound),

$$\sum_{t=1}^{T} \phi_t^N(\mathbf{0}) + S \sum_{i=1}^{N} w^{i*} \geq \sum_{t=1}^{T} \phi_t^0(\mathbf{s}_t^N) = \sum_{t=1}^{T} L^{Y_t}(\hat{\mathbf{y}}_t),$$

which completes the proof. □

Now we evaluate the efficiency of OnlineBMR by fixing a loss. Unfortunately, there is no canonical loss in MLR settings, but following *rank loss* is a strong candidate (cf. Cheng et al. [2010] and Gao and Zhou [2011]):

$$L_{\text{rnk}}^Y(\mathbf{s}) := w_Y \sum_{l \in Y} \sum_{r \notin Y} \mathbb{1}(\mathbf{s}[l] < \mathbf{s}[r]) + \frac{1}{2} \mathbb{1}(\mathbf{s}[l] = \mathbf{s}[r]),$$

where $w_Y = \frac{1}{|Y| \cdot |Y^c|}$ is a normalization constant that ensures the loss lies in $[0, 1]$. Note that this loss is not convex. In case weak learners are in fact single-label learners, we can simply use rank loss to compute potentials, but in more general case, we may use the following *hinge loss* to compute potentials:

$$L_{\text{hinge}}^Y(\mathbf{s}) := w_Y \sum_{l \in Y} \sum_{r \notin Y} (1 + \mathbf{s}[r] - \mathbf{s}[l])_+,$$

where $(\cdot)_+ := \max(\cdot, 0)$. It is convex and always greater than rank loss, and thus Theorem 3.2 can be used to bound rank loss. In Appendix B.1, we bound two terms in the RHS of (3.4) when the potentials are built upon rank and hinge losses. Here we record the results.

Table 3.1: Upper bounds for $\phi_t^N(\mathbf{0})$ and w^{i*}

loss	$\phi_t^N(\mathbf{0})$	w^{i*}
rank loss	$e^{-\frac{\gamma^2 N}{2}}$	$O(\frac{1}{\sqrt{N-i}})$
hinge loss	$(N+1)e^{-\frac{\gamma^2 N}{2}}$	2

For the case that we use rank loss, we can check

$$\sum_{i=1}^{N} w^{i*} \leq \sum_{i=1}^{N} O(\frac{1}{\sqrt{N-i}}) \leq O(\sqrt{N}).$$

Combining these results with Theorem 3.2, we get the following corollary.

Corollary 3.3. (BMR, rank loss bound) *For any T and $N \ll \frac{1}{\delta}$, OnlineBMR satisfies following rank loss bounds with probability $1 - N\delta$.*

With single-label learners, we have

$$\sum_{t=1}^{T} L_{rnk}^{Y_t}(\hat{\mathbf{y}}_t) \leq e^{-\frac{\gamma^2 N}{2}} T + O(\sqrt{N} S), \tag{3.6}$$

and with general learners, we have

$$\sum_{t=1}^{T} L_{rnk}^{Y_t}(\hat{\mathbf{y}}_t) \leq (N+1) e^{-\frac{\gamma^2 N}{2}} T + 2NS. \tag{3.7}$$

Remark. *When we divide both sides by T, we find the average loss is asymptotically bounded by the first term. The second term determines the sample complexity. In both cases, the first term decreases exponentially as N grows, which means the algorithm does not require too many learners to achieve a desired loss bound.*

Matching lower bounds From (3.6), we can deduce that to attain average loss less than ϵ, OnlineBMR needs $\Omega(\frac{1}{\gamma^2} \ln \frac{1}{\epsilon})$ learners and $\tilde{\Omega}(\frac{S}{\epsilon\gamma})$ samples. A natural question is whether these numbers are optimal. In fact the following theorem constructs a circumstance that matches these bounds up to logarithmic factors. Throughout the proof, we consider k as a fixed constant.

Theorem 3.4. *For any $\gamma \in (0, \frac{1}{2k})$, $\delta, \epsilon \in (0,1)$, and $S \geq \frac{k \ln(\frac{1}{\delta})}{\gamma}$, there exists an adversary with a family of learners satisfying OnlineWLC (δ, γ, S) such that to achieve error rate less than ϵ, any boosting algorithm requires at least $\Omega(\frac{1}{\gamma^2} \ln \frac{1}{\epsilon})$ learners and $\Omega(\frac{S}{\epsilon\gamma})$ samples.*

Proof. We introduce a sketch here and postpone the complete discussion to Appendix B.2. We assume that an adversary draws a label Y_t uniformly at random from $2^{[k]} - \{\emptyset, [k]\}$, and the weak learners generate single-label prediction l_t w.r.t. $\mathbf{p}_t \in \Delta[k]$. We manipulate $\mathbf{p}_t$ such that weak learners satisfy OnlineWLC (δ, γ, S) but the best possible performance is close to (3.6).

Boundedness conditions in $\mathcal{C}_0^{eor}$ and the Azuma-Hoeffding inequality provide that with probability $1 - \delta$,

$$\sum_{t=1}^{T} w_t \mathbf{c}_t[l_t] \leq \sum_{t=1}^{T} w_t \mathbf{c}_t \cdot \mathbf{p}_t + \frac{\gamma ||\mathbf{w}||_1}{k} + \frac{k \ln(\frac{1}{\delta})}{2\gamma}.$$

For the optimality of the number of learners, we let $\mathbf{p}_t = \mathbf{u}_{2\gamma}^{Y_t}$ for all t. The above inequality guarantees OnlineWLC is met. Then a similar argument of Schapire and Freund [2012, Section

13.2.6] can show that the optimal choice of weights over the learners is $(\frac{1}{N}, \cdots, \frac{1}{N})$. Finally, adopting the argument in the proof of Theorem 2.4, we can show

$$\mathbb{E}L^Y_{\text{rnk}}(\hat{\mathbf{y}}_t) \geq \Omega(e^{-4Nk^2\gamma^2}).$$

Setting this value equal to ϵ, we have $N \geq \Omega(\frac{1}{\gamma^2}\ln\frac{1}{\epsilon})$, considering k as a fixed constant. This proves the first part of the theorem.

For the second part, let $T_0 := \frac{S}{4\gamma}$ and define $\mathbf{p}_t = \mathbf{u}_0^{Y_t}$ for $t \leq T_0$ and $\mathbf{p}_t = \mathbf{u}_{2\gamma}^{Y_t}$ for $t > T_0$. Then OnlineWLC can be shown to be met in a similar fashion. Observing that weak learners do not provide meaningful information for $t \leq T_0$, we can claim any online boosting algorithm suffers a loss at least $\Omega(T_0)$. Therefore to obtain the certain accuracy ϵ, the number of instances T should be at least $\Omega(\frac{T_0}{\epsilon}) = \Omega(\frac{S}{\epsilon\gamma})$, which completes the second part of the proof. □

3.2.2 Adaptive algorithm

Despite the optimal loss bound, OnlineBMR has a few drawbacks when it is applied in practice. Firstly, potentials do not have a closed form, and their computation becomes a major bottleneck (cf. Table 3.3). Furthermore, the edge γ becomes an extra tuning parameter, which increases the runtime even more. Finally, it is possible that learners have different edges, and assuming a constant edge can lead to inefficiency. To overcome these drawbacks, rather than assuming positive edges for weak learners, our second algorithm chooses the weight α_t^i adaptively to handle variable edges.

Surrogate loss Like other adaptive boosting algorithms (e.g., Beygelzimer et al. [2015] and Freund et al. [1999]), our algorithm needs a surrogate loss. The choice of loss is broadly discussed in Chapter 2, and logistic loss seems to be a valid choice in online settings as its gradient is uniformly bounded. In this regard, we will use the following *logistic loss*:

$$L^Y_{\log}(\mathbf{s}) := w_Y \sum_{l\in Y}\sum_{r\notin Y} \log(1 + \exp(\mathbf{s}[r] - \mathbf{s}[l])).$$

It is proper and convex. We emphasize that booster's prediction suffers the rank loss, and this surrogate only plays an intermediate role in optimizing parameters.

Algorithm details The algorithm is inspired by Adaboost.OLM (Algorithm 2.2), and we call it by Ada.OLMR[2]. Since it internally aims to minimize the logistic loss, we set the cost vector to be

[2]Online, Logistic, Multi-label, and Ranking

the gradient of the surrogate:

$$\mathbf{c}_t^i := \nabla L_{\log}^{Y_t}(\mathbf{s}_t^{i-1}). \tag{3.8}$$

Next we present how to set the weights α_t^i. Essentially, Ada.OLMR wants to choose α_t^i to minimize the cumulative logistic loss:

$$\sum_t L_{\log}^{Y_t}(\mathbf{s}_t^{i-1} + \alpha_t^i \mathbf{h}_t^i).$$

After initializing α_1^i equals to 0, we use *online gradient descent* method, proposed by Zinkevich [2003], to compute the next weights. If we write $f_t^i(\alpha) := L_{\log}^{Y_t}(\mathbf{s}_t^{i-1} + \alpha \mathbf{h}_t^i)$, we want α_t^i to satisfy

$$\sum_t f_t^i(\alpha_t^i) \le \min_{\alpha \in F} \sum_t f_t^i(\alpha) + R^i(T),$$

where F is some *feasible set*, and $R^i(T)$ is a sublinear regret. To apply the result by Zinkevich [2003, Theorem 1], f_t^i needs to be convex, and F should be compact. The former condition is met by our choice of logistic loss, and we will use $F = [-2, 2]$ for the feasible set. Since the booster's loss is invariant under the scaling of weights, we can shrink the weights to fit in F.

Taking derivative, we can check $f_t^{i\prime}(\alpha) \le 1$. Now let $\Pi(\cdot)$ denote a projection onto F: $\Pi(\cdot) := \max\{-2, \min\{2, \cdot\}\}$. By setting

$$\alpha_{t+1}^i = \Pi(\alpha_t^i - \eta_t f_t^{i\prime}(\alpha_t^i)) \text{ where } \eta_t = \frac{1}{\sqrt{t}},$$

we get $R^i(T) \le 9\sqrt{T}$. Considering that $\mathbf{s}_t^i = \mathbf{s}_t^{i-1} + \alpha_t^i \mathbf{h}_t^i$, we can also write $f_t^{i\prime}(\alpha_t^i) = \mathbf{c}_t^{i+1} \cdot \mathbf{h}_t^i$.

Finally, it remains to address how to choose i_t. In contrast to OnlineBMR, we cannot show that the last expert is reliably sophisticated. Instead, what can be shown is that at least one of the experts is good enough. Thus we use classical *Hedge algorithm* (cf. Freund and Schapire [1997] and Littlestone and Warmuth [1989]) to randomly choose an expert at each iteration with adaptive probability distribution depending on each expert's prediction history. In particular, we introduce new variables v_t^i, which are initialized as $v_1^i = 1,\ \forall i$. At each iteration, i_t is randomly drawn such that

$$\mathbb{P}(i_t = i) \propto v_t^i,$$

and then v_t^i is updated based on the expert's rank loss:

$$v_{t+1}^i := v_t^i e^{-L_{\text{rnk}}^{Y_t}(\mathbf{s}_t^i)}.$$

The details are summarized in Algorithm 3.3.

Algorithm 3.3 Ada.OLMR details

1: **Initialize:** $\alpha_1^i = 0$ and $v_1^i = 1,\ \forall i \in [N]$
6: Randomly draw i_t s.t. $\mathbb{P}(i_t = i) \propto v_t^i$
9: Compute $\alpha_{t+1}^i = \Pi(\alpha_t^i - \frac{1}{\sqrt{t}} f_t^{i\prime}(\alpha_t^i)),\ \forall i \in [N]$
10: Compute $\mathbf{c}_t^i = \nabla L_{\log}^{Y_t}(\mathbf{s}_t^{i-1}),\ \forall i \in [N]$
13: Update $v_{t+1}^i = v_t^i e^{-L_{\text{rnk}}^{Y_t}(\mathbf{s}_t^i)},\ \forall i \in [N]$

Empirical edges As we are not imposing OnlineWLC, we need another measure of the learner's predictive power to prove the loss bound. From (3.8), it can be observed that the relevant labels have negative costs and the irrelevant ones have positive cost. Furthermore, the summation of entries of $\mathbf{c}_t^i$ is exactly 0. This observation suggests a new definition of weight:

$$\begin{aligned}\mathbf{w}^i[t] &:= w_{Y_t} \sum_{l \in Y_t} \sum_{r \notin Y_t} \frac{1}{1 + \exp(\mathbf{s}_t^{i-1}[l] - \mathbf{s}_t^{i-1}[r])} \\ &= -\sum_{l \in Y_t} \mathbf{c}_t^i[l] = \sum_{r \notin Y_t} \mathbf{c}_t^i[r] = \frac{||\mathbf{c}_t^i||_1}{2}.\end{aligned} \tag{3.9}$$

This does not directly correspond to the weight used in (3.3), but plays a similar role. Then we define the *empirical edge*:

$$\gamma_i := -\frac{\sum_{t=1}^T \mathbf{c}_t^i \cdot \mathbf{h}_t^i}{||\mathbf{w}^i||_1}. \tag{3.10}$$

The baseline learner $\mathbf{u}_\gamma^{Y_t}$ has this value exactly γ, which suggests that it is a good proxy for the edge defined in Definition 3.1.

Now we present the loss bound of Ada.OLMR.

Theorem 3.5. (Ada.OLMR, rank loss bound) *For any T and N, with probability $1 - \delta$, the rank loss suffered by Ada.OLMR is bounded as follows:*

$$\sum_{t=1}^T L_{rnk}^{Y_t}(\hat{\mathbf{y}}_t) \leq \frac{8}{\sum_i |\gamma_i|} T + \tilde{O}(\frac{N^2}{\sum_i |\gamma_i|}), \tag{3.11}$$

where $\tilde{O}$ notation suppresses dependence on $\log \frac{1}{\delta}$.

Proof. We start the proof by defining the rank loss suffered by expert i as below:

$$M_i := \sum_{t=1}^T L_{\text{rnk}}^{Y_t}(\mathbf{s}_t^i).$$

According to the formula, there is no harm to define $M_0 = \frac{T}{2}$ since $\mathbf{s}_t^0 = \mathbf{0}$. As the booster chooses an expert through the Hedge algorithm, a standard analysis (cf. [Cesa-Bianchi and Lugosi, 2006, Corollary 2.3]) along with the Azuma-Hoeffding inequality provides with probability $1 - \delta$,

$$\sum_{t=1}^{T} L_{\text{rnk}}^{Y_t}(\hat{\mathbf{y}}_t) \leq 2 \min_i M_i + 2 \log N + \tilde{O}(\sqrt{T}), \tag{3.12}$$

where $\tilde{O}$ notation suppresses dependence on $\log \frac{1}{\delta}$.

It is not hard to check that $\frac{1}{1+\exp(a-b)} \geq \frac{1}{2}\mathbb{1}(a \leq b)$, from which we can infer

$$\mathbf{w}^i[t] \geq \frac{1}{2} L_{\text{rnk}}^{Y_t}(\mathbf{s}_t^{i-1}) \text{ and } ||\mathbf{w}^i||_1 \geq \frac{M_{i-1}}{2}, \tag{3.13}$$

where $\mathbf{w}^i$ is defined in (3.9). Note that this relation holds for the case $i = 1$ as well.

Now let Δ_i denote the difference of the cumulative logistic loss between two consecutive experts:

$$\begin{aligned} \Delta_i &:= \sum_{t=1}^{T} L_{\log}^{Y_t}(\mathbf{s}_t^i) - L_{\log}^{Y_t}(\mathbf{s}_t^{i-1}) \\ &= \sum_{t=1}^{T} L_{\log}^{Y_t}(\mathbf{s}_t^{i-1} + \alpha_t^i \mathbf{h}_t^i) - L_{\log}^{Y_t}(\mathbf{s}_t^{i-1}). \end{aligned}$$

Then the online gradient descent algorithm provides

$$\begin{aligned} \Delta_i \leq \min_{\alpha \in [-2,2]} \sum_{t=1}^{T} [L_{\log}^{Y_t}(\mathbf{s}_t^{i-1} + \alpha \mathbf{h}_t^i) - L_{\log}^{Y_t}(\mathbf{s}_t^{i-1})] \\ + 9\sqrt{T}. \end{aligned} \tag{3.14}$$

Here we record an univariate inequality:

$$\begin{aligned} \log(1 + e^{s+\alpha}) - \log(1 + e^s) &= \log(1 + \frac{e^\alpha - 1}{1 + e^{-s}}) \\ &\leq \frac{1}{1 + e^{-s}}(e^\alpha - 1). \end{aligned}$$

We expand the difference to get

$$\begin{aligned}
&\sum_{t=1}^{T}[L_{\log}^{Y_t}(\mathbf{s}_t^{i-1}+\alpha\mathbf{h}_t^i)-L_{\log}^{Y_t}(\mathbf{s}_t^{i-1})]\\
&=\sum_{t=1}^{T}\sum_{l\in Y_t}\sum_{r\notin Y_t}\log\frac{1+e^{\mathbf{s}_t^{i-1}[r]-\mathbf{s}_t^{i-1}[l]+\alpha(\mathbf{h}_t^i[r]-\mathbf{h}_t^i[l])}}{1+e^{\mathbf{s}_t^{i-1}[r]-\mathbf{s}_t^{i-1}[l]}}\\
&\leq\sum_{t=1}^{T}\sum_{l\in Y_t}\sum_{r\notin Y_t}\frac{1}{1+e^{\mathbf{s}_t^{i-1}[l]-\mathbf{s}_t^{i-1}[r]}}(e^{\alpha(\mathbf{h}_t^i[r]-\mathbf{h}_t^i[l])}-1)\\
&=:f(\alpha).
\end{aligned}\tag{3.15}$$

We claim that $\min_{\alpha\in[-2,2]}f(\alpha)\leq-\frac{|\gamma_i|}{2}||\mathbf{w}^i||_1$. Let us rewrite $||\mathbf{w}^i||_1$ in (3.9) and γ_i in (3.10) as following.

$$\begin{aligned}
||\mathbf{w}^i||_1&=\sum_{t=1}^{T}\sum_{l\in Y_t}\sum_{r\notin Y_t}\frac{1}{1+e^{\mathbf{s}_t^{i-1}[l]-\mathbf{s}_t^{i-1}[r]}}\\
\gamma_i&=\sum_{t=1}^{T}\sum_{l\in Y_t}\sum_{r\notin Y_t}\frac{1}{||\mathbf{w}^i||_1}\frac{\mathbf{h}_t^i[l]-\mathbf{h}_t^i[r]}{1+e^{\mathbf{s}_t^{i-1}[l]-\mathbf{s}_t^{i-1}[r]}}.
\end{aligned}\tag{3.16}$$

For the ease of notation, let j denote an index that moves through all tuples of $(t,l,r)\in[T]\times Y_t\times Y_t^c$, and a_j and b_j denote following terms.

$$\begin{aligned}
a_j&=\frac{1}{||\mathbf{w}^i||_1}\frac{1}{1+e^{\mathbf{s}_t^{i-1}[l]-\mathbf{s}_t^{i-1}[r]}}\\
b_j&=\mathbf{h}_t^i[l]-\mathbf{h}_t^i[r].
\end{aligned}$$

Then from (3.16), we have $\sum_j a_j=1$ and $\sum_j a_jb_j=\gamma_i$. Now we express $f(\alpha)$ in terms of a_j and b_j as below.

$$\frac{f(\alpha)}{||\mathbf{w}^i||_1}=\sum_j a_j(e^{-\alpha b_j}-1)\leq e^{-\alpha\sum_j a_jb_j}-1=e^{-\alpha\gamma_i}-1,$$

where the inequality holds by Jensen's inequality. From this, we can deduce that

$$\min_{\alpha\in[-2,2]}\frac{f(\alpha)}{||\mathbf{w}^i||_1}\leq e^{-2|\gamma_i|}-1\leq-\frac{|\gamma_i|}{2},$$

where the last inequality can be checked by investigating $|\gamma_i|=0,1$ and observing the convexity of

the exponential function. This proves our claim that

$$\min_{\alpha\in[-2,2]} f(\alpha) \leq -\frac{|\gamma_i|}{2}||\mathbf{w}^i||_1. \tag{3.17}$$

Combining (3.13), (3.14), (3.15) and (3.17), we have

$$\Delta_i \leq -\frac{|\gamma_i|}{4}M_{i-1} + 9\sqrt{T}.$$

Summing over i, we get by telescoping rule

$$\begin{aligned}
&\sum_{t=1}^{T} L_{\log}^{Y_t}(\mathbf{s}_t^N) - \sum_{t=1}^{T} L_{\log}^{Y_t}(\mathbf{0}) \\
&\qquad \leq -\frac{1}{4}\sum_{i=1}^{N} |\gamma_i| M_{i-1} + 9N\sqrt{T} \\
&\qquad \leq -\frac{1}{4}\sum_{i=1}^{N} |\gamma_i| \min_i M_i + 9N\sqrt{T}.
\end{aligned}$$

Note that $L_{\log}^{Y_t}(\mathbf{0}) = \log 2$ and $L_{\log}^{Y_t}(\mathbf{s}_t^N) \geq 0$. Therefore we have

$$\min_i M_i \leq \frac{4\log 2}{\sum_i |\gamma_i|}T + \frac{36N\sqrt{T}}{\sum_i |\gamma_i|}.$$

Plugging this in (3.12), we get with probability $1-\delta$,

$$\begin{aligned}
\sum_{t=1}^{T} L_{\text{rnk}}^{Y_t}(\hat{\mathbf{y}}_t) &\leq \frac{8\log 2}{\sum_i |\gamma_i|}T + \tilde{O}\left(\frac{N\sqrt{T}}{\sum_i |\gamma_i|} + \log N\right) \\
&\leq \frac{8}{\sum_i |\gamma_i|}T + \tilde{O}\left(\frac{N^2}{\sum_i |\gamma_i|}\right),
\end{aligned}$$

where the last inequality holds from AM-GM inequality: $cN\sqrt{T} \leq \frac{c^2N^2+T}{2}$. This completes our proof. □

Comparison with OnlineBMR We finish this section by comparing our two algorithms. For a fair comparison, assume that all learners have edge γ. Since the baseline learner $\mathbf{u}_\gamma^Y$ has empirical edge γ, for sufficiently large T, we can deduce that $\gamma_i \geq \gamma$ with high probability. Using this relation,

(3.11) can be written as

$$\sum_{t=1}^{T} L_{\text{rnk}}^{Y_t}(\hat{\mathbf{y}}_t) \leq \frac{8}{N\gamma} T + \tilde{O}(\frac{N}{\gamma}).$$

Comparing this to either (3.6) or (3.7), we can see that OnlineBMR indeed has better asymptotic loss bound and sample complexity. Despite this sub-optimality (in upper bounds), Ada.OLMR shows comparable results in real data sets due to its adaptive nature.

3.3 Experiments

We performed an experiment on benchmark data sets taken from MULAN[3]. We chose these four particular data sets because Dembczynski and Hüllermeier [2012] already provided performances of batch setting boosting algorithms, giving us a benchmark to compare with. The authors in fact used five data sets, but *image* data set is no longer available from the source. Table 3.2 summarizes the basic statistics of data sets, including training and test set sizes, number of features and labels, and three statistics of the sizes of relevant sets. The data set *m-reduced* is a reduced version of *mediamill* obtained by random sampling without replacement. We keep the original split for training and test sets to provide more relevant comparisons.

Table 3.2: Summary of data sets

data	#train	#test	dim	k	min	mean	max
emotions	391	202	72	6	1	1.87	3
scene	1211	1196	294	6	1	1.07	3
yeast	1500	917	103	14	1	4.24	11
mediamill	30993	12914	120	101	0	4.38	18
m-reduced	1500	500	120	101	0	4.39	13

VFDT algorithms presented by Domingos and Hulten [2000] were used as weak learners. Every algorithm used 100 trees whose parameters were randomly chosen. VFDT is trained using single-label data, and we fed individual relevant labels along with importance weights that were computed as $\max_l \mathbf{c}_t - \mathbf{c}_t[l]$. Instead of using all covariates, the booster fed to trees randomly chosen 20 covariates to make weak predictions less correlated.

All computations were carried out on a Nehalem architecture 10-core 2.27 GHz Intel Xeon

[3] Tsoumakas et al. [2011], http://mulan.sourceforge.net/datasets.html

E7-4860 processors with 25 GB RAM per core. Each algorithm was trained at least ten times[4] with different random seeds, and the results were aggregated through mean. Predictions were evaluated by rank loss. The algorithm's loss was only recorded for test sets, but it kept updating its parameters while exploring test sets as well.

Since VFDT outputs a conditional distribution, which is not of a single-label format, we used hinge loss to compute potentials. Furthermore, OnlineBMR has an additional parameter of edge γ. We tried four different values[5], and the best result is recorded as *best BMR*. Table 3.3 summarizes the results.

Table 3.3: Average loss and runtime in seconds

data	batch[6]	Ada.OLMR		best BMR	
emotions	.1699	.1600	253	.1654	611
scene	.0720	.0881	341	.0743	1488
yeast	.1820	.1874	2675	.1836	9170
mediamill	.0665	.0508	69565	-	-
m-reduced	-	.0632	4148	.0630	288204

Two algorithms' average losses are comparable to each other and to batch setting results, but OnlineBMR requires much longer runtimes. Based on the fact that best BMR's performance is reported on the best edge parameter out of four trials, Ada.OLMR is far more favorable in practice. With large number of labels, runtime for OnlineBMR grows rapidly, and it was even impossible to run *mediamill* data within a week, and this was why we produced the reduced version. The main bottleneck is the computation of potentials as they do not have closed form.

[4]OnlineBMR for *m-reduced* was tested 10 times due to long runtimes, and others were tested 20 times
[5]$\{.2, .1, .01, .001\}$ for small k and $\{.05, .01, .005, .001\}$ for large k
[6]The best result from batch boosting algorithms in Dembczynski and Hüllermeier [2012]

CHAPTER 4

Online Boosting with Partial Information

Chapter 2 and 3 discuss online boosting algorithms in the *full information* setting, where the environment reveals the true label once prediction is made. However, when the number of labels becomes too large or the label itself involves a complex combinatorial structure, obtaining the true answer can be costly. For example, when the labels are ads or product recommendations on the web, the learner only receives feedback about whether its predicted label was correct (e.g., the user clicked on the ad or recommendation) or not (e.g., user did not click). Intuitively, training machine learning models under such partial feedback is challenging. A common approach is to convert a full information algorithm into a partial information version without incurring too much performance loss (see, for example, Kakade et al. [2008] and Beygelzimer et al. [2017] for work using the perceptron algorithm). This chapter will briefly discuss online boosting algorithms in the partial feedback settings [1]. The first part deals with the multi-class classification with bandit feedback and the second part discusses multi-label ranking with top-k feedback.

Designing a boosting algorithm with bandit feedback is particularly difficult as it is not clear how to update the weak learners. For example, suppose that a weak learner WL^1 predicts the label 1, another learner WL^2 predicts the label 2, and the boosting algorithm predicts the label 1, which turns out to be incorrect. We cannot even tell WL^2 whether its prediction is correct. Furthermore, top-k feedback is *not* even bandit feedback. Unlike the bandit multiclass setting, the learner does not even get to compute its own loss! Thus, a key challenge in this setting is to use the structure of the loss to design estimators that can produce unbiased estimates of the loss from only top-k feedback. This intricate interplay between loss functions and partial feedback does not occur in previous work on online boosting.

[1] This chapter is based on joint work with Daniel Zhang, who was an undergraduate student in the University of Michigan and is currently a software engineer at Facebook. The multi-class classification work appeared in AISTATS 2019 under the title "Online Multiclass Boosting with Bandit Feedback," and the multi-label ranking work is available as an arXiv preprint https://arxiv.org/abs/1910.10937.

In both settings, the key idea is to let the learner randomize its prediction and then estimate the loss using this randomness. In this way, one can compute an unbiased estimate of the loss, from which the booster can compute cost vectors and update weak learners. Quite surprisingly, partial information algorithms match their full information counterparts with respect to their asymptotic performance guarantees. The cost of partial feedback is only reflected to the increased sample complexities. That is to say, the partial information algorithms require more data instances to achieve the same accuracy with the full information algorithms. This can also be verified in the experiments.

4.1 Multi-class Classification with Bandit Feedback

The notation in this section adopts that in Chapter 2. The setting also resembles Chapter 2 except that the environment only tells the learner whether its prediction is correct or not.

4.1.1 Unbiased Estimate of the Zero-One Loss

It is naturally expected that the booster needs to estimate the final zero-one loss vector:

$$l_t^{0-1} = \mathbf{1} - \mathbf{e}_{y_t} \in \mathbb{R}^k. \tag{4.1}$$

As we are in the bandit setting, the booster only has limited information about this vector. In particular, unless its final prediction is correct, only a single entry of l_t^{0-1} is available.

A popular approach for algorithm design in the partial information setting is to obtain an unbiased estimate of the loss. To do so, many bandit algorithms randomize their prediction. In our setting, instead of making a deterministic prediction $\hat{y}$, the algorithm designs a sampling distribution $p_t \in \Delta_k$ as follows:

$$p_{t,i} = \begin{cases} 1 - \rho & \text{if } i = \hat{y}_t \\ \frac{\rho}{k-1} & \text{if } i \neq \hat{y}_t \end{cases}, \tag{4.2}$$

where ρ is a parameter that controls the exploration rate. This distribution puts a large weight on the label $\hat{y}_t$ and evenly distributes the remaining weight over the rest. The algorithm draws a final prediction $\tilde{y}_t$ based on p_t. In this way, the algorithm can build an estimator using the known

sampling distribution. A simplest unbiased estimate of the zero-one loss is

$$\hat{l}_t^{0-1} = \frac{\mathbb{1}(\tilde{y}_t = y_t)}{p_{t,\tilde{y}_t}}(\mathbf{1} - \mathbf{e}_{\tilde{y}_t}) \in \mathbb{R}^k. \tag{4.3}$$

It is easy to check that this is indeed unbiased. However, it is not necessarily the best because it becomes a zero vector when the booster makes a mistake. As the zero loss vector does not provide any useful information, the weak learners cannot update at this round. Therefore, it would be hard for the booster to escape the early training stage using the simple estimate.

As an alternative, we propose a new estimator

$$\hat{l}_{t,i}^{0-1} = \frac{\mathbb{1}(\tilde{y}_t = y_t)}{p_{t,\tilde{y}_t}}\mathbb{1}(y_t \neq i)\mathbb{1}(\hat{y}_t \neq i) + \frac{\mathbb{1}(\tilde{y}_t = \hat{y}_t)}{p_{t,\tilde{y}_t}}\mathbb{1}(\hat{y}_t \neq y_t)\mathbb{1}(\hat{y}_t = i). \tag{4.4}$$

We first emphasize that this quantity can be computed only using the bandit feedback. The following lemma shows that it is actually unbiased.

Lemma 4.1. *The estimator $\hat{l}_t^{0-1}$ in* (4.4) *is an unbiased estimator of the zero-one loss l_t^{0-1}:*

$$\mathbb{E}_{\tilde{y}_t \sim p_t} \hat{l}_t^{0-1} = l_t^{0-1}.$$

Proof. Since $\tilde{y}$ is drawn with respect to p_t, we can write

$$\begin{aligned}
\mathbb{E}_{\tilde{y}_t \sim p_t} \hat{l}_{t,i}^{0-1} &= \mathbb{1}(y_t \neq i)\mathbb{1}(\hat{y}_t \neq i) + \mathbb{1}(\hat{y}_t \neq y_t)\mathbb{1}(\hat{y}_t = i) \\
&= \mathbb{1}(y_t \neq i)\mathbb{1}(\hat{y}_t \neq i) + \mathbb{1}(i \neq y_t)\mathbb{1}(\hat{y}_t = i) \\
&= \mathbb{1}(y_t \neq i)(\mathbb{1}(\hat{y}_t \neq i) + \mathbb{1}(\hat{y}_t = i)) \\
&= \mathbb{1}(y_t \neq i),
\end{aligned}$$

where the last term is $l_{t,i}^{0-1}$, which completes the proof. □

This estimator resolves the main issue with the estimator in (4.3), viz. that the learner cannot update during a mistake round. In fact, it allows the weak learner to update on each instance with probability at least $1 - \rho$. Furthermore, the algorithms using this estimator empirically performed much better than ones using the estimator in (4.3). For these reasons, we will stick to the estimate in (4.4) from now on.

To apply concentration inequalities, we need to control the variance of estimators. We say a random vector Y is b-*bounded* if $||Y - \mathbb{E}Y||_\infty \leq b$ almost surely. Note that this definition

also applies to random variables (i.e., scalars), in which case the norm above simply becomes the absolute value. It is easy to check our estimator $\hat{l}_t^{0-1}$ is $\frac{k}{\rho}$-bounded.

Now suppose that a cost vector $c_t^i \in \mathbb{R}^k$ (to be fed into weak learner i at time t) requires the knowledge of the true label y_t. Since the label is usually unavailable, we also need to estimate the cost vector. We first compute a matrix $C_t^i \in \mathbb{R}^{k\times k}$, whose j^{th} column is the cost vector c_t^i assuming j is the correct label. Then we will use the following random cost vector:

$$\hat{c}_t^i = C_t^i \cdot (\mathbf{1} - \hat{l}_t^{0-1}). \tag{4.5}$$

Since C_t^i is a deterministic matrix, we can compute

$$\mathbb{E}_{\tilde{y}_t}\hat{c}_t^i = C_t^i \cdot (\mathbf{1} - l_t^{0-1}) = C_t^i \cdot \mathbf{e}_{y_t},$$

which is the y_t^{th} column of C_t^i. This shows that $\hat{c}_t^i$ is an unbiased estimate of c_t^i.

4.1.2 Algorithms

The proposed algorithms are essentially the same as the full information ones in Chapter 2. The only difference is that the bandit algorithms generate a random prediction $\tilde{y}$ based on its original prediction $\hat{y}$ and the exploration rate ρ. The computation of cost vectors remains same except that the bandit versions incorporate the unbiased estimate discussed in Section 4.1.1. As in the full information setting, we propose one optimal algorithm, BanditBBM, which extends OnlineMBBM, and one adaptive algorithm, AdaBandit, which extends Adaboost.OLM. The details of the algorithms are omitted in this manuscript but can be found in the complete paper.

4.1.3 Mistake Bounds

We investigate the mistake bounds of the bandit boosting algorithms and compare them with the full information algorithms. The proofs of the theorems can be found in the complete paper. We begin from BanditBBM.

Theorem 4.2 (Mistake Bound of BanditBBM)**.** *For any T, N satisfying $\delta \ll \frac{1}{N}$, the number of mistakes made by BanditBBM satisfies the following inequality with probability at least $1-(N+1)\delta$:*

$$\sum_{t=1}^{T} \mathbb{1}(\tilde{y}_t \neq y_t) \leq (k-1)e^{-\frac{\gamma^2 N}{2}}T + 2\rho T + \tilde{O}(\frac{k^{7/2}\sqrt{N}}{\rho}),$$

where $\tilde{O}$ suppresses dependence on $\log \frac{1}{\delta}$.

If we set the exploration rate $\rho = \frac{k^{7/4}N^{1/4}}{\sqrt{T}}$, then the bound becomes

$$(k-1)e^{-\frac{\gamma^2 N}{2}}T + \tilde{O}(k^{7/4}N^{1/4}\sqrt{T}).$$

Dividing by T, we can infer that $(k-1)e^{-\frac{\gamma^2 N}{2}}$ is the asymptotic error bound of the algorithm. This bound matches the bound of the full information counterpart, OnlineMBBM. Since it depends exponentially on N, BanditBBM does not require too many weak learners to obtain a desired accuracy. Theorem 2.4 also provide a lower bound in the full information setting, which shows that the exponential decay is the fastest rate one can expect for the asymptotic error bound. This result applies to the bandit setting as it is harder. It is worth noting that BanditBBM has larger sample complexity than OnlineMBBM, which results from less information provided to the learner.

Now we move on to the adaptive algorithm. We emphasize that the empirical edges in the next theorem are defined exactly in the same manner with those used in the full information bound.

Theorem 4.3 (Mistake Bound of AdaBandit)**.** *For any T, N satisfying $\delta \ll \frac{1}{N}$, the number of mistakes made by AdaBandit satisfies the following inequality with probability at least $1-(N+4)\delta$:*

$$\sum_{t=1}^{T} \mathbb{1}(\tilde{y}_t \neq y_t) \leq \frac{8k}{\sum_{i=1}^{N}\gamma_i^2}T + 2\rho T + \tilde{O}(\frac{k^3N^2}{\rho^2\sum_{i=1}^{N}\gamma_i^2}),$$

where $\tilde{O}$ suppresses dependence on $\log \frac{1}{\delta}$.

If we set the exploration rate $\rho = \frac{kN^{2/3}}{(T\sum_{i=1}^{N}\gamma_i^2)^{1/3}}$, then the bound becomes

$$\frac{8k}{\sum_{i=1}^{N}\gamma_i^2}T + \tilde{O}(\frac{kN^{\frac{2}{3}}}{(\sum_{i=1}^{N}\gamma_i^2)^{\frac{1}{3}}}T^{\frac{2}{3}}).$$

This implies that $\frac{8k}{\sum_{i=1}^{N}\gamma_i^2}$ becomes the asymptotic error bound of AdaBandit, which matches the bound of Adaboost.OLM. The difficulty of bandit feedback is again reflected to the increased sample complexity.

4.2 Multi-label Ranking with Top-k Feedback

This section discusses online boosting algorithms in the multi-label ranking problems with top-k feedback. The notation adopts that in Chapter 3. The setting also resembles Chapter 3 except that

the environment only tells the learner whether the top-k labels predicted by the learner are relevant or not. The readers should note that the learner can no longer infer the exact value of the loss it incurs with this feedback, which makes this setting much harder than the bandit setting.

4.2.1 Estimating a Loss Function

Because of top-k feedback, we require methods to estimate loss functions dependent on labels outside of the top-k labels from our score vector y_t. One common way of dealing with partial feedback is to introduce randomized predictions and construct an unbiased estimator of the loss using the known distribution of the prediction. This way, we can obtain a randomized loss function for our learner to use. Thus, we propose a novel unbiased estimator to randomize arbitrary y_t. This estimator requires some structure within the loss function it is estimating.

We require that loss to be writable as a sum of functions which only require as input the scores and relevance of two particular labels, each containing one relevant and irrelevant label. In particular, our loss must have the form

$$L(s, R) = \sum_{a \in R} \sum_{b \notin R} f(s[a], s[b]) =: \sum_{a,b \in [m]} f^{a,b}(s),$$
$$\text{where } f^{a,b}(s) = \mathbb{1}(a \in R)\mathbb{1}(b \notin R) f(s[a], s[b]).$$

Here s is an arbitrary score vector in $\mathbb{R}^m$, and f is a given function. We call this property *pairwise decomposability*. This decomposability allows us to individually estimate each $f^{a,b}$ and thus L.

In fact, various valid MLR loss functions are pairwise decomposable. An example is the *unweighted rank loss*

$$L^{\text{rnk}}(s, R_t) = \sum_{a \in R_t} \sum_{b \notin R_t} \mathbb{1}(s[a] \leq s[b]),$$

which has various surrogates, including the following *unweighted hinge rank loss*

$$L_{\text{hinge}}(s, R_t) = \sum_{a \in R_t} \sum_{b \notin R_t} \max\{0, s[b] - s[a] + 1\}.$$

It should be noted that the *weighted rank loss*

$$L^{\text{wrnk}}(s, R_t) = \frac{1}{|R_t|(m - |R_t|)} L^{\text{rnk}}(s, R_t)$$

cannot be computed using this strategy because its normalization weight is non-linearly dependent on $|R_t|$. In such cases, it is possible to upper bound the target loss function with a surrogate loss that is pairwise decomposable. For example, the unweighted rank loss is an obvious upper bound of the weighted one.

Returning to our estimator, we first elaborate on a method of randomized prediction given $r_t = \sigma(y_t)$ that will allow us to construct our unbiased estimator. This randomized prediction $\tilde{r}_t$ is paramaterized by the exploration rate $\rho \in (0, 1)$. After computing r_t, with probability $1 - \rho$ we use r_t as our final ranking. Otherwise, with probability ρ, we choose[2] two elements, denoted by A, from $\mathcal{T}^k(r_t)$ and two elements, denoted by B, from $\mathcal{T}^k(r_t)^c$, the set of labels which have rank lower than k. Then, we take the higher ranked labels from A and B and swap them, and do the same for the lower ranked labels, producing our final ranking. This process is more complicated than simply using a random ranking with probability ρ, but with our method, $\tilde{r}_t$ stays closer to r_t, which would be favorable provided r_t has a small loss. Figure 4.1 presents an example of this exploration step. In case the loss is a function of score vector instead of ranking, we can get a random score $\tilde{s}_t$ out of s_t in a similar manner.

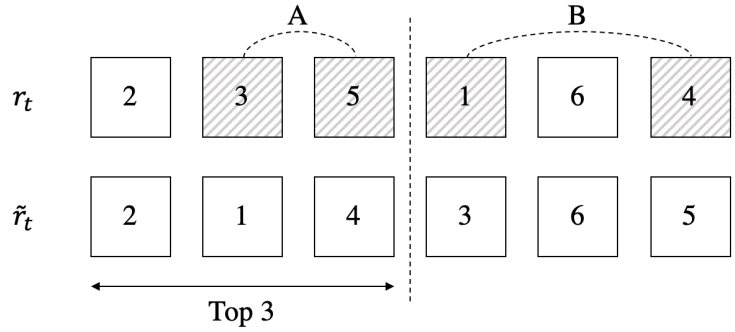

Figure 4.1: An example of the exploration step when $m = 6$, $k = 3$, and $r_t = (2, 3, 5, 1, 6, 4)$

We now present our unbiased estimator. Let $\tilde{r}_t$ be the random ranking from the previously described process, and let s be an arbitrary score vector in $\mathbb{R}^m$. We note that given any two distinct labels a and b, $\mathbb{P}[a, b \in \mathcal{T}^k(\tilde{r}_t)] > 0$. Since being in the top-k provides the learner with full information regarding the relevance and scores of the labels, we have this unbiased estimator using importance sampling

$$\hat{L}(s, R_t) = \sum_{a,b \in [m]} \frac{\mathbb{1}(a, b \in \mathcal{T}^k(\tilde{r}_t))}{\mathbb{P}[a, b \in \mathcal{T}^k(\tilde{r}_t)]} f^{a,b}(s). \tag{4.6}$$

It is not hard to check that this is an unbiased estimator. Our algorithms will use this unbiased

[2]It is this part of our construction that requires $k \geq 3$ and $m \geq 5$.

estimator to estimate certain surrogate functions which we construct to be pairwise decomposable.

Now suppose that the cost vector c_t^i (to be fed to the ith weak learner at time t) requires full knowledge of R_t to compute. If each of its entries is a function that is pairwise decomposable, we can use the same unbiased estimation strategy to obtain random cost vectors $\hat{c}_t^i$ that are in expectation equal to c_t^i.

4.2.2 Algorithms

The proposed algorithms are essentially the same as the full information ones in Chapter 3. The only difference is that the top-k algorithms generate a random prediction $\tilde{r}$ based on its original score vector and the exploration rate ρ. The computation of cost vectors remains same except that the top-k versions incorporate the unbiased estimate discussed in Section 4.2.1. As in the full information setting, we propose one optimal algorithm, TopkBBM, which extends OnlineBMR, and one adaptive algorithm, TopkAdaptive, which extends Ada.OLMR. The details of the algorithms are omitted in this manuscript but can be found in the complete preprint.

4.2.3 Loss Bounds

We can theoretically guarantee the performance of TopkBBM on any proper and pairwise decomposable loss function. However, this result only bounds the loss of the score vector, not the actual randomized prediction. Fortunately, we can bound the loss of the randomized prediction in the case of the rank loss. The omitted proofs in this section can be found in the complete preprint.

Theorem 4.4 (TopBBM, Rank Loss Bound)**.** *For any T and $N \ll \frac{1}{\delta}$, TopkBBM's randomized predictions $\tilde{y}_t$ satisfy the following bound on the rank loss with probability at least $1 - N\delta$*

$$\sum_{t=1}^{T} L_t^{rnk}(\tilde{y}_t) \leq \frac{m^2}{4}(N+1)\exp(-\frac{\gamma^2 N}{2})T + 2\rho m T + \tilde{\mathcal{O}}(\frac{2m^2-k^2}{\rho}N^2\sqrt{T}).$$

We can optimize $\rho \propto N\sqrt{\frac{2m^2-k^2}{m}}T^{-\frac{1}{4}}$ so that the first term in the bound becomes the asymptotic average loss bound. We can compare it to the asymptotic error bounds of OnlineBMR by multiplying the full information algorithm loss bounds by $\frac{m^2}{4}$, which is the maximum value of the rank loss normalization constant. Let s_t' be the score vectors produced by the full information algorithm.

Then we have that

$$\sum_{t=1}^{T} L_t^{\text{rnk}}(s'_t) \leq \frac{m^2}{4}(N+1)\exp(-\frac{\gamma^2 N}{2})T + \frac{m^2}{2}NS.$$

We see that the asymptotic losses, after optimizing ρ, are identical, so that the cost of top-k feedback appears only in the excess loss. Furthermore, since OnlineBMR is optimal in the number of weak learners it requires to achieve some asymptotic loss, TopkBBM is also optimal in this regard since the problem it faces is only harder because of partial information.

We now bound the cumulative rank loss of TopkAdaptive using the weak learner's empirical edges.

Theorem 4.5 (TopkAdaptive, Rank Loss Bound)**.** *For any T, N satisfying $\delta \ll \frac{1}{N}$, the cumulative rank loss of TopkAdaptive, $\sum_{t=1}^{T} L^{rnk}(\tilde{y}_t, R_t)$, satisfies the following bound with probability at least $1-(N+4)\delta$:*

$$\frac{2m^2}{\sum_i |\gamma_i|}T + 2\rho m T + \tilde{\mathcal{O}}(\frac{(2m^2-k^2)N\sqrt{T}}{\rho\sum_i |\gamma_i|}),$$

where $\tilde{\mathcal{O}}$ suppresses dependence on $\log\frac{1}{\delta}$.

By optimizing $\rho \propto \sqrt{\frac{(2m^2-k^2)N}{m\sum_i|\gamma_i|}}T^{-\frac{1}{4}}$, we get the first term of the bound as the asymptotic average loss bound. To compare it with Ada.OLMR, we again multiply the bound by $\frac{m^2}{4}$ to account for the normalization constant. Let s'_t be the scores of the full information adaptive algorithm at time t. Then we have

$$\sum_t L_t^{\text{rnk}}(s'_t) \leq \frac{2m^2}{\sum_i|\gamma_i|}T + \tilde{\mathcal{O}}(\frac{N^2m^2}{\sum_i|\gamma_i|}).$$

Which matches the asymptotic loss in TopkAdaptive, after optimizing for ρ. Thus, the cost of top-k feedback is again only present in the excess loss.

CHAPTER 5

Thompson Sampling in Episodic Restless Bandit Problems

Restless bandits [Whittle, 1988] are variants of multi-armed bandit (MAB) problems [Robbins, 1952] [1]. Unlike the classical MABs, the arms have non-stationary reward distributions. Specifically, we will focus on the class of restless bandits whose arms change their states based on Markov chains. Restless bandits are also distinguished from *rested bandits* where only the active arms evolve and the passive arms remain frozen. We will assume that each arm changes according to two different Markov chains depending on whether it is played or not. Because of their extra flexibility in modeling non-stationarity, restless bandits have been applied to practical problems such as dynamic channel access problems [Liu et al., 2011, 2013] and online recommendation systems [Meshram et al., 2017].

Due to the arms' non-stationary nature, playing the same set of arms for every round usually does not produce the optimal performance. This makes the optimal policy highly non-trivial, and Papadimitriou and Tsitsiklis [1999] show that it is generally PSPACE hard to identify the optimal policy for restless bandits. As a consequence, many researchers have been devoted to find an efficient way to approximate the optimal policy [Liu and Zhao, 2010, Meshram et al., 2018]. This line of work primarily focuses on the *optimization* perspective in that the system parameters are already known.

Since the true system parameters are unavailable in many cases, it becomes important to examine restless bandits from a *learning* perspective. Due to the learner's additional uncertainty, however, analyzing a learning algorithm in restless bandits is significantly challenging. Liu et al. [2011, 2013] and Tekin and Liu [2012] prove $\mathcal{O}(\log T)$ bounds for confidence bound based algorithms, but their competitor always selects a fixed set of actions, which is known to be weak (see Section 5.4 for an empirical example of the weakness of the best fixed action competitor). Dai et al. [2011, 2014]

[1] This chapter is based on the paper that appeared in NeurIPS 2019 [Jung and Tewari, 2019].

show $\mathcal{O}(\log T)$ bounds against the optimal policy, but their assumptions on the underlying model are very limited. Ortner et al. [2012] prove an $\tilde{\mathcal{O}}(\sqrt{T})$ bound in general restless bandits, but their algorithm is intractable in general.

In a different line of work, Osband et al. [2013] study Thompson sampling in the setting of a fully observable Markov decision process (MDP) and show the Bayesian regret bound of $\tilde{\mathcal{O}}(\sqrt{T})$ (hiding dependence on system parameters like state and action space size). Unfortunately, this result is not applicable in our setting as ours is partially observable due to bandit feedback. Following Ortner et al. [2012], it is possible to transform our setting to the fully observable case, but then we end up having exponentially many states, which restricts the practical utility of existing results.

In this work, we analyze Thompson sampling in restless bandits where the system resets at the end of every fixed-length episode and the rewards are binary. We emphasize that this episodic assumption simplifies our analysis as the problem boils down to a *finite time horizon* problem. This assumption can be arguably limited, but there are applications such as dynamic channel access problems where the channel provider might reset their system every night for a maintenance-related reason and the episodic assumption becomes natural. We directly tackle the partial observability and achieve a meaningful regret bound, which when restricted to the classical MABs matches the Thompson sampling result in that setting. We are not the first to analyze Thompson sampling in restless bandits, and Meshram et al. [2016] study this type of algorithm as well, but their regret analysis remains in the one-armed-case with a fixed reward of not pulling the arm. They explicitly mention that a regret analysis of Thompson sampling in the multi-armed case is an interesting open question.

5.1 Problem setting

We begin by introducing our setting. There are K arms indexed by $k = 1, \cdots, K$, and the algorithm selects N arms every round. We denote the learner's action at time t by a binary vector $A_t \in \{0,1\}^K$ where $||A_t||_1 = N$. We call the selected arms as *active* and the rest as *passive*. We assume each arm k has binary states, $\{0,1\}$, which evolve as a Markov chain with transition matrix either P_k^{active} or P_k^{passive}, depending on whether the learner pulled the arm or not.

At round t, pulling an arm k incurs a binary reward $X_{t,k}$, which is the arm's current state. As we are in the bandit setting, the learner only observes the rewards of active arms, which we denote by X_{t,A_t}, and does not observe the passive arms' rewards nor their states. This feature makes our setting to be a *partially observable Markov decision process*, or POMDP. We denote the history of the learner's actions and rewards up to time t by $\mathcal{H}_t = (A_1, X_{1,A_1}, \cdots, A_t, X_{t,A_t})$.

We assume the system resets every episode of length L, which is also known to the learner. This means that at the beginning of each episode, the states of the arms are drawn from an initial distribution. The entire time horizon is denoted by T, and for simplicity, we assume it is a multiple of L, or $T = mL$.

5.1.1 Bayesian regret and competitor policy

Let $\theta \in \Theta$ denote the entire parameters of the system. It includes transition matrices P^{active} and P^{passive}, and an initial distribution of each arm's state. The learner only knows the prior distribution of this parameter at the beginning and does not have access to the exact value.

In order to define a regret, we need a *competitor policy*, or a *benchmark*. We first define a class of deterministic policies and policy mappings.

Definition 5.1. *A deterministic policy π takes time index and history $(t, \mathcal{H}_{t-1})$ as an input and outputs a fixed action $A_t = \pi(t, \mathcal{H}_{t-1})$. A deterministic policy mapping μ takes a system parameter θ as an input and outputs a deterministic policy $\pi = \mu(\theta)$.*

We fix a deterministic policy mapping μ and let our algorithm compete against a deterministic policy $\pi^\star = \mu(\theta^\star)$, where $\theta^\star$ represents the true system parameter, which is unknown to the learner.

We keep our competitor policy abstract mainly because we are in the non-stationary setting. Unlike the classical (stationary) MABs, pulling the same set of arms with the largest expected rewards is not necessarily optimal. Moreover, it is in general PSPACE hard to compute the optimal policy when $\theta^\star$ is given. Regarding these statements, we refer the readers to the book by Gittins et al. [1989]. As a consequence, researchers have identified conditions that the (efficient) myopic policy is optimal [Ahmad et al., 2009] or proven that a tractable index-based policy has a reasonable performance against the optimal policy [Liu and Zhao, 2010].

We observe that most of proposed policies including the optimal policy, the myopic policy, or the index-based policy are deterministic. Therefore, researchers can plug in whatever competitor policy of their choice, and our regret bound will apply as long as the chosen policy mapping is deterministic.

Before defining the regret, we introduce a *value function*

$$V^{\theta}_{\pi,i}(\mathcal{H}) = \mathbb{E}_{\theta,\pi}[\sum_{j=i}^{L} A_j \cdot X_j | \mathcal{H}]. \tag{5.1}$$

This is the expected reward of running a policy π from round i to L where the system parameter is θ and the starting history is $\mathcal{H}$. Note that the benchmark policy $\pi^\star$ obtains $V^{\theta^\star}_{\pi^\star,1}(\emptyset)$ rewards per

Algorithm 5.1 Thompson sampling in restless bandits

Input prior Q, episode length L, policy mapping μ
Initialize posterior $Q_1 = Q$, history $\mathcal{H} = \emptyset$
for episodes $l = 1, \cdots, m$ **do**
  Draw a parameter $\theta_l \sim Q_l$ and compute the policy $\pi_l = \mu(\theta_l)$
  Set $\mathcal{H}_0 = \emptyset$
  for $t = 1, \cdots, L$ **do**
    Select N active arms $A_t = \pi_l(t, \mathcal{H}_{t-1})$
    Observe rewards X_{t,A_t} and update $\mathcal{H}_t$
  end for
  Append $\mathcal{H}_L$ to $\mathcal{H}$ and update posterior distribution Q_{l+1} using $\mathcal{H}$
end for

episode in expectation. Thus, we can define the regret as

$$R(T;\theta^\star) = mV^{\theta^\star}_{\pi^\star,1}(\emptyset) - \mathbb{E}_{\theta^\star}\sum_{t=1}^{T} A_t \cdot X_t. \tag{5.2}$$

If an algorithm chooses to fix a policy π_l for the entire episode l, which is the case of our algorithm, then the regret can be written as

$$R(T;\theta^\star) = mV^{\theta^\star}_{\pi^\star,1}(\emptyset) - \mathbb{E}_{\theta^\star}\sum_{l=1}^{m} V^{\theta^\star}_{\pi_l,1}(\emptyset) = \mathbb{E}_{\theta^\star}\sum_{l=1}^{m} V^{\theta^\star}_{\pi^\star,1}(\emptyset) - V^{\theta^\star}_{\pi_l,1}(\emptyset).$$

We particularly focus on the case where $\theta^\star$ is a random and bound the following *Bayesian regret*,

$$BR(T) = \mathbb{E}_{\theta^\star\sim Q} R(T;\theta^\star),$$

where Q is a prior distribution over the set of system parameters Θ. We assume that the prior is known to the learner. We caution our readers that there is at least one other regret definition in the literature, which is called either *frequentist regret* or *worst-case regret*. For this type of regret, one views $\theta^\star$ as a fixed unknown object and directly bounds $R(T;\theta^\star)$. Even though our primary interest is to bound the Bayesian regret, we can establish a connection to the frequentist regret in the special case where the prior Q has a finite support and the benchmark is the optimal policy (see Corollary 5.6).

5.2 Algorithm

Our algorithm is an instance of *Thompson sampling* or *posterior sampling*, first proposed by Thompson [1933]. At the beginning of episode l, the algorithm draws a system parameter θ_l from the posterior and plays $\pi_l = \mu(\theta_l)$ throughout the episode. Once an episode is over, it updates the posterior based on additional observations. Algorithm 5.1 describes the steps.

We want to point out that the history $\mathcal{H}$ fulfills two different purposes. One is to update the posterior Q_l, and the other is as an input to a policy π_l. For the latter, however, we do not need the entire history as the arms reset every episode. That is why we set $\mathcal{H}_0 = \emptyset$ (step 5) and feed $\mathcal{H}_{t-1}$ to π_l (step 7). Furthermore, as we assume that the arms evolve based on Markov chains, the history $\mathcal{H}_{t-1}$ can be summarized as

$$(r_1, n_1, \cdots, r_K, n_K), \tag{5.3}$$

which means that an arm k is played n_k rounds ago and r_k is the observed reward in that round. If an arm k is never played in the episode, then n_k becomes t, and r_k becomes the expected reward from the initial distribution based on θ_l. As we assume the episode length is fixed to be L, there are L possible values for n_k. Due to the binary reward assumption, r_k can take three values including the case where the arm k is never played. From these, we can infer that there are $(3L)^K$ possible tuples of $(r_1, n_1, \cdots, r_K, n_K)$. By considering these tuples as states and following the reasoning of Ortner et al. [2012], one can view our POMDP as a fully observable MDP. Then one can use the existing algorithms for fully observable MDPs (e.g., Osband et al. [2013]), but the regret bounds easily become vacuous since the number of states depends exponentially on the number of arms K. Additionally, as we assumed a policy mapping, one might argue to use existing *expert learning* or classical MAB algorithms considering potential policies as experts or arms. This is possible, but the number of potential policies corresponds to the size of Θ, which can be very large or even infinite. For this reason, existing algorithms are not efficient and/or their regret bounds become too loose.

Due to its generality, it is hard to analyze the time and space complexity of Algorithm 5.1. Two major steps are computing the policy (step 4) and updating posterior (step 10). Computing the policy depends on our choice of competitor mapping μ. If the competitor policy has better performance but is harder to compute, then our regret bound gets more meaningful as the benchmark is stronger, but the running time gets longer. Regarding the posterior update, the computational burden depends on the choice of the prior Q and its support. If there is a closed-form update, then the step is computationally cheap, but otherwise the burden increases with respect to the size of the support.

5.3 Regret bound

In this section, we bound the Bayesian regret of Algorithm 5.1 by $\tilde{\mathcal{O}}(\sqrt{T})$. A key idea in our analysis of Thompson sampling is that the distributions of $\theta^\star$ and θ_l are identical given the history up to the end of episode $l-1$ (e.g., see Lattimore and Szepesvári [2018, Chp. 36]). To state it more formally, let $\sigma(\mathcal{H})$ be the σ-algebra generated by the history $\mathcal{H}$. Then we call a random variable X is *$\sigma(\mathcal{H})$-measurable*, or simply *$\mathcal{H}$-measurable*, if its value is deterministically known given the information $\sigma(\mathcal{H})$. Similarly, we call a random function f is $\mathcal{H}$-measurable if its mapping is deterministically known given $\sigma(\mathcal{H})$. We record as a lemma an observation made by Russo and Van Roy [2014].

Lemma 5.2. (Expectation identity) *Suppose $\theta^\star$ and θ_l have the same distribution given $\mathcal{H}$. For any $\mathcal{H}$-measurable function f, we have*

$$\mathbb{E}[f(\theta^\star)|\mathcal{H}] = \mathbb{E}[f(\theta_l)|\mathcal{H}].$$

Recall that we assume the competitor mapping μ is deterministic. Furthermore, the value function $V^{\theta}_{\pi,i}(\emptyset)$ in (5.1) is deterministic given θ and π. This implies $\mathbb{E}[V^{\theta^\star}_{\pi^\star,i}(\emptyset)|\mathcal{H}] = \mathbb{E}[V^{\theta_l}_{\pi_l,i}(\emptyset)|\mathcal{H}]$, where $\mathcal{H}$ is the history up to the end of episode $l-1$. This observation leads to the following regret decomposition.

Lemma 5.3. (Regret decomposition) *The Bayesian regret of Algorithm 5.1 can be decomposed as*

$$BR(T) = \mathbb{E}_{\theta^\star\sim Q}\sum_{l=1}^{m}\mathbb{E}_{\theta_l\sim Q_l}[V^{\theta^\star}_{\pi^\star,1}(\emptyset) - V^{\theta^\star}_{\pi_l,1}(\emptyset)] = \mathbb{E}_{\theta^\star\sim Q}\sum_{l=1}^{m}\mathbb{E}_{\theta_l\sim Q_l}[V^{\theta_l}_{\pi_l,1}(\emptyset) - V^{\theta^\star}_{\pi_l,1}(\emptyset)].$$

Proof. The first equality is a simple rewriting of (5.2) because Algorithm 5.1 fixes a policy π_l for the entire episode l. Then we apply Lemma 5.2 along with the tower rule to get

$$\mathbb{E}_{\theta^\star\sim Q}\sum_{l=1}^{m}\mathbb{E}_{\theta_l\sim Q_l}V^{\theta^\star}_{\pi^\star,1}(\emptyset) = \mathbb{E}_{\theta^\star\sim Q}\sum_{l=1}^{m}\mathbb{E}_{\theta_l\sim Q_l}V^{\theta_l}_{\pi_l,1}(\emptyset). \qquad \square$$

Note that we can compute $V^{\theta_l}_{\pi_l,1}(\emptyset)$ as we know θ_l and π_l. We can also infer the value of $V^{\theta^\star}_{\pi_l,1}(\emptyset)$ from the algorithm's observations. The main point of Lemma 5.3 is to rewrite the Bayesian regret using terms that are relatively easy to analyze.

Next, we define the *Bellman operator*

$$\mathcal{T}^{\theta}_{\pi}V(\mathcal{H}_{t-1}) = \mathbb{E}_{\theta,\pi}[A_t\cdot X_t + V(\mathcal{H}_t)|\mathcal{H}_{t-1}].$$

It is not hard to check that $V^{\theta}_{\pi,i} = \mathcal{T}^{\theta}_{\pi} V^{\theta}_{\pi,i+1}$. The next lemma further decomposes the regret.

Lemma 5.4. (Per-episode regret decomposition) *Fix $\theta^\star$ and θ_l, and let $\mathcal{H}_0 = \emptyset$. Then we have*

$$V^{\theta_l}_{\pi_l,1}(\mathcal{H}_0) - V^{\theta^\star}_{\pi_l,1}(\mathcal{H}_0) = \mathbb{E}_{\theta^\star,\pi_l} \sum_{t=1}^{L} (\mathcal{T}^{\theta_l}_{\pi_l} - \mathcal{T}^{\theta^\star}_{\pi_l}) V^{\theta_l}_{\pi_l,t+1}(\mathcal{H}_{t-1}).$$

Proof. Using the relation $V^{\theta}_{\pi,i} = \mathcal{T}^{\theta}_{\pi} V^{\theta}_{\pi,i+1}$, we may write

$$\begin{aligned} V^{\theta_l}_{\pi_l,1}(\mathcal{H}_0) - V^{\theta^\star}_{\pi_l,1}(\mathcal{H}_0) &= (\mathcal{T}^{\theta_l}_{\pi_l} V^{\theta_l}_{\pi_l,2} - \mathcal{T}^{\theta^\star}_{\pi_l} V^{\theta^\star}_{\pi_l,2})(\mathcal{H}_0) \\ &= (\mathcal{T}^{\theta_l}_{\pi_l} - \mathcal{T}^{\theta^\star}_{\pi_l}) V^{\theta_l}_{\pi_l,2}(\mathcal{H}_0) + \mathcal{T}^{\theta^\star}_{\pi_l}(V^{\theta_l}_{\pi_l,2} - V^{\theta^\star}_{\pi_l,2})(\mathcal{H}_0). \end{aligned}$$

The second term can be written as $\mathbb{E}_{\theta^\star,\pi_l}[(V^{\theta_l}_{\pi_l,2} - V^{\theta^\star}_{\pi_l,2})(\mathcal{H}_1)|\mathcal{H}_0]$, and we can repeat this L times to obtain the desired equation. □

Now we are ready to prove our main theorem. A complete proof can be found in Appendix C.1.

Theorem 5.5. (Bayesian regret bound of Thompson sampling) *The Bayesian regret of Algorithm 5.1 satisfies the following bound*

$$BR(T) = \mathcal{O}(\sqrt{KL^3N^3T\log T}) = \mathcal{O}(\sqrt{mKL^4N^3\log(mL)}).$$

The constant hidden in $\mathcal{O}$ notation does not depend on the choice of policy mapping μ.

Remark. *If the system is the classical stationary MAB, then it corresponds to the case $L = 1, N = 1$, and our result reproduces the result of $\mathcal{O}(\sqrt{KT\log T})$ [Lattimore and Szepesvári, 2018, Chp. 36]. This suggests our bound is optimal in K and T up to a logarithmic factor. Further, when $N > \frac{K}{2}$, we can think of the problem as choosing the passive arms, and the smaller bound with N replaced by $K - N$ would apply. When $L = 1$, the problem becomes combinatorial bandits of choosing N active arms out of K. Cesa-Bianchi and Lugosi [2012] propose an algorithm with a regret bound $\mathcal{O}(\sqrt{KNT\log K})$ with an assumption that the loss is always bounded by 1. Since our reward can be as big as N, our bound has the same dependence on N with theirs, suggesting tight dependence of our bound on N.*

Proof Sketch. We fix an episode l and analyze the regret in this episode. Let $t_l = (l-1)L$ so that the episode starts at time $t_l + 1$. Define $N_l(k, r, n) = \sum_{t=1}^{t_l} \mathbb{1}\{A_{t,k} = 1, r_k = r, n_k = n\}$, which counts the number of rounds where the arm k was chosen by the learner with history $r_k = r$ and $n_k = n$ (see (5.3) for definition). Note that $k \in [K], r \in \{0, 1, \rho(k)\}$, and $n \in [L]$, where $\rho(k)$ is the initial success rate of the arm k. This implies there are $3KL$ tuples of (k, r, n).

Let $\omega^\theta(k,r,n)$ denote the conditional probability of $X_k = 1$ given a history (r,n) and a system parameter θ. Also let $\hat{\omega}(k,r,n)$ denote the empirical mean of this quantity (using $N_l(k,r,n)$ past observations and set the estimate to 0 if $N_l(k,r,n) = 0$). Then define

$$\Theta_l = \left\{ \theta \mid \forall (k,r,n),\ \ |(\hat{\omega} - \omega^\theta)(k,r,n)| < \sqrt{\frac{2\log(1/\delta)}{1 \vee N_l(k,r,n)}} \right\}.$$

Since $\hat{\omega}(k,r,n)$ is $\mathcal{H}_{t_l}$-measurable, so is the set Θ_l. Using the Hoeffding inequality, one can show $\mathbb{P}(\theta^\star \notin \Theta_l) = \mathbb{P}(\theta_l \notin \Theta_l) \leq 3\delta KL$. In other words, we can claim that with high probability, $|\omega^{\theta_l}(k,r,n) - \omega^{\theta^\star}(k,r,n)|$ is small for all (k,r,n).

We now turn our attention to the following Bellman operator

$$\mathcal{T}^\theta_{\pi_l} V^{\theta_l}_{\pi_l,t}(\mathcal{H}_{t-1}) = \mathbb{E}_{\theta,\pi_l}[A_{t_l+t} \cdot X_{t_l+t} + V^{\theta_l}_{\pi_l,t}(\mathcal{H}_t)|\mathcal{H}_{t-1}].$$

Since π_l is a deterministic policy, A_{t_l+t} is also deterministic given $\mathcal{H}_{t-1}$ and π_l. Let $(k_1,\ldots,k_N)$ be the active arms at time $t_l + t$ and write $\omega^\theta(k_i, r_{k_i}, n_{k_i}) = \omega_{\theta,i}$. Then we can rewrite

$$\mathcal{T}^\theta_{\pi_l} V^{\theta_l}_{\pi_l,t}(\mathcal{H}_{t-1}) = \sum_{i=1}^N \omega_{\theta,i} + \sum_{x \in \{0,1\}^N} P^\theta_x V^{\theta_l}_{\pi_l,t}(\mathcal{H}_{t-1} \cup (A_{t_l+t}, x)),$$

where $P^\theta_x = \prod_{i=1}^N \omega^{x_i}_{\theta,i}(1-\omega_{\theta,i})^{1-x_i}$. Under the event that $\theta^\star, \theta_l \in \Theta_l$, we have

$$|\omega_{\theta_l,i} - \omega_{\theta^\star,i}| < 1 \wedge \sqrt{\frac{8\log(1/\delta)}{1 \vee N_l(k_i, r_{k_i}, n_{k_i})}} =: \Delta_i(t_l + t),$$

where the dependence on $t_l + t$ comes from the mapping from i to k_i. When $\omega_{\theta_l,i}$ and $\omega_{\theta^\star,i}$ are close for all (k,r,n), we can actually bound the difference between the following Bellman operators as

$$|(\mathcal{T}^{\theta^\star}_{\pi_l} - \mathcal{T}^{\theta_l}_{\pi_l}) V^{\theta_l}_{\pi_l,t}(\mathcal{H}_{t-1})| \leq 3LN \sum_{i=1}^N \Delta_i(t_l + t).$$

Then by applying Lemma 5.4, we get $|V^{\theta_l}_{\pi_l,1}(\emptyset) - V^{\theta^\star}_{\pi_l,1}(\emptyset)| \leq 3LN\mathbb{E}_{\theta^\star,\pi_l} \sum_{t=1}^L \sum_{i=1}^N \Delta_i(t_l + t)$, which holds whenever $\theta^\star, \theta_l \in \Theta_l$. When $\theta^\star \notin \Theta_l$ or $\theta_l \notin \Theta_l$, which happens with probability less

than $6\delta KL$, we have a trivial bound $|V^{\theta_l}_{\pi_l,1}(\emptyset) - V^{\theta^\star}_{\pi_l,1}(\emptyset)| \leq LN$. We can deduce

$$|V^{\theta_l}_{\pi_l,1}(\emptyset) - V^{\theta^\star}_{\pi_l,1}(\emptyset)| \leq 3LN\mathbb{1}(\theta^\star, \theta_l \in \Theta_l)\mathbb{E}_{\theta^\star,\pi_l}\sum_{t=1}^{L}\sum_{i=1}^{N}\Delta_i(t_l+t) + 6\delta KL^2N.$$

Combining this with Lemma 5.3, we can show

$$BR(T) \leq 6\delta mKL^2N + \mathbb{E}_{\theta^\star\sim Q}3LN\sum_{l=1}^{m}\mathbb{1}(\theta^\star, \theta_l \in \Theta_l)\mathbb{E}_{\theta^\star,\pi_l}\sum_{t=1}^{L}\sum_{i=1}^{N}\Delta_i(t_l+t). \tag{5.4}$$

After some algebra, bounding sums of finite differences by integrals, and applying the Cauchy-Schwartz inequality, we can bound the second summation by

$$18KL^3N + 24\sqrt{3KL^3N^3T\log(1/\delta)}. \tag{5.5}$$

Combining (5.4), (5.5), and our assumption that $T = mL$, we obtain

$$BR(T) = \mathcal{O}(\delta KLNT + KL^3N + \sqrt{KL^3N^3T\log(1/\delta)}).$$

Since NT is a trivial upper bound of $BR(T)$, we may ignore the KL^3N term. Setting $\delta = \frac{1}{T}$ completes the proof. □

As discussed in Section 5.1, researchers sometimes pay more attention to the case where the true parameter $\theta^\star$ is deterministically fixed in advance, in which the frequentist regret becomes more relevant. It is not easy to directly extend our analysis to the frequentist regret in general, but we can achieve a meaningful bound with extra assumptions. Suppose our prior Q is discrete and the competitor is the optimal policy. Then we know $R(T;\theta^\star)$ is always non-negative due to the optimality of the benchmark and can deduce $qR(T;\theta^\star) \leq BR(T)$, where q is the probability mass on $\theta^\star$. This leads to the following corollary.

Corollary 5.6. (Frequentist regret bound of Thompson sampling) *Suppose the prior Q is discrete and puts a non-zero mass on the parameter $\theta^\star$. Additionally, assume that the competitor policy is the optimal policy. Then Algorithm 5.1 satisfies the following bound*

$$R(T;\theta^\star) = \mathcal{O}(\sqrt{KL^3N^3T\log T}) = \mathcal{O}(\sqrt{mKL^4N^3\log(mL)}).$$

5.4 Experiments

We empirically investigate the Gilbert-Elliott channel model, which is studied by Liu and Zhao [2010] in a restless bandit perspective[2]. This model can be broadly used in communication systems such as cognitive radio networks, downlink scheduling in cellular systems, opportunistic transmission over fading channels, and resource-constrained jamming and anti-jamming.

Each arm k has two parameters p_{01}^k and p_{11}^k, which determine the transition matrix. We assume $P^{\text{active}} = P^{\text{passive}}$ and each arm's transition matrix is independent on the learner's action. There are only two states, *good* and *bad*, and the reward of playing an arm is 1 if its state is good and 0 otherwise. Figure 5.1 summarizes this model. We assume the initial distribution of an arm k follows the stationary distribution. In other words, its initial state is good with probability $\omega_k = \frac{p_{01}^k}{p_{01}^k + 1 - p_{11}^k}$.

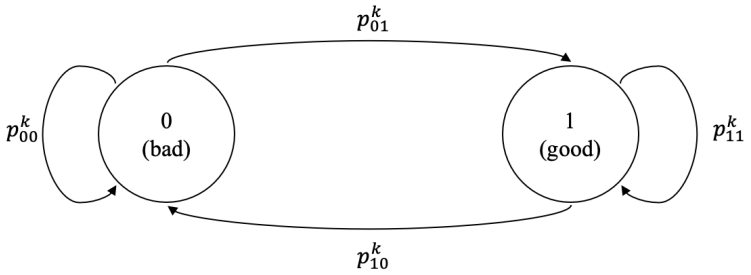

Figure 5.1: The Gilbert-Elliott channel model

We fix $L = 50$ and $m = 30$. We use Monte Carlo simulation with size 100 or greater to approximate expectations. As each arm has two parameters, there are $2K$ parameters. For these, we set the prior distribution to be uniform over a finite support $\{0.1, 0.2, \cdots, 0.9\}$.

5.4.1 Competitors

As mentioned earlier, one important strength of our result is that various policy mappings can be used as benchmarks. Here we test three different policies: the best fixed arm policy, the myopic policy, and the Whittle index policy. We want to emphasize again that these competitor policies know the system parameters while our algorithm does not.

The best fixed arm policy computes the stationary distribution $\omega_k = \frac{p_{01}^k}{p_{01}^k + 1 - p_{11}^k}$ for all k and pulls the arms with top N values. The myopic policy keeps updating the belief $\omega_k(t)$ for the arm k being in a good state and pulls the top N arms. Finally, the Whittle index policy computes the Whittle index of each arm and uses it to rank the arms. The Whittle index is proposed by Whittle

[2]Our code is available at https://github.com/yhjung88/ThompsonSamplinginRestlessBandits

[1988], and Liu and Zhao [2010] find a closed-form formula to compute the Whittle index in this particular setting. The Whittle index policy is very popular in optimization literature as it decouples the optimization process into K independent problems for each arm, which significantly reduces the computational complexity while maintaining a reasonable performance against the optimal policy.

One observation is that these three policies are reduced to the best fixed arm policy in the stationary case. However, the first two policies are known to be sub-optimal in general [Gittins et al., 1989]. Liu and Zhao [2010] justify both theoretically and empirically the performance of the Whittle index policy for the Gilbert-Elliott channel model.

5.4.2 Results

We first analyze the Bayesian regret. For this, we use $K = 8$ and $N = 3$. The value functions $V^{\theta}_{\pi,1}(\emptyset)$ of the best fixed arm policy, the myopic policy, and the Whittle index policy are $105.4, 110.3$, and 111.4, respectively. If a competitor policy has a weak performance, then Thompson sampling also uses this weak policy mapping to get a policy π_l for the episode l. This implies that the regret does not necessarily become negative when the benchmark policy is weak. Figure 5.2 shows the trend of the Bayesian regret as a function of episode indices. Regardless of the choice of policy mapping, the regret is sub-linear, and the slope of log-log plot is less than $\frac{1}{2}$, which agrees with Theorem 5.5.

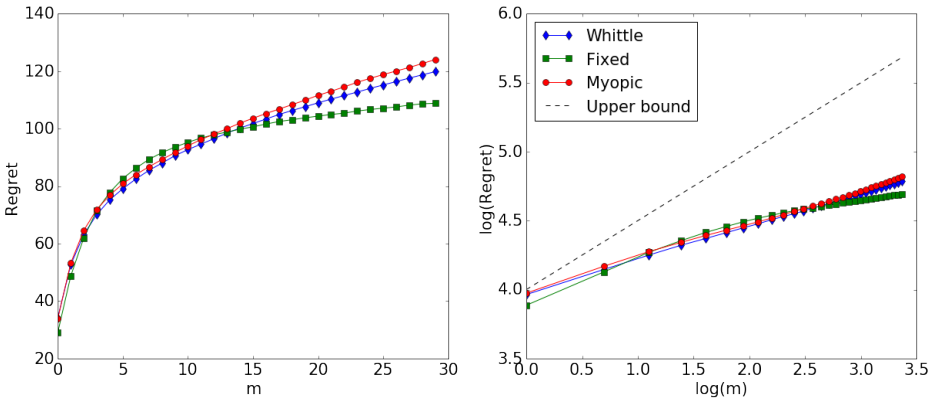

Figure 5.2: Bayesian regret of Thompson sampling versus episode (left) and its log-log plot (right)

Next we fix true parameters and investigate the model's behavior more closely. For this, we choose $K = 4$, $N = 2$, and $\{(p^k_{01}, p^k_{11})\}_{k=1,2,3,4} = \{(0.3, 0.7), (0.4, 0.6), (0.5, 0.5), (0.6, 0.4)\}$. This choice results in $\omega_k = 0.5$ for all k, and the best fixed arm policy becomes indifferent. Therefore achieving zero regret against the best fixed arm becomes trivial. We use the same uniform prior as the previous experiment. Figure 5.3 presents the trend of value functions and how Thompson sampling puts more posterior weights on the correct parameters as it proceeds. Three horizontal

lines in the left figure represent the values of the competitor policies. The values of the best fixed arm policy, the myopic policy, and the Whittle index policy are $50.2, 54.6$, and 55.6, respectively. It is a good example why one should not pull the same arms all the time in restless bandits. The value function of Thompson sampling successfully converges to the competitor value for every benchmark while the one with the myopic policy needs more episodes to fully converge. This supports Corollary 5.6 in that our model can be used even in the non-Bayesian setting as far as the prior has a non-zero weight on the true parameters. Also, the posterior weights on the correct parameters monotonically increase (Figure 5.3, right), which again confirms our model's performance. We measure these weights when the competitor map is the Whittle index policy.

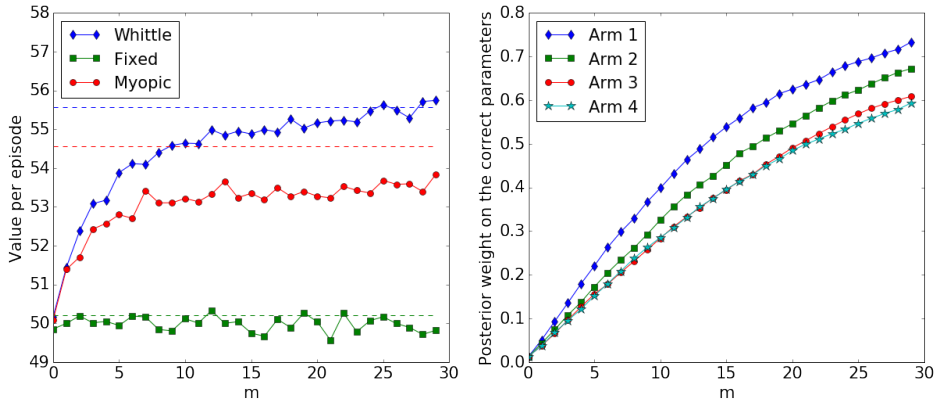

Figure 5.3: Average per-episode value versus episode and the benchmark values (left); the posterior weights of the correct parameters versus episode in the case of the Whittle index policy (right)

CHAPTER 6

Thompson Sampling in Non-Episodic Restless Bandits

In contrast to the classical multi-armed bandits (MABs), *restless multi-armed bandits* (RMABs), introduced by Whittle [1988], assume reward distributions that change along with the time [1]. Due to their non-stationary nature, RMABs can model more complicated systems and thus get more attention in practice and theoretical literature. In practice, they are used in a wide spectrum of applications including *sensor management* (Chapter 7 in Hero et al. [2007] and Chapter 5 in Biglieri et al. [2013]), *dynamic channel access* problems [Liu et al., 2011, 2013], and *online recommendation systems* [Meshram et al., 2017]. Theoretically, a variety of research communities have contributed to the literature on restless bandits, e.g., *complexity theory* [Blondel and Tsitsiklis, 2000], *applied probability* [Weber and Weiss, 1990], and *optimization* [Bertsimas and Niño-Mora, 2000].

In this setting, there are K independent arms indexed by $k \in [K]$.[2] Each arm is characterized by an internal state $s_k \in S_k$ which evolves in a *Markovian fashion* according to the (possibly distinct) transition matrices P_k^{active} and P_k^{passive} depending on whether the arm is pulled (i.e., *active*) or not (i.e., *passive*). The reward of pulling an arm k depends on its state s_k^t, which brings the non-stationarity.

We aggregate the transition matrices as $\theta \in \Theta$ and consider this problem as a *Reinforcement Learning* problem where θ is unknown to the learner. This problem has a complication in defining the baseline competitor against which the learner competes. It is not guaranteed, without additional assumptions, that the optimal policy exists, and even if it exists, Papadimitriou and Tsitsiklis [1999] show that it is generally PSPACE hard to compute the optimal policy.

Researchers take different paths to tackle this challenge. Some define the regret using a simpler policy, which can be easily computed (e.g., see Tekin and Liu [2012], Liu et al. [2013]). They

[1] This chapter is based on joint work (with the same title) with Marc Abeille currently available as an arXiv preprint https://arxiv.org/abs/1910.05654.

[2] For an integer n, we denote the set $\{1, \cdots, n\}$ by $[n]$.

compare the learner's reward to a policy that pulls a fixed set of arms every round. Their algorithm is efficient and has a strong regret guarantee, $\mathcal{O}(\log T)$, but this baseline policy is known to be weak in the RMAB setting, which makes the regret less meaningful. Our empirical results in Section 6.6 also show the weakness of this policy. Another breakthrough is made by Ortner et al. [2012] who show a sub-linear regret bound against the optimal policy. However, they ignore the computational burden of their algorithm.

In Chapter 5, we propose another interesting direction in that they introduce a deterministic *policy mapping* μ. It takes the system parameter θ as an input and outputs a deterministic stationary policy $\pi = \mu(\theta)$. Then the learner competes against the policy $\pi^\star = \mu(\theta^\star)$, where $\theta^\star$ denotes the true system. This framework is general enough to include the best fixed arm policy and the optimal policy that are mentioned earlier. That being said, one can achieve an efficient algorithm by choosing an efficient mapping μ or make the regret more meaningful with a stronger policy. In fact, there are different lines of work (e.g., Whittle [1988], Liu and Zhao [2010], Meshram et al. [2017]) that study an efficient way, namely the *Whittle index policy*, to approximate the optimal algorithm. Using this policy as a mapping, one can obtain an efficient algorithm with a meaningful regret simultaneously.

In this chapter, we also adopt the policy mapping from Chapter 5 and answer an open question raised by them. Specifically, they prove the regret bound of *Thompson sampling* in the episodic restless bandits where the system periodically resets. From the episodic assumption, the problem boils down to a *finite horizon* problem, which makes the analysis simpler. However, there are many cases (e.g., online recommendations) where the periodic reset is not natural, and they mention the analysis of a learning algorithm in the *infinite time horizon* as an open question.

We identify explicit conditions in Section 6.4 that ensure the *Bellman* equation of the entire Markov decision process (MDP). It is hard to analyze the vanilla Thompson sampling in this setting, and we adapt *Thompson sampling with dynamic episodes* (TSDE) of Ouyang et al. [2017] in the fully observable MDP. TSDE (Algorithm 6.1) has one deterministic and one random termination conditions and switches to a new episode if one of these is met. At the beginning of each episode, TSDE draws a system parameter using the posterior distribution from which it computes a policy and runs this policy throughout the episode. We theoretically prove a sub-linear regret bound of this algorithm and empirically test it on a simulated dynamic channel access problem.

6.1 Main result

As mentioned earlier, our learner competes against the policy $\pi^\star = \mu(\theta^\star)$ without the knowledge of $\theta^\star \in \Theta$. We denote the *average long term reward* of $\pi^\star$ on the system $\theta^\star$ by $J_{\pi^\star}(\theta^\star)$, which is a well-defined notion under certain assumptions that will be discussed later. Then we define the *frequentist regret* by

$$R(T;\theta^\star) = J_{\pi^\star}(\theta^\star)\cdot T - \mathbb{E}_{\theta^\star}\sum_{t=1}^{T} r_t, \tag{6.1}$$

where r_t is the learner's reward at time t. We focus on bounding the following *Bayesian regret*

$$BR(T) = \mathbb{E}_{\theta^\star\sim Q} R(T;\theta^\star), \tag{6.2}$$

where Q is a prior distribution over Θ and is known to the learner. Our main result is to bound the Bayesian regret of TSDE.

Theorem 6.1. *The Bayesian regret of TSDE satisfies the following bound*

$$BR(T) = \mathcal{O}(\sqrt{T}\log T),$$

where the exact upper bound appears later in Section 6.5.

6.2 Preliminaries

We begin by formally defining our problem setting.

6.2.1 Problem setting

As stated earlier, we focus on a Bayesian framework where the true system, denoted as $\theta^\star$, is a random object that is drawn from a prior distribution Q before the interaction with the system begins. In line with Ouyang et al. [2017], we assume that the prior is known to the learner, and we denote its support by Θ.

At each time step t, the learner selects N arms from $[K]$ which become *active* while the others remain *passive*. Following Ortner et al. [2012], we impose the *passive* Markov chains to be irreducible and aperiodic. As a result, we can associated with each arm k the *mixing time* of P_k^{passive}. Let $p_k^t(s)$ be the distributions of the state s_k of arm k starting from a state s and remaining passive

for t steps, and let p_k be the stationary distribution. Then, we define

$$T_k^{\text{mix}}(\epsilon) = \inf\left\{t \geq 1 \text{ s.t.} \max_{s \in S_k} \|p_k^t(s) - p_k\|_1 \leq \epsilon\right\}, \tag{6.3}$$

and work under the assumption of known mixing time[3].

Assumption 6.2 (Mixing times). *For all $k \in [K]$ and $\theta \in \Theta$, $P_k^{passive}$ is irreducible and aperiodic, and $T^{mix}(\frac{1}{4}) := \max_{k,\theta} T_k^{mix}(\frac{1}{4})$ is known to the learner.*

The learner's action at time t is written as $A_t \in \{0,1\}^K$, 1 indicating the active action. For all the chosen arms, the learner observes the state s_k^t and receives a reward $r_k(s_k^t)$, where the rewards are deterministic *known* functions of the state $r_k : S_k \to [0,1]$ for all $k \in [K]$. The objective of the learner is to choose the best sequence of arms, given the history (state and actions) observed so far, which maximizes the long term average reward

$$\limsup_{t \to T} \frac{1}{T} \mathbb{E}\left(\sum_{t=1}^{T} \sum_{k:A_{t,k}=1} r_k(s_k^t)\right). \tag{6.4}$$

6.2.2 From POMDP to MDP

By nature, the RMAB problem we consider is a *partially observable Markov decision process* (POMDP) since the arms evolve in a Markovian fashion and we only observe the states of the active arms. Nonetheless, one can turn this POMDP into a fully observable *Markov decision process* (MDP) by introducing belief states, i.e., distributions over states given the history. Notice that the number of belief states become therefore (countably) infinite even if the original problem is finite. Following Ortner et al. [2012], we track the history introducing a *meta-state* ξ_t, fully observed at time t, from which we can reconstruct the belief states. Formally, we define $\xi_t = (\xi_t^s, \xi_t^n)$ where

$$\xi_t^s = (\sigma_1^t, \cdots, \sigma_K^t) \text{ and } \xi_t^n = (n_1^t, \cdots, n_K^t).$$

For each $k \in [K]$, σ_k^t is the last observation of the state process $\{s_k^t\}_{t \geq 1}$ before time t, n_k^t is time elapsed from this last observation. Further, it is clear that $\{\xi_t\}_{t \geq 1}$ is a Markov process on a countably infinite state space S. As a result, the maximization of the partially observable problem in (6.4) is

[3]The knowledge of $T^{mix}(\frac{1}{4})$ maybe relaxed to the knowledge of an upper bound of it, without affecting our result.

equivalent to the maximization of the fully observable one

$$\limsup_{t\to T} \frac{1}{T}\mathbb{E}\Big(\sum_{t=1}^{T} r_{\theta^\star}(\xi_t, A_t)\Big), \tag{6.5}$$

where

$$r_\theta(\xi_t, A_t) = \sum_{k:A_{t,k}=1} \mathbb{E}_\theta[r_k(s_k^t)|\xi_t, A_t]. \tag{6.6}$$

We use the notation $\mathbb{E}_\theta$ and r_θ to emphasize that the random behavior of s_k^t is governed by the system θ. We also assume that the initial state ξ_1 is known to the learner.

6.2.3 Policy mapping

To maximize the long term average reward in (6.5), Ortner et al. [2012] construct a finite approximation of the countable MDP which allows them, under a bounded diameter assumption, to compute ϵ-optimal policy for a given θ. However, their computational complexity is prohibitive for practical applications. As explained in the introduction, we follow a different approach, in line with Chapter 5, which achieves both tractability and optimality through the use of a policy mapping $\mu : \Theta \to \Pi$. It associates each parameter θ with a stationary deterministic policy π_θ. To ensure the well-posedness of the long-term average reward, we impose the following assumption on μ.

Assumption 6.3 (Bounded span). *For all $\theta \in \Theta$, the parameter/policy pair (θ, π_θ) satisfies Condition 6.5.*

Condition 6.5 is formalized and discussed in detail in Section 6.4. Assumption 6.3 should be understood as the counterpart of the bounded diameter assumption made by Ortner et al. [2012] or the bounded span assumption by Ouyang et al. [2017] adapted to our policy mapping approach.

6.3 Algorithm

Algorithm 6.1 builds on *Thompson Sampling with Dynamic Episodes* (TSDE) of Ouyang et al. [2017]. At the beginning of each episode i, we draw system parameters θ_i from the latest posterior Q_{t_i}, compute the policy $\pi_i = \mu(\theta_i)$, and run π_i throughout the episode. We proceed to the next episode if one of the termination conditions, which will appear shortly, occurs.

Before introducing the termination conditions, let us discuss Assumption 6.2. As pointed out in Ortner et al. [2012, (1)], we have $T_k^{\text{mix}}(\epsilon) \leq \log_2(1/\epsilon) T^{\text{mix}}(\frac{1}{4})$ for all $k \in [K]$ and $\epsilon > 0$. As we

Algorithm 6.1 TSDE in restless bandits

Input prior Q, policy mapping μ, mixing time T^{mix}, initial state ξ_1
Initialize $Q_1 = Q$, $t = 1$, $t_0 = 1$
for episodes $i = 1, 2, \cdots$ **do**
  Set $t_i = t$ and $T_{i-1} = t_i - t_{i-1}$
  Draw $\theta_i \sim Q_t$ and compute $\pi_i = \mu(\theta_i)$
  while not termination condition (6.7) **do**
    Select active arms $A_t = \pi_i(\xi_t)$
    Observe states s_k^t for active arms k
    Update ξ_t to ξ_{t+1} and Q_t to Q_{t+1}
    Increment $t = t + 1$
  end while
end for

want the accuracy of $\frac{1}{T}$, which will not affect the regret significantly, we define

$$T^{\text{mix}} := (\log_2 T) T^{\text{mix}}(\frac{1}{4}) \geq T^{\text{mix}}(\frac{1}{T}).$$

Here we assume the time horizon T is known. When it is unknown, we can use the *doubling trick* and get the same regret bound up to a constant factor. We remark that $T^{\text{mix}} = \mathcal{O}(\log T)$.

For tuples (k, s, n), we define

$$\tilde{N}_t(k, s, n) = \sum_{\tau=1}^{t-1} \mathbb{1}(A_{\tau,k} = 1, \sigma_k^t = s, n_k^t = n).$$

Then we introduce the truncated counter

$$N_t(k, s, n) = \begin{cases} \tilde{N}_t(k, s, n) & \text{if } n < T^{\text{mix}} \\ \sum_{n' \geq T^{\text{mix}}} \tilde{N}_t(k, s, n') & \text{if } n \geq T^{\text{mix}} \end{cases}.$$

The intuition behind this aggregation is that the distribution of the states remains similar for sufficiently large n, thanks to the mixing time. As a result, the possible number of tuples (k, s, n) with $n \leq T^{\text{mix}}$ is at most $\sum_k |S_k| \cdot T^{\text{mix}}$. When there is no ambiguity, we write $(k, s, n) = \zeta$ for brevity and let Z be the set of all possible values of ζ.

We *terminate* the episode i if

$$t > t_i + T_{i-1} \text{ or } N_t(\zeta) > 2N_{t_k}(\zeta) \text{ for some } \zeta \in Z, \tag{6.7}$$

where T_i represents the length of episode i. This quantity can differ for each episode. This is where the name dynamic episodes comes from. In addition, the second condition makes the quantity T_i random, and one recovers the well-known *lazy update scheme* from this condition [Jaksch et al., 2010, Ouyang et al., 2017]. The underlying intuition is that one should update the policy only after gathering enough additional information over the unknown Markov process.

6.4 Planning problem

The MDP reformulation in Section 6.2.2 reduces the objective to maximizing (6.5). However, we inherit from the original POMDP problem severe difficulties in the *planning* task. For example, given the parametrization $\theta^\star$, how to efficiently compute a *stationary* and *deterministic* policy π (i.e., π maps a state ξ to an action A in a deterministic manner) that maximizes the average long term reward, and more importantly, does such policy exist? Unfortunately, the average reward POMDP problem is not well understood in contrast with the finite state average reward MDP. In particular, it is known [Bertsekas, 1995] that the long term average reward may not be constant w.r.t. the initial state. Even when this holds, **1)** The Bellman equation may not have a solution. **2)** Value Iteration may fail to converge to the optimal average reward. **3)** There may not exist an optimal policy, stationary or non-stationary. **4)** Finally, even when the optimal policy exists, Papadimitriou and Tsitsiklis [1999] show that it is generally PSPACE hard to compute it.

To overcome this difficulty, Ortner et al. [2012] perform a state aggregation to reduce the countably infinite MDP into a finite one, which under the bounded diameter assumption can be solved using standard techniques. Although this reduction allows them to compute an ϵ-optimal policy, the computational complexity of their approach remains prohibitive for practical application. On the other hand, a significant amount of work has been done to design *good* policies in the RMAB framework, for instance the best fixed arm policy (that is optimal in the classical MAB framework), the myopic policy [Javidi et al., 2008], or the Whittle index policy [Whittle, 1988, Liu and Zhao, 2010]. We leverage this prior knowledge following an alternative approach that consists in competing with the best policy within some known class of policies. Formally, let Π be the set of stationary deterministic policies, and we assume a policy mapping $\mu : \Theta \to \Pi$ is given and known to the learner. This set of deterministic mappings is quite rich in that the optimal policy can be

also represented when it exists. If one cares more about the efficiency, one can use some efficient mappings while there is a trade-off of weakening the competitor.

Finally, in contrast to Ortner et al. [2012], our approach does not turn the countable MDP problem into a finite one. Hence, it requires a further condition on the parameter space Θ and the policy mapping μ for the average reward criterion in (6.5) to be well-posed. More precisely, we expect the average reward to be independent of the initial state and associated to a Bellman equation, with a bias function of a bounded span. For a given $\theta \in \Theta$ and associated policy $\pi_\theta = \mu(\theta)$, we introduce the following conditions.

Condition 6.4. *Let $\mathcal{V}$ be the set of bounded span real-valued function. There exists $v \in \mathcal{V}$ and a constant g which satisfy for all $\xi \in S$,*

$$g + v(\xi) = r_\theta(\xi, \pi_\theta(\xi)) + \mathbb{E}_\theta[v(\xi')|\xi, \pi_\theta(\xi)],$$

where the expectation is taken over ξ' evolving from ξ given the action $\pi_\theta(\xi)$ and the system θ.

Under Condition 6.4, it is known (see Proposition 6.6) that the long term average reward of π_θ is well-defined (the $\limsup$ reduces to the standard $\lim$), independent of the initial state ξ_1, and associated with the Bellman equation with a bounded span bias function. However, Condition 6.4 is implicit and uneasy to assert as it relies on the existence result[4]. This motivates the alternative condition, known as the *discounted approach* in the literature.

Condition 6.5. *For any $\beta \in (0, 1)$, let $v^\beta_{\pi_\theta}$ be the discounted infinite horizon value function defined as*

$$v^\beta_{\pi_\theta}(\xi) = \mathbb{E}_\theta \left(\sum_{t=1}^{\infty} \beta^t r_\theta(\xi_t, \pi_\theta(\xi_t)) | \xi_1 = \xi \right).$$

Then $\sup_{(\xi,\xi') \in S^2} v^\beta_{\pi_\theta}(\xi) - v^\beta_{\pi_\theta}(\xi')$ is uniformly bounded for all $\beta \in (0, 1)$.

The introduction of the discount factor $\beta \in (0, 1)$ guarantees that $v^\beta_{\pi_\theta}$ is a well-defined function, and hence Condition 6.5 is reduced to assert the uniform boundedness of a known family of function. Further, it also guarantees that the long term average reward is well-defined as it implies Condition 6.4.

Proposition 6.6. *Let $\theta \in \Theta$ be a system parameter and $\pi_\theta = \mu(\theta)$ be a policy. Then the followings hold.*

- *Condition 6.5 implies Condition 6.4.*

[4] If a function v satisfies Condition 6.4, it is not unique since adding any constant to v still meet the requirement.

- *Under Condition 6.4 (or Condition 6.5), the quantity*

$$J_{\pi_\theta}(\theta) = \lim_{T\to\infty} \frac{1}{T}\mathbb{E}_\theta\left(\sum_{t=1}^{T} r_\theta(\xi_t, \pi_\theta(\xi_t))|\xi_1 = \xi\right) \tag{6.8}$$

is constant and independent of the initial state. Further, there exists a non-negative function h_θ, with bounded span $C_\theta = \sup_{(\xi,\xi')\in S^2} h_\theta(\xi) - h_\theta(\xi') < \infty$, such that for any $\xi \in S$,

$$J_{\pi_\theta}(\theta) + h_\theta(\xi) = r_\theta(\xi, \pi_\theta(\xi)) + \mathbb{E}_\theta[h_\theta(\xi')|\xi, \pi_\theta]. \tag{6.9}$$

We denote as $H = \sup_{\theta\in\Theta} C_\theta$ the uniform upper bound on the span.

The proof of Proposition 6.6 can be adapted from Puterman [2014, Theorem8.10.7] for a given (i.e., not necessarily optimal) policy. We postpone the proof to Appendix D.1.

6.5 Regret bound

In this section, we bound the Bayesian regret of TSDE (Algorithm 6.1). The analysis crucially relies on four distinct properties: **1)** the Bellman equation in (6.9) satisfied by the average cost at each policy update, **2)** the Thompson sampling algorithm which samples parameters θ_i according to the posterior, hence ensuring that $\theta^\star$ and θ_i are conditionally identical in distribution, **3)** the concentration of the empirical estimates around the $\theta^\star$, and **4)** the update scheme in (6.7) which controls the number of episodes while preserving sufficient measurability of the termination times.

We provide here a proof sketch to explain how we leverage those properties and how they translate in key intermediate results that allow us to obtain the final bound. The formal proofs can be found in Appendix D.2.

6.5.1 Regret decomposition

Under Assumption 6.3, Proposition 6.6 ensures that each sampled parameter policy pair (θ_i, π_i) satisfies the Bellman equation (6.9):

$$r_{\theta_i}(\xi, \pi_i(\xi)) = J_{\pi_i}(\theta_i) + v_{\theta_i}(\xi) - \mathbb{E}_{\theta_i}[v_{\theta_i}(\xi')|\pi_i, \xi].$$

As a result, we can decompose on each episode i the frequentist regret and obtain over T,

$$R(T;\theta^\star) = J_{\pi^\star}(\theta^\star)\cdot T - \mathbb{E}_{\theta^\star}\sum_{i=1}^{M_T}\sum_{t=t_i}^{t_{i+1}-1} r_{\theta^\star}(\xi_t, A_t)$$
$$=: R_0 + R_1 + R_2 + R_3,$$

where

$$R_0 = J_{\pi^\star}(\theta^\star)\cdot T - \mathbb{E}_{\theta^\star}\sum_{i=1}^{M_T} J_{\pi_i}(\theta_i)\cdot T_i$$
$$R_1 = \mathbb{E}_{\theta^\star}\sum_{i=1}^{M_T}\sum_{t=t_i}^{t_{i+1}-1} v_{\theta_i}(\xi_{t+1}) - v_{\theta_i}(\xi_t)$$
$$R_2 = \mathbb{E}_{\theta^\star}\sum_{i=1}^{M_T}\sum_{t=t_i}^{t_{i+1}-1} \mathbb{E}_{\theta_i}[v_{\theta_i}(\xi')|\pi_i,\xi_t] - v_{\theta_i}(\xi_{t+1})$$
$$R_3 = \mathbb{E}_{\theta^\star}\sum_{i=1}^{M_T}\sum_{t=t_i}^{t_{i+1}-1} (r_{\theta_i} - r_{\theta^\star})(\xi_t, \pi_i(\xi_t)).$$

See Appendix D.2 for a more detailed derivation.

Bounding R_0. The first regret term is addressed thanks to the well-known expectation identity (see Russo and Van Roy [2014]), leveraging that conditionally, $\theta^\star \overset{d}{=} \theta_i$.

Lemma 6.7 (Expectation identity). *Suppose $\theta^\star$ and θ_i have the same distribution given a history $\mathcal{H}$. For any $\mathcal{H}$-measurable function f, we have*

$$\mathbb{E}[f(\theta^\star)|\mathcal{H}] = \mathbb{E}[f(\theta_i)|\mathcal{H}].$$

As pointed out in Ouyang et al. [2017], one cannot apply Lemma 6.7 directly to $J_{\pi_i}(\theta_i)$ and $J_{\pi^\star}(\theta^\star)$ because of the measurability issue arising from the lazy-update scheme in (6.7). In line with Ouyang et al. [2017], we overcome this difficulty thanks to the first deterministic termination rule in (6.7). Taking the expectation w.r.t. $\theta^\star$ leads to the following lemma.

Lemma 6.8 (Ouyang et al. [2017], Lemma 3 and 4).

$$\mathbb{E}_{\theta^\star\sim Q} R_0 \le N \cdot \mathbb{E}_{\theta^\star\sim Q} M_T,$$

where M_T is the total number of episodes until time T.

Bounding R_1. Clearly, R_1 involves telescopic sums over each episode i. As a result, it solely depends on the number of policy switches and on the uniform span bound H in Proposition 6.6.

Lemma 6.9.

$$R_1 \leq H \cdot \mathbb{E}M_T.$$

As a result, both R_0 and R_1 reduce to a fine bound over the number of episodes, M_T.

Bounding R_2 and R_3. Finally, the last regret terms are dealing with the model misspecification. That is to say, they depend on the *on-policy* error between the empirical estimate and the true transition model. Formally, Lemma 6.10 and 6.11 show that they scale with

$$\Delta_T = \sum_{i=1}^{M_T} \sum_{t=t_i}^{t_{i+1}-1} \sum_{\text{active arms } k} ||(\hat{p}_{t_i} - p_{\theta^\star})(k, \sigma_k^t, n_k^t)||_1,$$

where $p_\theta(\cdot; k, \sigma, n)$ is the probability distribution of arm k's state under parametrization θ and $\hat{p}_{t_i}$ is its empirical estimate at the beginning of episode i. The core of the proofs thus lies in deriving a high-probability confidence set whose associated on-policy error Δ_T is cumulatively bounded by $\sqrt{T}$. We state the lemmas here and postpone the proofs to Appendix D.2.

Lemma 6.10. *R_2 satisfies the following bound*

$$R_2 \leq 28H \sum_{k=1}^{K} |S_k| \sqrt{NT^{mix}T \log(T^{mix}T)}.$$

Lemma 6.11. *R_3 satisfies the following bound*

$$R_3 \leq 28 \sum_{k=1}^{K} |S_k| \sqrt{NT^{mix}T \log(T^{mix}T)}.$$

We detail the construction and probabilistic argument of the confidence set later in the section.

6.5.2 Bounding the number of episodes

As breifly discussed in Section 6.3, each episode has a random length T_i, and the number of episodes M_T also becomes random. In order to bound R_0 and R_1, we first bound this quantity. As discussed in Osband and Van Roy [2014], the specific structure of our problem due to the MDP formulation of the original POMDP problem allows us to guarantee a tighter bound w.r.t. the number of states than straightforwardly applying the TSDE analysis on the meta-state ξ. In particular, we leverage this

structure to obtain a bound that depends on the number of states through the summation $\sum_{k=1}^{K} |S_k|$ instead of the product $\prod_{k=1}^{K} |S_k|$.

Lemma 6.12. *The number of episodes M_T satisfies the following inequality almost surely*

$$M_T \leq 2\sqrt{(\sum_{k=1}^{K} |S_k|)T^{mix}T \log NT}.$$

Proof. Following Ouyang et al. [2017], we define macro episodes with start times t_{n_i} for a subsequence $\{n_i\} \subset [M_T]$ such that $n_1 = 1$ and

$$t_{n_{i+1}} = \min\{t_k > t_{n_i} | N_{t_k}(\zeta) > 2N_{t_{k-1}}(\zeta) \text{ for some } \zeta\}.$$

Note that the macro episode starts when the second termination criterion happens. Ouyang et al. [2017] prove in their Lemma 1 that

$$M_T \leq \sqrt{2MT}, \tag{6.10}$$

where M is the number of macro episodes. We claim

$$M \leq 2(\sum_{k=1}^{K} |S_k|)T^{\text{mix}} \log NT, \tag{6.11}$$

which prove our lemma when combined with (6.10).

For each $\zeta = (k, s, n) \in Z$, we define

$$M(\zeta) = |\{i \leq M_T | N_{t_i}(\zeta) > 2N_{t_{i-1}}(\zeta)\}|.$$

This means that $N_t(\zeta)$ gets doubled $M(\zeta)$ times out of M_T episodes. It leads to the following inequality

$$2^{M(\zeta)} \leq N_{T+1}(\zeta), \text{ or } M(\zeta) \leq 2 \log N_{T+1}(\zeta).$$

Then we have

$$\begin{aligned} M &\le 1 + \sum_{\zeta \in Z} M(\zeta) \\ &\le 1 + 2 \sum_{\zeta \in Z} \log N_{T+1}(\zeta) \\ &\le 1 + 2(\sum_{k=1}^{K} |S_k|) T^{\text{mix}} \log \frac{\sum_\zeta N_{T+1}(\zeta)}{(\sum_k |S_k|) T^{\text{mix}}} \\ &= 1 + 2(\sum_{k=1}^{K} |S_k|) T^{\text{mix}} \log \frac{NT}{(\sum_k |S_k|) T^{\text{mix}}} \\ &\le 2(\sum_{k=1}^{K} |S_k|) T^{\text{mix}} \log NT, \end{aligned}$$

where we added 1 to account for the initial case $n_1 = 1$ and the third inequality holds due to Jensen's inequality along with the fact that $|Z| \le \sum_k |S_k| \cdot T^{\text{mix}}$. The equality holds because $\sum_\zeta N_{T+1}(\zeta)$ is the total number of active arms until time T. This proves our claim (6.11) and therefore the lemma. □

6.5.3 Confidence set

To bound R_2 and R_3, we construct a confidence set for the system parameters θ. Recall that ζ represents (k, s, n). Suppose at time t, the state of arm k was observed to be s in n rounds ago. Let $p_\theta(\zeta)$ denote the probability distribution of the arm's state if the true system were θ. For an individual probability weight, we write $p_\theta(s'; \zeta) = p_\theta(s'; k, s, n)$ for $s' \in S_k$. Using the $N_t(\zeta)$ samples collected so far, we can also compute an empirical distribution $\hat{p}_t(\zeta)$. We construct a confidence set as a collection of θ such that $p_\theta(\zeta)$ is close to $\hat{p}_t(\zeta)$. Namely in episode i, we define

$$\Theta_i = \{\theta \in \Theta | \forall \zeta \in Z, ||(p_\theta - \hat{p}_{t_i})(\zeta)||_1 \le c_i(\zeta)\},$$

where $c_i(\zeta) = c_i(k, s, n) = \sqrt{\frac{8|S_k| \log 1/\delta}{1 \vee N_{t_i}(\zeta)}}$.

Since Θ_i is $\mathcal{H}_{t_i}$-measurable, Lemma 6.7 provides

$$\mathbb{P}(\theta^\star \notin \Theta_i | \mathcal{H}_{t_i}) = \mathbb{P}(\theta_i \notin \Theta_i | \mathcal{H}_{t_i}).$$

The following lemma bounds this probability.

Lemma 6.13. *For every episode i, we can bound*

$$\mathbb{P}(\theta^\star \notin \Theta_i | \mathcal{H}_{t_i}) = \mathbb{P}(\theta_i \notin \Theta_i | \mathcal{H}_{t_i}) \leq \sum_{k=1}^{K} |S_k| \cdot \delta T^{mix}.$$

Proof. For an episode i, pick $(k, s, n) = \zeta \in Z$ and let $m = N_{t_i}(\zeta)$. If m equals to 0, then $c_i(\zeta) > 1$ and the inequality $||(p_\theta - \hat{p})(\zeta)||_1 \leq c_i(\zeta)$ becomes trivial. Suppose $m > 0$. We first analyze the case $n < T^{\text{mix}}$. Weissman et al. [2003] show that

$$\mathbb{P}(||(p_\theta - \hat{p})(\zeta)||_1 \geq \epsilon) \leq 2^{|S_k|} \exp(-\frac{m\epsilon^2}{2}). \tag{6.12}$$

Setting $\epsilon = c_i(\zeta) = \sqrt{\frac{8|S_k| \log 1/\delta}{n}}$, we get

$$\mathbb{P}(||(p_\theta - \hat{p})(\zeta)||_1 \geq c_i(\zeta)) \leq \delta. \tag{6.13}$$

For the case $n = T^{\text{mix}}$, we want to prove the same probability bound in (6.13) but cannot directly use (6.12) due to aggregation. We can still show a similar bound by using the proof technique by Weissman et al. [2003].

For simplicity, write $p_\theta(\zeta) = p$, $\hat{p}(\zeta) = \hat{p}$, and $c_i(\zeta) = c$. Then it can be easily checked that

$$||p - \hat{p}||_1 = 2 \max_{A \subset S_k} p(A) - \hat{p}(A).$$

Using this and the union bound, we can write

$$\mathbb{P}(||p - \hat{p}||_1 \geq c) \leq \sum_{A \subset S_k} \mathbb{P}(p(A) - \hat{p}(A) \geq \frac{c}{2}). \tag{6.14}$$

By the definition of T^{mix}, we have

$$|p(A) - \mathbb{E}\hat{p}(A)| < \frac{1}{T} < \frac{c}{4}.$$

Then Hoeffding's inequality implies that

$$\begin{aligned} \mathbb{P}(p(A) - \hat{p}(A) \geq \frac{c}{2}) &\leq \mathbb{P}(\mathbb{E}\hat{p}(A) - \hat{p}(A) \geq \frac{c}{4}) \\ &\leq \exp(-\frac{mc^2}{8}). \end{aligned}$$

Plugging this in (6.14), we get

$$\mathbb{P}(||p - \hat{p}||_1 \geq c) \leq 2^{|S_k|} \exp(-\frac{mc^2}{8}) \leq \delta,$$

which shows (6.13) for the case $n = T^{\text{mix}}$.

Since $|Z| \leq \sum_{k=1}^{K} |S_k| \cdot T^{\text{mix}}$, applying the union bound finishes the proof. □

Furthermore, the confidence set satisfies that the cumulative on-policy error Δ_T (see Section 6.5.1) is bounded.

Lemma 6.14. *On the high-probability event $\theta^\star \in \cap_{i \leq M_T} \Theta_i$, we can show*

$$\Delta_T \leq 12\sqrt{NT^{mix}T \log 1/\delta} \sum_{k=1}^{K} |S_k|.$$

The proof of Lemma 6.14 is postponed to Appendix D.2. We want to emphasize that the set Θ_i only appears in the proof and it has nothing to do with running TSDE. For example, we can set an arbitrary value for δ to make the proof works. The main idea of bounding R_2 and R_3 is that the event $\theta^\star, \theta_i \in \Theta_i$ happens with high probability, and if so, then $\pi^\star$ and π_i behave similarly.

6.5.4 Putting everything together

Plugging Lemma 6.8, 6.9, 6.10, and 6.11 into the regret decomposition, we prove our main result.

Theorem 6.1 (Exact regret bound, restated). *The Bayesian regret of TSDE is bounded by*

$$2(H+N)\sqrt{(\sum_{k=1}^{K} |S_k|)T^{mix}T \log NT} + 28(H+1)(\sum_{k=1}^{K} |S_k|)\sqrt{NT^{mix}T \log(T^{mix}T)},$$

where $T^{mix} = (\log_2 T)T^{mix}(\frac{1}{4}) = \mathcal{O}(\log T)$.

6.6 Experiments

We empirically evaluated TSDE (Algorithm 6.1) on simulated data. Following Chapter 5, we chose the Gilbert-Elliott channel model in Figure 5.1 to model each arm. This model assumes binary states and is widely used in communication systems (e.g., see Liu and Zhao [2010]).

For simplicity, we assumed $P^{\text{active}} = P^{\text{passive}}$ and $r_k(s) = s$. This means that the learner's action does not affect the transition matrix and the binary reward equals one if and only if the state is good. We also assumed the initial states of the arms are all good. Each arm has two parameters: p^k_{01} and p^k_{11}. We set the prior to be uniform over a finite set $\{.1, .2, \cdots, .9\}$. Expectations are approximated by the Monte Carlo simulation with size 100 or greater.

We investigated three *index-based policies*: the best fixed arm policy, the myopic policy, and the Whittle index policy. Index-based policies compute an index for each arm only using the samples from this arm and choose the top N arms. Due to their decoupling nature, these policies are computationally efficient. The best fixed arm policy computes the expected reward according to the stationary distribution. The myopic policy maximizes the expected regret of the current round. The Whittle index policy is first introduced by Whittle [1988] and shown to be powerful in this particular setting by Liu and Zhao [2010]. The Whittle index policy is very popular in RMABs as it can efficiently approximate the optimal policy in many different settings. As a remark, all these policies are reduced to the best fixed arm policy in the stationary bandits.

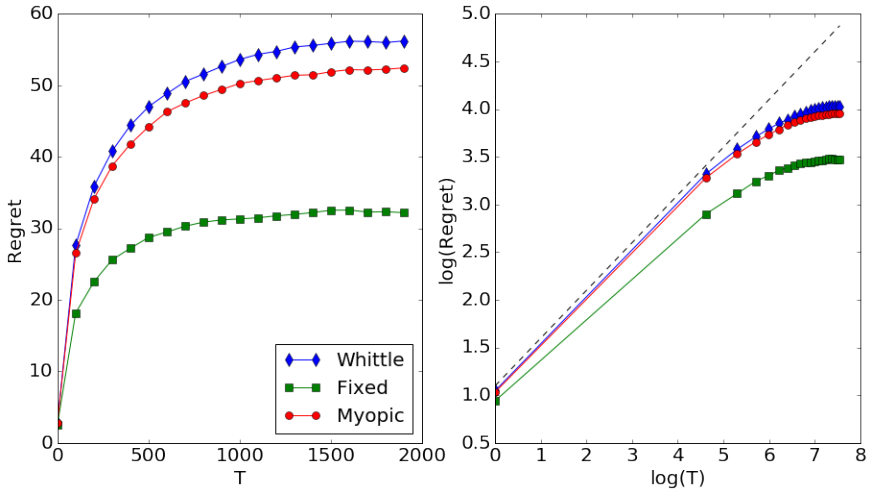

Figure 6.1: Bayesian regrets of TSDE (left) and their log-log plots (right)

We first analyzed the Bayesian regret. Here we used $T = 2000$, $K = 8$, and $N = 3$. The true system $\theta^\star$ was actually drawn from the uniform prior. The average rewards smoothed by the prior, $\mathbb{E}_{\theta^\star \sim Q} J_{\pi^\star}(\theta^\star)$, were 2.05 (fixed), 2.16 (myopic), and 2.17 (Whittle), showing the power of the Whittle index policy. As described in Figure 6.1, the Bayesian regrets were sub-linear regardless of the competitor policy. The log-log plot shows that they are indeed $\tilde{\mathcal{O}}(\sqrt{T})$ as the dotted line has

a slope of 0.5.

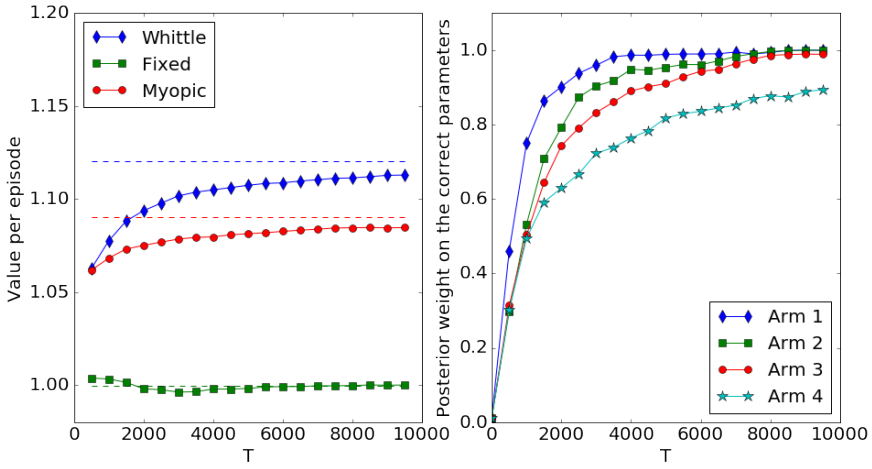

Figure 6.2: Average rewards of TSDE converge to their benchmarks (left); Posterior weights of the true parameters monotonically increase to one (right)

Then we tested the frequentist setting to empirically validate that TSDE still performs well in this setting even though our theory only bounds the Bayesian regret. We chose $T = 10000$, $K = 4$, $N = 2$, and

$$\{(p_{01}^k, p_{11}^k)\}_{k=1,2,3,4} = \{(.3, .7), (.4, .6), (.5, .5), (.6, .4)\}.$$

We again adopted the setting from the previous chapter. This $\theta^\star$ is particularly interesting because each arm has the same stationary distribution of $(.5, .5)$. This means that the best fixed arm policy becomes indifferent among the arms. The average rewards, $J_{\pi^\star}(\theta^\star)$, were 1.00 (fixed), 1.09 (myopic), and 1.12 (Whittle), again justifying the power of Whittle index policy. On the left plot of Figure 6.2, three horizontal dotted lines represent $J_{\pi^\star}(\theta^\star)$ for each of the competitors. The solid lines show the time-averaged cumulative rewards, $\frac{1}{t}\mathbb{E}_{\theta^\star} \sum_{\tau=1}^{t} r_{\theta^\star}(\xi_\tau, A_\tau)$. Every solid line converged to the dotted line. The right figure plots the posterior probability of the true parameters using the Whittle index policy. For all arms, these probabilities monotonically increased to one, illustrating that TSDE were learning $\theta^\star$ properly. From this, we can assert that TSDE still performs reasonably well at least when the true parameters lie on the support of the prior.

CHAPTER 7

Conclusion

This manuscript discussed several interesting topics in online learning and explore new directions. This chapter briefly summarizes the previous contents and presents future directions.

In the full information online boosting (Chapter 2 and Chapter 3), one optimal algorithm and one adaptive algorithm are proposed. The optimal algorithm requires the minimal number of weak learners for a desired accuracy, and the adaptive algorithm is computationally more feasible and shows competitive results in experiments. The algorithms are quite flexible in their choice of weak learners in that various types of learners can be combined to produce a strong learner. MLR boosting algorithms (Chapter 3) even allow the weak learners to have different prediction formats. Two prediction problems (multi-class classification and multi-label ranking) are considered, and designing boosting algorithms in other settings will be valuable. Even in the MLR setting, developing boosting algorithms for other ranking losses remain open (The rank loss is a strong candidate, but it is hard to claim that it is the canonical ranking loss). Another interesting open question is whether there is an *optimal* adaptive boosting algorithm.

Chapter 4 added one more flexibility, namely partial feedback, to online boosting framework. With randomized prediction and unbiased estimate of the loss, the full information algorithms from Chapter 2 and Chapter 3 are naturally extended to this setting. The unbiased estimate successfully resolved the issue of updating weak learners under the constraint of limited information. Again, an optimal and an adaptive algorithms are presented for two different prediction problems, and their performance guarantees matched those of the full information counterparts. The cost of partial feedback is reflected to increased sample complexities.

Chapter 5 and Chapter 6 tackled non-stationarity in MABs. Restless bandit problems are instances of non-stationary MABs assuming the state of the arms evolves according to a Markov process. The episodic case (Chapter 5) and the non-episodic case (Chapter 6) are discussed. Thompson sampling algorithm and its slight modification are analyzed, and the Bayesian regret can be theoretically bounded as $\tilde{\mathcal{O}}(\sqrt{T})$, which naturally extends the results in the stationary MABs.

One primary strength of this analysis is that the bound applies to arbitrary deterministic competitor policy mappings, which include the optimal policy and many other practical policies. This allows the researchers to obtain computationally tractable algorithm with a reliable theoretical guarantee. In the special case where the prior has a discrete support and the benchmark is the optimal policy, the results extend to the frequentist regret, which is also supported by empirical results. One interesting open question is to bound the frequentist regret of Thompson sampling in the general case. In stationary MABs, it has been shown that Thompson sampling enjoys the frequentist regret bound of $\tilde{\mathcal{O}}(\sqrt{T})$ with additional assumptions [Lattimore and Szepesvári, 2018, Chp. 36]. Extending this to the restless bandit setting will be an interesting and very challenging problem.

APPENDIX A

Details for Online Multiclass Boosting

A.1 Link between batch and online weak learning conditions

Let us begin the section by introducing the weak learning condition in the batch setting. Mukherjee and Schapire [2013] have identified necessary and sufficient condition for boostability. We will focus on a sufficient condition due to reasons of computational tractability. In the batch setting, the entire training set is revealed. Let $D := \{(\mathbf{x}_t, y_t) \mid t = 1, \cdots, T\}$ be the training set and define a family of cost matrices:

$$\mathcal{C}^{eor} := \{\mathbf{C} \in \mathbb{R}^{T \times k} \mid \forall t,\ \mathbf{C}[t, y_t] = \min_{l \in [k]} \mathbf{C}[t, l]\}.$$

The superscript "eor" stands for "edge-over-random." We warn the readers not to confuse $\mathcal{C}^{eor}$ with $\mathcal{C}_1^{eor}$. They both impose similar row constraints, but the matrices in these sets have different dimensions: $T \times k$ and $k \times k$ respectively. $\mathcal{C}_1^{eor}$ also has additional an normalization constraint. Note that $\mathcal{C}^{eor}$ provides one cost vector for an instance whereas $\mathcal{C}_1^{eor}$ provides a matrix. This is necessary because if an adversary passes only a vector to an online learner, then the learner can simply make the prediction which minimizes the cost. Furthermore, in the online boosting setting, the booster does not know the true label when it computes a cost matrix.

The authors prove that if a weak learning space $\mathcal{H}$ satisfies the condition described in Definition A.1, then it is boostable, which means there exists a convex linear combination of hypotheses in $\mathcal{H}$ that perfectly classifies D.

Definition A.1. (Batch setting weak learning condition, Mukherjee and Schapire [2013]) *Suppose D is fixed and $\mathcal{C}^{eor}$ is defined as above. A weak learning space $\mathcal{H}$ is said to satisfy weak*

learning condition $(\mathcal{C}^{eor}, \boldsymbol{U}_\gamma)$ *if* $\forall \boldsymbol{C} \in \mathcal{C}^{eor}$*, one can find a weak hypothesis* $h \in \mathcal{H}$ *such that*

$$\sum_{t=1}^{T} \boldsymbol{C}[t, h(\boldsymbol{x}_t)] \leq \boldsymbol{C} \bullet \boldsymbol{U}'_\gamma. \tag{A.1}$$

Now we present how our online weak learning condition (Definition 2.1) is naturally derived from the batch setting counterpart (Definition A.1). We extend the arguments of Beygelzimer et al. [2015]. The batch setting condition (A.1) can be interpreted as making the following two implicit assumptions:

1. (Richness condition) For any $\mathbf{C} \in \mathcal{C}^{eor}$, there is some hypothesis $h \in \mathcal{H}$ such that

$$\sum_{t=1}^{T} \mathbf{C}[t, h(\mathbf{x}_t)] \leq \mathbf{C} \bullet \mathbf{U}'_\gamma.$$

2. (Agnostic learnability) For any $\mathbf{C} \in \mathcal{C}^{eor}$ and $\epsilon \in (0, 1)$, there is an algorithm which can compute a nearly optimal hypothesis $h \in \mathcal{H}$, i.e.

$$\sum_{t=1}^{T} \mathbf{C}[t, h(\mathbf{x}_t)] \leq \inf_{h' \in \mathcal{H}} \sum_{t=1}^{T} \mathbf{C}[t, h'(\mathbf{x}_t)] + \epsilon T.$$

For the online setting, we will keep the richness assumption with $\mathbf{C}$ being the matrix consisting of rows of $w_t \mathbf{C}_t[y_t]$, and the data being drawn by a fixed adversary. That is to say, it is the online richness condition that imposes a restriction on adversary because the condition cannot be met by any $\mathcal{H}$ with fully adaptive adversary. For example, suppose an adversary draws samples uniformly at random from the set $\{(\mathbf{x}, 1), \cdots, (\mathbf{x}, k)\}$ for some fixed $\mathbf{x} \in \mathcal{X}$. There does not exist weak learning space $\mathcal{H}$ that satisfies the online richness condition with this adversary. The agnostic learnability assumption is also replaced by online agnostic learnability assumption. We present online versions of the above two assumptions:

1′. (Online richness condition) For any sample length T, any sequence of labeled examples $\{(\mathbf{x}_t, y_t) \mid t = 1, \cdots, T\}$ generated by a fixed adversary, and any series of pairs of weight and cost matrix $\{(w_t, \mathbf{C}_t) \in [0, 1] \times \mathcal{C}_1^{eor} \mid t = 1, \cdots, T\}$, there is some hypothesis $h \in \mathcal{H}$ such that

$$\sum_{t=1}^{T} w_t \mathbf{C}_t[y_t, h(\mathbf{x}_t)] \leq \mathbf{C} \bullet \mathbf{U}'_\gamma, \tag{A.2}$$

where $\mathbf{C} \in \mathbb{R}^{T \times k}$ consists of rows of $w_t \mathbf{C}_t[y_t]$.

2′. (Online agnostic learnability) For any sample length T, $\delta \in (0,1)$, and for any adaptively chosen series of pairs of weight and cost matrix $\{(w_t, \mathbf{C}_t) \in [0,1] \times \mathcal{C}_1^{eor} \mid t = 1, \cdots, T\}$, there is an online algorithm which can generate predictions $\hat{y}_t$ such that with probability $1-\delta$,

$$\sum_{t=1}^{T} w_t \mathbf{C}_t[y_t, \hat{y}_t] \leq \inf_{h \in \mathcal{H}} \sum_{t=1}^{T} w_t \mathbf{C}_t[y_t, h(\mathbf{x}_t)] + R_\delta(T), \tag{A.3}$$

where $R_\delta : \mathbb{N} \to \mathbb{R}$ is a sublinear regret.

Daniely et al. [2011] extensively investigates agnostic learnability in online multiclass problems by introducing the following generalized Littlestone dimension (Littlestone [1988]) of a hypothesis family $\mathcal{H}$. Consider a binary rooted tree RT whose internal nodes are labeled by elements from $\mathcal{X}$ and whose edges are labeled by elements from $[k]$ such that two edges from a same parent have different labels. The tree RT is *shattered* by $\mathcal{H}$ if, for every path from root to leaf which traverses the nodes $\mathbf{x}_1, \cdots, \mathbf{x}_k$, there is a hypothesis $h \in \mathcal{H}$ such that $h(\mathbf{x}_i)$ corresponds to the label of the edge from $\mathbf{x}_i$ to $\mathbf{x}_{i+1}$. The *Littlestone dimension* of $\mathcal{H}$ is the maximal depth of complete binary tree that is shattered by $\mathcal{H}$ (or ∞ if one can build a arbitrarily deep shattered tree). The authors prove that an optimal online algorithm has a sublinear regret under the expected (w.r.t. the randomness of the algorithm) 0-1 loss if Littlestone dimension of $\mathcal{H}$ is finite.

Similarly we prove in Lemma A.2 that the condition (A.3) is satisfied if $\mathcal{H}$ has a finite Littlestone dimension. We need to slightly modify their result in two ways. One is to replace expectation by probabilistic argument, and the other is to replace 0-1 loss by our cost matrix framework. Both questions can be resolved by replacing an auxiliary lemma used by Daniely et al. [2011] without changing the main structure.

Lemma A.2. *Suppose a weak learning space $\mathcal{H}$ has a finite Littlestone dimension d and an adversary chooses examples in fully adaptive manner. For any sample length T and for any adaptively chosen series of pairs of weight and cost matrix $\{(w_t, \boldsymbol{C}_t) \in [0,1] \times \mathcal{C}_1^{eor} \mid t = 1, \cdots, T\}$, with probability $1-\delta$, the online agnostic learnability condition (A.3) is satisfied with following sublinear regret*

$$R_\delta(T) = \sqrt{(Td \ln Tk)/2} + \sqrt{(T \ln 1/\delta)/2}.$$

Proof. We first introduce an online algorithm with experts. Suppose we have a fixed pool of experts of size N. We keep our cost matrix framework. Each expert f^i would suffer cumulative cost $C_T^i := \sum_{t=1}^{T} w_t \mathbf{C}_t[y_t, f^i(\mathbf{x}_t)]$. At each iteration, an online algorithm chooses to follow one expert and incurs a cost $w_t \mathbf{C}_t[y_t, \hat{y}_t]$, and its goal is to perform as well as the best expert. That is to say, the algorithm wants to keep its cumulative cost $\sum_{t=1}^{T} w_t \mathbf{C}_t[y_t, \hat{y}_t]$ not too much larger than $\min_{i \in [N]} C_T^i$.

Algorithm A.1 Learning with Expert Advice (LEA)

Input T: time horizon, N: number of experts
Set $\eta = \sqrt{(8 \ln N)/T}$
Set $C_0^i = 0$ for all i
for $t = 1, \cdots, T$ **do**
    Receive example $\mathbf{x}_t$
    Receive expert advices $(f_t^1, \cdots, f_t^N) \in [k]^N$
    Predict $\hat{y}_t = f_t^i$ with probability proportional to $\exp(-\eta C_{t-1}^i)$
    Receive true label y_t
    Update $C_t^i = C_{t-1}^i + w_t \mathbf{C}_t[y_t, f_t^i]$ for all i
end for

This learning framework is called *weighted majority algorithm* and is thoroughly investigated by several researchers (e.g., Littlestone and Warmuth [1989] and Vovk [1990]). We will specifically use Algorithm A.1 (LEA), which is shown to achieve a sublinear regret $\sqrt{(T \ln N)/2} + \sqrt{(T \ln 1/\delta)/2}$ with probability $1 - \delta$ (cf. Cesa-Bianchi and Lugosi [2006, Corollary 4.2]). The authors require the loss to be bounded, which is also satisfied in our cost matrix framework. Readers might raise a question that our loss function changes for each iteration, but the proof still works as long as it is bounded. Interested readers might refer Hazan et al. [2016, Section 1.3.3].

To apply this result in our case, we need to construct a finite set of experts whose best performance is as good as that of hypotheses in $\mathcal{H}$. In fact, in the proof of Daniely et al. [2011, Theorem 25], the authors construct a set E of size $N \leq (Tk)^d$ such that for every hypothesis $h \in \mathcal{H}$, there is an expert $f \in E$ which coincides with h subject to the given examples $\mathbf{x}_1, \cdots, \mathbf{x}_T$.

Applying the LEA result on E shows that with probability $1 - \delta$, the regret is bounded above by $\sqrt{(Td \ln Tk)/2} + \sqrt{(T \ln 1/\delta)/2}$, which concludes the proof. □

One remark is that the proof of Lemma A.2 only uses the boundedness condition of $\mathcal{C}_1^{eor}$.

Now we are ready to demonstrate that our online weak learning condition is indeed naturally derived from the batch setting counterpart. The following Theorem shows that two conditions (A.2) and (A.3) directly imply the online weak learning condition (2.1). In other words, if the weak learning space $\mathcal{H}$ accompanied by an adversary is rich enough to contain a hypothesis that slightly outperforms a random guess and has a reasonably small dimension, then we can find an excess loss S that satisfies (2.1). This is a generalization of Beygelzimer et al. [2015, Lemma 2]. Note that we impose an additional assumption that $w_t \geq m > 0$, $\forall t$. In case the learner encounters zero weight, it can simply ignore the instance, and the above assumption is not too artificial.

Theorem A.3. (Link between batch and online weak learning conditions) *Suppose a pair of weak learning space $\mathcal{H}$ and an adversary satisfies online richness assumption (A.2) with edge*

2γ *and online agnostic learnability assumption (A.3) with mistake probability* δ *and sublinear regret* $R_\delta(\cdot)$*. Additionally we assume there exists a positive constant* m *that satisfies* $w_t \geq m$, $\forall t$*. Then the online learning algorithm satisfies the online weak learning condition (2.1), with mistake probability* δ*, edge* γ*, and excess loss* $S = \max_T(R_\delta(T) - \frac{\gamma m T}{k})$.

Proof. Fix $\delta \in (0,1)$ and a series of pairs of weight and cost matrix $\{(w_t, \mathbf{C}_t) \in [0,1] \times \mathcal{C}_1^{eor} \mid t = 1, \cdots, T\}$, and let $\mathbf{C} \in \mathbb{R}^{T \times k}$ consist of rows of $w_t \mathbf{C}_t[y_t]$. First note that by sublinearity of $R_\delta(\cdot)$, S is finite. According to (A.3), the online learning algorithm can generate predictions $\hat{y}_t$ such that, with probability $1 - \delta$,

$$\sum_{t=1}^{T} w_t \mathbf{C}_t[y_t, \hat{y}_t] \leq \mathbf{C} \bullet \mathbf{U}'_{2\gamma} + R_\delta(T).$$

Thus it suffices to show that

$$\mathbf{C} \bullet \mathbf{U}'_{2\gamma} + R_\delta(T) \leq \mathbf{C} \bullet \mathbf{U}'_{\gamma} + S. \tag{A.4}$$

Since the correct label gets zero cost and the row $\mathbf{C}[r]$ has ℓ_1 norm w_t, we have

$$\mathbf{C} \bullet \mathbf{U}'_{\gamma} = \frac{1-\gamma}{k} ||\mathbf{C}||_1 = \frac{1-\gamma}{k} \sum_{t=1}^{T} w_t.$$

By plugging this in (A.4), we get

$$\mathbf{C} \bullet \mathbf{U}'_{2\gamma} - \mathbf{C} \bullet \mathbf{U}'_{\gamma} + R_\delta(T) = -\frac{\gamma}{k} \sum_{t=1}^{T} w_t + R_\delta(T) \leq -\frac{\gamma}{k} mT + R_\delta(T) \leq S.$$

The first inequality holds because $w_t \geq m$, and the second inequality holds by definition of S, which completes the proof. □

Lemma A.2 and Theorem A.3 suggest an implicit relation between δ and S in (2.1). If we want probabilistically stronger weak learning condition, $R_\delta(T)$ in Lemma A.2 gets bigger, which results in larger $S = \max_T(R_\delta(T) - \frac{\gamma T}{k})$.

A.2 Detailed discussion of OnlineMBBM

A.2.1 Proof of Theorem 2.2

Proof. For ease of notation, we will assume the edge is equal to γ and the true label is r unless otherwise specified. That is to say, $\mathbf{u}$ stands for $\mathbf{u}_\gamma^r$ and ϕ_i for ϕ_i^r. By rewriting (2.3),

$$\begin{aligned}\phi_{N-i+1}(\mathbf{s}_t^{i-1}) &= \mathbb{E}_{l\sim\mathbf{u}}\phi_{N-i}(\mathbf{s}_t^{i-1}+\mathbf{e}_l)\\ &= \mathbf{C}_t^i[r]\bullet\mathbf{u}\\ &= \mathbf{C}_t^i[r]\bullet(\mathbf{u}-\mathbf{e}_{l_t^i})+\phi_{N-i}(\mathbf{s}_t^i),\end{aligned}$$

where $\mathbf{C}_t^i$ is defined in (2.4). The last equation holds due to the relation $\mathbf{s}_t^i=\mathbf{s}_t^{i-1}+\mathbf{e}_{l_t^i}$. Also note that $||\mathbf{u}||_1=||\mathbf{e}_r||_1=1$, and thus subtracting common numbers from each component of $\mathbf{C}_t^i[r]$ does not affect the dot product term. Therefore, by introducing normalized cost matrix $\mathbf{D}_t^i$ as in (2.5) and $\mathbf{w}^i[t]$ as in Algorithm 2.1, we may write

$$\begin{aligned}\phi_{N-i+1}^{y_t}(\mathbf{s}_t^{i-1}) &= \mathbf{w}^i[t]\mathbf{D}_t^i[y_t]\bullet(\mathbf{u}_\gamma^{y_t}-\mathbf{e}_{l_t^i})+\phi_{N-i}^{y_t}(\mathbf{s}_t^i)\\ &= \mathbf{w}^i[t]\mathbf{D}_t^i[y_t]\bullet\mathbf{u}_\gamma^{y_t}-\mathbf{w}^i[t]\mathbf{D}_t^i[y_t,l_t^i]+\phi_{N-i}^{y_t}(\mathbf{s}_t^i)\\ &= \mathbf{w}^i[t]\frac{1-\gamma}{k}-\mathbf{w}^i[t]\mathbf{D}_t^i[y_t,l_t^i]+\phi_{N-i}^{y_t}(\mathbf{s}_t^i).\end{aligned}\tag{A.5}$$

The last equality holds because $\mathbf{D}_t^i$ is normalized and $\mathbf{D}_t^i[y_t,y_t]=0$. If $\mathbf{D}_t^i[y_t]$ is a zero vector, then by definition $\mathbf{w}^i[t]=0$, and the equality still holds. Then by summing (A.5) over t, we get

$$\sum_{t=1}^T\phi_{N-i+1}^{y_t}(\mathbf{s}_t^{i-1})=\frac{1-\gamma}{k}||\mathbf{w}^i||_1-\sum_{t=1}^T\mathbf{w}^i[t]\mathbf{D}_t^i[y_t,l_t^i]+\sum_{t=1}^T\phi_{N-i}^{y_t}(\mathbf{s}_t^i).$$

By online weak learning condition, we have with probability $1-\delta$, (recall that w^{i*} estimates $||\mathbf{w}^i||_\infty$)

$$\sum_{t=1}^T\frac{\mathbf{w}^i[t]}{w^{i*}}\mathbf{D}_t^i[y_t,l_t^i]\le\frac{1-\gamma}{k}\frac{||\mathbf{w}^i||_1}{w^{i*}}+S.$$

From this, we can argue that

$$\sum_{t=1}^T\phi_{N-i+1}^{y_t}(\mathbf{s}_t^{i-1})+Sw^{i*}\ge\sum_{t=1}^T\phi_{N-i}^{y_t}(\mathbf{s}_t^i).$$

Since the above inequality holds for any i, summing over i gives

$$\sum_{t=1}^{T} \phi_N^{y_t}(\mathbf{0}) + S \sum_{i=1}^{N} w^{i*} \geq \sum_{t=1}^{T} \phi_0^{y_t}(\mathbf{s}_t^N),$$

which holds with probability $1 - N\delta$ by union bound. By symmetry, $\phi_N^{y_t}(\mathbf{0}) = \phi_N^1(\mathbf{0})$ regardless of the true label y_t, and by definition of potential function (2.3), $\phi_0^{y_t}(\mathbf{s}_t^N) = L^{y_t}(\mathbf{s}_t^N)$, which completes the proof.

□

A.2.2 Bounding the terms in general bound under 0-1 loss

Even though OnlineMBBM has a promising theoretical justification, it would be infeasible if the computation of potential functions takes too long or if the behavior of asymptotic error rate $\phi_N^1(\mathbf{0})$ is too complicated to be approximated. Fortunately for the 0-1 loss, we can get a computationally tractable algorithm with vanishing error rate. The use of potential functions in binary boosting setup is thoroughly discussed by Schapire [2001]. In binary setting under 0-1 loss, potential function has a closed form which dramatically reduces the computational complexity. Unfortunately, the multiclass version does not have a closed form, but Mukherjee and Schapire [2013] introduce a heuristic to compute it in reasonable time:

$$\phi_i^r(\mathbf{s}) = 1 - \sum_{(x_1,\cdots,x_k)\in A} \binom{i}{x_1,\cdots,x_k} \prod_{l=1}^{k} u_l^{x_l}, \tag{A.6}$$

where $A := \{(x_1, \cdots x_k) \in \mathbb{Z}^k \mid x_1 + \cdots x_k = i,\ \forall l \ : \ x_l \geq 0, x_l + \mathbf{s}[l] < x_r + \mathbf{s}[r]\}$, and $\mathbf{u}_\gamma^r = (u_1, \cdots, u_k)$. By using dynamic programming, the RHS of (A.6) can be computed in polynomial time in i, k, and $||\mathbf{s}||_1$. In our setting where the number of learners is fixed to be N, the computation can be done in polynomial time in k and N because $||\mathbf{s}||_1$ is bounded by N. To the best of our knowledge, there is no way to compute the potential function in polynomial time if we start from necessary and sufficient weak learning condition (the algorithm given by Mukherjee and Schapire [2013] takes exponential time in the number of learners), and this is the main reason that we use the sufficient condition. Recall from (2.6) that $\phi_N^1(\mathbf{0})$ plays a role of asymptotic error rate and the second term determines the sample complexity. The following two lemmas provide bounds for both terms.

By applying the Hoeffding's inequality, we can prove in Lemma A.4 that $\phi_N^1(\mathbf{0})$ vanishes exponentially fast as N grows. That is to say, to get a satisfactory accuracy, we do not need too

many learners. We also note that we can decide N before the learning process begins, which is logically plausible.

Lemma A.4. *Under the same setting as in Theorem 2.2 but with the particular choice of 0-1 loss, we may bound $\phi_N^1(\boldsymbol{0})$ as follows:*

$$\phi_N^1(\boldsymbol{0}) \leq (k-1)\exp(-\frac{\gamma^2 N}{2}). \tag{A.7}$$

Proof. We reinterpret $\phi_N^1(\mathbf{0})$ in (A.6). Imagine that we draw numbers N times from $[k]$ where the probability that a number i is drawn is $\mathbf{u}_\gamma^1[i]$. That is to say, 1 has highest probability of $\frac{1-\gamma}{k}+\gamma$, and other numbers have equal probability of $\frac{1-\gamma}{k}$. Then $\phi_N^1(\mathbf{0})$ can be interpreted as a probability that the number that is drawn for the most time out of N draws is not 1. Let A_i denote the event that the number i gets more votes than the number 1. Then we have by union bound,

$$\begin{aligned}\phi_N^1(\mathbf{0}) &= \mathbb{P}(A_2 \cup \cdots \cup A_k) \\ &\leq \sum_{l=2}^{k} \mathbb{P}(A_i) \\ &= (k-1)\mathbb{P}(A_2)\end{aligned} \tag{A.8}$$

The last equality holds by symmetry. To compute $\mathbb{P}(A_2)$, imagine that we draw 1 with probability $\frac{1-\gamma}{k}+\gamma$, -1 with probability $\frac{1-\gamma}{k}$, and 0 otherwise. $\mathbb{P}(A_2)$ is equal to the probability that after independent N draws, the summation of N i.i.d. random numbers is non-positive. Thus by the Hoeffding's inequality, we get

$$\mathbb{P}(A_2) \leq \exp(-\frac{\gamma^2 N}{2}) \tag{A.9}$$

Combining (A.8) and (A.9) completes the proof. □

Now we have fixed N based on the desired asymptotic accuracy. Since 0-1 loss is bounded in $[0,1]$, so are potential functions. Then by definition of weights (cf. Algorithm 2.1), $||\mathbf{w}^i||_\infty$ is trivially bounded above by k, which means we can use $w^{i*} = k \ \forall i$. Thus the second term of (2.6) is bounded above by kNS, which is valid. However, Lemma A.5 allows a tighter bound.

Lemma A.5. *Under the same setting as in Theorem 2.2 but with the particular choice of 0-1 loss and an additional constraint of $\gamma < \frac{1}{2}$, we may bound $||\boldsymbol{w}^i||_\infty$ by*

$$||\boldsymbol{w}^i||_\infty \leq \frac{ck^{5/2}}{\sqrt{N-i}}, \tag{A.10}$$

where c is a universal constant that can be determined before the algorithm begins.

Proof. We will start by providing a bound on $\phi_m^r(\mathbf{s}+\mathbf{e}_l) - \phi_m^r(\mathbf{s}+\mathbf{e}_r)$. First note that it is non-negative as potential functions are proper. Again by using random draw framework as in the proof of Lemma A.4 (now r has the largest probability to be drawn), this value corresponds to the probability that after m draws, the number r wins the majority votes if the count starts from $\mathbf{s}+\mathbf{e}_r$ but loses if the count starts from $\mathbf{s}+\mathbf{e}_l$. Let $X_1, \cdots, X_k$ denote the number of draws of each number out of m draws and define the events $A_l := \{(X_r + \mathbf{s}[r]) - (X_l + \mathbf{s}[l]) \in \{0,1\}\}$. Then it can be checked that

$$\begin{aligned}
&\phi_m^r(\mathbf{s}+\mathbf{e}_l) - \phi_m^r(\mathbf{s}+\mathbf{e}_r) \\
&= \mathbb{P}(\exists l' \text{ s.t. } X_{l'} + \mathbf{s}[l'] + \mathbf{e}_l[l'] \geq X_r + \mathbf{s}[r]) - \mathbb{P}(\exists l' \text{ s.t. } X_{l'} + \mathbf{s}[l'] \geq X_r + \mathbf{s}[r] + 1) \\
&\leq \mathbb{P}(\exists l' \text{ s.t. } X_{l'} + \mathbf{s}[l'] + \mathbf{e}_l[l'] \geq X_r + \mathbf{s}[r] \text{ and } \forall l',\ X_r + \mathbf{s}[r] \geq X_{l'} + \mathbf{s}[l']) \\
&\leq \mathbb{P}(\exists l' \text{ s.t. } X_{l'} + \mathbf{s}[l'] + \mathbf{e}_l[l'] \geq X_r + \mathbf{s}[r] \geq X_{l'} + \mathbf{s}[l']) \\
&= \mathbb{P}(\bigcup_{l \neq r} A_l) \leq \sum_{l \neq r} \mathbb{P}(A_l).
\end{aligned} \tag{A.11}$$

The first inequality holds by $\mathbb{P}(A) - \mathbb{P}(B) \leq \mathbb{P}(A - B)$. Individual probabilities can be written as

$$\begin{aligned}
\mathbb{P}(A_l) &= \mathbb{P}(X_r - X_l = \mathbf{s}[l] - \mathbf{s}[r]) + \mathbb{P}(X_r - X_l = \mathbf{s}[l] - \mathbf{s}[r] + 1) \\
&\leq 2 \max_n \mathbb{P}(X_r - X_l = n).
\end{aligned} \tag{A.12}$$

We can prove by applying the Berry-Esseen theorem that the last probability is $O(\frac{1}{\sqrt{m}})$. Let $Y_1, \cdots, Y_m$ be a sequence of i.i.d. random variables such that $Y_j \in \{-1, 0, 1\}$ and

$$\begin{aligned}
\mathbb{P}(Y_j = 1) &= \frac{1-\gamma}{k} + \gamma, \\
\mathbb{P}(Y_j = -1) &= \frac{1-\gamma}{k}.
\end{aligned}$$

Note that $\mathbb{E}Y_j = \gamma$ and $Var(Y_j) = \frac{2(1-\gamma)}{k} + \gamma(1-\gamma) =: \sigma^2$. It can be easily checked that $Y := \sum_{j=1}^m Y_j$ has same distribution with $X_r - X_l$. Now we approximate Y by a Gaussian random variable $W \sim N(m\gamma, m\sigma^2)$. Let F_W and F_Y denote CDF of W and Y, respectively, and let f denote the density of W. First note that

$$\begin{aligned}
|\mathbb{P}(Y = n) - \int_{n-1}^{n} f(w)dw| &= |(F_Y(n) - F_Y(n-1)) - (F_W(n) - F_W(n-1))| \\
&\leq |F_Y(n) - F_W(n)| + |F_Y(n-1) - F_W(n-1)|.
\end{aligned}$$

We can apply the Berry-Esseen theorem to the last CDF differences, which provides

$$|\mathbb{P}(Y = n) - \int_{n-1}^{n} f(w)dw| \leq \frac{2C\rho}{\sigma^3\sqrt{m}}, \tag{A.13}$$

where C is the universal constant that appears in Berry-Esseen and $\rho := \mathbb{E}|Y_j - \gamma|^3$. As Y_j is a bounded random variable, we have

$$\rho = \mathbb{E}|Y_j - \gamma|^3 \leq (1+\gamma)\mathbb{E}|Y_j - \gamma|^2 = (1+\gamma)\sigma^2 \leq 2\sigma^2.$$

Plugging this in (A.13) gives

$$|\mathbb{P}(Y = n) - \int_{n-1}^{n} f(w)dw| \leq \frac{4C}{\sigma\sqrt{m}}$$

By simple algebra, we can deduce

$$\begin{aligned}\mathbb{P}(Y = n) &\leq \int_{n-1}^{n} f(w)dw + \frac{4C}{\sigma\sqrt{m}} \\ &\leq \sup_{w\in\mathbb{R}} f(w) + \frac{4C}{\sigma\sqrt{m}} \\ &= \frac{1}{\sqrt{2\pi m}\sigma} + \frac{4C}{\sigma\sqrt{m}}.\end{aligned} \tag{A.14}$$

Using the fact that $\gamma < \frac{1}{2}$, we can show

$$\sigma^2 = \frac{2(1-\gamma)}{k} + \gamma(1-\gamma) \geq \frac{1}{k}$$

Plugging this in (A.14) gives

$$\mathbb{P}(Y = n) \leq \frac{1}{\sigma\sqrt{m}}(\frac{1}{\sqrt{2\pi}} + 4C) \leq C'\sqrt{\frac{k}{m}}, \tag{A.15}$$

where $C' = \frac{1}{\sqrt{2\pi}} + 4C$. By combining (A.11), (A.12), (A.15), and the fact that Y and $X_r - X_l$ have same distribution, we prove

$$\phi_m^r(\mathbf{s} + \mathbf{e}_l) - \phi_m^r(\mathbf{s} + \mathbf{e}_r) \leq 2C'k\sqrt{\frac{k}{m}}. \tag{A.16}$$

The proof is complete by observing that $\mathbf{w}^i[t] = \sum_{l=1}^{k}[\phi_{N-i}^{y_t}(\mathbf{s}_t^{i-1} + \mathbf{e}_l) - \phi_{N-i}^{y_t}(\mathbf{s}_t^{i-1} + \mathbf{e}_{y_t})]$. □

Remark. *By summing (A.10) over i, we can bound the second term of (2.6) by $O(k^{5/2}\sqrt{N})S$. Comparing this to the aforementioned bound kNS, Lemma A.5 reduces the dependency on N, but as a tradeoff the dependency on k is increased. The optimal bound for this term remains open, but in the case that the number of classes k is fixed to be moderate, Lemma A.5 provides a better bound.*

Corollary 2.3 is a simple consequence of plugging Lemma A.4 and A.5 to Theorem 2.2.

A.2.3 Proof of lower bounds and discussion of gap

We begin by proving Theorem 2.4.

Proof. At time t, an adversary draws a label y_t uniformly at random from $[k]$, and the weak learners independently make predictions with respect to the probability distribution $\mathbf{p}_t \in \Delta[k]$. This can be achieved if the adversary draws $\mathbf{x}_t \in \mathbb{R}^N$ where $\mathbf{x}_t[1], \cdots, \mathbf{x}_t[N]|y_t$'s are conditionally independent with conditional distribution of $\mathbf{p}_t$ and WL^i predicts $\mathbf{x}_t[i]$. The booster can only make a final decision by weighted majority votes of N weak learners. We will manipulate $\mathbf{p}_t$ in such a way that weak learners satisfy (2.1), but the booster's performance is close to that of Online MBBM.

First we note that since $\mathbf{C}_t[y_t, \hat{y}_t]$ used in (2.1) is bounded in $[0, 1]$, the Azuma-Hoeffding inequality implies that if a weak learner makes prediction $\hat{y}_t$ according to the probability distribution $\mathbf{p}_t$ at time t, then with probability $1 - \delta$, we have

$$\begin{aligned}\sum_{t=1}^{T} w_t \mathbf{C}_t[y_t, \hat{y}_t] &\leq \sum_{t=1}^{T} w_t \mathbf{C}_t[y_t] \bullet \mathbf{p}_t + \sqrt{2||\mathbf{w}||_2^2 \ln(\frac{1}{\delta})} \\ &\leq \sum_{t=1}^{T} w_t \mathbf{C}_t[y_t] \bullet \mathbf{p}_t + \frac{\gamma||\mathbf{w}||_2^2}{k} + \frac{k \ln(\frac{1}{\delta})}{2\gamma} \\ &\leq \sum_{t=1}^{T} w_t \mathbf{C}_t[y_t] \bullet \mathbf{p}_t + \frac{\gamma||\mathbf{w}||_1}{k} + \frac{k \ln(\frac{1}{\delta})}{2\gamma},\end{aligned} \tag{A.17}$$

where the second inequality holds by arithmetic mean and geometric mean relation and the last inequality holds due to $w_t \in [0, 1]$.

We start from providing a lower bound on the number of weak learners. Let $\mathbf{p}_t = \mathbf{u}_{2\gamma}^{y_t}$ for all t. This can be done by the constraint $\gamma < \frac{1}{4}$. Then the last line of (A.17) becomes

$$\sum_{t=1}^{T} w_t \mathbf{C}_t[y_t] \bullet \mathbf{u}_{2\gamma}^{y_t} + \frac{\gamma||\mathbf{w}||_1}{k} + \frac{k \ln(\frac{1}{\delta})}{2\gamma} = \frac{1 - 2\gamma}{k}||\mathbf{w}||_1 + \frac{\gamma||\mathbf{w}||_1}{k} + \frac{k \ln(\frac{1}{\delta})}{2\gamma} \leq \frac{1 - \gamma}{k}||\mathbf{w}||_1 + S,$$

where the first equality follows by the fact that $\mathbf{C}_t[y_t, y_t] = 0$ and $||\mathbf{C}_t[y_t]||_1 = 1$. Thus the weak learners indeed satisfy the online weak learning condition with edge γ and excess loss S. Now suppose a booster imposes weights on weak learners by α^i. WLOG, we may assume the weights are normalized such that $\sum_{i=1}^N \alpha^i = 1$. Adopting the argument of Schapire and Freund [2012, Section 13.2.6], we prove that the optimal choice of weights is $(\frac{1}{N}, \cdots, \frac{1}{N})$. Fix t, and let l^i denote the prediction made by WL^i. By noting that $\mathbb{P}(y_t = y) = \frac{1}{k}$, which is constant, we can deduce

$$\begin{aligned}\mathbb{P}(y_t = y|l^1, \cdots, l^N) &= \frac{\mathbb{P}(l^1, \cdots, l^N|y_t = y)\mathbb{P}(y_t = y)}{\mathbb{P}(l^1, \cdots, l^N)} \\ &\propto \mathbb{P}(l^1, \cdots, l^N|y_t = y) \\ &= \prod_{i=1}^N p^{\mathbb{1}(l^i = y)} q^{\mathbb{1}(l^i \neq y)},\end{aligned}$$

where $f \propto g$ means $f(y)/g(y)$ does not depend on y, $p = \mathbf{u}_{2\gamma}^{y_t}[y_t] = \frac{1-2\gamma}{k} + 2\gamma$, and $q = \mathbf{u}_{2\gamma}^{y_t}[l] = \frac{1-2\gamma}{k}$. By taking log, we get

$$\begin{aligned}\log \mathbb{P}(y_t = y|l^1, \cdots, l^N) &= C + \log p \sum_{i=1}^N \mathbb{1}(l^i = y) + \log q \sum_{i=1}^N \mathbb{1}(l^i \neq y) \\ &= C + N \log q + \log \frac{p}{q} \sum_{i=1}^N \mathbb{1}(l^i = y).\end{aligned}$$

Therefore, the optimal decision after observing $l^1, \cdots, l^N$ is to choose y that maximizes $\sum_{i=1}^N \mathbb{1}(l^i = y)$, or equivalently, to take simple majority votes.

To compute a lower bound for the error rate, we again introduce random draw framework as in the proof of Lemma A.4. WLOG, we may assume that the true label is 1. Let A_i denote the event that the number i beats 1 in the majority votes. Then we have

$$\mathbb{P}(\text{booster makes error}) \geq \mathbb{P}(A_2). \tag{A.18}$$

Now we need a lower bound for $\mathbb{P}(A_2)$. To do so, let $\{Y_i\}$ be the series of i.i.d. random variables such that $Y_i \in \{-1, 0, 1\}$ and

$$\begin{aligned}\mathbb{P}(Y_j = 1) &= \frac{1-2\gamma}{k} + 2\gamma =: p_1, \\ \mathbb{P}(Y_j = -1) &= \frac{1-2\gamma}{k} =: p_{-1}.\end{aligned}$$

Then $\mathbb{P}(A_2) = \mathbb{P}(Y < 0)$ where $Y := \sum_{i=1}^{N} Y_i$.

Now let M be the number of j such that $Y_j \neq 0$. By conditioning on M, we can write

$$\mathbb{P}(Y < 0 | M = m) = \mathbb{P}(B \leq \frac{m}{2}),$$

where $B \sim binom(m, \frac{p_1}{p_1+p_{-1}})$. By Slud's inequality [Slud, 1977, Theorem 2.1], we have

$$\mathbb{P}(B \leq \frac{m}{2}) \geq \mathbb{P}(Z \geq \sqrt{m}\frac{p - \frac{1}{2}}{\sqrt{p(1-p)}}),$$

where Z follows a standard normal distribution and $p = \frac{p_1}{p_1+p_{-1}}$. Now using tail bound on normal distribution, we get

$$\begin{aligned}
\mathbb{P}(B \leq \frac{m}{2}) &\geq \Omega(\exp(-\frac{m(p-1/2)^2}{p(1-p)})) \\
&= \Omega(\exp(-\frac{m(p_1-p_{-1})^2}{4p_1p_{-1}})) \\
&= \Omega(\exp(-\frac{m\gamma^2}{p_1p_{-1}})) \\
&\geq \Omega(\exp(-4mk^2\gamma^2)) \\
&\geq \Omega(\exp(-4Nk^2\gamma^2)).
\end{aligned} \tag{A.19}$$

We intentionally drop $\frac{1}{2}$ from the power, which makes the bound smaller. The second inequality holds because $p_1p_{-1} \geq \frac{(1-2\gamma)^2}{k^2} \geq \frac{1}{4k^2}$. Integrating w.r.t. m gives

$$\mathbb{P}(\text{booster makes error}) \geq \mathbb{P}(Y < 0) \geq \Omega(\exp(-4Nk^2\gamma^2)).$$

By setting this value equal to ϵ, we have $N \geq \Omega(\frac{1}{k^2\gamma^2}\ln\frac{1}{\epsilon})$, which proves the first part of the theorem.

Now we turn our attention to the optimality of sample complexity. Let $T_0 := \frac{kS}{4\gamma}$ and define $\mathbf{p}_t = \mathbf{u}_0^{y_t}$ for $t \leq T_0$ and $\mathbf{p}_t = \mathbf{u}_{2\gamma}^{y_t}$ for $t > T_0$. Then for $T \leq T_0$, (A.17) implies

$$\sum_{t=1}^{T} w_t \mathbf{C}_t[y_t, \hat{y}_t] \leq \frac{1+\gamma}{k}||\mathbf{w}||_1 + \frac{k\ln(\frac{1}{\delta})}{2\gamma} \leq \frac{1-\gamma}{k}||\mathbf{w}||_1 + S, \tag{A.20}$$

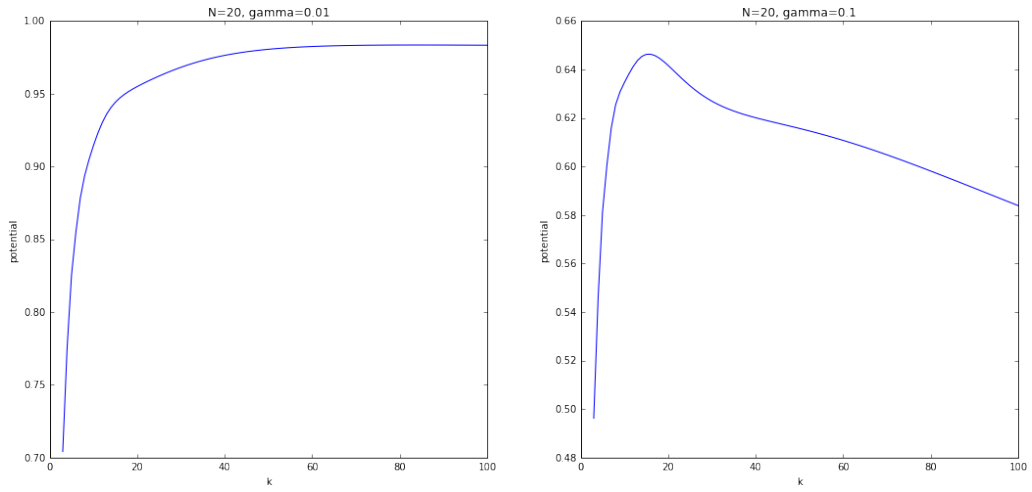

Figure A.1: Plot of $\phi_N^1(\mathbf{0})$ computed with distribution $\mathbf{u}_\gamma^1$ versus the number of labels k. N is fixed to be 20, and the edge γ is set to be 0.01 (left) and 0.1 (right). The graph is not monotonic for larger edge. This hinders the approximation of potential functions with respect to k.

where the last inequality holds because $||\mathbf{w}||_1 \leq T_0 = \frac{kS}{4\gamma}$. For $T > T_0$, again (A.17) implies

$$\begin{aligned}\sum_{t=1}^{T} w_t \mathbf{C}_t[y_t, \hat{y}_t] &\leq \frac{1}{k}\sum_{t=1}^{T_0} w_t + \frac{1-2\gamma}{k}\sum_{t=T_0+1}^{T} w_t + \frac{\gamma||\mathbf{w}||_1}{k} + \frac{k\ln(\frac{1}{\delta})}{2\gamma} \\ &\leq \frac{2\gamma}{k}T_0 + \frac{1-\gamma}{k}||\mathbf{w}||_1 + \frac{k\ln(\frac{1}{\delta})}{2\gamma} \\ &\leq \frac{1-\gamma}{k}||\mathbf{w}||_1 + S.\end{aligned} \tag{A.21}$$

(A.20) and (A.21) prove that the weak learners indeed satisfy (2.1). Now note that combining weak learners does not provide meaningful information for $t \leq T_0$, and thus any online boosting algorithm has errors at least $\Omega(T_0)$. Therefore to get the desired asymptotic error rate, the number of observations T should be at least $\Omega(\frac{T_0}{\epsilon}) = \Omega(\frac{k}{\epsilon\gamma}S)$, which proves the second part of the theorem. □

Even though the gap for the number of weak learners between Corollary 2.3 and Theorem 2.4 is merely polynomial in k, readers might think it is counter-intuitive that N is increasing in k in the upper bound while decreasing in the lower bound. This phenomenon occurs due to the difficulty in approximating potential functions. Recall that Lemma A.4 and Theorem 2.4 utilize upper and lower bound of $\phi_N^1(\mathbf{0})$.

At first glance, considering that $\phi_N^1(\mathbf{0})$ implies the error rate of majority votes out of N independent random draws with distribution $\mathbf{u}_\gamma^1$, the potential function seems to be increasing in k as the

task gets harder with bigger set of options. This is the case of left panel of Figure A.1. However, as it is shown in the right panel, it can also start decreasing in k when γ is larger. This can happen because the probability that a wrong label is drawn vanishes as k grows while the probability that the correct label is drawn remains bigger than γ. In this regard, even though the number of wrong labels gets larger, the error rate actually decreases as $\mathbf{u}_\gamma^1[1]$ dominates other probabilities.

After acknowledging that $\phi_N^1(\mathbf{0})$ might not be a monotonic function of k, the linear upper bound (A.7) turns out to be quite naive, and this is the main reason for the conflicting dependence on k in upper bound and lower bound for N. As the relation among k, N, and γ in $\phi_N^1(\mathbf{0})$ is quite intricate, the issue of deriving better approximation of potential functions remains open.

A.3 Proof of Theorem 2.5

We first introduce a lemma that will be used in the proof.

Lemma A.6. *Suppose $A, B \geq 0$, $B - A = \gamma \in [-1, 1]$, and $A + B \leq 1$. Then we have*

$$\min_{\alpha \in [-2,2]} A(e^\alpha - 1) + B(e^{-\alpha} - 1) \leq -\frac{\gamma^2}{2}.$$

Proof. We divide into three cases with respect to the range of $\frac{B}{A}$.

First suppose $e^{-4} \leq \frac{B}{A} \leq e^4$. In this case, the minimum is attained at $\alpha = \frac{1}{2}\log\frac{B}{A}$, and the minimum becomes

$$\begin{aligned} -(A+B) + 2\sqrt{AB} &= -(\sqrt{A} - \sqrt{B})^2 \\ &= -(\frac{A-B}{\sqrt{A}+\sqrt{B}})^2 \\ &= -\frac{\gamma^2}{(\sqrt{A}+\sqrt{B})^2} \\ &\leq -\frac{\gamma^2}{2(A+B)} \leq -\frac{\gamma^2}{2}. \end{aligned}$$

Now suppose $\frac{B}{A} > e^4 > 51$. From $B - A = \gamma$, we have $\gamma > 50A \geq 0$. Choosing $\alpha = \log 6$, we

get the minimum is bounded above by

$$\begin{aligned} 5A - \frac{5}{6}B &= \frac{25}{6}A - \frac{5}{6}\gamma \\ &< \frac{25}{6}\frac{\gamma}{50} - \frac{5}{6}\gamma \\ &= -\frac{3}{4}\gamma < -\frac{\gamma^2}{2}. \end{aligned}$$

The last inequality hold due to $\gamma \leq 1$.

Finally suppose $\frac{A}{B} > e^4 > 51$. From $B - A = \gamma$, we have $-\gamma > 50B \geq 0$. Choosing $\alpha = -\log 6$, we get the minimum is bounded above by

$$\begin{aligned} -\frac{5}{6}A + 5B &= \frac{25}{6}B + \frac{5}{6}\gamma \\ &< -\frac{25}{6}\frac{\gamma}{50} + \frac{5}{6}\gamma \\ &= \frac{3}{4}\gamma < -\frac{\gamma^2}{2}. \end{aligned}$$

The last inequality hold due to $\gamma \geq -1$. This completes the proof. □

Now we provide a proof of Theorem 2.5.

Proof. Let M_i denote the number of mistakes made by expert i: $M_i = \sum_t \mathbb{1}(y_t \neq \hat{y}_t^i)$. We also let $M_0 = T$ for the ease of presentation. As Adaboost.OLM is using the Hedge algorithm among N experts, the Azuma-Hoeffding inequality and a standard analysis (cf. Cesa-Bianchi and Lugosi [2006, Corollary 2.3]) provide with probability $1 - \delta$,

$$\sum_t \mathbb{1}(y_t \neq \hat{y}_t) \leq 2 \min_i M_i + 2 \log N + \tilde{O}(\sqrt{T}), \tag{A.22}$$

where $\tilde{O}$ notation suppresses dependence on $\log \frac{1}{\delta}$.

Now suppose the expert $i - 1$ makes a mistake at iteration t. That is to say, in a conservative way, $\mathbf{s}_t^{i-1}[y_t] \leq \mathbf{s}_t^{i-1}[l]$ for some $l \neq y_t$. This implies that among $k - 1$ terms in the summation of $-\mathbf{C}_t^i[y_t, y_t]$ in (2.9), at least one term is not less than $\frac{1}{2}$. Thus we can say $-\mathbf{C}_t^i[y_t, y_t] \geq \frac{1}{2}$ if the expert $i - 1$ makes a mistake at $\mathbf{x}_t$. This leads to the inequality:

$$-\sum_t \mathbf{C}_t^i[y_t, y_t] \geq \frac{M_{i-1}}{2}. \tag{A.23}$$

Note that by definition of M_0 and $\mathbf{C}_t^1$, the above inequality holds for $i = 1$ as well. For ease of notation, let us write $w^i := -\sum_t \mathbf{C}_t^i[y_t, y_t]$.

Now let Δ_i denote the difference of the cumulative logistic loss between two consecutive experts:

$$\Delta_i = \sum_t L^{y_t}(\mathbf{s}_t^i) - L^{y_t}(\mathbf{s}_t^{i-1}) = \sum_t L^{y_t}(\mathbf{s}_t^{i-1} + \alpha_t^i \mathbf{e}_{l_t^i}) - L^{y_t}(\mathbf{s}_t^{i-1}).$$

Then Online Gradient Descent algorithm provides

$$\Delta_i \leq \min_{\alpha \in [-2,2]} \sum_t [L^{y_t}(\mathbf{s}_t^{i-1} + \alpha \mathbf{e}_{l_t^i}) - L^{y_t}(\mathbf{s}_t^{i-1})] + 4\sqrt{2}(k-1)\sqrt{T}. \tag{A.24}$$

By simple algebra, we can check

$$\log(1 + e^{s+\alpha}) - \log(1 + e^s) = \log(1 + \frac{e^\alpha - 1}{1 + e^{-s}}) \leq \frac{1}{1 + e^{-s}}(e^\alpha - 1).$$

From this, we can deduce that

$$L^{y_t}(\mathbf{s}_t^{i-1} + \alpha \mathbf{e}_{l_t^i}) - L^{y_t}(\mathbf{s}_t^{i-1}) \leq \begin{cases} \mathbf{C}_t^i[y_t, l_t^i](e^\alpha - 1) & , \text{if } l_t^i \neq y_t \\ \mathbf{C}_t^i[y_t, l_t^i](-e^{-\alpha} + 1) & , \text{if } l_t^i = y_t \end{cases}.$$

Summing over t, we have

$$\sum_t L^{y_t}(\mathbf{s}_t^{i-1} + \alpha \mathbf{e}_{l_t^i}) - L^{y_t}(\mathbf{s}_t^{i-1}) \leq w^i(A(e^\alpha - 1) + B(e^{-\alpha} - 1)),$$

where

$$A = \sum_{l_t \neq y_t} \mathbf{C}_t[y_t, l_t]/w^i, \; B = -\sum_{l_t = y_t} \mathbf{C}_t[y_t, l_t]/w^i.$$

Note that A and B are non-negative and $B - A = \gamma_i \in [-1, 1], \; A + B \leq 1$. Lemma A.6 provides

$$\min_{\alpha \in [-2,2]} \sum_t [L^{y_t}(\mathbf{s}_t^{i-1} + \alpha \mathbf{e}_{l_t^i}) - L^{y_t}(\mathbf{s}_t^{i-1})] \leq -\frac{{\gamma_i}^2}{2} w^i. \tag{A.25}$$

Combining (A.23), (A.24), and (A.25), we have

$$\Delta_i \leq -\frac{{\gamma_i}^2}{4} M_{i-1} + 4\sqrt{2}(k-1)\sqrt{T}.$$

Summing over i, we get by telescoping rule

$$\sum_t L^{y_t}(\mathbf{s}_t^N) - \sum_t L^{y_t}(\mathbf{0}) \leq -\frac{1}{4}\sum_i \gamma_i^2 M_{i-1} + 4\sqrt{2}(k-1)N\sqrt{T}$$
$$\leq -\frac{1}{4}\sum_i \gamma_i^2 \min_i M_i + 4\sqrt{2}(k-1)N\sqrt{T}.$$

Note that $L^{y_t}(\mathbf{0}) = (k-1)\log 2$ and $L^{y_t}(\mathbf{s}_t^N) \geq 0$. Therefore we have

$$\min_i M_i \leq \frac{4(k-1)\log 2}{\sum_i \gamma_i^2} T + \frac{16\sqrt{2}(k-1)N}{\sum_i \gamma_i^2}\sqrt{T}.$$

Plugging this in (A.22), we get with probability $1-\delta$,

$$\sum_t \mathbb{1}(y_t \neq \hat{y}_t) \leq \frac{8(k-1)\log 2}{\sum_i \gamma_i^2} T + \tilde{O}(\frac{kN\sqrt{T}}{\sum_i \gamma_i^2} + \log N)$$
$$\leq \frac{8(k-1)}{\sum_i \gamma_i^2} T + \tilde{O}(\frac{kN^2}{\sum_i \gamma_i^2}),$$

where the last inequality holds from AM-GM inequality: $cN\sqrt{T} \leq \frac{c^2N^2+T}{2}$.

□

A.4 Adaptive algorithms with different surrogate losses

In this section, we present similar adaptive boosting algorithms with Adaboost.OLM but with two different surrogate losses: exponential loss and square hinge loss. We keep the main structure, but the unique properties of each loss result in little difference in details.

A.4.1 Exponential loss

As discussed in Section 2.3.1, exponential loss is useful in batch setting because it provides a closed form for the potential function. We will use following multiclass version of exponential loss:

$$L^r(\mathbf{s}) := \sum_{l \neq r} \exp(\mathbf{s}[l] - \mathbf{s}[r]). \tag{A.26}$$

From this, we can compute the cost matrix and $f_t^{i'}$ for the online gradient descent as below:

$$\mathbf{C}_t^i[r,l] = \begin{cases} \exp(\mathbf{s}_t^{i-1}[l] - \mathbf{s}_t^{i-1}[r]) & \text{, if } l \neq r \\ -\sum_{j \neq r} \exp(\mathbf{s}_t^{i-1}[j] - \mathbf{s}_t^{i-1}[r]) & \text{, if } l = r \end{cases} \tag{A.27}$$

$$f_t^{i\prime}(\alpha) = \begin{cases} \exp(\mathbf{s}_t^{i-1}[l_t^i] + \alpha - \mathbf{s}_t^{i-1}[y_t]) & \text{, if } l_t^i \neq y_t \\ -\sum_{j \neq y_t} \exp(\mathbf{s}_t^{i-1}[j] - \alpha - \mathbf{s}_t^{i-1}[y_t]) & \text{, if } l_t^i = y_t. \end{cases} \tag{A.28}$$

With this gradient, if we set the learning rate $\eta_t^i = \frac{2\sqrt{2}}{(k-1)\sqrt{t}} e^{-i}$, a standard analysis provides $R^i(T) \leq 4\sqrt{2}(k-1)e^i\sqrt{T}$. Note that with exponential loss, we have different learning rate for each weak learner. We keep the algorithm same as Algorithm 2.2, but with different cost matrix and learning rate. Now we state the theorem for the mistake bound.

Theorem A.7. (Mistake bound with exponential loss) *For any T and N, the number of mistakes made by Algorithm 2.2 with above cost matrix and learning rate satisfies the following inequality with high probability:*

$$\sum_t \mathbb{1}(y_t \neq \hat{y}_t) \leq \frac{4k}{\sum_i \gamma_i^2} T + \tilde{O}(\frac{ke^{2N}}{\sum_i \gamma_i^2}).$$

Proof. The proof is almost identical to that of Theorem 2.5, and we only state the different steps. With cost matrix defined in (A.27), we can show

$$-\sum_t \mathbf{C}_t^i[y_t, y_t] \geq M_{i-1}.$$

Furthermore, we have following identity (which was inequality in the original proof):

$$L^{y_t}(\mathbf{s}_t^{i-1} + \alpha \mathbf{e}_{l_t^i}) - L^{y_t}(\mathbf{s}_t^{i-1}) = \begin{cases} \mathbf{C}_t^i[y_t, l_t^i](e^\alpha - 1) & \text{, if } l_t^i \neq y_t \\ \mathbf{C}_t^i[y_t, l_t^i](-e^{-\alpha} + 1) & \text{, if } l_t^i = y_t \end{cases}.$$

This leads to

$$\Delta_i \leq -\frac{{\gamma_i}^2}{2} M_{i-1} + 4\sqrt{2}(k-1)e^i\sqrt{T}.$$

Summing over i, we get

$$\begin{aligned}\frac{\sum_i \gamma_i^2}{2} \min_i M_i &\leq (k-1)T + 4\sqrt{2}(k-1)e\frac{e^N-1}{e-1}\sqrt{T} \\ &\leq (k-1)T + 9ke^N\sqrt{T}.\end{aligned}$$

Plugging this in (A.22), we get with high probability,

$$\begin{aligned}\sum_t \mathbb{1}(y_t \neq \hat{y}_t) &\leq \frac{4(k-1)}{\sum_i \gamma_i^2}T + \tilde{O}(\frac{ke^N\sqrt{T}}{\sum_i \gamma_i^2} + \log N) \\ &\leq \frac{4k}{\sum_i \gamma_i^2}T + \tilde{O}(\frac{ke^{2N}}{\sum_i \gamma_i^2}),\end{aligned}$$

which completes the proof. We also used AM-GM inequality for the last step. □

Comparing to Theorem 2.5, we get a better coefficient for the first term, which is asymptotic error rate, but the exponential function in the second term makes the bound significantly loose. The exponential term comes from the larger variability of f_t^i associated with exponential loss. It should also be noted that the empirical edge γ_i is measured with different cost matrices, and thus direct comparison is not fair. In fact, as discussed in Section 2.3.1, γ_i is closer to 0 with exponential loss than with logistic loss due to larger variation in weights, which is another huge advantage of logistic loss.

A.4.2 Square hinge loss

Another popular surrogate loss is square hinge loss. We begin the section by introducing multiclass version of it:

$$L^r(\mathbf{s}) := \frac{1}{2}\sum_{l\neq r}(\mathbf{s}[l] - \mathbf{s}[r] + 1)_+^2, \tag{A.29}$$

where $f_+ := \max\{0, f\}$. From this, we can compute the cost matrix and $f_t^{i\prime}$ for the online gradient descent as below:

$$\mathbf{C}_t^i[r,l] = \begin{cases} (\mathbf{s}_t^{i-1}[l] - \mathbf{s}_t^{i-1}[r] + 1)_+ & \text{, if } l \neq r \\ -\sum_{j\neq r}(\mathbf{s}_t^{i-1}[j] - \mathbf{s}_t^{i-1}[r] + 1)_+ & \text{, if } l = r \end{cases} \tag{A.30}$$

$$f_t^{i\prime}(\alpha) = \begin{cases} (\mathbf{s}_t^{i-1}[l_t^i] + \alpha - \mathbf{s}_t^{i-1}[y_t] + 1)_+ & \text{, if } l_t^i \neq y_t \\ -\sum_{j \neq y_t} (\mathbf{s}_t^{i-1}[j] - \alpha - \mathbf{s}_t^{i-1}[y_t] + 1)_+ & \text{, if } l_t^i = y_t. \end{cases} \tag{A.31}$$

With square hinge loss, we do not use Lemma A.6 in the proof of mistake bound, and thus the feasible set F can be narrower. In fact, we will set $F = [-c, c]$, where the parameter c will be optimized later. With this F, we have $|f_t^{i\prime}(\alpha)| \leq (k-1) + ci \leq (k-1) + cN$, and the standard analysis of online gradient descent algorithm with learning rate $\eta_t = \frac{\sqrt{2}c}{((k-1)+cN)\sqrt{t}}$ provides that $R^i(T) \leq 2\sqrt{2}(k-1+cN)\sqrt{T}$. Now we are ready to prove the mistake bound.

Theorem A.8. (Mistake bound with square hinge loss) *For any T and N, with the choice of $c = \frac{1}{\sqrt{N}}$, the number of mistakes made by Algorithm 2.2 with above cost matrix and learning rate satisfies the following inequality with high probability:*

$$\sum_t \mathbb{1}(y_t \neq \hat{y}_t) \leq \frac{2k\sqrt{N}}{\sum_i |\gamma_i|} T + \tilde{O}(\frac{(k^2+N)N\sqrt{N}}{\sum_i |\gamma_i|}).$$

Proof. With cost matrix defined in (A.30), we can show

$$-\sum_t \mathbf{C}_t^i[y_t, y_t] \geq M_{i-1}.$$

We can also check that

$$\frac{1}{2}[(s+\alpha)_+^2 - s_+^2] \leq s_+\alpha + \frac{\alpha^2}{2},$$

by splitting the cases with the sign of each term. Using this, we can deduce that

$$L^{y_t}(\mathbf{s}_t^{i-1} + \alpha \mathbf{e}_{l_t^i}) - L^{y_t}(\mathbf{s}_t^{i-1}) \leq \mathbf{C}_t^i[y_t, l_t^i]\alpha + \frac{(k-1)\alpha^2}{2}.$$

Summing over t gives

$$\sum_t L^{y_t}(\mathbf{s}_t^{i-1} + \alpha \mathbf{e}_{l_t^i}) - L^{y_t}(\mathbf{s}_t^{i-1}) \leq \sum_t \mathbf{C}_t^i[y_t, y_t]\gamma_i\alpha + \frac{(k-1)\alpha^2}{2} T.$$

The RHS is a quadratic in α, and the minimizer is $\alpha^* = -\frac{\sum_t \mathbf{C}_t^i[y_t, y_t]\gamma_i}{(k-1)T}$. Since the magnitude of $\mathbf{C}_t^i[y_t, y_t]$ grows as a function of c, there is no guarantee that this minimizer lies in the feasible set

$F = [-c, c]$. Instead, we will bound the minimum by plugging in $\alpha = \pm c$:

$$\begin{aligned}\min_{\alpha\in[-c,c]} \sum_t L^{y_t}(\mathbf{s}_t^{i-1} + \alpha \mathbf{e}_{l_t^i}) - L^{y_t}(\mathbf{s}_t^{i-1}) &\leq \frac{(k-1)c^2}{2}T + c|\gamma_i| \sum_t \mathbf{C}_t^i[y_t, y_t] \\ &\leq \frac{(k-1)c^2}{2}T - c|\gamma_i| M_{i-1}.\end{aligned}$$

From this, we get

$$\Delta_i \leq -c|\gamma_i| M_{i-1} + \frac{(k-1)c^2}{2}T + 2\sqrt{2}(k-1+cN)\sqrt{T}.$$

Summing over i, we get

$$c\sum_i |\gamma_i| \min_i M_i \leq \frac{k-1}{2}T + \frac{(k-1)c^2 N}{2}T + 2\sqrt{2}(k-1+cN)N\sqrt{T}.$$

By rearranging terms, we conclude

$$\min_i M_i \leq \frac{(k-1)}{2\sum_i |\gamma_i|}(\frac{1}{c} + cN)T + \frac{2\sqrt{2}(k-1+cN)N}{\sum_i |\gamma_i|}\sqrt{T}.$$

It is the first term from the RHS that provides an optimal choice of $c = \frac{1}{\sqrt{N}}$, and this value gives

$$\min_i M_i \leq \frac{(k-1)\sqrt{N}}{\sum_i |\gamma_i|}T + \frac{2\sqrt{2}(k-1+\sqrt{N})N}{\sum_i |\gamma_i|}\sqrt{T}.$$

Plugging this in (A.22), we get with high probability,

$$\begin{aligned}\sum_t \mathbb{1}(y_t \neq \hat{y}_t) &\leq \frac{2(k-1)\sqrt{N}}{\sum_i |\gamma_i|}T + \tilde{O}(\frac{(k+\sqrt{N})N}{\sum_i |\gamma_i|}\sqrt{T} + \log N) \\ &\leq \frac{2k\sqrt{N}}{\sum_i |\gamma_i|}T + \tilde{O}(\frac{(k^2+N)N\sqrt{N}}{\sum_i |\gamma_i|}),\end{aligned}$$

which completes the proof. We also used AM-GM inequality for the last step. □

By Cauchy-Schwartz inequality, we have $N\sum_i \gamma_i^2 \geq (\sum_i |\gamma_i|)^2$. From this, we can deduce $(\frac{\sqrt{N}}{\sum_i |\gamma_i|})^2 \geq \frac{1}{\sum_i \gamma_i^2}$. If LHS is greater than 1, then the bound in Theorem A.8 is meaningless. Otherwise, we have

$$\frac{\sqrt{N}}{\sum_i |\gamma_i|} \geq (\frac{\sqrt{N}}{\sum_i |\gamma_i|})^2 \geq \frac{1}{\sum_i \gamma_i^2},$$

which validates that the bound with logistic loss is tighter. Furthermore, square hinge loss also produces more variable weights over instances, which results in worse empirical edges.

A.5 Detailed description of experiment

Testing was performed on a variety of data sets described in Table A.1. All are from the UCI data repository (Blake and Merz [1998], Higuera C [2015], Ugulino et al. [2012]) with a few adjustments made to deal with missing data and high dimensionality. These changes are noted in the table below. Many of the data sets are the same as used in the Oza [2005], with the addition of a few sets with larger numbers of data points and predictors. We report the average performance on both the entire data set and on the final 20% of the data set. The two accuracy measures help understand both the "burn in period", or how quickly the algorithm improves as observations are recorded, and the "accuracy plateau", or how well the algorithm can perform given sufficient data. Different applications may emphasize each of these two algorithmic characteristics, so we choose to provide both to the reader. We also report average run times. All computations were carried out on a Nehalem architecture 10-core 2.27 GHz Intel Xeon E7-4860 processors with 25 GB RAM per core. For all but the last two data sets, results are averaged over 27 reordering of the data. Due to computational constraints, Movement was run just nine times and ISOLET just once.

Table A.1: Data set details

Data sets	Number of data points	Number of predictors	Number of classes
Balance	625	4	3
Mice	1080	82*	8
Cars	1728	6	4
Mushroom	8124	22	2
Nursery	12960	8	4
ISOLET	7797	50**	26
Movement	165631***	12***	5

* Missing data was replaced with 0.
** The original 617 predictors were projected onto their first 50 principal components, which contained 80% of the variation.
*** User information was removed, leaving only sensor position predictors. Single data point with missing value removed.

In all the experiments we used Very Fast Decision Trees (VFDT) from Domingos and Hulten [2000] as weak learners. VFDT has several tuning parameters which relate to the frequency

with which the tree splits. In all methods we assigned these randomly for each tree. Specifically for our implementation the tuning parameter `grace_period` was chosen randomly between 5 and 20 and the tuning parameters `split_confidence` and `hoeffding_tie_threshold` randomly between 0.01 and 0.9. It is likely that this procedure would produce trees which do not perform well on specific data sets. In practice for the Adaboost.OLM it is possible to restart poorly performing trees using parameters similar to better performing trees in an automated and online (although ad hoc) fashion using the α_t^i, and this tends to produce superior performance (as well as allow adaptivity to changes in the data distribution). However for these experiments, we did not take advantage of this to better examine the benefits of just the cost matrix framework.

Several algorithms were tested using the above specifications, but with slightly different conditions. The first three are directly comparable since they all use the same weak learners and do not require knowledge of the edge of the weak learners. DT is the best result from running 100 VFDT independently. The best was chosen after seeing the performance on the entire data set and final 20% respectively. However the time reported was the average time for running all 100 VFDT. This was done to better see the additional cost of running the boosting framework on top of the training of the raw weak learners. OLB is an implementation of the Online Boosting algorithm in Oza [2005, Figure 2] with 100 VFDT. AdaOLM stands for Adaboost.OLM, again with 100 VFDT.

The next five algorithms (MB) tested were all variants of the OnlineMBBM but with different edge γ values. In practice this value is never known ahead of time, but we want to explore how different edges affect the performance of the algorithm. For the ease of computation, instead of exactly finding the value of (A.6), we estimated the potential functions by Monte Carlo (MC) simulations.

The final two algorithms are slightly different implementations of the One VS All (OvA) ensemble method. In this framework multiple binary classifiers are used to solve a multiclass problem by viewing different classes as the positive class, and all others as the negative class. They then predict whether a data point is their positive class or not, and the results are used together to make a final classification. Both use VFDT as their weak learners, but with $100 \times k$ binary trees. The first method (OvA) uses k versions of Adaboost.OL, each viewing one of the classes as the positive class. Recall that Adaboost.OLM in the binary setting is just Adaboost.OL by Beygelzimer et al. [2015]. The second (AdaOVA) produces 100 weak multiclass classifiers by grouping a k binary classifiers, one for each class, and then uses Adaboost.OLM to get the final learner, treating the 100 single tree OvA's as its weak learners. In the table below we have partitioned the methods in terms of the number of weak learners since, while they all tackle the same problem, algorithms within each partition are more directly comparable since they use the same weak learners.

Table A.2: Comparison of algorithms on final 20% of data set

	100 multiclass trees								$100k$ binary trees	
Data sets	DT	OLB	AdaOLM	MB .3	MB .1	MB .05	MB .01	MB .001	OvA	AdaOVA
Balance	0.768	0.772	0.754	0.788	0.821	0.819	0.805	0.752	0.786	0.795
Mice	0.608	0.399	0.561	0.572	0.695	0.663	0.502	0.467	0.742	0.667
Cars	0.924	0.914	0.930	0.914	0.885	0.870	0.836	0.830	0.946	0.919
Mushroom	0.999	1.000	1.000	0.997	1.000	1.000	0.999	0.998	1.000	1.000
Nursery	0.953	0.941	0.966	0.965	0.969	0.964	0.948	0.940	0.974	0.965
ISOLET	0.515	0.149	0.521	0.453	0.626	0.635	0.226	0.165	0.579	0.570
Movement	0.915	0.870	0.962	0.975	0.987	0.988	0.984	0.981	0.947	0.970

Table A.3: Comparison of algorithms on full data set

	100 multiclass trees								$100k$ binary trees	
Data sets	DT	OLB	AdaOLM	MB .3	MB .1	MB .05	MB .01	MB .001	OvA	AdaOVA
Balance	0.734	0.747	0.698	0.751	0.769	0.759	0.736	0.677	0.724	0.730
Mice	0.499	0.315	0.454	0.457	0.507	0.449	0.356	0.343	0.586	0.530
Cars	0.848	0.839	0.865	0.842	0.829	0.814	0.767	0.762	0.881	0.853
Mushroom	0.996	0.997	0.995	0.991	0.995	0.994	0.993	0.992	0.996	0.995
Nursery	0.921	0.909	0.928	0.932	0.936	0.932	0.918	0.912	0.939	0.932
ISOLET	0.395	0.104	0.456	0.333	0.486	0.461	0.152	0.111	0.507	0.472
Movement	0.898	0.864	0.942	0.954	0.972	0.973	0.959	0.957	0.927	0.952

Table A.4: Comparison of algorithms total run time in seconds

	100 multiclass trees								$100k$ binary trees	
Data sets	DT	OLB	AdaOLM	MB .3	MB .1	MB .05	MB .01	MB .001	OvA	AdaOVA
Balance	8	19	20	26	42	47	50	51	66	43
Mice	105	263	416	783	2173	3539	3579	3310	3092	3013
Cars	39	27	59	56	105	146	165	152	195	143
Mushroom	241	169	355	318	325	326	324	321	718	519
Nursery	526	302	735	840	1510	2028	2181	1984	2995	1732
ISOLET	470	1497	2422	18732	38907	64707	62492	50700	37300	33328
Movement	1960	3437	5072	13018	17608	18676	16739	16023	30080	21389

A.5.1 Analysis

It is worth beginning by noting the strength of the VFDT without any boosting framework. While the results above are for the best performing tree in hindsight, which is not a valid strategy in practice, in many applications it would be possible to collect some data beforehand activating the system, and use that to pick tuning parameters. It is also worth noting that many of the weaknesses of the above methods, such as their poor scaling with the number of predictors, are also inherited from the VFDT. Nonetheless in almost all cases Adaboost.OLM algorithm outperforms both the best tree and the preexisting Online Boosting algorithm (and is often comparable to the OnlineMBBM algorithms), as well as provide theoretical guarantees. In particular these performance gains seem to be greater on the final 20% of the data and in data sets with larger number of data points n, leading us to believe that Adaboost.OLM has a longer burn in period, but higher accuracy plateau. This performance does come at additional computational cost, but this cost is relatively mild, especially compared to the costs of OnlineMBBM and the OvA methods.

The OnlineMBBM methods use additional assumptions about the power of their weak learners, and are able to leverage that additional information to produce more accurate, with one of these algorithms often achieving the highest accuracy on each data set. However they can be sensitive to the choice of γ, with the worst choice of γ often underperforming both pure trees and Adaboost.OLM, and with no single γ value always producing the best result. These methods are also much slower than Adaboost.OLM, likely due to computational burden in estimating the potential functions.

Finally our two OvA algorithms tend to perform very well, often beating the other adaptive methods. However this performance is likely due to the use of many times more weak learners than the other adaptive methods used, which results in high computational cost. Again we see that as n increases the implementation of OvA using our cost matrix framework performs better compared to the vanilla implementation, reinforcing our belief that the cost matrix framework requires more data to come online but has a higher accuracy plateau.

APPENDIX B

Details for Online Boosting Algorithms for Multi-label Ranking

B.1 Specific bounds for OnlineBMR

We begin this section by introducing a *random walk framework* to compute potentials. Suppose $\mathbf{X}^i := (X_1, \cdots, X_k)$ is a random vector that tracks the number of draws of each label among i i.i.d. random draws w.r.t. $\mathbf{u}_\gamma^{Y_t}$. Then according to (3.1), we may write

$$\phi_t^i(\mathbf{s}) = \mathbb{E}L^{Y_t}(\mathbf{s} + \mathbf{X}).$$

This framework will appear frequently throughout the proofs. We start from rank loss.

Lemma B.1. *Under the same setting as in Theorem 3.2 but with potentials built upon rank loss, we may bound $\phi_t^N(\boldsymbol{0})$ as following:*

$$\phi_t^N(\boldsymbol{0}) \leq e^{-\frac{\gamma^2 N}{2}}.$$

Proof. For simplicity, we drop t in the proof. Let $\mathbf{X}^N$ be the aforementioned random vector. Then we may write the potential by

$$\begin{aligned}
\phi^N(\mathbf{0}) &= \mathbb{E}L_{\text{rnk}}^Y(\mathbf{X}^N) \\
&\leq w_Y \sum_{l \in Y} \sum_{r \notin Y} \mathbb{E}\mathbb{1}(X_r \geq X_l) \\
&= w_Y \sum_{l \in Y} \sum_{r \notin Y} \mathbb{P}(X_r - X_l \geq 0).
\end{aligned}$$

Fix $l \in Y$ and $r \notin Y$. By definition of $\mathbf{u}_\gamma^Y$, we have

$$a := \mathbf{u}_\gamma^Y[l] = \mathbf{u}_\gamma^Y[r] + \gamma =: b.$$

Now suppose we draw 1 with probability a, -1 with probability b, and 0 otherwise. Then $\mathbb{P}(X_r - X_l \geq 0)$ equals the probability that the summation of N i.i.d. random numbers is non-negative. Then we can apply the Hoeffding's inequality to get

$$\mathbb{P}(X_r - X_l \geq 0) \leq e^{-\frac{\gamma^2 N}{2}}.$$

Since w_Y is the inverse of the number of pairs (l, r), this proves our assertion. □

Lemma B.2. *Under the same setting as in Theorem 3.2 but with potentials built upon rank loss, we can show that* $\forall i,\ w^{i*} \leq O(\frac{1}{\sqrt{N-i}})$.

Proof. First we fix t and i. We also fix $l^* \in Y_t$ and $r^* \in Y_t^c$. Then write $\mathbf{s}_1 := \mathbf{s}_t^{i-1} + \mathbf{e}_{l^*}$ and $\mathbf{s}_2 := \mathbf{s}_t^{i-1} + \mathbf{e}_{r^*}$. Again we introduce $\mathbf{X}^{N-i}$. Then we may write

$$\begin{aligned}
\mathbf{c}_t^i[r^*] - \mathbf{c}_t^i[l^*] &= \phi_t^{N-i}(\mathbf{s}_2) - \phi_t^{N-i}(\mathbf{s}_1) \\
&= \mathbb{E}[L_{\text{rnk}}^{Y_t}(\mathbf{s}_2 + \mathbf{X}^{N-i}) - L_{\text{rnk}}^{Y_t}(\mathbf{s}_1 + \mathbf{X}^{N-i})] \\
&\leq w_{Y_t} \sum_{l \in Y_t} \sum_{r \notin Y_t} f(r, l),
\end{aligned}$$

where

$$\begin{aligned}
f(r,l) := \mathbb{E}[&\mathbb{1}(\mathbf{s}_2[r] + X_r \geq \mathbf{s}_2[l] + X_l) \\
&- \mathbb{1}(\mathbf{s}_1[r] + X_r > \mathbf{s}_1[l] + X_l)].
\end{aligned}$$

Here we intentionally include and exclude equality for the ease of computation. Changing the order of terms, we can derive

$$\begin{aligned}
f(r,l) &\leq \mathbb{P}(\mathbf{s}_1[l] - \mathbf{s}_1[r] \geq X_r - X_l \geq \mathbf{s}_2[l] - \mathbf{s}_2[r]) \\
&\leq 3 \max_n \mathbb{P}(X_r - X_l = n),
\end{aligned}$$

where the last inequality is deduced from the fact that

$$(\mathbf{s}_1[l] - \mathbf{s}_1[r]) - (\mathbf{s}_2[l] - \mathbf{s}_2[r]) \in \{0, 1, 2\}.$$

Using Berry-Esseen theorem, it is shown in Lemma A.5 that $\max_n \mathbb{P}(X_r - X_l = n) \leq O(\frac{1}{\sqrt{N-i}})$, which implies that

$$\mathbf{c}_t^i[r^*] - \mathbf{c}_t^i[l^*] \leq O(\frac{1}{\sqrt{N-i}}).$$

Since l^* and r^* are arbitrary, and the bound does not depend on t, the last inequality proves our assertion. □

Now we provide similar bounds when the potentials are computed from hinge loss.

Lemma B.3. *Under the same setting as in Theorem 3.2 but with potentials built upon hinge loss, we may bound $\phi_t^N(\boldsymbol{0})$ as following:*

$$\phi_t^N(\boldsymbol{0}) \leq (N+1)e^{-\frac{\gamma^2 N}{2}}.$$

Proof. Again we drop t in the proof and introduce $\mathbf{X}^N$. Then we may write the potential by

$$\begin{aligned}
\phi^N(\mathbf{0}) &= \mathbb{E}L_{\text{hinge}}^Y(\mathbf{X}^N) \\
&= w_Y \sum_{l\in Y} \sum_{r\notin Y} \mathbb{E}(1 + X_r - X_l)_+ \\
&= w_Y \sum_{l\in Y} \sum_{r\notin Y} \sum_{n=0}^{N} \mathbb{P}(X_r - X_l \geq n) \\
&\leq w_Y \sum_{l\in Y} \sum_{r\notin Y} (N+1)\mathbb{P}(X_r - X_l \geq 0).
\end{aligned}$$

We already checked in Lemma B.1 that

$$\mathbb{P}(X_r - X_l \geq 0) \leq e^{-\frac{\gamma^2 N}{2}},$$

which concludes the proof. □

Lemma B.4. *Under the same setting as in Theorem 3.2 but with potentials built upon hinge loss, we can show that $\forall i,\ w^{i*} \leq 2$.*

Proof. First we fix t and i. We also fix $l^* \in Y_t$ and $r^* \in Y_t^c$. Then write $\mathbf{s}_1 := \mathbf{s}_t^{i-1} + \mathbf{e}_{l*}$ and

$\mathbf{s}_2 := \mathbf{s}_t^{i-1} + \mathbf{e}_{r*}$. Again with $\mathbf{X}^{N-i}$, we may write

$$\begin{aligned}\mathbf{c}_t^i[r^*] - \mathbf{c}_t^i[l^*] &= \phi_t^{N-i}(\mathbf{s}_2) - \phi_t^{N-i}(\mathbf{s}_1) \\ &= \mathbb{E}[L_{\text{hinge}}^{Y_t}(\mathbf{s}_2 + \mathbf{X}^{N-i}) - L_{\text{hinge}}^{Y_t}(\mathbf{s}_1 + \mathbf{X}^{N-i})] \\ &= w_{Y_t} \sum_{l \in Y_t} \sum_{r \notin Y_t} f(r, l),\end{aligned}$$

where

$$\begin{aligned}f(r,l) := \mathbb{E}[&(1 + (\mathbf{s}_2 + \mathbf{X}^{N-i})[r] - (\mathbf{s}_2 + \mathbf{X}^{N-i})[l])_+ \\ &- (1 + (\mathbf{s}_1 + \mathbf{X}^{N-i})[r] - (\mathbf{s}_1 + \mathbf{X}^{N-i})[l])_+].\end{aligned}$$

It is not hard to check that the term inside the expectation is always bounded above by 2. This fact along with the definition of w_{Y_t} provides that $\mathbf{c}_t^i[r^*] - \mathbf{c}_t^i[l^*] \leq 2$. Since our choice of l^* and r^* are arbitrary, this proves $\mathbf{w}^i[t] \leq 2$, which completes the proof. □

B.2 Complete proof of Theorem 3.4

Proof. We assume that an adversary draws a label Y_t uniformly at random from $2^{[k]} - \{\emptyset, [k]\}$, and the weak learners generate single-label predictions w.r.t. $\mathbf{p}_t \in \Delta[k]$. Any boosting algorithm can only make a final decision by weighted cumulative votes of N weak learners. We manipulate $\mathbf{p}_t$ such that weak learners satisfy OnlineWLC (δ, γ, S) but the best possible performance is close to (3.6).

As we are assuming single-label predictions, $\mathbf{h}_t = \mathbf{e}_{l_t}$ for some $l_t \in [k]$ and $\mathbf{c}_t \cdot \mathbf{h}_t = \mathbf{c}_t[l_t]$. Furthermore, the bounded condition of $\mathcal{C}_0^{eor}$ ensures $\mathbf{c}_t[l_t]$ is contained in $[0, 1]$. The Azuma-Hoeffding inequality provides that with probability $1 - \delta$,

$$\begin{aligned}\sum_{t=1}^{T} w_t \mathbf{c}_t[l_t] &\leq \sum_{t=1}^{T} w_t \mathbf{c}_t \cdot \mathbf{p}_t + \sqrt{2||\mathbf{w}||_2^2 \ln(\frac{1}{\delta})} \\ &\leq \sum_{t=1}^{T} w_t \mathbf{c}_t \cdot \mathbf{p}_t + \frac{\gamma||\mathbf{w}||_2^2}{k} + \frac{k \ln(\frac{1}{\delta})}{2\gamma} \qquad \text{(B.1)} \\ &\leq \sum_{t=1}^{T} w_t \mathbf{c}_t \cdot \mathbf{p}_t + \frac{\gamma||\mathbf{w}||_1}{k} + \frac{k \ln(\frac{1}{\delta})}{2\gamma},\end{aligned}$$

where the second inequality holds by arithmetic mean and geometric mean relation and the last

inequality holds due to $w_t \in [0,1]$.

We start from providing a lower bound on the number of weak learners. Let $\mathbf{p}_t = \mathbf{u}_{2\gamma}^{Y_t}$ for all t. This can be done by the constraint $\gamma < \frac{1}{4k}$. From the condition of $\mathcal{C}_0^{eor}$ that $\min_l \mathbf{c}[l] = 0, \max_l \mathbf{c} = 1$ along with the fact that $Y \notin \{\emptyset, [k]\}$, we can show that $\mathbf{c} \cdot (\mathbf{u}_\gamma^Y - \mathbf{u}_{2\gamma}^Y) \geq \frac{\gamma}{k}$. Then the last line of (B.1) becomes

$$\begin{aligned}
&\sum_{t=1}^{T} w_t \mathbf{c}_t \cdot \mathbf{u}_{2\gamma}^{Y_t} + \frac{\gamma ||\mathbf{w}||_1}{k} + \frac{k \ln(\frac{1}{\delta})}{2\gamma} \\
&\qquad \leq \sum_{t=1}^{T} (w_t \mathbf{c}_t \cdot \mathbf{u}_\gamma^{Y_t} - \frac{\gamma w_t}{k}) + \frac{\gamma ||\mathbf{w}||_1}{k} + \frac{k \ln(\frac{1}{\delta})}{2\gamma} \\
&\qquad \leq \sum_{t=1}^{T} w_t \mathbf{c}_t \cdot \mathbf{u}_\gamma^{Y_t} + S,
\end{aligned}$$

which validates that weak learners indeed satisfy OnlineWLC (δ, γ, S). Following the argument of Schapire and Freund [2012, Section 13.2.6], we can also prove that the optimal choice of weights over the learners is $(\frac{1}{N}, \cdots, \frac{1}{N})$.

Now we compute a lower bound for the booster's loss. Let $\mathbf{X} := (X_1, \cdots, X_k)$ be a random vector that tracks the number of labels drawn from N i.i.d. random draws w.r.t. $\mathbf{u}_{2\gamma}^Y$. Then the expected rank loss of the booster can be written as:

$$\mathbb{E} L_{\text{rnk}}^Y(\mathbf{X}) \geq w_Y \sum_{l \in Y} \sum_{r \notin Y} \mathbb{P}(X_l < X_r).$$

Adopting the arguments in the proof of Theorem 2.4, we can show that

$$\mathbb{P}(X_l < X_r) \geq \Omega(e^{-4Nk^2\gamma^2}).$$

This shows $\mathbb{E} L_{\text{rnk}}^Y(\mathbf{X}) \geq \Omega(e^{-4Nk^2\gamma^2})$. Setting this value equal to ϵ, we have $N \geq \Omega(\frac{1}{\gamma^2} \ln \frac{1}{\epsilon})$, considering k as a fixed constant. This proves the first part of the theorem.

Now we move on to the optimality of sample complexity. We record another inequality that can be checked from the conditions of $\mathcal{C}_0^{eor}$: $\mathbf{c} \cdot (\mathbf{u}_0^Y - \mathbf{u}_\gamma^Y) \leq \gamma$. Let $T_0 := \frac{S}{4\gamma}$ and define $\mathbf{p}_t = \mathbf{u}_0^{Y_t}$ for

$t \leq T_0$ and $\mathbf{p}_t = \mathbf{u}_{2\gamma}^{Y_t}$ for $t > T_0$. Then for $T \leq T_0$, (B.1) implies

$$\begin{aligned}
&\sum_{t=1}^{T} w_t \mathbf{c}_t[l_t] \\
&\quad \leq \sum_{t=1}^{T} w_t \mathbf{c}_t \cdot \mathbf{u}_0^{Y_t} + \frac{\gamma ||\mathbf{w}||_1}{k} + \frac{k \ln(\frac{1}{\delta})}{2\gamma} \\
&\quad \leq \sum_{t=1}^{T} w_t \mathbf{c}_t \cdot \mathbf{u}_\gamma^{Y_t} + \gamma(1 + \frac{1}{k})||\mathbf{w}||_1 + \frac{k \ln(\frac{1}{\delta})}{2\gamma} \\
&\quad \leq \sum_{t=1}^{T} w_t \mathbf{c}_t \cdot \mathbf{u}_\gamma^{Y_t} + S.
\end{aligned} \tag{B.2}$$

where the last inequality holds because $||\mathbf{w}||_1 \leq T_0 = \frac{S}{4\gamma}$. For $T > T_0$, again (B.1) implies

$$\begin{aligned}
\sum_{t=1}^{T} w_t \mathbf{c}_t[l_t] &\leq \sum_{t=1}^{T_0} w_t \mathbf{c}_t \cdot \mathbf{u}_0^{Y_t} + \sum_{t=T_0+1}^{T} w_t \mathbf{c}_t \cdot \mathbf{u}_{2\gamma}^{Y_t} \\
&\qquad + \frac{\gamma ||\mathbf{w}||_1}{k} + \frac{k \ln(\frac{1}{\delta})}{2\gamma} \\
&\leq \sum_{t=1}^{T} w_t \mathbf{c}_t \cdot \mathbf{u}_\gamma^{Y_t} + \frac{k+1}{k} \gamma T_0 + \frac{k \ln(\frac{1}{\delta})}{2\gamma} \\
&\leq \sum_{t=1}^{T} w_t \mathbf{c}_t \cdot \mathbf{u}_\gamma^{Y_t} + S.
\end{aligned} \tag{B.3}$$

(B.2) and (B.3) prove that the weak learners indeed satisfy OnlineWLC (δ, γ, S). Observing that weak learners do not provide meaningful information for $t \leq T_0$, we can claim any online boosting algorithm suffers a loss at least $\Omega(T_0)$. Therefore to get the certain accuracy, the number of instances T should be at least $\Omega(\frac{T_0}{\epsilon}) = \Omega(\frac{S}{\epsilon\gamma})$, which completes the second part of the proof. □

APPENDIX C

Details for Thompson Sampling in Episodic Restless Bandit Problems

C.1 Proof of Theorem 5.5

We begin by introducing a technical lemma.

Lemma C.1. *Let $a_i, b_i \in [0,1]$ and $|a_i - b_i| \leq \Delta_i$ for $i \in [k]$. Then we can show*

$$\sum_{x \in \{0,1\}^k} |\prod_i a_i^{x_i}(1-a_i)^{1-x_i} - \prod_i b_i^{x_i}(1-b_i)^{1-x_i}| \leq 2\sum_{j=1}^{k} \Delta_j. \tag{C.1}$$

Proof. Fix a binary vector x. For simplicity, let $c_i = a_i^{x_i}(1-a_i)^{1-x_i}$ and $d_i = b_i^{x_i}(1-b_i)^{1-x_i}$. Since x_i is either 0 or 1, we have $|c_i - d_i| = |a_i - b_i| \leq \Delta_i$. Then we can deduce

$$\begin{aligned}
|\prod_{i=1}^{k} c_i - \prod_{i=1}^{k} d_i| &\leq (\prod_{i=1}^{k-1} c_i)|c_k - d_k| + |\prod_{i=1}^{k-1} c_i - \prod_{i=1}^{k-1} d_i| d_k \\
&\leq (\prod_{i=1}^{k-1} c_i)\Delta_k + |\prod_{i=1}^{k-1} c_i - \prod_{i=1}^{k-1} d_i| d_k \\
&\leq (\prod_{i=1}^{k-1} c_i)\Delta_k + (\prod_{i=1}^{k-2} c_i)\Delta_{k-1} d_k + |\prod_{i=1}^{k-2} c_i - \prod_{i=1}^{k-2} d_i| d_{k-1} d_k \\
&\leq \cdots \\
&\leq \sum_{j=1}^{k} (\prod_{i=1}^{j-1} c_i)\Delta_j (\prod_{i=j+1}^{k} d_i).
\end{aligned}$$

When summing up for all binary vectors x, we can write the coefficient of Δ_j as

$$\begin{aligned}\sum_{x\in\{0,1\}^k}(\prod_{i=1}^{j-1}c_i)(\prod_{i=j+1}^{k}d_i) &= (\prod_{i=1}^{j-1}\sum_{x_i\in\{0,1\}}c_i)(\sum_{x_j\in\{0,1\}}1)(\prod_{i=j+1}^{k}\sum_{x_i\in\{0,1\}}d_i)\\ &= (\prod_{i=1}^{j-1}1)2(\prod_{i=j+1}^{k}1)\\ &= 2,\end{aligned}$$

where the second equality holds because $\sum_{x\in\{0,1\}}a^x(1-a)^{1-x} = a+(1-a) = 1$. This completes the proof. □

Now we prove the main theorem.

Theorem 5.5. (Bayesian regret bound of Thompson sampling) *The Bayesian regret of Algorithm 5.1 satisfies the following bound*

$$BR(T) = \mathcal{O}(\sqrt{KL^3N^3T\log T}) = \mathcal{O}(\sqrt{mKL^4N^3\log(mL)}).$$

Proof. We fix an episode l and analyze the regret in this episode. Let $t_l = (l-1)L$ so that the episode starts at time t_l+1. Define

$$N_l(k,r,n) = \sum_{t=1}^{t_l}\mathbb{1}\{A_{t,k}=1, r_k = r, n_k = n\}.$$

It counts the number of rounds where the arm k was chosen by the learner with history $r_k = r$ and $n_k = n$ (see (5.3) for definition). Note that

$$k\in[K], r\in\{0,1,\rho(k)\}, \text{ and } n\in[L],$$

where $\rho(k)$ is the initial success rate of the arm k. This implies there are $3KL$ tuples of (k,r,n).

Let $\omega^\theta(k,r,n)$ denote the conditional probability of $X_k = 1$ given a history (r,n) and a system parameter θ. Also let $\hat{\omega}(k,r,n)$ denote the empirical mean of this quantity (using $N_l(k,r,n)$ past observations and set the estimate to 0 if $N_l(k,r,n) = 0$). Then define

$$\Theta_l = \{\theta \mid \forall(k,r,n),\ \ |(\hat{\omega}-\omega^\theta)(k,r,n)| < \sqrt{\frac{2\log(1/\delta)}{1\vee N_l(k,r,n)}}\}.$$

Since $\hat{\omega}(k, r, n)$ is $\mathcal{H}_{t_l}$-measurable, so is the set Θ_l. Using the Hoeffding inequality, one can show $\mathbb{P}(\theta^\star \notin \Theta_l) = \mathbb{P}(\theta_l \notin \Theta_l) \leq 3\delta KL$.

We now turn our attention to the following Bellman operator

$$\mathcal{T}_{\pi_l}^{\theta} V_{\pi_l,t}^{\theta_l}(\mathcal{H}_{t-1}) = \mathbb{E}_{\theta,\pi_l}[A_{t_l+t} \cdot X_{t_l+t} + V_{\pi_l,t}^{\theta_l}(\mathcal{H}_t)|\mathcal{H}_{t-1}].$$

Since π_l is a deterministic policy, A_{t_l+t} is also deterministic given $\mathcal{H}_{t-1}$ and π_l. Let $(k_1, \ldots, k_N)$ be the active arms at time $t_l + t$ and write $\omega^\theta(k_i, r_{k_i}, n_{k_i}) = \omega_{\theta,i}$. Then we can rewrite

$$\mathcal{T}_{\pi_l}^{\theta} V_{\pi_l,t}^{\theta_l}(\mathcal{H}_{t-1}) = \sum_{i=1}^{N} \omega_{\theta,i} + \sum_{x \in \{0,1\}^N} P_x^\theta V_{\pi_l,t}^{\theta_l}(\mathcal{H}_{t-1} \cup (A_{t_l+t}, x)), \tag{C.2}$$

where $P_x^\theta = \prod_{i=1}^N \omega_{\theta,i}^{x_i}(1 - \omega_{\theta,i})^{1-x_i}$. Under the event that $\theta^\star, \theta_l \in \Theta_l$, we have

$$|\omega_{\theta_l,i} - \omega_{\theta^\star,i}| < 1 \wedge \sqrt{\frac{8\log(1/\delta)}{1 \vee N_l(k_i, r_{k_i}, n_{k_i})}} =: \Delta_i(t_l + t), \tag{C.3}$$

where the dependence on $t_l + t$ comes from the mapping from i to k_i. Lemma C.1 provides

$$\sum_{x \in \{0,1\}^N} |P_x^{\theta_l} - P_x^{\theta^\star}| \leq 2 \sum_{i=1}^{N} \Delta_i(t_l + t). \tag{C.4}$$

From (C.2), (C.4), and the fact that $|V_{\pi,t}^\theta| \leq LN$, we obtain given $\mathcal{H}_{t-1}$ and the event $\theta^\star, \theta_l \in \Theta_l$,

$$|(\mathcal{T}_{\pi_l}^{\theta^\star} - \mathcal{T}_{\pi_l}^{\theta_l}) V_{\pi_l,t}^{\theta_l}(\mathcal{H}_{t-1})| \leq (2LN + 1) \sum_{i=1}^{N} \Delta_i(t_l + t) \leq 3LN \sum_{i=1}^{N} \Delta_i(t_l + t).$$

Then by applying Lemma 5.4, we get

$$|V_{\pi_l,1}^{\theta_l}(\emptyset) - V_{\pi_l,1}^{\theta^\star}(\emptyset)| \leq 3LN\mathbb{E}_{\theta^\star,\pi_l} \sum_{t=1}^{L} \sum_{i=1}^{N} \Delta_i(t_l + t).$$

The above inequality holds whenever $\theta^\star, \theta_l \in \Theta_l$. When $\theta^\star \notin \Theta_l$ or $\theta_l \notin \Theta_l$, which happens with probability less than $6\delta KL$, we have a trivial bound $|V_{\pi_l,1}^{\theta_l}(\emptyset) - V_{\pi_l,1}^{\theta^\star}(\emptyset)| \leq LN$. We can deduce

$$|V_{\pi_l,1}^{\theta_l}(\emptyset) - V_{\pi_l,1}^{\theta^\star}(\emptyset)| \leq 3LN\mathbb{1}(\theta^\star, \theta_l \in \Theta_l)\mathbb{E}_{\theta^\star,\pi_l} \sum_{t=1}^{L} \sum_{i=1}^{N} \Delta_i(t_l + t) + 6\delta KL^2N.$$

Combining this with Lemma 5.3, we can show

$$BR(T) \leq 6\delta mKL^2N + \mathbb{E}_{\theta^\star \sim Q} 3LN \sum_{l=1}^{m} \mathbb{1}(\theta^\star, \theta_l \in \Theta_l) \mathbb{E}_{\theta^\star, \pi_l} \sum_{t=1}^{L} \sum_{i=1}^{N} \Delta_i(t_l + t). \tag{C.5}$$

We further analyze the summation to finish the argument. Note that for this summation, we have $\theta^\star, \theta_l \in \Theta_l$. We shorten $N_l(k_i, r_{k_i}, n_{k_i})$ to N_l for simplicity. By the definition of Δ_i in (C.3), we get

$$\begin{aligned}\sum_{l=1}^{m} \sum_{t=1}^{L} \sum_{i=1}^{N} \Delta_i(t_l + t) &\leq \sum_{l=1}^{m} \sum_{t=1}^{L} \sum_{i=1}^{N} \mathbb{1}\{N_l \leq L\} + \Delta_i \mathbb{1}\{N_l > L\} \\ &\leq 6KL^2 + \sum_{l=1}^{m} \sum_{t=1}^{L} \sum_{i=1}^{N} \mathbb{1}\{N_l > L\} \sqrt{\frac{8\log(1/\delta)}{N_l}},\end{aligned} \tag{C.6}$$

where the second inequality holds because there are $3KL$ possible tuples of (k, r, n) and a tuple can contribute at most $2L$ to the first summation.

We can bound the second term as follows

$$\begin{aligned}\sum_{l=1}^{m} \sum_{t=1}^{L} \sum_{i=1}^{N} \mathbb{1}\{N_l > L\} \sqrt{\frac{1}{N_l}} &= \sum_{l=1}^{m} \sum_{(k,r,n)} \mathbb{1}\{N_l > L\}(N_{l+1} - N_l) \sqrt{\frac{1}{N_l}} \\ &\leq \sum_{l=1}^{m} \sum_{(k,r,n)} (N_{l+1} - N_l) \sqrt{\frac{2}{N_{l+1}}} \\ &\leq \sqrt{8} \sum_{(k,r,n)} \sqrt{N_{m+1}(k, r, n)} \\ &\leq \sqrt{24KLNT}.\end{aligned} \tag{C.7}$$

For the first inequality, we use $N_{l+1} \leq N_l + L \leq 2N_l$. The second inequality holds due to the integral trick. Finally, the last inequality holds by the Cauchy-Schwartz inequality along with the fact that $\sum_{(k,r,n)} N_{m+1}(k, r, n) = NT$.

Combining (C.5), (C.6), (C.7), and our assumption that $T = mL$, we obtain

$$BR(T) = \mathcal{O}(\delta KLNT + KL^3N + \sqrt{KL^3N^3T\log(1/\delta)}).$$

Since NT is a trivial upper bound of $BR(T)$, we may ignore the KL^3N term. Setting $\delta = \frac{1}{T}$ completes the proof. □

APPENDIX D

Details for Thompson Sampling in Non-Episodic Restless Bandits

D.1 Proof of Proposition 6.6

We first prove that Condition 6.4 guarantees the constant average cost and the associated Bellman equation and then show that Condition 6.5 implies Condition 6.4.

1. Let $\theta \in \Theta$ and $\pi_\theta = \mu(\theta)$ satisfy Condition 6.4 for some bounded function $v \in \mathcal{V}$ and constant g. Then, for all $\xi \in S$,

$$r_\theta(\xi, \pi_\theta(\xi)) = g + v(\xi) - \mathbb{E}_\theta[v(\xi')|\xi],$$

where the next "meta"-state $\xi' \sim \mathbb{P}_\theta(\cdot|\xi, A = \pi_\theta(\xi))$ is drawn according to the Markov transition probability of $\{\xi_t\}_{t\geq 1}$ knowing the current state ξ and the action $A = \pi_\theta(\xi)$ under parametrization θ. Thus,

$$\begin{aligned}\sum_{t=1}^{T} r_\theta(\xi_t, \pi_\theta(\xi_t)) &= Tg + \sum_{t=1}^{T} v(\xi_t) - \mathbb{E}_\theta[v(\xi_{t+1})|\xi_t, A_t = \pi_\theta(\xi_t)] \\ &= Tg + \sum_{t=1}^{T} v(\xi_{t+1}) - \mathbb{E}_\theta[v(\xi_{t+1})|\xi_t, A_t = \pi_\theta(\xi_t)] + v(\xi_1) - v(\xi_{T+1}).\end{aligned}$$

Multiplying by $\frac{1}{T}$ both sides of the equation and taking the expectation given ξ_1 leads to

$$\frac{1}{T}\mathbb{E}_\theta\left(\sum_{t=1}^{T} r_\theta(\xi_t, \pi_\theta(\xi_t))|\xi_1\right) = g + \frac{1}{T}\mathbb{E}_\theta\Big(v(\xi_1) - v(\xi_{T+1})|\xi_1\Big).$$

Finally, since v is bounded, letting $T \to \infty$ one has $\frac{1}{T}\mathbb{E}_\theta\Big(v(\xi_1) - v(\xi_{T+1})|\xi_1\Big) \to 0$ and thus

$$J_{\pi_\theta}(\theta) := \lim_{T\to\infty} \frac{1}{T}\mathbb{E}_\theta\left(\sum_{t=1}^{T} r_\theta(\xi_t, \pi_\theta(\xi_t))|\xi_1\right) = g.$$

Futhermore, since g is constant, it ensures that $J_{\pi_\theta}(\theta)$ is independent of the initial state. Replacing g by $J_{\pi_\theta}(\theta)$ in Condition 6.4, we directly obtain that J is associated with the Bellman equation. Since the function v is arbitrary up to constant term (it still satisfies the Bellman equation and does not affect the span), we can set it without loss of generality to be non-negative defining $h_\theta(\xi) = v(\xi) - \inf_\xi v(\xi)$ and the pair $(J_{\pi_\theta}(\theta), h_\theta)$ satisfies the Bellman equation ((6.9)). Additionally, we have

$$C_\theta = \sup_{(\xi,\xi')\in S^2} h_\theta(\xi) - h_\theta(\xi') = \sup_{(\xi,\xi')\in S^2} v(\xi) - v(\xi') < \infty.$$

2. We now show that Condition 6.5 implies Condition 6.4. The proof is adapted from Puterman [2014, Theorem8.10.7] which is derived for optimal policies. The core idea is to consider a sequence of discount factor $\beta_n \to 1$ and to choose an appropriate subsequence (also indexed by n for ease of notation) to assert the existence of g and $v \in \mathcal{V}$ thanks to the uniform boundedness of $|v^\beta_{\pi_\theta}|$.
   First, notice that for all $\xi \in S$, $r_\theta(\xi, \pi_\theta(\xi)) \in [0, N]$ and thus that $v^\beta_{\pi_\theta}(\xi) \in [0, \frac{N}{1-\beta}]$ for all $\beta \in (0, 1)$. Also, it is well known that $v^\beta_{\pi_\theta}(\xi)$ satisfies the discounted Bellman equation:

$$v^\beta_{\pi_\theta} = \mathcal{T}_\beta(v^\beta_{\pi_\theta}), \qquad \text{where } \mathcal{T}_\beta(v^\beta_{\pi_\theta})(\xi) = r_\theta(\xi, \pi_\theta(\xi)) + \beta\mathbb{E}_\theta\big(v^\beta_{\pi_\theta}(\xi')|\xi\big).$$

   Let $\bar{\xi} \in S$ be an arbitrary state and define $\bar{v}^\beta(\xi) = v^\beta_{\pi_\theta}(\xi) - v^\beta_{\pi_\theta}(\bar{\xi})$. Clearly, $\bar{v}^\beta$ is uniformly bounded and $\bar{v}^\beta$ satisfies

$$\bar{v}^\beta + (1-\beta)v^\beta_{\pi_\theta}(\bar{\xi}) = \mathcal{T}_\beta(\bar{v}^\beta). \tag{D.1}$$

   Since $\bar{v}^\beta$ and r_θ are uniformly bounded, so is $(1-\beta)v^\beta_{\pi_\theta}(\bar{\xi})$. Further, the Bolzano-Weierstrass theorem for bounded sequence together with a standard diagonal argument ensures that there exists a subsequence $\beta_n \to 1$ such that

   - $(1-\beta_n)v^{\beta_n}_{\pi_\theta}(\bar{\xi}) \to g$
   - $\bar{v}^{\beta_n}$ converges pointwise to some function $\bar{v}$.

Finally, since $\sup_{(\xi,\xi')\in S^2} v^{\beta}_{\pi_\theta}(\xi) - v^{\beta}_{\pi_\theta}(\xi') = C_\theta$ is uniformly bounded so is $\bar{v}$:

$$\sup_{(\xi,\xi')\in S^2} \bar{v}(\xi) - \bar{v}(\xi') \leq \sup_{(\xi,\xi')\in S^2} \sup_{n\geq 1} \bar{v}^{\beta_n}(\xi) - \bar{v}^{\beta_n}(\xi') \leq \sup_{n\geq 1} \sup_{(\xi,\xi')\in S^2} \bar{v}^{\beta_n}(\xi) - \bar{v}^{\beta_n}(\xi') \leq 2C_\theta.$$

We are now left to check that the pair $(g, \bar{v})$ satisfies the Bellman equation in Condition 6.4. It relies on the following lemma (Lemma 3 in Platzman [1980]).

Lemma D.1. *If $\bar{v}^{\beta_n}$ converges to $\bar{v}$ pointwise, then $\mathcal{T}_1(\bar{v}^{\beta_n})$ converges to $\mathcal{T}_1(\bar{v})$ pointwise.*

Proof. We provide the proof for the sake of completeness. The objective is to prove that for an arbitrary fixed $\xi \in S$, $\epsilon > 0$, there exits a constant M such that $|[\mathcal{T}_1(\bar{v}^{\beta_n}) - \mathcal{T}_1(\bar{v})](\xi)| < \epsilon$ for all $n \geq M$.
Let $\xi \in S$ be an arbitrary state, and define

$$S_\xi = \{\xi' \in S \text{ s.t. } \mathbb{P}_\theta(\xi_{t+1} = \xi' | \xi_t = \xi, A_t = \pi_\theta(\xi)) > 0\}.$$

Notice that since A_t is fully determined by ξ, so is ξ^n_{t+1} and S_ξ is a finite non-empty set of state. Thus, there exists M such that $|\bar{v}^{\beta_n}(\xi') - \bar{v}(\xi')| < \epsilon$ for all $\xi' \in S_\xi$, $n \geq M$. Finally, it leads to

$$|[\mathcal{T}_1(\bar{v}^{\beta_n}) - \mathcal{T}_1(\bar{v})](\xi)| \leq \mathbb{E}_\theta \left(|\bar{v}^{\beta_n}(\xi') - \bar{v}(\xi')| \;\middle|\; \xi \right) \leq \max_{\xi'\in S_\xi} |\bar{v}^{\beta_n}(\xi') - \bar{v}(\xi')| < \epsilon,$$

which proves the desired result. □

Finally, the uniform boundedness of $\bar{v}^{\beta_n}$ implies

$$|\mathcal{T}_{\beta_n}(\bar{v}^{\beta_n}) - \mathcal{T}_1(\bar{v}^{\beta_n})| \to 0,$$

which in addition to Lemma D.1 ensures that

$$|\mathcal{T}_{\beta_n}(\bar{v}^{\beta_n}) - \mathcal{T}_1(\bar{v})| \leq |\mathcal{T}_{\beta_n}(\bar{v}^{\beta_n}) - \mathcal{T}_1(\bar{v}^{\beta_n})| + |\mathcal{T}_1(\bar{v}^{\beta_n}) - \mathcal{T}_1(\bar{v})| \to 0.$$

Taking the limit in (D.1) concludes the proof

$$\bar{v}^{\beta_n} + (1-\beta_n) v^{\beta_n}_{\pi_\theta}(\bar{\xi}) - \mathcal{T}_{\beta_n}(\bar{v}^{\beta_n}) = 0 \quad \Rightarrow \quad \bar{v} + g - \mathcal{T}_1(\bar{v}) = 0.$$

D.2 Regret bound proofs

In this section, we provide full proofs that are sketched in Section 6.5.

D.2.1 Regret decomposition

Let (θ_i, π_i) be the sampled parameter-policy pair used in episode i. From (6.9), one has

$$\begin{aligned}\sum_{t=t_i}^{t_{i+1}-1} r_{\theta^\star}(\xi_t, A_t) &= \sum_{t=t_i}^{t_{i+1}-1} [r_{\theta_i} + (r_{\theta^\star} - r_{\theta_i})](\xi_t, \pi_i(\xi_t)) \\ &= \sum_{t=t_i}^{t_{i+1}-1} [J_{\pi_i}(\theta_i) + v_{\theta_i}(\xi_t) - \mathbb{E}_{\theta_i}[v_{\theta_i}(\xi')|\pi_i, \xi_t] + (r_{\theta^\star} - r_{\theta_i})(\xi_t, \pi_i(\xi_t))].\end{aligned}$$

Using this, we can rewrite the frequentist regret by

$$\begin{aligned}R(T; \theta^\star) &= J_{\pi^\star}(\theta^\star) \cdot T - \mathbb{E}_{\theta^\star} \sum_{i=1}^{M_T} \sum_{t=t_i}^{t_{i+1}-1} r_{\theta^\star}(\xi_t, A_t) \\ &=: R_0 + R_1 + R_2 + R_3,\end{aligned}$$

where

$$\begin{aligned}R_0 &= J_{\pi^\star}(\theta^\star) \cdot T - \mathbb{E}_{\theta^\star} \sum_{i=1}^{M_T} J_{\pi_i}(\theta_i) \cdot T_i \\ R_1 &= \mathbb{E}_{\theta^\star} \sum_{i=1}^{M_T} \sum_{t=t_i}^{t_{i+1}-1} v_{\theta_i}(\xi_{t+1}) - v_{\theta_i}(\xi_t) \\ R_2 &= \mathbb{E}_{\theta^\star} \sum_{i=1}^{M_T} \sum_{t=t_i}^{t_{i+1}-1} \mathbb{E}_{\theta_i}[v_{\theta_i}(\xi')|\pi_i, \xi_t] - v_{\theta_i}(\xi_{t+1}) \\ R_3 &= \mathbb{E}_{\theta^\star} \sum_{i=1}^{M_T} \sum_{t=t_i}^{t_{i+1}-1} (r_{\theta_i} - r_{\theta^\star})(\xi_t, \pi_i(\xi_t)).\end{aligned}$$

D.2.2 Confidence set

We begin with a useful result that is induced by Assumption 6.2.

Proposition D.2. *For any arm $k \in [K]$ and $\theta \in \Theta$, let*

$$p_\theta(s'; k, s, n) = \mathbb{P}_\theta(s_k^t = s' | s_k^{t-n} = s)$$

and $p_\theta(k, s, n)$ be the corresponding distribution over S_k. For any $\epsilon > 0$ and $n, n' > \log_2(1/\epsilon) T^{mix}(\frac{1}{4})$, we have

$$\|p_\theta(k, s, n) - p_\theta(k, s, n')\|_1 \leq 2|S_k|\epsilon.$$

Proof. For any $n \geq 1$, we can write

$$p_\theta(k, s, n) = \left(P_k^{passive}\right)^{n-1} P^{active} e^s,$$

where e^s is a binary vector of size $|S_k|$ with 1 on the s entry and 0 elsewhere. We can deduce

$$\begin{aligned}
\|p_\theta(k, s, n) - p_\theta(k, s, n')\|_1 &\leq |S_k| \max_{s \in S_k} \|\left(\left(P_k^{passive}\right)^{n-1} - \left(P_k^{passive}\right)^{n'-1}\right) e^s\|_1 \\
&\leq 2|S_k| \max_{s \in S_k} \|p_k^{T^{mix}}(s) - p_k\|_1 \\
&\leq 2|S_k|\epsilon,
\end{aligned}$$

where we used the fact $T_k^{\text{mix}}(\epsilon) \leq \log_2(1/\epsilon) T^{\text{mix}}(\frac{1}{4})$, discussed by Ortner et al. [2012, (1)]. □

Now we prove Lemma 6.14.

Lemma 6.14. *On the high-probability event $\theta^\star \in \cap_{i \leq M_T} \Theta_i$, we can show*

$$\Delta_T \leq 12\sqrt{N T^{mix} T \log 1/\delta} \sum_{k=1}^{K} |S_k|.$$

Proof. We work on the high-probability event where $\theta^\star \in \Theta_i$ for all $i \leq M_T$. Thus, from Lemma 6.13 we have $||(\hat{p}_{t_i} - p_{\theta^\star})(\zeta)||_1 \leq c_i(\zeta)$ for all ζ. Hence, we obtain

$$\Delta_T \leq \sum_{i=1}^{M_T} \sum_{t=t_i}^{t_{i+1}-1} \sum_{\text{active arms } k} c_i(k, \sigma_k^t, n_k^t).$$

By the second stopping criterion of TSDE, we have $N_t(\zeta) \leq 2N_{t_i}(\zeta)$ for all t in episode i. Using

this, we can write

$$\sum_{i=1}^{M_T}\sum_{t=t_i}^{t_{i+1}-1}\sum_{\text{active arms } k} c_i(k,\sigma_k^t,n_k^t) \le \sum_{i=1}^{M_T}\sum_{t=t_i}^{t_{i+1}-1}\sum_{\text{active arms } k}\sqrt{\frac{16|S_k|\log 1/\delta}{1\vee N_t(k,\sigma_k^t,n_k^t)}}$$
$$= \sum_{k=1}^{K}\sum_{t=1}^{T}\mathbb{1}(A_{t,k}=1)\sqrt{\frac{16|S_k|\log 1/\delta}{1\vee N_t(k,\sigma_k^t,n_k^t)}}.$$

For each $\zeta=(k,s,n)$, it appears in the above summation exactly $N_{T+1}(\zeta)$ times. That is to say, the above equation can be written as

$$\sum_{\zeta\in Z}\sqrt{16|S_k|\log 1/\delta}\cdot\sum_{j=1}^{N_{T+1}(\zeta)}\frac{1}{\sqrt{1\vee(j-1)}} \le \sum_{\zeta\in Z}12\sqrt{|S_k|N_{T+1}(\zeta)\log 1/\delta}. \tag{D.2}$$

The number of $\zeta=(k,s,n)$ for a fixed k is bounded by $|S_k|T^{\text{mix}}$. Also, we have $\sum_{\zeta\in Z}N_{T+1}(\zeta)=NT$. The Cauchy-Schwartz inequality provides

$$\sum_{s\in S_k}\sum_{n=1}^{T^{\text{mix}}}\sqrt{N_{T+1}(k,s,n)} \le \sqrt{|S_k|T^{\text{mix}}NT}.$$

Finally, we obtain

$$\Delta_T \le 12\sqrt{NT^{\text{mix}}T\log 1/\delta}\sum_{k=1}^{K}|S_k|.$$

□

D.2.3 Bounding R_0 and R_1

Lemma 6.8 (Ouyang et al. [2017], Lemma 3 and 4).

$$\mathbb{E}_{\theta^\star\sim Q}R_0 \le N\cdot\mathbb{E}_{\theta^\star\sim Q}M_T,$$

where M_T is the total number of episodes until time T.

Proof. By definition in (6.8), we have $0\le J_\pi(\theta)\le N$ for all π and θ. For ease of analysis, let us write

$$J_{\pi^\star}(\theta^\star)=N-J^\star \text{ and } J_{\pi_i}(\theta_i)=N-J_i.$$

Since $M_T \leq T$ almost surely, we can rewrite

$$\begin{aligned} R_0 &= J_{\pi^\star}(\theta^\star) \cdot T - \mathbb{E}_{\theta^\star} \sum_{i=1}^{T} \mathbb{1}(t_i \leq T) J_{\pi_i}(\theta_i) \cdot T_i \\ &= \mathbb{E}_{\theta^\star} \sum_{i=1}^{T} \mathbb{1}(t_i \leq T) J_i \cdot T_i - J^\star \cdot T. \end{aligned}$$

Due to the first stopping criterion of TSDE, we have $T_i \leq T_{i-1} + 1$ for all i. Using this, we can deduce

$$R_0 \leq \mathbb{E}_{\theta^\star} \sum_{i=1}^{T} \mathbb{1}(t_i \leq T) J_i \cdot (T_{i-1} + 1) - J^\star \cdot T.$$

In the meantime, note that $\mathbb{1}(t_i \leq T) J_i \cdot (T_{i-1} + 1)$ is a $\mathcal{H}_{t_i}$-measurable function of θ_i. Thus Lemma 6.7 implies

$$\mathbb{E}\mathbb{1}(t_i \leq T) J_i \cdot (T_{i-1} + 1) = \mathbb{E}\mathbb{1}(t_i \leq T) J^\star \cdot (T_{i-1} + 1).$$

Using this, we obtain

$$\mathbb{E}_{\theta^\star \sim Q} R_0 \leq \mathbb{E} \sum_{i=1}^{T} \mathbb{1}(t_i \leq T) J^\star \cdot (T_{i-1} + 1) - J^\star \cdot T = \mathbb{E} J^\star \cdot M_T.$$

Since $J^\star \leq N$ almost surely, this completes the proof. □

Lemma 6.9.

$$R_1 \leq H \cdot \mathbb{E} M_T.$$

Proof. For a fixed episode i, the telescope rule gives

$$\sum_{t=t_i}^{t_{i+1}-1} v_{\theta_i}(\xi_{t+1}) - v_{\theta_i}(\xi_t) = v_{\theta_i}(\xi_{t_{i+1}}) - v_{\theta_i}(\xi_{t_i}),$$

which is less than H by the assumption. Summing over the episodes concludes the argument. □

D.2.4 Bounding R_2 and R_3.

Before delving into bounding R_2 and R_3, we record a technical lemma, which generalizes Lemma C.1.

Lemma D.3. *Suppose a_k and b_k are probability distributions over a set $[n_k]$ for $k \in [K]$. Then we have*

$$\sum_{x \in \otimes_{k=1}^{K}[n_k]} |\prod_{k=1}^{K} a_{k,x_k} - \prod_{k=1}^{K} b_{k,x_k}| \leq \sum_{k=1}^{K} ||a_k - b_k||_1.$$

Proof. Fix a vector x. For simplicity, let $\alpha_k = a_{k,x_k}$, $\beta_k = b_{k,x_k}$, and $\delta_k = |\alpha_k - \beta_k|$. We may write

$$\begin{aligned}
|\prod_{k=1}^{K} \alpha_k - \prod_{k=1}^{K} \beta_k| &\leq (\prod_{k=1}^{K-1} \alpha_k)|\alpha_K - \beta_K| + |\prod_{k=1}^{K-1} \alpha_k - \prod_{k=1}^{K-1} \beta_k|\beta_K \\
&= (\prod_{k=1}^{K-1} \alpha_k)\delta_K + |\prod_{k=1}^{K-1} \alpha_k - \prod_{k=1}^{K-1} \beta_k|\beta_K \\
&\leq \cdots \\
&\leq \sum_{k=1}^{K} (\prod_{j=1}^{k-1} \alpha_j)\delta_k(\prod_{j=k+1}^{K} \beta_j) \\
&= \sum_{k=1}^{K} (\prod_{j=1}^{k-1} a_{j,x_j})|a_{k,x_k} - b_{k,x_k}|(\prod_{j=k+1}^{K} b_{j,x_j}).
\end{aligned}$$

When summing the last term for all possible vectors x, the coefficient of $|a_{k,x_k} - b_{k,x_k}|$ becomes 1 because a_k and b_k are probability distributions. Then we get the desired inequality. □

Lemma 6.10. *R_2 satisfies the following bound*

$$R_2 \leq 28H \sum_{k=1}^{K} |S_k| \sqrt{NT^{mix}T \log(T^{mix}T)}.$$

Proof. In episode i, ξ_{t+1} evolves from ξ_t on the system $\theta^\star$ with the action $A_t = \pi_i(\xi_t)$. From this, we can rewrite

$$R_2 = \mathbb{E}_{\theta^\star} \sum_{i=1}^{M_T} \sum_{t=t_i}^{t_{i+1}-1} (\mathbb{E}_{\theta_i} - \mathbb{E}_{\theta^\star})[v_{\theta_i}(\xi')|\pi_i, \xi_t].$$

Since $|v_{\theta_i}(\xi')| \leq H$, the individual difference becomes

$$\sum_{\xi' \in S} (\mathbb{P}_{\pi_i(\xi_t)}(\xi_t, \xi'|\theta_i) - \mathbb{P}_{\pi_i(\xi_t)}(\xi_t, \xi'|\theta^\star))v_{\theta_i}(\xi') \leq H \sum_{\xi' \in S} |\mathbb{P}_{\pi_i(\xi_t)}(\xi_t, \xi'|\theta_i) - \mathbb{P}_{\pi_i(\xi_t)}(\xi_t, \xi'|\theta^\star)|.$$

Once the action $\pi_i(\xi_t)$ is fixed, $\xi_t^n = (n_1^t, \cdots, n_K^t)$ evolves in a deterministic manner. Only σ_k^t for

the active arms k will be updated. Then we may write

$$\mathbb{P}_{\pi_i(\xi_t)}(\xi_t, \xi'|\theta) = \prod_{\text{active arms } k} p_\theta(\sigma'_k; k, \sigma_k^t, n_k^t),$$

where $p_\theta(k, s, n)$ is defined earlier in the section. Using Lemma D.3, we obtain

$$\sum_{\xi' \in S} |\mathbb{P}_{\pi_i(\xi_t)}(\xi_t, \xi'|\theta_i) - \mathbb{P}_{\pi_i(\xi_t)}(\xi_t, \xi'|\theta^\star)| \leq \sum_{\text{active arms } k} ||(p_{\theta_i} - p_{\theta^\star})(k, \sigma_k^t, n_k^t)||_1. \tag{D.3}$$

If $\theta^\star, \theta_i \in \Theta_i$, we can apply Lemma 6.14 to upper bound the cumulative sum. If not, the entire summation is bounded by 2. From these and Lemma 6.13 and 6.14, we obtain

$$\begin{aligned} R_2 &\leq H(R_2^0 + R_2^1), \text{ where} \\ R_2^0 &= 24\sqrt{NT^{\text{mix}}T \log 1/\delta} \sum_{k=1}^K |S_k|, \\ R_2^1 &= 4\delta T^{\text{mix}} T \sum_{k=1}^K |S_k|. \end{aligned} \tag{D.4}$$

We finish the proof by setting $\delta = \frac{1}{T^{\text{mix}}T}$ in (D.4). □

Lemma 6.11. *R_3 satisfies the following bound*

$$R_3 \leq 28 \sum_{k=1}^K |S_k| \sqrt{NT^{mix}T \log(T^{mix}T)}.$$

Proof. We begin by investigating the individual term

$$\begin{aligned} (r_{\theta_i} - r_{\theta^\star})(\xi_t, \pi_i(\xi_t)) &= \sum_{\text{active arms } k} (\mathbb{E}_{\theta_i} - \mathbb{E}_{\theta^\star})[r_k(s_k^t)|\xi_t, \pi_i(\xi_t)] \\ &= \sum_{\text{active arms } k} \sum_{s' \in S_k} r_k(s')(p_{\theta_i} - p_{\theta^\star})(s'; k, \sigma_k^t, n_k^t) \\ &\leq \sum_{\text{active arms } k} \sum_{s' \in S_k} ||(p_{\theta_i} - p_{\theta^\star})(k, \sigma_k^t, n_k^t)||_1, \end{aligned}$$

where the last inequality holds by the assumption $r_k(s_k) \leq 1$. The last term actually appears in (D.3) from the proof of Lemma 6.10, and we can use the same argument to obtain the desired bound. □

BIBLIOGRAPHY

Jacob D Abernethy, Young Hun Jung, Chansoo Lee, Audra McMillan, and Ambuj Tewari. Online learning via the differential privacy lens. In *Advances in Neural Information Processing Systems*, pages 8892–8902, 2019.

Sahand Haji Ali Ahmad, Mingyan Liu, Tara Javidi, Qing Zhao, and Bhaskar Krishnamachari. Optimality of myopic sensing in multichannel opportunistic access. *IEEE Transactions on Information Theory*, 55(9):4040–4050, 2009.

Erin L Allwein, Robert E Schapire, and Yoram Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1(Dec):113–141, 2000.

Dimitri P Bertsekas. *Dynamic programming and optimal control*, volume 2. Athena scientific Belmont, MA, 1995.

Dimitris Bertsimas and José Niño-Mora. Restless bandits, linear programming relaxations, and a primal-dual index heuristic. *Operations Research*, 48(1):80–90, 2000.

Alina Beygelzimer, Satyen Kale, and Haipeng Luo. Optimal and adaptive algorithms for online boosting. In *Proceedings of the International Conference on Machine Learning*, 2015.

Alina Beygelzimer, Francesco Orabona, and Chicheng Zhang. Efficient online bandit multiclass learning with $\tilde{O}(\sqrt{T})$ regret. In *Proceedings of the International Conference on Machine Learning*, pages 488–497, 2017.

Ezio Biglieri, Andrea J Goldsmith, Larry J Greenstein, Narayan B Mandayam, and H Vincent Poor. *Principles of cognitive radio*. Cambridge University Press, 2013.

C.L. Blake and C.J. Merz. UCI machine learning repository, 1998. URL http://archive.ics.uci.edu/ml.

Vincent D Blondel and John N Tsitsiklis. A survey of computational complexity results in systems and control. *Automatica*, 36(9):1249–1274, 2000.

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

Nicolo Cesa-Bianchi and Gábor Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012.

Shang-Tse Chen, Hsuan-Tien Lin, and Chi-Jen Lu. An online boosting algorithm with theoretical justifications. In *Proceedings of the International Conference on Machine Learning*, 2012.

Shang-Tse Chen, Hsuan-Tien Lin, and Chi-Jen Lu. Boosting with online binary learners for the multiclass bandit problem. In *Proceedings of The International Conference on Machine Learning*, pages 342–350, 2014.

Haibin Cheng, Roelof van Zwol, Javad Azimi, Eren Manavoglu, Ruofei Zhang, Yang Zhou, and Vidhya Navalpakkam. Multimedia features for click prediction of new ads in display advertising. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 777–785, 2012.

Weiwei Cheng, Eyke Hüllermeier, and Krzysztof J Dembczynski. Bayes optimal multilabel classification via probabilistic classifier chains. In *Proceedings of the International Conference on Machine Learning*, 2010.

Wenhan Dai, Yi Gai, Bhaskar Krishnamachari, and Qing Zhao. The non-bayesian restless multi-armed bandit: A case of near-logarithmic regret. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2940–2943. IEEE, 2011.

Wenhan Dai, Yi Gai, and Bhaskar Krishnamachari. Online learning for multi-channel opportunistic access over unknown markovian channels. In *IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 64–71. IEEE, 2014.

Amit Daniely, Sivan Sabato, Shai Ben-David, and Shai Shalev-Shwartz. Multiclass learnability and the erm principle. In *Conference on Learning Theory*, pages 207–232, 2011.

Krzysztof Dembczynski and Eyke Hüllermeier. Consistent multilabel ranking through univariate loss minimization. In *Proceedings of the International Conference on Machine Learning*, 2012.

Pedro Domingos and Geoff Hulten. Mining high-speed data streams. In *Proceedings of the International Conference on Knowledge Discovery and Data mining*, 2000.

Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

Yoav Freund, Robert Schapire, and Naoki Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

Wei Gao and Zhi-Hua Zhou. On the consistency of multi-label learning. In *Proceedings of the annual Conference on Learning Theory*, 2011.

John C Gittins, Kevin D Glazebrook, Richard Weber, and Richard Weber. *Multi-armed bandit allocation indices*, volume 25. Wiley Online Library, 1989.

Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

Alfred Olivier Hero, David Castañón, Doug Cochran, and Keith Kastella. *Foundations and applications of sensor management*. Springer Science & Business Media, 2007.

Cios KJ Higuera C, Gardiner KJ. Self-organizing feature maps identify proteins critical to learning in a mouse model of down syndrome. *PLoS ONE*, 2015. URL https://doi.org/10.1371/journal.pone.0129126.

Hanzhang Hu, Wen Sun, Arun Venkatraman, Martial Hebert, and Andrew Bagnell. Gradient boosting on stochastic data streams. In *Artificial Intelligence and Statistics*, pages 595–603, 2017.

Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(Apr):1563–1600, 2010.

Tara Javidi, Bhaskar Krishnamachari, Qing Zhao, and Mingyan Liu. Optimality of myopic sensing in multi-channel opportunistic access. In *2008 IEEE International Conference on Communications*, pages 2107–2112. IEEE, 2008.

Young Hun Jung and Ambuj Tewari. Online boosting algorithms for multi-label ranking. In *Artificial Intelligence and Statistics*, 2018.

Young Hun Jung and Ambuj Tewari. Regret bounds for thompson sampling in episodic restless bandit problems. In *Advances in Neural Information Processing Systems*, pages 9005–9014, 2019.

Young Hun Jung, Jack Goetz, and Ambuj Tewari. Online multiclass boosting. In *Advances in neural information processing systems*, pages 919–928, 2017.

Young Hun Jung, Marc Abeille, and Ambuj Tewari. Thompson sampling in non-episodic restless bandits. *arXiv preprint arXiv:1910.05654*, 2019.

Sham M Kakade, Shai Shalev-Shwartz, and Ambuj Tewari. Efficient bandit algorithms for online multiclass prediction. In *Proceedings of the International Conference on Machine Learning*, pages 440–447. ACM, 2008.

Marcin Korytkowski, Leszek Rutkowski, and Rafał Scherer. Fast image classification by boosting fuzzy classifiers. *Information Sciences*, 327:175–182, 2016.

Oluwasanmi O Koyejo, Nagarajan Natarajan, Pradeep K Ravikumar, and Inderjit S Dhillon. Consistent multilabel classification. In *Advances in Neural Information Processing Systems*, 2015.

Tor Lattimore and Csaba Szepesvári. Bandit algorithms. *preprint*, 2018.

Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine learning*, 2(4):285–318, 1988.

Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. In *Foundations of Computer Science, 1989., 30th Annual Symposium on*. IEEE, 1989.

Haoyang Liu, Keqin Liu, and Qing Zhao. Logarithmic weak regret of non-bayesian restless multi-armed bandit. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1968–1971. IEEE, 2011.

Haoyang Liu, Keqin Liu, and Qing Zhao. Learning in a changing world: Restless multiarmed bandit with unknown dynamics. *IEEE Transactions on Information Theory*, 59(3):1902–1916, 2013.

Keqin Liu and Qing Zhao. Indexability of restless bandit problems and optimality of whittle index for dynamic multichannel access. *IEEE Transactions on Information Theory*, 56(11):5547–5567, 2010.

Rahul Meshram, Aditya Gopalan, and D Manjunath. Optimal recommendation to users that react: Online learning for a class of pomdps. In *IEEE 55th Conference on Decision and Control (CDC)*, pages 7210–7215. IEEE, 2016.

Rahul Meshram, Aditya Gopalan, and D Manjunath. Restless bandits that hide their hand and recommendation systems. In *IEEE International Conference on Communication Systems and Networks (COMSNETS)*, pages 206–213. IEEE, 2017.

Rahul Meshram, D Manjunath, and Aditya Gopalan. On the whittle index for restless multiarmed hidden markov bandits. *IEEE Transactions on Automatic Control*, 63(9):3046–3053, 2018.

Indraneel Mukherjee and Robert E Schapire. A theory of multiclass boosting. *Journal of Machine Learning Research*, 14(Feb):437–497, 2013.

Ronald Ortner, Daniil Ryabko, Peter Auer, and Rémi Munos. Regret bounds for restless markov bandits. In *International Conference on Algorithmic Learning Theory*, pages 214–228. Springer, 2012.

Ian Osband and Benjamin Van Roy. Near-optimal reinforcement learning in factored mdps. In *Advances in Neural Information Processing Systems*, pages 604–612, 2014.

Ian Osband, Daniel Russo, and Benjamin Van Roy. (more) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems*, pages 3003–3011, 2013.

Yi Ouyang, Mukul Gagrani, Ashutosh Nayyar, and Rahul Jain. Learning unknown markov decision processes: A thompson sampling approach. In *Advances in Neural Information Processing Systems*, pages 1333–1342, 2017.

Nikunj C Oza. Online bagging and boosting. In *2005 IEEE international conference on systems, man and cybernetics*, volume 3. IEEE, 2005.

Christos H Papadimitriou and John N Tsitsiklis. The complexity of optimal queuing network control. *Mathematics of Operations Research*, 24(2):293–305, 1999.

Loren K Platzman. Optimal infinite-horizon undiscounted control of finite probabilistic systems. *SIAM Journal on Control and Optimization*, 18(4):362–380, 1980.

Martin L Puterman. *Markov Decision Processes.: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952.

Daniel Russo and Benjamin Van Roy. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243, 2014.

Robert E Schapire. Drifting games. *Machine Learning*, 43(3):265–291, 2001.

Robert E Schapire and Yoav Freund. *Boosting: Foundations and algorithms*. MIT press, 2012.

Robert E Schapire and Yoram Singer. Boostexter: A boosting-based system for text categorization. *Machine learning*, 39(2-3):135–168, 2000.

Eric V Slud. Distribution inequalities for the binomial law. *The Annals of Probability*, pages 404–412, 1977.

Cem Tekin and Mingyan Liu. Online learning of rested and restless bandits. *IEEE Transactions on Information Theory*, 58(8):5588–5611, 2012.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

Grigorios Tsoumakas, Eleftherios Spyromitros-Xioufis, Jozef Vilcek, and Ioannis Vlahavas. Mulan: A java library for multi-label learning. *Journal of Machine Learning Research*, 12:2411–2414, 2011.

Wallace Ugulino, Débora Cardador, Katia Vega, Eduardo Velloso, Ruy Milidiú, and Hugo Fuks. Wearable computing: Accelerometers' data classification of body postures and movements. In *Advances in Artificial Intelligence-SBIA 2012*, pages 52–61. Springer, 2012.

Volodimir G Vovk. Aggregating strategies. In *Proc. Third Workshop on Computational Learning Theory*, pages 371–383. Morgan Kaufmann, 1990.

Richard R Weber and Gideon Weiss. On an index policy for restless bandits. *Journal of Applied Probability*, 27(3):637–648, 1990.

Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdu, and Marcelo J Weinberger. Inequalities for the l1 deviation of the empirical distribution. *Hewlett-Packard Labs, Tech. Rep*, 2003.

Peter Whittle. Restless bandits: Activity allocation in a changing world. *Journal of applied probability*, 25(A):287–298, 1988.

Daniel T Zhang, Young Hun Jung, and Ambuj Tewari. Online boosting for multilabel ranking with top-k feedback. *arXiv preprint arXiv:1910.10937*, 2019.

Xiao-Lei Zhang and DeLiang Wang. Boosted deep neural networks and multi-resolution cochleagram features for voice activity detection. In *INTERSPEECH*, pages 1534–1538, 2014.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the International Conference on Machine Learning*, 2003.