Topics in Learning Theory: Prediction, Estimation, and Partial Information

by

Vinod Raman

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Statistics)
in The University of Michigan
2025

Doctoral Committee:

Professor Ambuj Tewari, Chair
Professor Saptarshi Chakraborty
Professor Wei Hu
Professor Clayton Scott

Vinod Raman
vkraman@umich.edu
ORCID iD: 0000-0002-3221-1908

DEDICATION

I dedicate this thesis to my parents and younger brother.

ACKNOWLEDGEMENTS

My PhD journey could not have been possible without the support of countless number of people. These include, but are not limited to: Ambuj Tewari, Unique Subedi, Yash Patel, Saptarshi Roy, Seamus Somerstep, Sahana Rayan, Marc Brooks, Huey Sun, Kerby Shedden, Judy McDonald, Becca Usoff, Kunal Talwar, Hilal Asi, Satyen Kale, Matthew Joseph, Travis Dick, Shaddin Dughmi, Steve Hanneke, Andrej Lenert, Mahdi Cheraghchi, Sindhu Kutty, my parents and younger brother, and perhaps most important of all, my partner in crime, Amanda Gregolynskyj. I am indebted to every one of you for believing in me when I could not believe in myself. Thank you.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

LIST OF APPENDICES

ABSTRACT

Prediction and estimation are two fundamental pillars of modern statistics and machine learning, unified by the challenge of partial information. In many learning scenarios, one must make predictions without always observing the true response, or estimate unknown quantities from only a finite sample. Even when feedback is available, it may be incomplete or live in a space that does not match the predictions. This thesis investigates how partial information shapes what can and cannot be learned or estimated, focusing on online learning and sample-dependent (missing) mass estimation.

The first three chapters study sequential prediction under full-information feedback. Chapter 2 characterizes multiclass online learnability when the label space is unbounded, resolving an open question posed by Daniely et al. [2015]. Chapter 3 connects learning-augmented algorithms with online classification, showing that predictions of future covariates can strictly improve online performance. Chapter 4 moves beyond classification to sequential prediction in discrete-time dynamical systems, revealing that predicting future system states behaves fundamentally differently from standard prediction tasks such as classification and regression.

The next three chapters examine online learning under various forms of partial information. Chapter 5 provides a complete characterization of multiclass online learnability under bandit feedback, even with unbounded label spaces. Chapter 6 revisits the classical apple tasting model of Helmbold et al. [2000a], where the true label is revealed only when the learner predicts "1," and develops new combinatorial tools that yield sharper characterizations of learnability under such asymmetric feedback. Chapter 7 studies multilabel ranking, a setting where the learner predicts a permutation but receives only a binary relevance vector, and gives a full combinatorial description of learnability in both realizable and agnostic regimes.

The final chapter turns from learning to estimation, focusing on missing mass estimation in a general, sample-dependent form. Given a sample from an unknown distribution, the goal is to estimate the probability mass of a subset of the domain determined by the sample itself. We introduce a broad framework for such tasks, specified by functions $g : \mathcal{X}^\star \to 2^{\mathcal{X}}$, and identify wide conditions under which simple estimators achieve vanishing minimax error.

We also construct examples demonstrating inherent limits to estimability, highlighting the boundary between what can and cannot be inferred from finite data.

CHAPTER 1

Introduction

Prediction and estimation are two of the central themes in learning theory. In both cases, a learner or statistician must make inferences from limited information. In many realistic settings, the learner cannot always observe the true outcome of a prediction, and the statistician cannot fully recover the structure of an unknown distribution from a finite sample. Although these two problems arise in different forms, they are linked by a common challenge: how to reason and act under partial information.

This thesis studies this challenge from a theoretical perspective. It develops a unified view of learning and estimation when feedback is missing, incomplete, or mismatched with the quantities being predicted. The thesis focuses on two broad themes:

1. **Sequential prediction (online learning)**, where decisions must be made over time against an adaptive or adversarial environment.

2. **Sample-dependent mass estimation**, where the quantity to be estimated can depend on the portion of the distribution that the sample fails to reveal.

Across these settings, the work identifies combinatorial principles that determine what is learnable or estimable, and develops new tools that precisely characterize the limitations imposed by partial information. In online learning, we examine how different feedback models (full-information, bandit, asymmetric, and mismatched) lead to different complexity measures and different notions of learnability. In estimation, we explore a general framework in which the statistician must infer the mass of a sample-dependent subset of the domain – a broad generalization of the classical missing-mass problem.

The results give sharp thresholds for when learning or estimation is possible, and reveal several surprising phenomena. For instance, in multiclass online learning, the label space may be infinite without affecting learnability; in apple tasting, achievable mistake rates fall into only three possible regimes; in multilabel ranking, large families of ranking losses are equivalent from the standpoint of learnability; and in sample-dependent mass estimation,

simple estimators succeed for many problems, but certain targets provably cannot be estimated at all.

1.1 Overview of Chapters

The thesis is organized into seven research chapters. Each chapter is self-contained, with its own set of notations and definitions. Accordingly, we omit preliminaries here.

Chapters 2–4: Sequential prediction with full-information feedback

Chapter 2. This chapter settles a fundamental question in multiclass online learning: when is a concept class learnable in the agnostic setting? Prior work resolved this question when the label space is finite. This chapter proves that the answer is unchanged even when the label space is infinite: a class is agnostically online learnable if and only if it has finite Littlestone dimension. The proof introduces a new expert-based algorithm constructed from restricted executions of the Standard Optimal Algorithm. The chapter also shows that online uniform convergence is not equivalent to online learnability, and introduces the sequential graph dimension as the right measure for controlling sequential Rademacher complexity in multiclass prediction.

Chapter 3. This chapter studies online classification when the learner receives predictions about future inputs. Motivated by learning-augmented algorithms, the chapter shows that such side-information can improve accuracy, sometimes dramatically. It proves faster rates in the realizable case, establishes matching lower bounds, and clarifies how future information changes the difficulty of online classification.

Chapter 4. Chapter 4 extends sequential prediction well beyond classification, turning to discrete-time dynamical systems. This chapter defines appropriate complexity measures, new definitions of regret, including Markovian and flow regret, and proves that prediction in dynamical systems behaves in ways sharply distinct from classical prediction problems. In particular, realizable and agnostic learnability hinge on new dimensions tailored to state-transition structure, thereby linking online learning with modern control theory.

Chapters 5–7: Online learning with partial or mismatched feedback

Chapter 5. This chapter studies multiclass online learning under bandit feedback, where the learner is told only whether its prediction was correct. Earlier results showed that the

Bandit Littlestone dimension characterizes learnability when the label space is finite. This chapter extends the characterization to the general case: finite Bandit Littlestone dimension is necessary and sufficient for learnability even with infinitely many labels. The chapter also demonstrates that sequential uniform convergence, while necessary, is not sufficient in the bandit setting.

Chapter 6. This chapter revisits the classical apple tasting problem, where the true label is revealed only when the learner predicts "1." Despite its long history, key questions about its structure were unresolved. This chapter gives a complete characterization of online learnability under apple tasting feedback. In the agnostic setting, regret is controlled by the Littlestone dimension. In the realizable case, a new combinatorial parameter – the Effective Width – captures the difficulty of the problem and leads to a clean trichotomy: for any hypothesis class, achievable mistake rates grow like $\Theta(1)$, $\Theta(\sqrt{T})$, or $\Theta(T)$.

Chapter 7. This chapter focuses on multilabel ranking, where the learner predicts a permutation of objects but receives only a binary relevance vector as feedback. Because the prediction and feedback spaces differ, the problem raises basic questions about what can be learned from incomplete feedback. The chapter gives complete learnability characterizations for a broad family of ranking losses in both the PAC and online settings, and identifies two equivalence classes of losses that capture most cases used in practice.

Chapter 8: Sample-dependent mass estimation

Chapter 8. The final research chapter moves from learning to estimation. It introduces a general framework for sample-dependent mass estimation, where the target of estimation is the probability mass of a subset of the domain that depends on the observed sample. This framework includes the missing-mass problem but is much broader. The chapter identifies large classes of sample-dependent functionals that can be estimated by simple methods such as empirical or leave-one-out estimators, and also constructs sample-dependent targets that cannot be estimated at all, thereby clarifying the limits of sample-dependent inference.

Chapter 9: Future directions

Chapter 9. The final chapter discusses future directions and open problems suggested by the previous chapters.

CHAPTER 2

Multiclass Online Learning and Uniform Convergence

In this chapter, we study multiclass classification in the agnostic adversarial online learning setting. As our main result, we prove that any multiclass concept class is agnostically learnable if and only if its Littlestone dimension is finite. This solves an open problem studied by Daniely, Sabato, Ben-David, and Shalev-Shwartz (2011,2015) who handled the case when the number of classes (or labels) is bounded. We also prove a separation between online learnability and online uniform convergence by exhibiting an easy-to-learn class whose sequential Rademacher complexity is unbounded.

Our learning algorithm uses the multiplicative weights algorithm, with a set of experts defined by executions of the Standard Optimal Algorithm on subsequences of size Littlestone dimension. We argue that the best expert has regret at most Littlestone dimension relative to the best concept in the class. This differs from the well-known covering technique of Ben-David, Pál, and Shalev-Shwartz (2009) for binary classification, where the best expert has regret zero.

2.1 Introduction

Many important machine learning tasks involve a large prediction space; for instance, in language models the prediction space corresponds to the language size (number of words). Other examples include recommendation systems, image object recognition, protein folding prediction, and more.

Consequently, multiclass prediction problems have been studied extensively in the literature. Natarajan and Tadepalli [1988] and Natarajan [1989] initiated the study of multiclass prediction in the basic PAC setting. They characterized the concept classes that satisfy uniform convergence via a natural combinatorial parameter called the graph dimension. They further characterized PAC learnability in the case when the number of labels is bounded via another combinatorial parameter called the Natarajan dimension. Whether the Natarajan dimension characterizes learnability in general (i.e., even when the number of labels can be infinite) has remained open, until recently Brukhim, Carmon, Dinur, Moran, and Yehudayoff [2022] exhibited an unlearnable concept class with Natarajan dimension 1. Brukhim et al. [2022] showed that multiclass PAC learnability is in fact captured by a different combinatorial parameter which they called the Daniely Shalev-Shwartz (DS) dimension, after Daniely and Shalev-Shwartz [2014] who defined it.

Remarkably, the nature of multiclass PAC learnability with a bounded label space is very different than the one when the number of labels is infinite. For instance, in the former PAC learnability and uniform convergence are equivalent[1]. In contrast, already in the 80's, Natarajan [1988] demonstrated an easy-to-learn concept class over an infinite label space which does not satisfy uniform convergence. More recently, Daniely, Sabato, Ben-David, and Shalev-Shwartz [2011], Daniely and Shalev-Shwartz [2014] studied variants of the ERM principle in multiclass learning, and even demonstrated a PAC learnable class which cannot be learned properly (i.e., by learners whose hypothesis is always an element of the concept class).

How about agnostic versus realizable PAC learning? Here, it turns out that the two are equivalent; that is, any class that is learnable in the realizable case is also learnable in the agnostic case. This was shown by David, Moran, and Yehudayoff [2016] using sample compression arguments.

Perhaps surprisingly, the corresponding questions in the setting of online multiclass classification are still open. Daniely, Sabato, Ben-David, and Shalev-Shwartz [2011] initiated the study of online multiclass classification and show that, like in the binary case, the Littlestone dimension characterizes learnability in the realizable setting. They also studied the agnostic

[1]This equivalence yields the fundamental empirical risk minimization principle in PAC learning.

setting, and showed that when the number of labels is finite, then agnostic learnability is equivalent to realizable case learnability. They left open whether this equivalence extends to unbounded label space (or equivalently whether the dependence on the number of labels in the optimal regret can be removed). In this work we resolve this question by showing that agnostic- and realizable-case learnability remain equivalent, even when the number of labels is infinite.

How about uniform convergence versus learnability? In the past 15 years an online analogue of uniform convergence has emerged from the introduction of the sequential Rademacher complexity [Rakhlin, Sridharan, and Tewari, 2010, 2015a], and of the adversarial laws of large numbers framework [Alon, Ben-Eliezer, Dagan, Moran, Naor, and Yogev, 2021b]. This raises the question whether online uniform convergence and online learnability are equivalent. When the number of labels is finite, known results imply that indeed the two are equivalent. How about classes with an infinite number of labels? In this work we resolve this question by introducing a combinatorial parameter which we call the sequential graph dimension that characterizes online uniform convergence in the multiclass setting. Furthermore we identify an online learnable class with an unbounded sequential graph dimension, thus separating online uniform convergence from online learnability.

In both the PAC and online settings, our interest in studying unbounded label spaces has multiple motivations. One main interest is in establishing sharp enough guarantees to reflect the intuitive fact that the optimal performance should not inherently depend on the number of possible labels: that is, that the latter has no explicit significance to the optimal sample complexity (in PAC learning) or regret (in online learning), even when finite. As many modern learning problems have enormous label spaces (e.g., face recognition), this is quite relevant, and has recently been studied in the machine learning literature under the name "extreme classification", where the number of possible labels is vast (possibly exceeding even the data set size). More abstractly, often in mathematics it is the case that infinities clarify concepts and phenomena (e.g., the notion of continuity, based on limits). Similarly, focusing on infinite label spaces is a natural way to abstract away an irrelevant detail, which helps to clarify what is the "correct" way to approach the problem.

2.1.1 Setting and Summary of Main Results

We begin with the basic setup. Let $\mathcal{X}$ and $\mathcal{Y}$ be arbitrary non-empty sets, called the instance space and label space, respectively. Let $\mathbb{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be an arbitrary set of functions, called the concept class. Our results will hold for any $(\mathcal{X}, \mathcal{Y}, \mathbb{H})$, and hence are fully general.

Denote by $\Pi(\mathcal{Y})$ the set of probability measures on $\mathcal{Y}$.[2] A learning algorithm is a mapping $\mathbb{A} : (\mathcal{X} \times \mathcal{Y})^* \times \mathcal{X} \to \Pi(\mathcal{Y})$. Intuitively, $\mathbb{A}(X_1, Y_1, \ldots, X_{t-1}, Y_{t-1}, X_t)$ outputs a prediction $\hat{y}_t$ of the label Y_t of the test point X_t, after observing the history $(X_1, Y_1), \ldots, (X_{t-1}, Y_{t-1})$ and the test point X_t. The output space $\Pi(\mathcal{Y})$ generalizes this to allow for randomized algorithms: that is, the predicted label $\hat{y}_t$ may be randomized. This is known to be necessary for agnostic online learning [Cesa-Bianchi and Lugosi, 2006]. For simplicity, when $\mathbb{A}$ and $X_1, Y_1, \ldots, X_{t-1}, Y_{t-1}, X_t$ are clear from the context, we denote by $p_t \in \Pi(\mathcal{Y})$ the output $\mathbb{A}(X_1, Y_1, \ldots, X_{t-1}, Y_{t-1}, X_t)$, and for any $y \in \mathcal{Y}$ we denote by $p_t(y)$ the probability of the singleton set $\{y\}$ under the probability measure p_t.

For any $T \in \mathbb{N}$, called the time horizon, the regret of a learning algorithm $\mathbb{A}$ is defined as

$$\text{regret}(\mathbb{A}, T) = \sup_{(X_1,Y_1),\ldots,(X_T,Y_T)} \left(\sum_{t=1}^{T} (1 - p_t(Y_t)) \right) - \left(\min_{h \in \mathbb{H}} \sum_{t=1}^{T} \mathbb{1}[h(X_t) \neq Y_t] \right).$$

As is well-known, this may be interpreted as the worst-case value of the difference between the expected number of mistakes the algorithm $\mathbb{A}$ makes in its predictions $\hat{y}_t$ (i.e., how many times t have $\hat{y}_t \neq Y_t$) and the number of mistakes made by the best function in the class $\mathbb{H}$ (where "best" means the function making fewest mistakes on the sequence).

This is often also interpreted as a sequential game between the learner and an adversary. On each round t, the adversary first chooses a value for X_t, the learner observes this value and chooses a probability measure p_t; the adversary observes this p_t and chooses a class label Y_t, and the learner suffers a loss $1 - p_t(Y_t)$, representing the probability that its randomized prediction $\hat{y}_t$ is incorrect. The overall objective of the learner is to achieve low regret on this sequence of plays, that is, $\left(\sum_{t=1}^{T} (1 - p_t(Y_t)) \right) - \left(\min_{h \in \mathbb{H}} \sum_{t=1}^{T} \mathbb{1}[h(X_t) \neq Y_t] \right)$, whereas the adversary's objective is to maximize this quantity. $\text{regret}(\mathbb{A}, T)$ represents the value of this objective when the adversary plays optimally against the algorithm $\mathbb{A}$.

A concept class $\mathbb{H}$ is said to be agnostically online learnable if

$$\inf_{\mathbb{A}} \ \text{regret}(\mathbb{A}, T) = o(T).$$

The key quantity of interest in online learning (be it binary or multiclass) is the Littlestone dimension, $\text{L}(\mathbb{H})$ (see Section 2.2 for the definition), whose finiteness is known to be necessary and sufficient for online learnability in the realizable case (i.e., when the subtracted term in $\text{regret}(\mathbb{A}, T)$ is zero). Daniely, Sabato, Ben-David, and Shalev-Shwartz [2011] effectively asked the following question regarding whether this fact extends to the agnostic setting:

[2]The associated σ-algebra is of little consequence here, except that singleton sets $\{y\}$ should be measurable.

Is any concept class $\mathbb{H}$ agnostically online learnable if and only if $\mathrm{L}(\mathbb{H}) < \infty$?

Daniely, Sabato, Ben-David, and Shalev-Shwartz [2011] proved the necessity direction (which, as they note, follows readily from arguments of Littlestone, 1987), that is, that any class with $\mathrm{L}(\mathbb{H}) = \infty$ is not online learnable (indeed, even in the realizable case). Daniely, Sabato, Ben-David, and Shalev-Shwartz [2015] further establish a lower bound when $\mathrm{L}(\mathbb{H}) < \infty$: for any $\mathbb{A}$ and any $T \geq \mathrm{L}(\mathbb{H})$,

$$\mathrm{regret}(\mathbb{A}, T) = \Omega\left(\sqrt{\mathrm{L}(\mathbb{H})T}\right). \tag{2.1}$$

However, for the sufficiency direction, their upper bound has a dependence on the number of classes, and hence only establishes sufficiency for a bounded number of classes. This is analogous to the well-known gap in PAC learnability, which was only recently resolved by Brukhim, Carmon, Dinur, Moran, and Yehudayoff [2022]. As our main result, we prove the following theorem.

Theorem 2.1.1. ***(Main Result)*** *Any concept class $\mathbb{H}$ is agnostically online learnable iff $\mathrm{L}(\mathbb{H}) < \infty$.*
Moreover, for any $T \in \mathbb{N}$, there is an online learning algorithm $\mathbb{A}$ satisfying

$$\mathrm{regret}(\mathbb{A}, T) = \tilde{O}\left(\sqrt{\mathrm{L}(\mathbb{H})T}\right).$$

In light of the lower bound (2.1), this further establishes that the optimal achievable regret for $T \geq \mathrm{L}(\mathbb{H})$ satisfies

$$\min_{\mathbb{A}} \mathrm{regret}(\mathbb{A}, T) = \tilde{\Theta}\left(\sqrt{\mathrm{L}(\mathbb{H})T}\right).$$

While the result of Daniely, Sabato, Ben-David, and Shalev-Shwartz [2015], which has a dependence on the number of classes, is based on an extension of the algorithm of Ben-David, Pál, and Shalev-Shwartz [2009] for binary classification, our result modifies this approach. We take inspiration from the PAC setting, where the only known proof that the agnostic sample complexity is characterized by the DS dimension proceeds by a reduction to the realizable case, wherein the algorithm first identifies a maximal subset of the data, and then applies a compression scheme for the realizable case to this subset [David, Moran, and Yehudayoff, 2016]. In our case, this maximal realizable subset appears only in the analysis, but serves an important role. Specifically, our algorithm applies the well-known multiplicative weights experts algorithm, with a family of experts defined by predictions of all possible executions of the SOA (see Section 2.2.1) that are constrained to only update their predictor

in at most $\mathrm{L}(\mathbb{H})$ pre-specified time steps. The important property is that one of these experts corresponds precisely to executing the SOA on a maximal realizable subsequence of the data sequence, and hence makes at most $\mathrm{L}(\mathbb{H})$ mistakes on this subsequence (see Section 2.2.1). This expert therefore has regret only $\mathrm{L}(\mathbb{H})$ compared to the best function in $\mathbb{H}$, and a regret bound for the overall algorithm then follows from classical analysis of prediction with expert advice. We present the detailed proof in Section 2.3.

The $\tilde{O}$ in Theorem 2.1.1 hides a factor $\sqrt{\log(T/\mathrm{L}(\mathbb{H}))}$. An analogous factor has recently been removed from the best known regret bound for binary classification [Alon, Ben-Eliezer, Dagan, Moran, Naor, and Yogev, 2021b], yielding a regret bound that is optimal up to numerical constants. It remains open whether an analogous refinement is possible in the multiclass setting. Specifically, we pose the following open problem.

Open Problem 1. *Is it true that, for any concept class $\mathbb{H}$, the optimal regret is $\Theta\Big(\sqrt{\mathrm{L}(\mathbb{H})T}\Big)$?*

In addition to the above results for learnability, we also study the related question of adversarial uniform laws of large numbers; see Section 2.4 for definitions. It was shown in the PAC setting that multiclass learnability is not equivalent to a uniform law of large numbers [Natarajan, 1988, Daniely, Sabato, Ben-David, and Shalev-Shwartz, 2011, 2015], in contrast to binary classification where it has long been established that these are equivalent [Vapnik and Chervonenkis, 1974a]. Again in the binary classification setting, Alon, Ben-Eliezer, Dagan, Moran, Naor, and Yogev [2021b] have established an equivalence between a type of adversarial uniform law of large numbers (AULLN), online learnability, and convergence of the sequential Rademacher complexity (the latter two were already known to be equivalent, Rakhlin, Sridharan, and Tewari, 2015a). In the present work, we find that the analogy between PAC and online learning settings holds true once again. That is, in the multiclass setting with unbounded label space, while the AULLN is again equivalent to convergence of the sequential Rademacher complexity, in this case these properties are not equivalent to online learnability. Indeed, we show (Theorem 2.4.1) that AULLN and convergence of sequential Rademacher complexity are characterized by finiteness of a different combinatorial parameter $\mathrm{d}_{\mathrm{SG}}(\mathbb{H})$ called the sequential graph dimension (Definition 2.4.2), and we exhibit an example in which $\mathrm{L}(\mathbb{H}) = 1$ but $\mathrm{d}_{\mathrm{SG}}(\mathbb{H}) = \infty$. All of this remains perfectly analogous to known results for multiclass PAC learning. Carrying the parallel further, in the case of bounded label spaces, we relate these two parameters by $\mathrm{d}_{\mathrm{SG}}(\mathbb{H}) = O(\mathrm{L}(\mathbb{H})\log(|\mathcal{Y}|))$ (Theorem 2.4.2). Based on the above, we also state new bounds on the achievable regret (Theorem 2.4.3): namely, $\mathrm{regret}(\mathbb{A}, T) = O\Big(\sqrt{\mathrm{d}_{\mathrm{SG}}(\mathbb{H})T}\Big)$, which in the finite $|\mathcal{Y}|$ case, further implies $\mathrm{regret}(\mathbb{A}, T) = O\Big(\sqrt{\mathrm{L}(\mathbb{H})T\log(|\mathcal{Y}|)}\Big)$.

2.2 Definitions

We begin with some basic useful notation. For any sequence $z_1, z_2, \ldots$, for any $t \in \mathbb{N} \cup \{0\}$, we denote by $z_{\leq t} = (z_1, \ldots, z_t)$ and $z_{<t} = (z_1, \ldots, z_{t-1})$, interpreting $z_{\leq 0} = z_{<1} = ()$, the empty sequence. For simplicity, for a sequence $(x_1, y_1), \ldots, (x_n, y_n)$, we denote by $(x_{<t}, y_{<t}) = ((x_1, y_1), \ldots, (x_{t-1}, y_{t-1}))$ and $(x_{\leq t}, y_{\leq t}) = ((x_1, y_1), \ldots, (x_t, y_t))$. For any $h \in \mathbb{H}$, $n \in \mathbb{N}$, and $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{X}^n$, denote by $h(\mathbf{x}) = (h(x_1), \ldots, h(x_n))$. For any $V \subseteq \mathbb{H}$, $n \in \mathbb{N}$, and any sequence $(\mathbf{x}, \mathbf{y}) = ((x_1, y_1), \ldots, (x_n, y_n)) \in (\mathcal{X} \times \mathcal{Y})^n$, denote by $V_{(\mathbf{x},\mathbf{y})} = \{h \in V : h(\mathbf{x}) = \mathbf{y}\}$, called a version space of V. We say a sequence $(\mathbf{x}, \mathbf{y}) \in (\mathcal{X} \times \mathcal{Y})^n$ is realizable by $\mathbb{H}$ if $\mathbb{H}_{(\mathbf{x},\mathbf{y})} \neq \emptyset$: that is, $\exists h \in \mathbb{H}$ with $h(\mathbf{x}) = \mathbf{y}$. For any value $\alpha \in \mathbb{R}$, denote by $\log(\alpha) = \ln(\max\{\alpha, e\})$.

The following is the primary definition of dimension central to this work: namely, the Littlestone dimension $\mathrm{L}(\mathbb{H})$ of a concept class $\mathbb{H}$. The definition is due to Daniely, Sabato, Ben-David, and Shalev-Shwartz [2011, 2015], representing the natural generalization of the classic definition of Littlestone [1987] (for binary classification) to multiclass.

We first state a clear intuitive definition in terms of binary trees. Specifically, a Littlestone tree T is a rooted binary tree, where each internal node is labeled by an instance $x \in \mathcal{X}$, and each edge between that node and a child is labeled by (x, y) for a class label $y \in \mathcal{Y}$, with the restriction that if the node has two children, then the corresponding class labels y, y' must be distinct: i.e., if the edges are labeled $(x, y), (x, y')$, respectively, then $y \neq y'$. A finite-depth Littlestone tree T is shattered by $\mathbb{H}$ if, for every leaf node of any depth d, the sequence $(\mathbf{x}, \mathbf{y})$ of labels of the edges along the path from the root to this leaf is realizable by $\mathbb{H}$. The Littlestone dimension, $\mathrm{L}(\mathbb{H})$, is then the maximum depth of a perfect Littlestone tree shattered by $\mathbb{H}$, where the term perfect means that every node has 2 children and every leaf has equal depth. If there are arbitrarily large depths d for which there exist perfect Littlestone trees shattered by $\mathbb{H}$, then $\mathrm{L}(\mathbb{H}) = \infty$.

We state this definition formally (giving notation to the labels of nodes and edges) as follows.

Definition 2.2.1. *The Littlestone dimension of $\mathbb{H}$, denoted $\mathrm{L}(\mathbb{H})$, is defined as the largest $n \in \mathbb{N} \cup \{0\}$ for which $\exists\{(x_{\mathbf{b}}, y_{(\mathbf{b},0)}, y_{(\mathbf{b},1)}) : \mathbf{b} \in \{0,1\}^t, t \in \{0, \ldots, n-1\}\} \subseteq \mathcal{X} \times \mathcal{Y}^2$ (interpreting $\{0,1\}^0 = \{()\}$) with the property that $\forall b_1, \ldots, b_n \in \{0,1\}$, $\exists h \in \mathbb{H}$ with*

$$h(x_{b_{<1}}, x_{b_{<2}}, \ldots, x_{b_{<n}}) = (y_{b_{\leq 1}}, y_{b_{\leq 2}}, \ldots, y_{b_{\leq n}}).$$

If no such largest n exists, define $\mathrm{L}(\mathbb{H}) = \infty$. Also define $\mathrm{L}(\emptyset) = -1$.

When $\mathrm{L}(\mathbb{H}) < \infty$*, one can show that* $\mathrm{L}(\mathbb{H})$ *can equivalently be defined inductively as*

$$\max_{x} \max_{y_0 \neq y_1} \min_{i \in \{0,1\}} \mathrm{L}(\mathbb{H}_{(x,y_i)}) + 1.$$

To see the correspondence between Definition 2.2.1 and the definition in terms of shattered Littlestone trees, we take the nodes at depth $t \geq 0$ (counting the root as depth 0) to be labeled $x_{b_{\leq t}}$, with the edge connecting to its left child labeled $(x_{b_{\leq t}}, y_{(b_{\leq t},0)})$ and the edge connecting to its right child labeled $(x_{b_{\leq t}}, y_{(b_{\leq t},1)})$.

2.2.1 Background

It was shown by Daniely, Sabato, Ben-David, and Shalev-Shwartz [2011, 2015] that $\mathbb{H}$ is online learnable in the realizable case if and only if $\mathrm{L}(\mathbb{H}) < \infty$. That is, there is an algorithm guaranteeing a bounded number of mistakes on all $(X_1, Y_1), \ldots, (X_T, Y_T)$ realizable by $\mathbb{H}$. Specifically, they employ a natural multiclass generalization of Littlestone's Standard Optimal Algorithm (SOA), defined as follows. For any $(X_1, Y_1), \ldots, (X_{t-1}, Y_{t-1}), X_t$, define

$$\mathrm{SOA}(X_{<t}, Y_{<t}, X_t) = \arg\max_{y \in \mathcal{Y}} \mathrm{L}(\mathbb{H}_{((X_{<t}, Y_{<t}), (X_t, y))}),$$

where ties are broken arbitrarily. In other words, it (deterministically) predicts a label $\hat{y}_t$ which maximizes the Littlestone dimension of the version space constrained by that label. Following the classic argument of Littlestone [1987] for binary classification, Daniely, Sabato, Ben-David, and Shalev-Shwartz [2011, 2015] showed that SOA makes at most $\mathrm{L}(\mathbb{H})$ mistakes on any sequence $(X_1, Y_1), \ldots, (X_T, Y_T)$ realizable by $\mathbb{H}$: that is,

$$\sum_{t=1}^{T} \mathbb{1}[\mathrm{SOA}(X_{<t}, Y_{<t}, X_t) \neq Y_t] \leq \mathrm{L}(\mathbb{H}).$$

The reason is clear from the inductive version of the definition of $\mathrm{L}(\mathbb{H})$. When $\mathrm{L}(\mathbb{H}) < \infty$, for any $V \subseteq \mathbb{H}$ and $x \in \mathcal{X}$, at most one label y can have $\mathrm{L}(V_{(x,y)}) = \mathrm{L}(V)$, and hence (since L is integer-valued) every other y' must have $\mathrm{L}(V_{(x,y')}) \leq \mathrm{L}(V) - 1$. Thus, for $V = \mathbb{H}_{(X_{<t}, Y_{<t})}$, by predicting the label $\hat{y}_t$ with maximum $\mathrm{L}(\mathbb{H}_{((X_{<t}, Y_{<t}), (X_t, \hat{y}_t))})$, we are guaranteed that if $\hat{y}_t \neq Y_t$, then Y_t cannot have $\mathrm{L}(\mathbb{H}_{(X_{\leq t}, Y_{\leq t})}) = \mathrm{L}(\mathbb{H}_{(X_{<t}, Y_{<t})})$. That is, every mistake guarantees that the version space $\mathbb{H}_{(X_{\leq t}, Y_{\leq t})}$ has a smaller Littlestone dimension (by at least 1) than $\mathbb{H}_{(X_{<t}, Y_{<t})}$. Since $(X_1, Y_1), \ldots, (X_T, Y_T)$ is realizable by $\mathbb{H}$, we always have $\mathbb{H}_{(X_{\leq t}, Y_{\leq t})} \neq \emptyset$, and hence $\mathrm{L}(\mathbb{H}_{(X_{\leq t}, Y_{\leq t})}) \geq 0$, so that we can have $\hat{y}_t \neq Y_t$ at most $\mathrm{L}(\mathbb{H})$ times.

Again following Littlestone [1987], the work of Daniely, Sabato, Ben-David, and Shalev-

Shwartz [2011, 2015] also shows a lower bound, establishing that for any deterministic online learning algorithm, there exists a realizable sequence where it makes at least $\mathrm{L}(\mathbb{H})$ mistakes, so that SOA is optimal in this regard among all deterministic online learning algorithms. Moreover, even for any randomized learning algorithm, there always exists a sequence realizable by $\mathbb{H}$ for which the expected number of mistakes is at least $\mathrm{L}(\mathbb{H})/2$. Thus, there exists an algorithm guaranteeing a bounded number of mistakes on all realizable sequences if and only if $\mathrm{L}(\mathbb{H}) < \infty$: that is, $\mathbb{H}$ is online learnable in the realizable case iff $\mathrm{L}(\mathbb{H}) < \infty$.

In the case of agnostic online learning for multiclass classification, Daniely, Sabato, Ben-David, and Shalev-Shwartz [2015] establish a lower bound (for $T \geq \mathrm{L}(\mathbb{H})$), $\mathrm{regret}(\mathbb{A}, T) = \Omega(\sqrt{\mathrm{L}(\mathbb{H})T})$, holding for any algorithm $\mathbb{A}$. Moreover, in the case of $|\mathcal{Y}| < \infty$, they propose a learning algorithm $\mathbb{A}$ guaranteeing $\mathrm{regret}(\mathbb{A}, T) = O\Big(\sqrt{\mathrm{L}(\mathbb{H})T\log(T|\mathcal{Y}|)}\Big)$. Thus, in the case of bounded label spaces, the Littlestone dimension characterizes agnostic learnability. They left open the question of whether this remains true for unbounded label spaces. Note that the dependence on $|\mathcal{Y}|$ in the above regret bound makes the bound vacuous for infinite $\mathcal{Y}$ spaces. We resolve this question (positively) in the present work (Theorem 2.1.1), and refine the above finite label bound as well (Theorem 2.4.2).

2.3 Agnostic Online Learnability of Littlestone Classes

This section presents the main result and algorithm in detail. As one of the main components of the algorithm, we make use of the following classic result for learning from expert advice [e.g., Vovk, 1990, 1992, Littlestone and Warmuth, 1994]; see Theorem 2.2 of Cesa-Bianchi and Lugosi [2006].

Lemma 2.3.1. *[Cesa-Bianchi and Lugosi, 2006, Theorem 2.2] For any $N, T \in \mathbb{N}$ and an array of values $e_{i,t} \in \{0,1\}$, with $i \in \{1,\ldots,N\}$, $t \in \{1,\ldots,T\}$, letting $\eta = \sqrt{(8/T)\ln(N)}$, for any $y_1,\ldots,y_T \in \{0,1\}$, letting $w_{i,1} = 1$ and $w_{i,t} = e^{-\eta\sum_{s<t}\mathbb{1}[e_{i,s}\neq y_s]}$ for each $t \in \{2,\ldots,T\}$ and $i \in \{1,\ldots,N\}$, letting $p_t = \sum_{i=1}^N w_{i,t}e_{i,t} / \sum_{i'=1}^N w_{i',t}$, it holds that*

$$\sum_{t=1}^T |p_t - y_t| - \min_{1\leq i\leq N}\sum_{t=1}^T \mathbb{1}[e_{i,t}\neq y_t] \leq \sqrt{(T/2)\ln(N)}.$$

We are now ready to describe our agnostic online learning algorithm, denoted by $\mathbb{A}_{\mathrm{AG}}$, specified based on a given time horizon $T \in \mathbb{N}$. As above, let $(X_1,Y_1),\ldots,(X_T,Y_T)$ denote the data sequence. For any $J \subseteq \{1,\ldots,T\}$, denote by $X_J = \{X_t : t \in J\}$ and $Y_J = \{Y_t : t \in J\}$, and for any $t \in \mathbb{N}$, denote by $J_{<t} = J \cap \{1,\ldots,t-1\}$. We consider a set of experts defined as follows. Let $\mathcal{J} = \{J \subseteq \{1,\ldots,T\} : |J| \leq \mathrm{L}(\mathbb{H})\}$. For each $J \in \mathcal{J}$, for each $t \in \mathbb{N}$,

define

$$g_t^J = \mathrm{SOA}(X_{J_{<t}}, Y_{J_{<t}}, X_t).$$

That is, g_t^J is the prediction the SOA would make on X_t, given that its previous sequence of examples were $(X_{J_{<t}}, Y_{J_{<t}})$: namely, $\arg\max_y \mathrm{L}(\mathbb{H}_{((X_{J_{<t}}, Y_{J_{<t}}),(X_t,y))})$. Based on the set of experts $\{g^J : J \in \mathcal{J}\}$, we apply the multiplicative weights algorithm. Explicitly, letting $\eta = \sqrt{(8/T)\ln(|\mathcal{J}|)}$, and defining $w_{J,1} = 1$ and $w_{J,t} = e^{-\eta \sum_{s<t} \mathbb{1}[g_s^J \neq Y_s]}$ for $t \in \{2, \ldots, T\}$ and $J \in \mathcal{J}$, we define the prediction at time t as

$$p_t = \frac{\sum_{J \in \mathcal{J}} w_{J,t} g_t^J}{\sum_{J' \in \mathcal{J}} w_{J',t}}.$$

Note that the values g_t^J of the experts at time t only depend on $X_{<t}, Y_{<t}, X_t$, so that this is a valid prediction. We have the following result, representing the main theorem of this work. In particular, in conjunction with the necessity of $\mathrm{L}(\mathbb{H}) < \infty$ for online learnability, established by Daniely, Sabato, Ben-David, and Shalev-Shwartz [2015], our Theorem 2.1.1 immediately follows from this.

Theorem 2.3.2. *For any concept class $\mathbb{H}$ and $T \geq 2\mathrm{L}(\mathbb{H})$, the algorithm $\mathbb{A}_{\mathrm{AG}}$ defined above satisfies*

$$\mathrm{regret}(\mathbb{A}_{\mathrm{AG}}, T) = O\left(\sqrt{\mathrm{L}(\mathbb{H}) T \log\left(\frac{T}{\mathrm{L}(\mathbb{H})}\right)}\right).$$

Proof. Let $h^* \in \mathbb{H}$ satisfy $\sum_{t=1}^T \mathbb{1}[h^*(X_t) \neq Y_t] = \min_{h \in \mathbb{H}} \sum_{t=1}^T \mathbb{1}[h(X_t) \neq Y_t]$. Denote by $R^* = \{t \in \{1, \ldots, T\} : h^*(X_t) = Y_t\}$, and note that the subsequence (X_{R^*}, Y_{R^*}) is realizable by $\mathbb{H}$. Define a sequence j_r inductively, as follows. Let $j_1 = \min\{t \in R^* : \mathrm{SOA}(\emptyset, \emptyset, X_t) \neq Y_t\}$ if it exists. For $r \geq 2$, if j_{r-1} is defined, let

$$j_r = \min\{t \in R^* : t > j_{r-1} \text{ and } \mathrm{SOA}(X_{j_{<r}}, Y_{j_{<r}}, X_t) \neq Y_t\}$$

if it exists. Finally, define

$$J^* = \{j_r : r \in \mathbb{N} \text{ and } j_r \text{ exists}\}.$$

In other words, J^* represents the sequence of mistakes SOA would make on the sequence R^* using conservative updates: that is, only adding an example (X_t, Y_t) to its history if it is a mistake point.

In particular, note that if $J^* = \emptyset$, then every $t \in R^*$ satisfies $g_t^{J^*} = \mathrm{SOA}(X_{J^*_{<t}}, Y_{J^*_{<t}}, X_t) = \mathrm{SOA}(\emptyset, \emptyset, X_t) = Y_t$. If $J^* \neq \emptyset$, then any $t \in R^*_{<j_1}$ has $g_t^{J^*} = \mathrm{SOA}(X_{J^*_{<t}}, Y_{J^*_{<t}}, X_t) =$

$\mathrm{SOA}(\emptyset, \emptyset, X_t) = Y_t$; similarly, any $r \in \{1, \ldots, |J^*| - 1\}$ and $t \in R^*_{<j_{r+1}} \setminus R^*_{\leq j_r}$ has $g_t^{J^*} = \mathrm{SOA}(X_{J^*_{<t}}, Y_{J^*_{<t}}, X_t) = \mathrm{SOA}(X_{j_{<r+1}}, Y_{j_{<r+1}}, X_t) = Y_t$; also, for $r = |J^*|$ (the largest r for which j_r is defined), any $t \in R^* \setminus R^*_{\leq j_r}$ has $g_t^{J^*} = \mathrm{SOA}(X_{J^*_{<t}}, Y_{J^*_{<t}}, X_t) = \mathrm{SOA}(X_{j_{<r+1}}, Y_{j_{<r+1}}, X_t) = Y_t$. In other words, $g_t^{J^*} = Y_t$ for every $t \in R^* \setminus J^*$. Therefore,

$$\sum_{t=1}^{T} \mathbb{1}[g_t^{J^*} \neq Y_t] \leq |J^*| + (T - |R^*|) = |J^*| + \min_{h \in \mathbb{H}} \sum_{t=1}^{T} \mathbb{1}[h(X_t) \neq Y_t].$$

Moreover, since (X_{R^*}, Y_{R^*}) is realizable by $\mathbb{H}$, and $J^* \subseteq R^*$, the subsequence (X_{J^*}, Y_{J^*}) is also realizable by $\mathbb{H}$. By definition, SOA makes a mistake on every time when run through the subsequence (X_{J^*}, Y_{J^*}): that is, $\mathrm{SOA}(X_{j_{<r}}, Y_{j_{<r}}, X_{j_r}) \neq Y_{j_r}$ for every $j_r \in J^*$. Thus, by the guaranteed mistake bound $\mathrm{L}(\mathbb{H})$ for SOA on realizable sequences, we conclude that $|J^*| \leq \mathrm{L}(\mathbb{H})$. In particular, this implies $J^* \in \mathcal{J}$. Altogether, by Lemma 2.3.1, we have that

$$\begin{aligned}\sum_{t=1}^{T} |p_t - Y_t| &\leq \left(\sum_{t=1}^{T} \mathbb{1}[g_t^{J^*} \neq Y_t]\right) + \sqrt{(T/2)\ln(|\mathcal{J}|)} \\ &\leq \min_{h \in \mathbb{H}} \sum_{t=1}^{T} \mathbb{1}[h(X_t) \neq Y_t] + \mathrm{L}(\mathbb{H}) + \sqrt{(T/2)\mathrm{L}(\mathbb{H}) \ln\left(\frac{eT}{\mathrm{L}(\mathbb{H})}\right)},\end{aligned}$$

$$\implies \mathrm{regret}(\mathbb{A}_{\mathrm{AG}}, T) \leq \mathrm{L}(\mathbb{H}) + \sqrt{(T/2)\mathrm{L}(\mathbb{H}) \ln\left(\frac{eT}{\mathrm{L}(\mathbb{H})}\right)} = O\left(\sqrt{\mathrm{L}(\mathbb{H}) T \log\left(\frac{T}{\mathrm{L}(\mathbb{H})}\right)}\right). \quad \blacksquare$$

2.4 Online Uniform Convergence Versus Online Learnability

In PAC learning for binary classification, for a given data distribution, a class $\mathbb{H}$ satisfies the uniform law of large numbers (i.e., is P-Glivenko-Cantelli) if and only if the (normalized) Rademacher complexity converges to 0 in sample size. Moreover, the rate of uniform convergence is dominated by a converging distribution-free function of sample size if and only if the VC dimension is finite. By a chaining argument [Talagrand, 1994, van der Vaart and Wellner, 1996], the optimal form for this rate is $\Theta(\sqrt{\mathrm{VC}(\mathbb{H})/n})$, where n is the sample size and $\mathrm{VC}(\mathbb{H})$ is the VC dimension of $\mathbb{H}$.

Similarly, in adversarial online learning for binary classification, Alon, Ben-Eliezer, Dagan, Moran, Naor, and Yogev [2021b] showed that a class $\mathbb{H}$ satisfies an adversarial uniform law of

large numbers if and only if the (normalized) sequential Rademacher complexity converges to 0 in the sequence length.[3] The rate of adversarial uniform convergence, uniform over sequences, is then controlled by the Littlestone dimension. Again by a chaining argument [Alon et al., 2021b], the optimal form for this rate is $\Theta(\sqrt{\mathrm{L}(\mathbb{H})T})$.[4]

Importantly, for both of these facts, the complexity measure controlling the rate of uniform convergence is the same as the complexity measure that determines learnability: for PAC learning, the VC dimension, and for online learning, the Littlestone dimension. In particular, together with separately established lower bounds for learning, these facts imply that the optimal rate of convergence of expected excess error in agnostic PAC learning is $\Theta(\sqrt{\mathrm{VC}(\mathbb{H})/n})$ [Talagrand, 1994], whereas the optimal regret bound for agnostic online learning is $\Theta(\sqrt{\mathrm{L}(\mathbb{H})T})$ [Alon et al., 2021b].

In the case of PAC learning for multiclass classification, it again holds that the classification losses satisfy the uniform law of large numbers if and only if the Rademacher complexity converges to 0 in sample size. However, in this case, there exists a distribution-free bound on the rate of uniform convergence if and only if the graph dimension is finite [Ben-David, Cesa-Bianchi, Haussler, and Long, 1995, Daniely, Sabato, Ben-David, and Shalev-Shwartz, 2011, 2015], and the optimal such bound is $\Theta(\sqrt{d_G(\mathbb{H})/n})$, where n is the sample size and $d_G(\mathbb{H})$ is the graph dimension. Notably, the graph dimension does not control PAC learnability of the class [Natarajan, 1988, Daniely et al., 2011, 2015]; rather, a recent result of Brukhim et al. [2022] established that multiclass PAC learnability (including agnostic learnability) is controlled by a quantity they call the DS dimension (originally proposed by Daniely and Shalev-Shwartz, 2014, who proved it provides a lower bound), and there are simple examples where the DS dimension is finite while the graph dimension is infinite (necessarily with an infinite number of possible class labels). Thus, in the case of multiclass classification, we see that the uniform law of large numbers, and PAC learnability, are controlled by different parameters of the class, and there are PAC learnable classes which do not satisfy the uniform law of large numbers. Thus, PAC learning algorithms generally cannot rely on a uniform law of large numbers, in contrast to binary classification.

In this section, we note an analogous result holds for the adversarial uniform law of large

[3] Rakhlin, Sridharan, and Tewari [2010, 2015a,b] also studied a notion of sequential uniform law of large numbers. Though formulated somewhat differently, that notion was also shown to be equivalent to convergence of sequential Rademacher complexity to 0, and hence is satisfied iff the notion of AULLN studied by Alon, Ben-Eliezer, Dagan, Moran, Naor, and Yogev [2021b] is satisfied.

[4] This result of Alon et al. [2021b] was in fact only shown under a further technical restriction on $\mathbb{H}$ (rooted in the work of Rakhlin et al., 2010, 2015a), so that the rounds of the online learning game satisfy the minimax theorem. As the results in this section are based on this result of Alon et al. [2021b], we also suppose this condition is appropriately satisfied. It remains open whether this $\sqrt{\mathrm{L}(\mathbb{H})T}$ regret for binary classification, and consequently our theorems below, remain valid without any restrictions on $\mathbb{H}$.

numbers. We find that, while the adversarial uniform law of large numbers is again satisfied if and only if the sequential Rademacher complexity converges to 0 in sequence length, the optimal sequence-independent bound on the convergence depends on the sequential graph dimension. Thus, in light of our Theorem 2.1.1, establishing that agnostic online learnability is controlled by the Littlestone dimension, we again see that the parameter controlling the adversarial uniform law of large numbers differs from the parameter controlling online learnability (including agnostic learnability). Moreover, we provide a simple example where the sequential graph dimension is infinite while the Littlestone dimension is finite, thus showing that not all online learnable classes satisfy the AULLN.

We begin by recalling the following definitions of Alon, Ben-Eliezer, Dagan, Moran, Naor, and Yogev [2021b]. An adversarially uniform law of large numbers (AULLN) can be viewed as a sequential game between a sampler $\mathcal{S}$ and an adversary. On each round t, the adversary chooses $(x_t, y_t) \in \mathcal{X} \times \mathcal{Y}$, and the sampler decides whether to include (x_t, y_t) in its "sample": a subsequence K. The sampler may be randomized, and the adversary may observe and adapt to the sampler's past decisions (though not its internal random bits).

Definition 2.4.1 (Alon et al., 2021b)**.** *A concept class $\mathbb{H}$ satisfies the adversarial uniform law of large numbers (AULLN) if, for any $\varepsilon, \delta \in (0,1)$, there exists $k(\varepsilon, \delta) \in \mathbb{N}$ and a sampler $\mathcal{S}$ such that, for any adversarially-produced sequence $(\bar{x}, \bar{y}) = \{(x_t, y_t)\}_{t=1}^T$ of any length T, the sample K selected by $\mathcal{S}$ always satisfies $|K| \leq k(\varepsilon, \delta)$, and with probability at least $1 - \delta$, K forms an ε-approximation of $(\bar{x}, \bar{y})$ (with respect to $\mathbb{H}$), meaning*[5]

$$\sup_{h \in \mathbb{H}} \left| \frac{1}{|K|} \sum_{(x_t, y_t) \in K} \mathbb{1}[h(x_t) \neq y_t] - \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}[h(x_t) \neq y_t] \right| \leq \varepsilon.$$

The AULLN was shown by Alon, Ben-Eliezer, Dagan, Moran, Naor, and Yogev [2021b] to be intimately related to the sequential Rademacher complexity of the class of indicator functions, in this case, $(x, y) \mapsto \mathbb{1}[h(x) \neq y]$, $h \in \mathbb{H}$. Formally, we recall the definition of sequential Rademacher complexity from the work of Rakhlin, Sridharan, and Tewari [2010, 2015a]. Let $\varepsilon = \{\varepsilon_t\}_{t \in \mathbb{N}}$ be i.i.d. Uniform($\{-1, 1\}$) random variables. Let $z = \{(x_t, y_t)\}_{t \in \mathbb{N}}$ be any sequence of functions $(x_t, y_t) : \{-1, 1\}^{t-1} \to \mathcal{X} \times \mathcal{Y}$, denoting by $(x_t(\varepsilon_{<t}), y_t(\varepsilon_{<t}))$ the value of this function on $\varepsilon_{<t}$. For any $T \in \mathbb{N}$, the sequential Rademacher complexity is defined as

$$\mathrm{Rad}_T(\mathbb{H}) = \sup_z \mathbb{E}\left[\sup_{h \in \mathbb{H}} \frac{1}{T} \sum_{t=1}^{T} \varepsilon_t \mathbb{1}[h(x_t(\varepsilon_{<t})) \neq y_t(\varepsilon_{<t})] \right].$$

[5]To be clear, the sum over $(x_t, y_t) \in K$ treats duplicates as distinct: that is, there are $|K|$ terms in the sum.

Alon, Ben-Eliezer, Dagan, Moran, Naor, and Yogev [2021b] proved that, for binary classification, the best achievable bound on the sequential Rademacher complexity is the Littlestone dimension of $\mathbb{H}$. However, binary classifiation enjoys a special property that composition of $\mathbb{H}$ with the 0-1 loss does not change the complexity. In contrast, in multiclass classification, the Littlestone dimension of $\mathbb{H}$ composed with the 0-1 loss is a different quantity, which we term the sequential graph dimension:

Definition 2.4.2. *The sequential graph dimension of* $\mathbb{H}$*, denoted* $\mathrm{d}_{\mathrm{SG}}(\mathbb{H})$*, is defined as*

$$\mathrm{d}_{\mathrm{SG}}(\mathbb{H}) = \mathrm{L}(\{(x, y) \mapsto \mathbb{1}[h(x) \neq y] : h \in \mathbb{H}\}).$$

Explicitly, $\mathrm{d}_{\mathrm{SG}}(\mathbb{H})$ *is the largest* $n \in \mathbb{N} \cup \{0\}$ *s.t.* $\exists \{(x_{\mathbf{b}}, y_{\mathbf{b}}) : \mathbf{b} \in \{0,1\}^t, t \in \{0, \ldots, n-1\}\} \subseteq \mathcal{X} \times \mathcal{Y}$ *with the property that* $\forall b_1, \ldots, b_n \in \{0,1\}$*,* $\exists h \in \mathbb{H}$ *with*

$$(\mathbb{1}[h(x_{b_{<1}}) \neq y_{b_{<1}}], \mathbb{1}[h(x_{b_{<2}}) \neq y_{b_{<2}}], \ldots, \mathbb{1}[h(x_{b_{<n}}) \neq y_{b_{<n}}]) = (b_1, \ldots, b_n).$$

If no such largest n *exists, define* $\mathrm{d}_{\mathrm{SG}}(\mathbb{H}) = \infty$*. Also define* $\mathrm{d}_{\mathrm{SG}}(\emptyset) = -1$*.*

Here we state an extension of the result of Alon, Ben-Eliezer, Dagan, Moran, Naor, and Yogev [2021b] for AULLN to the multiclass setting. The result follows immediately by applying Theorems 2.2 and 2.3 of Alon, Ben-Eliezer, Dagan, Moran, Naor, and Yogev [2021a] to the class of binary functions $\mathcal{G} = \{(x, y) \mapsto \mathbb{1}[h(x) \neq y] : h \in \mathbb{H}\}$, along with (for the final claim about $\mathrm{Rad}_T(\mathbb{H})$) the analogous application to $\mathcal{G}$ of Corollary 12 of Rakhlin, Sridharan, and Tewari [2015a] and the upper bound on sequential Rademacher complexity in the proof of Theorem 12.1 of Alon, Ben-Eliezer, Dagan, Moran, Naor, and Yogev [2021a].

Theorem 2.4.1. *For any concept class* $\mathbb{H}$*, the following are equivalent:*

1. $\mathbb{H}$ *satisfies the adversarial uniform law of large numbers*
2. $\mathrm{Rad}_T(\mathbb{H}) \to 0$
3. $\mathrm{d}_{\mathrm{SG}}(\mathbb{H}) < \infty$.

Moreover, if $\mathbb{H}$ *satisfies AULLN, then Defintion 2.4.1 is satisfied with* $k(\varepsilon, \delta) = O\left(\frac{\mathrm{d}_{\mathrm{SG}}(\mathbb{H}) + \log(1/\delta)}{\varepsilon^2}\right)$*, and the minimal achievable* $k(\varepsilon, \delta)$ *satisfies* $k(\varepsilon, \delta) = \Omega\left(\frac{\mathrm{d}_{\mathrm{SG}}(\mathbb{H})}{\varepsilon^2}\right)$*. Additionally, for* $T \geq \mathrm{d}_{\mathrm{SG}}(\mathbb{H})$*, it holds that* $\mathrm{Rad}_T(\mathbb{H}) = \Theta\left(\sqrt{\frac{\mathrm{d}_{\mathrm{SG}}(\mathbb{H})}{T}}\right)$*.*

While Theorem 2.4.1 indeed expresses a fairly tight relation between AULLN, seq. Rademacher complexity, and the seq. graph dimension, the corresponding result of Alon,

Ben-Eliezer, Dagan, Moran, Naor, and Yogev [2021b] for binary classification additionally found an equivalence to online learnability and agnostic online learnability. Since our Theorem 2.1.1 establishes that agnostic online learnability is characterized by finiteness of the Littlestone dimension (and this is true for realizable online learning as well), rather than the sequential graph dimension, we see a separation in multiclass classification between AULLN and online learnability. To formally establish this separation, we present the following example, exhibiting a concept class with finite Littlestone dimension but infinite sequential graph dimension.

Example 1: The following construction is identical to a known example of Daniely, Sabato, Ben-David, and Shalev-Shwartz [2011, 2015], originally constructed to show a class that is PAC learnable but has infinite (non-sequential) graph dimension. For completeness, we include the full details of the construction here. Let $\mathcal{X}$ be a countable set, let $\mathcal{Y} = 2^{\mathcal{X}} \cup \{*\}$, and for each $A \subseteq \mathcal{X}$, define

$$h_A(x) = \begin{cases} A & \text{if } x \in A \\ * & \text{otherwise} \end{cases}.$$

Define $\mathbb{H} = \{h_A : A \subseteq \mathcal{X}\}$. For any $x \in \mathcal{X}$ and distinct $y_0, y_1 \in \mathcal{Y}$, it must be that one of y_0, y_1 is equal some $A \subseteq \mathcal{X}$. Since only one $h \in \mathbb{H}$ could possibly have $h(x) = A$ (namely, h_A), and even then only if $x \in A$, we would have $\mathrm{L}(\mathbb{H}_{(x,A)}) \in \{-1, 0\}$. It follows that $\mathrm{L}(\mathbb{H}) \leq 1$. It can be seen that $\mathrm{L}(\mathbb{H}) = 1$ by choosing $y_0 = *$ and $y_1 = A$ for any A such that $x \in A$. On the other hand, for any $n \in \mathbb{N}$ and distinct $x_1, \ldots, x_n \in \mathcal{X}$, for any $b_1, \ldots, b_n \in \{0, 1\}$, letting $x_{b_{<t}} = x_t$ and $y_{b_{<t}} = *$ for each $t \leq n$, the function h_A, with $A = \{x_t : b_t = 1\}$, satisfies $\mathbb{1}[h_A(x_{b_{<t}}) \neq y_{b_{<t}}] = b_t$ for every $t \in \{1, \ldots, n\}$. Thus, $\mathrm{d_{SG}}(\mathbb{H}) \geq n$. Since this is true of every $n \in \mathbb{N}$, we see that $\mathrm{d_{SG}}(\mathbb{H}) = \infty$. In particular, in light of Theorems 2.1.1 and 2.4.1, this class $\mathbb{H}$ is agnostically online learnable (and realizable online learnable), but does not satisfy the adversarial uniform law of large numbers (nor satisfy $\mathrm{Rad}_T(\mathbb{H}) \to 0$). This is precisely analogous to the findings of Daniely, Sabato, Ben-David, and Shalev-Shwartz [2011, 2015] that this class is PAC learnable but does not satisfy the (non-adversarial) uniform law of large numbers.

2.4.1 Finite Label Spaces

In regard to agnostic online learning in the case $|\mathcal{Y}| < \infty$, the best previous regret bound, due to Daniely, Sabato, Ben-David, and Shalev-Shwartz [2015], is $O(\sqrt{\mathrm{L}(\mathbb{H})T\log(T|\mathcal{Y}|)})$. Based on Theorem 2.4.1, in this section, we improve this regret to $O(\sqrt{\mathrm{d_{SG}}(\mathbb{H})T})$. Further, we prove in Theorem 2.4.2 that $\mathrm{d_{SG}}(\mathbb{H}) = O(\mathrm{L}(\mathbb{H})\log(|\mathcal{Y}|))$, so that this additionally implies

a guarantee $\mathrm{regret}(\mathbb{A}, T) = O(\sqrt{\mathrm{L}(\mathbb{H})T\log(|\mathcal{Y}|)})$. On the one hand, this improves over Theorem 2.3.2 by removing a factor $\sqrt{\log(T/\mathrm{L}(\mathbb{H}))}$, but on the other hand, includes a factor $\sqrt{\log(|\mathcal{Y}|)}$ not present in Theorem 2.1.1. In light of the lower bound $\Omega(\sqrt{\mathrm{L}(\mathbb{H})T})$ of Daniely, Sabato, Ben-David, and Shalev-Shwartz [2015] holding for any $\mathbb{A}$, we see that this $O(\sqrt{\mathrm{L}(\mathbb{H})T\log(|\mathcal{Y}|)})$ regret guarantee is optimal up to the $\sqrt{\log(|\mathcal{Y}|)}$ factor. As stated in Open Problem 1, it remains an open problem to determine whether the optimal regret is always of the form $\Theta(\sqrt{\mathrm{L}(\mathbb{H})T})$. Formally:

Theorem 2.4.2. *If $|\mathcal{Y}| < \infty$, for any concept class $\mathbb{H}$, $\mathrm{d_{SG}}(\mathbb{H}) = O(\mathrm{L}(\mathbb{H})\log(|\mathcal{Y}|))$.*

Proof. We follow a strategy of adaptive experts, rooted in the work of Ben-David, Pál, and Shalev-Shwartz [2009] (and similar to the extension thereof used by Daniely, Sabato, Ben-David, and Shalev-Shwartz, 2015 in their multiclass agnostic online learner). Let $n \in \mathbb{N}$ be any number with $n \leq \mathrm{d_{SG}}(\mathbb{H})$. Let $\mathcal{Q}$ denote the set of all (J, Y) such that $J \subseteq \{1, \ldots, n\}$ with $|J| \leq \mathrm{L}(\mathbb{H})$, and $Y = \{Y_j\}_{j\in J}$ is a sequence of values in $\mathcal{Y}$. For any $(J, Y) \in \mathcal{Q}$ and any $t \in \{1, \ldots, n\}$ and $x_1, \ldots, x_t \in \mathcal{X}$, define a value $y_t^{J,Y}(x_{\leq t})$ inductively, as

$$y_t^{J,Y}(x_{\leq t}) = \begin{cases} \mathrm{SOA}(x_{<t}, y_{<t}^{J,Y}, x_t) & \text{if } j \notin J \\ Y_j & \text{if } j \in J \end{cases}.$$

Consider any set $\{(x_{\mathbf{b}}, y_{\mathbf{b}}) : \mathbf{b} \in \{0,1\}^t, t \in \{0, \ldots, n-1\}\} \subseteq \mathcal{X} \times \mathcal{Y}$ satisfying the property in Definition 2.4.2. Now we inductively construct a sequence $b_1, \ldots, b_n \in \{0, 1\}$ as follows. For $t \in \{1, \ldots, n\}$, suppose we have already defined $b_1, \ldots, b_{t-1}$, and let

$$V_{<t} = \{(J, Y) \in \mathcal{Q} : \forall i \in \{1, \ldots, t-1\}, \mathbb{1}[y_i^{J,Y}(x_{b_{<i}}) \neq y_{b_{<i}}] = b_i\}.$$

Define

$$b_t = \operatorname*{arg\,min}_{b\in\{0,1\}} |\{(J, Y) \in V_{<t} : \mathbb{1}[y_t^{J,Y}(x_{b_{<1}}, \ldots, x_{b_{<t}}) \neq y_{b_{<t}}] = b\}|.$$

This completes the inductive definition of $b_1, \ldots, b_n$. In particular, note that for every $t \in \{1, \ldots, n\}$ satisfies $|V_t| \leq \frac{1}{2}|V_{t-1}|$, so that $|V_n| \leq 2^{-n}|\mathcal{Q}|$.

On the other hand, by definition of $(x_{\mathbf{b}}, y_{\mathbf{b}})$, there exists $h \in \mathbb{H}$ with $\mathbb{1}[h(x_{b_{<t}}) \neq y_{b_{<t}}] = b_t$ simultaneously for every $t \in \{1, \ldots, n\}$. Let

$$J^* = \{t \in \{1, \ldots, n\} : \mathrm{SOA}(x_{b_{<t-1}}, h(x_{b_{<t-1}}), x_{b_{<t}}) \neq h(x_{b_{<t}})\},$$

interpreting $(x_{b_{<0}}, h(x_{b_{<0}}) = (\emptyset, \emptyset)$. Recall from Section 2.2.1 that Daniely, Sabato, Ben-David, and Shalev-Shwartz [2011, 2015] proved a mistake bound of $\mathrm{L}(\mathbb{H})$ for SOA, and

hence $|J^*| \leq \mathrm{L}(\mathbb{H})$. For each $t \in J^*$, define $Y^*_t = h(x_{b_{<t}})$, and let $Y^* = \{Y^*_t\}_{t \in J^*}$. By definition of (J^*, Y^*), we have

$$\forall t \in \{1, \ldots, n\}, y_t^{J^*, Y^*}(x_{b_{<1}}, \ldots, x_{b_{<t}}) = h(x_{b_{<t}}).$$

In particular, this implies $(J^*, Y^*) \in V_n$, so that $|V_n| \geq 1$. Altogether, we have

$$1 \leq |V_n| \leq 2^{-n}|\mathcal{Q}| = 2^{-n} \sum_{i=1}^{\mathrm{L}(\mathbb{H})} \binom{n}{i} |\mathcal{Y}|^i \leq 2^{-n} \left(\frac{en}{\mathrm{L}(\mathbb{H})}\right)^{\mathrm{L}(\mathbb{H})} |\mathcal{Y}|^{\mathrm{L}(\mathbb{H})}.$$

Multiplying the leftmost and rightmost expressions by 2^n and taking logarithms yields

$$n \leq \mathrm{L}(\mathbb{H}) \log_2\left(\frac{2n}{\mathrm{L}(\mathbb{H})}\right) + \mathrm{L}(\mathbb{H}) \log_2(|\mathcal{Y}|)\,. \tag{2.2}$$

Solving for an upper bound on n reveals that $n \leq 2\mathrm{L}(\mathbb{H}) \log_2(e|\mathcal{Y}|)$. Specifically, this claim holds trivially when $|\mathcal{Y}| = 1$, and for $|\mathcal{Y}| \geq 2$ it follows from (2.2) by Lemma 4.6 of Vidyasagar [2003]. ■

Theorem 2.4.3. *If $|\mathcal{Y}| < \infty$, for any concept class $\mathbb{H}$ with $\mathrm{L}(\mathbb{H}) < \infty$, there exists an algorithm $\mathbb{A}$ satisfying*

$$\mathrm{regret}(\mathbb{A}, T) = O\left(\sqrt{\mathrm{d_{SG}}(\mathbb{H})T}\right).$$

Moreover, this implies $\mathrm{regret}(\mathbb{A}, T) = O\left(\sqrt{\mathrm{L}(\mathbb{H})T \log(|\mathcal{Y}|)}\right)$.

Proof. The proof follows identically the proof of Theorem 12.1 of Alon, Ben-Eliezer, Dagan, Moran, Naor, and Yogev [2021a]. Specifically, Theorem 7 of Rakhlin, Sridharan, and Tewari [2015a] provides that $\mathrm{regret}(\mathbb{A}, T) \leq 2\mathrm{Rad}_T(\mathbb{H})$. The theorem then follows directly from Theorems 2.4.1 and 2.4.2. ■

CHAPTER 3

Online Classification with Predictions

In this chapter, we study online classification when the learner has access to predictions about future examples. We design an online learner whose expected regret is never worse than the worst-case regret, gracefully improves with the quality of the predictions, and can be significantly better than the worst-case regret when the predictions of future examples are accurate. As a corollary, we show that if the learner is always guaranteed to observe data where future examples are easily predictable, then online learning can be as easy as transductive online learning. Our results complement recent work in online algorithms with predictions and smoothed online classification, which go beyond a worse-case analysis by using machine-learned predictions and distributional assumptions respectively.

3.1 Introduction

In online classification, Nature plays a game with a learner over $T \in \mathbb{N}$ rounds. In each round $t \in [T]$, Nature selects a labeled example $(x_t, y_t) \in \mathcal{X} \times \mathcal{Y}$ and reveals just the example x_t to the learner. The learner, using the history of the game $(x_1, y_1), ..., (x_{t-1}, y_{t-1})$ and the current example x_t, makes a potentially randomized prediction $\hat{y}_t \in \mathcal{Y}$. Finally, Nature reveals the true label y_t and the learner suffers the loss $\mathbb{1}\{\hat{y}_t \neq y_t\}$. Given access to a hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ consisting of functions $h : \mathcal{X} \to \mathcal{Y}$, the goal of the learner is to minimize its regret, the difference between its cumulative mistake and that of the best fixed hypothesis $h \in \mathcal{H}$ in hindsight. We say a class $\mathcal{H}$ is online learnable if there exists a learning algorithm that achieves vanishing average regret for *any*, potentially adversarial chosen, stream of labeled examples $(x_1, y_1), ..., (x_T, y_T)$. Canonically, one also distinguishes between the realizable and agnostic settings. In the realizable setting, Nature must choose a stream $(x_1, y_1), ..., (x_T, y_T)$ such that there exists a $h \in \mathcal{H}$ for which $h(x_t) = y_t$ for all $t \in [T]$. On the other hand, in the agnostic setting, no such assumptions on the stream are placed.

Due to applications in spam filtering, image recognition, and language modeling, online classification has had a long, rich history in statistical learning theory. In a seminal work, Littlestone [1987] provided a sharp quantitative characterization of which binary hypothesis classes $\mathcal{H} \subseteq \{0, 1\}^{\mathcal{X}}$ are online learnable in the realizable setting. This characterization was in terms of the finiteness of a combinatorial dimension called the Littlestone dimension. Twenty-two years later, Ben-David et al. [2009] proved that the Littlestone dimension continues to characterize the online learnability of binary hypothesis classes in the agnostic setting. Later, Daniely et al. [2011] generalized the Littlestone dimension to multiclass hypothesis classes $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, and showed that it fully characterizes multiclass online learnability when the label space $\mathcal{Y}$ is finite. More recently, Hanneke et al. [2023] extended this result to show that the multiclass Littlestone dimension continues to characterize multiclass online learnability even when $\mathcal{Y}$ is unbounded.

While elegant, the characterization of online classification in terms of the Littlestone dimension is often interpreted as an impossibility result [Haghtalab, 2018]. Indeed, due to the restrictive nature of the Littlestone dimension, even simple classes like the 1-dimensional thresholds $\mathcal{H}_{\text{thresh}} = \{x \mapsto \mathbb{1}\{x \geq a\} : a \in \mathbb{N}\}$ are not online learnable in the realizable setting. This hardness arises mainly due to a worst-case analysis: the adversary is allowed to choose any sequence of labeled examples, even possibly adapting to the learner's strategy. In many situations, however, the sequence of data is "easy" and a worst-case analysis is too pessimistic. For example, if one were to use the daily temperatures to predict snowfall, it is unlikely that temperatures will vary rapidly within a given week. Even so, one might have to

access to temperature forecasting models that can accurately predict future temperatures. This motivates a beyond-worst-case analysis of online classification algorithms by proving guarantees that adapt to the "easiness" of the example stream.

The push for a beyond-worst-case analysis has its roots in classical algorithm design [Roughgarden, 2021]. Of recent interest is Algorithms with Predictions (AwP), a specific sub-field of beyond-worst-case analysis of algorithms [Mitzenmacher and Vassilvitskii, 2022]. Here, classical algorithms are given additional information about the problem instance in the form of machine-learned predictions. Augmented with these predictions, the algorithm's goal is to perform optimally on a per-input basis when the predictions are good (known as consistency), while always ensuring optimal worst-case guarantees (known as robustness). Ideally, algorithms are also smooth, obtaining performance guarantees that interpolate between instance and worst-case optimality as a function of prediction quality. After a successful application to learning index structures [Kraska et al., 2018], there has been an explosion of work designing algorithms whose guarantees depend on the quality of available, machine-learned predictions Mitzenmacher and Vassilvitskii [2022]. For example, machine-learned predictions have been used to achieve more efficient data-structures [Lin et al., 2022], faster runtimes [Chen et al., 2022, Ergun et al., 2021], better accuracy-space tradeoffs for streaming algorithms [Hsu et al., 2019], and improved performance bounds for online algorithms [Purohit et al., 2018].

Despite this vast literature, the accuracy benefits of machine-learned predictions for online classification are, to the best of our knowledge, unknown. In this work, we bridge the gap between AwP and online classification. In contrast to previous work, which go beyond a worst-case analysis in online classification through smoothness or other distributional assumptions [Haghtalab et al., 2020, Block et al., 2022, Wu et al., 2023], we give the learner access to a Predictor, a forecasting algorithm that predicts future examples in the data stream. The learner, before predicting a label $\hat{y}_t$, can query the Predictor and receive predictions $\hat{x}_{t+1}, ..., \hat{x}_T$ on the future examples. The learner can then use the history of the game $(x_1, y_1), ..., (x_{t-1}, y_{t-1})$, the current example x_t, and the predictions $\hat{x}_{t+1}, ..., \hat{x}_T$ to output a label $\hat{y}_t$. We allow Predictors to be adaptive - they can change their predictions of future examples based on the actual realizations of past examples. From this perspective, Predictors are also online learners, and we quantify the predictability of example streams through their mistake-bounds.

In this work, we seek to design online learning algorithms whose expected regret, given black-box access to a Predictor, degrades gracefully with the quality of the Predictor's predictions. By doing so, we are also interested in understanding how access to a Predictor may impact the characterization of online learnability. In particular, given a Predictor, when can

online learnability become easier than in the standard, worst-case setup? Guided by these objectives, we make the following contributions.

(1) In the realizable and agnostic settings, we design online learners that, using black-box access to a Predictor, adapt to the "easiness" of the example stream. When the predictions of the Predictor are good, our learner's expected mistakes/regret significantly improves upon the worst-case guarantee. When the Predictor's predictions are bad, the expected mistakes/regret of our learner matches the optimal worst-case expected mistake-bound/regret. Finally, our learner's expected mistake-bound/regret degrades gracefully with the quality of the Predictor's predictions.

(2) We show that having black-box access to a good Predictor can make learning much easier than the standard, worst-case setting. More precisely, good Predictors allow "offline" learnable classes to become online learnable. In this chapter, we take the "offline" setting to be transductive online learning [Ben-David et al., 1997, Hanneke et al., 2024] where Nature reveals the entire sequence of examples $x_1, ..., x_T$ (but not the labels $y_1, ..., y_T$) to the learner before the game begins. Many "offline" learnable classes are not online learnable. For example, when $\mathcal{Y} = \{0, 1\}$, transductive online learnability is characterized by the finiteness of the VC dimension, the same dimension that characterizes PAC learnability. Thus, our result is analogous to that in smoothed online classification, where PAC learnability is also sufficient for online learnability [Haghtalab et al., 2020, Block et al., 2022].

A notable property of our realizable and agnostic online learners is their use of black-box access to a transductive online learner to make predictions. In this sense, our proof strategies involve reducing online classification with predictions to transductive online learning. For both contributions (1) and (2), we consider only the realizable setting in the main text. The results and arguments for the agnostic setting are nearly identical and thus deferred to Appendix A.5.

3.1.1 Related Works

Online Algorithms with Predictions. Online Algorithms with Predictions (OAwP) has emerged as an important paradigm lying at the intersection of classical online algorithm design and machine learning. Many fundamental online decision-making problems including ski rental [Gollapudi and Panigrahi, 2019, Wang et al., 2020, Bamas et al., 2020], online scheduling [Lattanzi et al., 2020, Wei and Zhang, 2020, Scully et al., 2021], online facility location [Almanza et al., 2021, Jiang et al., 2021], caching [Lykouris and Vassilvitskii, 2021,

Elias et al., 2024], and metrical task systems [Antoniadis et al., 2023], have been analyzed under this framework. Recently, Elias et al. [2024] consider a model where the predictor is allowed to learn and adapt its predictions based on the observed data. This is contrast to previous work on learning-augmented online algorithms, where predictions are made from machine learning models trained on historical data, and thus their predictions are static and non-adaptive to the current task at hand. Elias et al. [2024] study a number of fundamental problems, like caching and scheduling, and show how explicitly designed predictors can lead to improved performance bounds. In this work, we consider a model similar to Elias et al. [2024], where the predictions available to the learning algorithms are not fixed, but rather adapt to the true sequence of data processed by the learning algorithm. However, unlike Elias et al. [2024], we do not hand-craft these predictions, but rather assume our learning algorithms have black-box access to a machine-learned prediction algorithm.

Transductive Online Learning. In the Transductive Online Learning setting, Nature reveals the entire sequence of examples $x_1, ..., x_T$ to the learner before the game begins. The goal of the learner is to predict the corresponding labels $y_1, ..., y_T$ in order, receiving the true label y_t only after making the prediction $\hat{y}_t$ for example x_t. First studied by Ben-David et al. [1997], recent work by Hanneke et al. [2024] has established the minimax rates on expected mistakes/regret in the realizable/agnostic settings. In the context of online classification with predictions, one can think of the transductive online learning setting as a special case where the Predictor never makes mistakes.

Smoothed Online Classification. In addition to AwP, smoothed analysis [Spielman and Teng, 2009] is another important sub-field of beyond-worst-case analysis of algorithms. By placing distributional assumptions on the input, one can typically go beyond computational and information-theoretic bottlenecks due to worst-case inputs. To this end, Rakhlin et al. [2011], Haghtalab [2018], Haghtalab et al. [2020], Block et al. [2022] consider a smoothed online classification model. Here, the adversary has to choose and draw examples from sufficiently anti-concentrated distributions. For binary classification, Haghtalab [2018] and Haghtalab et al. [2020] showed that smoothed online learnability is as easy as PAC learnability. That is, the finiteness of a smaller combinatorial parameter called the VC dimension is sufficient for smoothed online classification. In this work, we also go beyond the worst-case analysis standard in online classification, but consider a different model where the adversary is constrained to reveal a sequence of examples that are predictable. In this model, we also show that the VC dimension can be sufficient for online learnability.

3.2 Preliminaries

Let $\mathcal{X}$ denote an example space and $\mathcal{Y}$ denote the label space. We make no assumptions about $\mathcal{Y}$, so it can be unbounded (e.g., $\mathcal{Y} = \mathbb{N}$). Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ denote a hypothesis class. For a set A, let $A^\star = \bigcup_{n=0}^{\infty} A^n$ denote the set of all finite sequences of elements in A. Moreover, we let $A^{\leq n}$ denote the set of all sequences of elements in A of size at most n. Then, $\mathcal{X}^\star$ denotes the set of all finite sequences of examples in $\mathcal{X}$ and $\mathcal{Z} \subseteq \mathcal{X}^\star$ denotes a particular family of such sequences. We abbreviate a sequence $z_1, ..., z_T$ by $z_{1:T}$. Finally, for $a, b, c \in \mathbb{R}$, we let $a \wedge b \wedge c = \min\{a, b, c\}$.

3.2.1 Online Classification

In online classification, a learner $\mathcal{A}$ plays a repeated game against Nature over $T \in \mathbb{N}$ rounds. In each round $t \in [T]$, Nature picks a labeled example $(x_t, y_t) \in \mathcal{X} \times \mathcal{Y}$ and reveals x_t to the learner. The learner makes a randomized prediction $\hat{y}_t \in \mathcal{Y}$. Finally, Nature reveals the true label y_t and the learner suffers the 0-1 loss $\mathbb{1}\{\hat{y}_t \neq y_t\}$. Given a hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, the goal of the learner is to minimize its expected regret

$$\mathrm{R}_{\mathcal{A}}(T, \mathcal{H}) := \sup_{x_{1:T} \in \mathcal{X}} \sup_{y_{1:T} \in \mathcal{Y}^T} \left(\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}(x_t) \neq y_t\} \right] - \inf_{h \in \mathcal{H}} \sum_{t=1}^{T} \mathbb{1}\{h(x_t) \neq y_t\} \right),$$

where the expectation is only over the randomness of the learner. A hypothesis class $\mathcal{H}$ is said to be online learnable if there exists an (potentially randomized) online learning algorithm $\mathcal{A}$ such that $\mathrm{R}_{\mathcal{A}}(T, \mathcal{H}) = o(T)$. If it is guaranteed that the learner always observes a sequence of examples labeled by some hypothesis $h \in \mathcal{H}$, then we say we are in the realizable setting and the goal of the learner is to minimize its expected cumulative mistakes,

$$\mathrm{M}_{\mathcal{A}}(T, \mathcal{H}) := \sup_{x_{1:T} \in \mathcal{X}^T} \sup_{h \in \mathcal{H}} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}(x_t) \neq h(x_t)\} \right],$$

where again the expectation is taken only with respect to the randomness of the learner. It is well known that the finiteness of the multiclass extension of the Littlestone dimension (Ldim) characterizes realizable and agnostic online learnability [Littlestone, 1987, Daniely et al., 2011, Hanneke et al., 2023]. See Appendix A.1 for complete definitions.

3.2.2 Online Classification with Predictions

Motivated by the fact that real-world example streams $x_{1:T}$ are far from worst-case, we give our learner $\mathcal{A}$ black-box access to a Predictor $\mathcal{P}$, defined algorithmically in Algorithm 1 and formally in Definition 3.2.1. In the rest of the chapter, we abuse notation by not explicitly indicating that $\mathcal{P}$ takes its own past predictions as input. That is, given a sequence $x_{1:T} \in \mathcal{X}^T$, we will let $\mathcal{P}(x_{1:t})$ denotes its prediction on the t'th round.

Definition 3.2.1 (Predictor)**.** *A Predictor $\mathcal{P} : (\mathcal{X} \times \mathcal{X}^T)^\star \to \Pi(\mathcal{X}^T)$ is a map that takes in a sequence of instances $x_1, x_2, ...$, its own past predictions $\hat{x}^1_{1:T}, \hat{x}^2_{1:T}, ...$, and outputs a distribution $\hat{\mu} \in \Pi(\mathcal{X}^T)$. The Predictor make its next prediction by sampling $\hat{x}_{1:T} \sim \hat{\mu}$.*

Algorithm 1 Predictor $\mathcal{P}$

Require: Time horizon T
1: **for** $t = 1, ..., T$ **do**
2: Nature reveals the true example x_t.
3: Observe x_t, update, and make a (potentially randomized) prediction $\hat{x}^t_{1:T}$.
4: **end for**

Remark. We highlight that our Predictors are very general and can also use side information, in addition to the past examples, to make predictions about future examples. For example, if the examples are daily average temperatures, then Predictors can also use other covariates, like humidity, precipitation, and wind speed, to predict future temperatures.

In each round $t \in [T]$, the learner $\mathcal{A}$ can query the Predictor $\mathcal{P}$ to get a sense of what examples it will observe in the future. Then, the learner $\mathcal{A}$ can use the history $(x_1, y_1), .., (x_{t-1}, y_{t-1})$, the current example x_t, and the future predicted examples to classify the current example. Protocol 2 makes explicit the interaction between the learner, the Predictor, and Nature.

Algorithm 2 Online Learning with a Predictor

Require: Predictor $\mathcal{P}$, Hypothesis class $\mathcal{H}$, Time horizon T
1: **for** $t = 1, ..., T$ **do**
2: Nature reveals the true example x_t.
3: The Predictor $\mathcal{P}$ observes x_t, updates, and reveals its predictions $\hat{x}^t_{1:T}$.
4: Learner makes a randomized prediction $\hat{y}_t$ using $\hat{x}^t_{1:T}, x_t$, and $(x_1, y_1), ..., (x_{t-1}, y_{t-1})$.
5: Nature reveals the true label y_t to the learner.
6: Learner suffers loss $\mathbb{1}\{y_t \neq \hat{y}_t\}$ and updates itself.
7: **end for**

Note that, in every round $t \in [T]$, the Predictor $\mathcal{P}$ makes a prediction about the entire

sequence of T examples, even those that it has observed in the past. This is mainly for notational convenience as we assume that our Predictors are consistent.

Assumption 1 (Consistency)**.** *A Predictor is consistent if for every sequence $x_{1:T} \in \mathcal{X}^T$ and every time point $t \in [T]$, the prediction $\hat{x}_{1:T} = \mathcal{P}(x_{1:t})$ satisfies the property that $\hat{x}_{1:t} = x_{1:t}$.*

Although stated as an assumption, it is without loss of generality that Predictors are consistent - any inconsistent Predictor can be made consistent by hard coding its input into its output. In addition to consistency, we assume that our Predictors are lazy.

Assumption 2 (Laziness)**.** *A Predictor is lazy if for every sequence $x_{1:T} \in \mathcal{X}^T$ and every $t \in [T]$, if $\mathcal{P}(x_{1:t-1})_t = x_t$, then $\mathcal{P}(x_{1:t}) = \mathcal{P}(x_{1:t-1})$. That is, $\mathcal{P}$ does not change its prediction if it is correct.*

Since Predictors are also online learners, the assumption of laziness is also mild: non-lazy online learners can be generically converted into lazy ones [Littlestone, 1987, 1989]. We always assume that Predictors are consistent and lazy and drop these pronouns for the rest of the chapter.

Remark. We highlight that Predictors are adaptive and change their predictions based on the realizations of past examples. This is contrast to existing literature in OAwP, where machine-learned predictions are often static. Nevertheless, our framework is more general and captures the setting where predictions of examples are made once and fixed throughout the game. Indeed, consider the consistent, lazy Predictor that fixes a sequence $z_{1:T} \in \mathcal{X}^T$ before the game begins, and for every $t \in [T]$, outputs the predictions $\hat{x}^t_{1:T}$ such that $\hat{x}^t_{1:t} = x_{1:t}$ and $\hat{x}^t_{t+1:T} = z_{t+1:T}$.

Ideally, when given access to a Predictor $\mathcal{P}$, the expected regret of $\mathcal{A}$ should degrade gracefully with the quality of $\mathcal{P}$'s predictions. To this end, we quantify the performance of a Predictor $\mathcal{P}$ through

$$\mathrm{M}_{\mathcal{P}}(x_{1:T}) := \mathbb{E}\left[\sum_{t=2}^{T} \mathbb{1}\{\mathcal{P}(x_{1:t-1})_t \neq x_t\}\right],$$

the expected number of mistakes that $\mathcal{P}$ makes on a sequence of examples $x_{1:T} \in \mathcal{X}^T$. In Section 3.3, we design an online learner whose expected regret/mistake-bound on a stream $(x_1, y_1), ..., (x_T, y_T)$ can be written in terms of $\mathrm{M}_{\mathcal{P}}(x_{1:T})$.

3.2.3 Predictability

Predictors and their mistake bounds offer us to ability to define and quantify a notion of "easiness" for example streams $x_{1:T}$. In particular, we can distinguish between example

streams that are predictable and unpredictable. To do so, let $\mathcal{Z} \subseteq \mathcal{X}^\star$ denote a collection of finite sequences of examples. By restricting Nature to playing examples streams in $\mathcal{Z}$, we can define analogous notions of minimax expected regret

$$\mathrm{R}_{\mathcal{A}}(T, \mathcal{H}, \mathcal{Z}) := \sup_{x_{1:T} \in \mathcal{Z}} \sup_{y_{1:T} \in \mathcal{Y}^T} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}(x_t) \neq y_t\} - \inf_{h \in \mathcal{H}} \sum_{t=1}^{T} \mathbb{1}\{h(x_t) \neq y_t\}\right],$$

and minimax expected mistakes,

$$\mathrm{M}_{\mathcal{A}}(T, \mathcal{H}, \mathcal{Z}) := \sup_{x_{1:T} \in \mathcal{Z}} \sup_{h \in \mathcal{H}} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}(x_t) \neq h(x_t)\}\right].$$

As usual, we say that a tuple $(\mathcal{H}, \mathcal{Z})$ is online and realizable online learnable if $\inf_{\mathcal{A}} \mathrm{R}_{\mathcal{A}}(T, \mathcal{H}, \mathcal{Z}) = o(T)$ and $\inf_{\mathcal{A}} \mathrm{M}_{\mathcal{A}}(T, \mathcal{H}, \mathcal{Z}) = o(T)$ respectively. If $\mathcal{Z} = \mathcal{X}^\star$, then the definitions above recover the standard, worst-case online classification setup. However, in the more general case where $\mathcal{Z}^\star \subseteq \mathcal{X}^\star$, we can use the existence of good Predictors $\mathcal{P}$ and their mistake bounds to quantify the "easiness" of a stream class $\mathcal{Z}$. That is, we say $\mathcal{Z}$ is predictable if there exists a consistent, lazy Predictor $\mathcal{P}$ such that $\mathrm{M}_{\mathcal{P}}(T, \mathcal{Z}) := \sup_{x_{1:T} \in \mathcal{Z}} \mathrm{M}_{\mathcal{P}}(x_{1:T}) = o(T)$.

Definition 3.2.2 (Predictability)**.** *A class $\mathcal{Z} \subseteq \mathcal{X}^\star$ is predictable if and only if* $\inf_{\mathcal{P}} \mathrm{M}_{\mathcal{P}}(T, \mathcal{Z}) = o(T)$.

Definition 3.2.2 provides a qualitative definition of what it means for a sequence of examples to be predictable, and therefore "easy". If $\mathcal{Z} \subseteq \mathcal{X}^\star$ is a predictable class of example streams, then a stream $x_{1:T} \in \mathcal{X}^T$ is predictable if $x_{1:T} \in \mathcal{Z}$. By having access to a good Predictor, sequences of examples that previously exhibited "worst-case" behavior, now become predictable. One natural predictable collection of streams are those induced by easy-to-learn discrete-time dynamical systems [Raman et al., 2024b]. That is, let $\mathcal{X}$ be the state space for a finite collection $\mathcal{G}$ of transition functions. Then, given an initial state $x_0 \in \mathcal{X}$, one can consider the stream class $\mathcal{Z}$ to be the set of all trajectories induced by transition functions in $\mathcal{G}$. In Section 3.3, we show that for such classes of predictable examples, "offline" learnability is sufficient for online learnability.

3.2.4 Offline Learnability

In the classical analysis of online algorithms, one competes against the best "offline" solution. In the context of online classification, this amounts to comparing online learnability to "offline" learnability, where we interpret the "offline" setting as the case where Nature reveals

the sequence of examples $(x_1, ..., x_T)$ before the game begins. In particular, compared to the standard online learning setting, in the "offline" version, the learner knows the sequence of examples $x_1, ..., x_T$ before the game begins, and its goal is to predict the corresponding labels $y_1, ..., y_T$. This setting was recently named "Transductive Online Learning" [Hanneke et al., 2024] and the minimax rates in both the realizable and agnostic setting have been established [Ben-David et al., 1997, Hanneke et al., 2023]. For the remainder of the chapter, we will use offline and transductive online learnability interchangeably.

For a randomized offline learner $\mathcal{B}$, we let

$$\mathrm{R}_{\mathcal{B}}(T, \mathcal{H}) := \sup_{x_{1:T} \in \mathcal{X}^T} \sup_{y_{1:T} \in \mathcal{Y}^T} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{B}_{x_{1:T}}(x_t) \neq y_t\} - \inf_{h \in \mathcal{H}} \sum_{t=1}^{T} \mathbb{1}\{h(x_t) \neq y_t\}\right]$$

denote its minimax expected regret and

$$\mathrm{M}_{\mathcal{B}}(T, \mathcal{H}) := \sup_{h \in \mathcal{H}} \sup_{x_{1:T} \in \mathcal{X}^T} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{B}_{x_{1:T}}(x_t) \neq h(x_t)\}\right].$$

denote its minimax expected mistakes. We use the notation $\mathcal{B}_{x_{1:T}}$ to indicate that $\mathcal{B}$ was initialized with the sequence $x_{1:T}$ before the game begins. If $\mathrm{M}_{\mathcal{B}}(T, \mathcal{H}) = o(T)$ or $\mathrm{R}_{\mathcal{B}}(T, \mathcal{H}) = o(T)$, then we say that $\mathcal{B}$ is a no-regret offline learner. It turns out that realizable and agnostic offline learnability are equivalent [Hanneke et al., 2024]. That is, $\mathrm{M}_{\mathcal{B}}(T, \mathcal{H}) = o(T) \Leftrightarrow \mathrm{R}_{\mathcal{B}}(T, \mathcal{H}) = o(T)$. Thus, we say a class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ is offline learnable if and only if there exists a no-regret offline learner for $\mathcal{H}$.

When $|\mathcal{Y}| = 2$, Ben-David et al. [1997] and Hanneke et al. [2023] show that the finiteness of a combinatorial dimension called the Vapnik–Chervonenkis (VC) dimension (or equivalently PAC learnability) is sufficient for offline learnability (see Appendix A.1 for complete definitions).

Lemma 3.2.1 (Ben-David et al. [1997], Hanneke et al. [2024])**.** *For every $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$, there exists a deterministic offline learner $\mathcal{B}$ such that*

$$\mathrm{M}_{\mathcal{B}}(T, \mathcal{H}) = O\Big(\mathrm{VC}(\mathcal{H}) \log_2 T\Big)$$

where $\mathrm{VC}(\mathcal{H})$ is the VC dimension of $\mathcal{H}$.

In Section 3.3, we use this upper bound in Lemma 3.2.1 to prove that PAC learnability of $\mathcal{H}$ implies $(\mathcal{H}, \mathcal{Z})$ online learnability when $\mathcal{Z}$ is predictable. Interestingly, Hanneke et al. [2024] also establish a trichotomy in the minimax expected mistakes for offline learning in the realizable setting. That is, for any $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ with $|\mathcal{Y}| < \infty$, the quantity $\mathrm{M}_{\mathcal{B}}(T, \mathcal{H})$ can

only be $\Theta(1)$, $\Theta(\log_2 T)$, or $\Theta(T)$. On the other hand, in the agnostic setting, $\mathrm{R}_{\mathcal{B}}(T, \mathcal{H})$ can be $\tilde{\Theta}(\sqrt{T})$ or $\Theta(T)$, where $\tilde{\Theta}$ hides poly-log terms in T.

Our main result in Section 3.3 shows that offline learnability is sufficient for online learnability under predictable examples. The following technical lemma will be important when proving so.

Lemma 3.2.2. *[Woess, 2017, Lemma 5.17] Let $g : \mathbb{Z}_+ \mapsto \mathbb{R}_+$ be a positive sublinear function. Then, g is bounded from above by a concave sublinear function $f : \mathbb{R}_+ \mapsto \mathbb{R}_+$.*

In light of Lemma 3.2.2, we let $\bar{f}$ denote the smallest concave sublinear function upper bounding the positive sublinear function f. For example, our regret bounds in Section 3.3 will often be in terms of $\overline{\mathrm{M}}_{\mathcal{B}}(T, \mathcal{H})$. Although in full generality $\mathrm{M}_{\mathcal{B}}(T, \mathcal{H}) \leq \overline{\mathrm{M}}_{\mathcal{B}}(T, \mathcal{H})$, in many cases we have equality. For example, when $|\mathcal{Y}| = 2$, the trichotomy of expected minimax rates established by Theorem 4.1 in Hanneke et al. [2024] shows that $\mathrm{M}_{\mathcal{B}}(T, \mathcal{H}) = \overline{\mathrm{M}}_{\mathcal{B}}(T, \mathcal{H})$.

3.3 Adaptive Rates in the Realizable Setting

In this section, we design learning algorithms whose expected mistake bounds, given black-box access to a Predictor $\mathcal{P}$ and offline learner $\mathcal{B}$, adapt to the quality of predictions by $\mathcal{P}$ and $\mathcal{B}$. Our main quantitative result is stated below.

Theorem 3.3.1 (Realizable upper bound)**.** *For every $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, Predictor $\mathcal{P}$, and no-regret offline learner $\mathcal{B}$, there exists an online learner $\mathcal{A}$ such that for every realizable stream $(x_1, y_1), ..., (x_T, y_T)$, $\mathcal{A}$ makes at most*

$$3\Bigg(\underbrace{\mathrm{L}(\mathcal{H})}_{(i)} \wedge \underbrace{(\mathrm{M}_{\mathcal{P}}(x_{1:T}) + 1)\,\mathrm{M}_{\mathcal{B}}(T, \mathcal{H})}_{(ii)} \wedge \underbrace{6\Big((\mathrm{M}_{\mathcal{P}}(x_{1:T}) + 1)\,\overline{\mathrm{M}}_{\mathcal{B}}\Big(\frac{T}{\mathrm{M}_{\mathcal{P}}(x_{1:T}) + 1} + 1, \mathcal{H}\Big) + \log_2 T\Big)}_{(iii)}\Bigg) + 5$$

mistakes in expectation, where $\mathrm{L}(\mathcal{H})$ is the Littlestone dimension of $\mathcal{H}$.

We highlight some important consequences of Theorem 3.3.1. Firstly, when $\mathrm{M}_{\mathcal{P}}(x_{1:T}) = 0$, the expected mistake bound of $\mathcal{A}$ matches (up to constant factors) that of the offline learner $\mathcal{B}$. Thus, when $\mathrm{M}_{\mathcal{P}}(x_{1:T}) = 0$ and $\mathcal{B}$ is a minimax optimal offline learner, our learner $\mathcal{A}$ performs as well as the best offline learner. Secondly, the expected mistake bound of $\mathcal{A}$ is always at most $3\,\mathrm{L}(\mathcal{H}) + 5$; the minimax worst-case mistake bound up to constant

factors. Thus, our learner $\mathcal{A}$ never does worse than the worst-case mistake bound. Thirdly, the expected mistake bound of $\mathcal{A}$ gracefully interpolates between the offline and worst-case optimal rates as a function of $\mathrm{M}_{\mathcal{P}}(x_{1:T})$. In Section 3.3.3, we show that the dependence of $\mathcal{A}$'s mistake bound on $\mathrm{M}_{\mathcal{P}}(x_{1:T})$ and $\mathrm{M}_{\mathcal{B}}(T, \mathcal{H})$ can be tight. Lastly, we highlight that Theorem 3.3.1 makes no assumption about the size of $\mathcal{Y}$.

With respect to learnability, Corollary 3.3.2 shows that offline learnability of $\mathcal{H}$ is sufficient for online learnability under predictable examples.

Corollary 3.3.2 (Offline learnability $\implies$ Realizable Online learnability with Predictable Examples)**.** *For every $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ and $\mathcal{Z} \subseteq \mathcal{X}^{\star}$,*

$$\mathcal{Z} \text{ is predictable and } \mathcal{H} \text{ is offline learnable} \implies (\mathcal{H}, \mathcal{Z}) \text{ is realizable online learnable.}$$

This follows from a slight modification of the proof of Theorem 3.3.1 along with the fact that the term $(\mathrm{M}_{\mathcal{P}}(T, \mathcal{Z}) + 1)\overline{\mathrm{M}}_{\mathcal{B}}\Big(\frac{T}{\mathrm{M}_{\mathcal{P}}(T,\mathcal{Z})+1}, \mathcal{H}\Big) = o(T)$ when $\mathrm{M}_{\mathcal{B}}(T, \mathcal{H}) = o(T)$ and $\mathrm{M}_{\mathcal{P}}(T, \mathcal{Z}) = o(T)$. In addition, we can also establish a quantitative version of Corollary 3.3.2 for VC classes.

Corollary 3.3.3. *For every $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$, Predictor $\mathcal{P}$ and $\mathcal{Z} \subseteq \mathcal{X}^{\star}$, there exists an online learner $\mathcal{A}$ such that*

$$\mathrm{M}_{\mathcal{A}}(T, \mathcal{H}, \mathcal{Z}) = O\left(\mathrm{VC}(\mathcal{H})(\mathrm{M}_{\mathcal{P}}(T, \mathcal{Z}) + 1)\log_2\Big(\frac{T}{\mathrm{M}_{\mathcal{P}}(T, \mathcal{Z}) + 1}\Big) + \log_2 T\right).$$

We prove both Corollary 3.3.2 and 3.3.3 in Appendix A.3.1. Corollary 3.3.3 shows that PAC learnability implies online learnability under predictable examples. Moreover, for VC classes, when $\mathrm{M}_{\mathcal{P}}(x_{1:T}) = 0$, the upper bound in Corollary 3.3.3 exactly matches that of Lemma 3.2.1. An analogous corollary in terms of the Natarajan dimension (see Appendix A.1 for definition) holds when $|\mathcal{Y}| < \infty$.

The remainder of this section is dedicated to proving Theorem 3.3.1. The proof involves constructing three different online learners, with expected mistake bounds (i), (ii), and (iii) respectively, and then running the Deterministic Weighted Majority Algorithm (DWMA) using these learners as experts [Arora et al., 2012]. The following guarantee of DWMA along with upper bounds (i), (ii), and (iii) gives the upper bound in Theorem 3.3.1 (see Appendix A.4.1 for complete proof).

Lemma 3.3.4 (DWMA guarantee [Arora et al., 2012])**.** *The DWMA run with N experts and learning rate $\eta = 1/2$ makes at most $3(\min_{i\in[N]} M_i + \log_2 N)$ mistakes, where M_i is the number of mistakes made by expert $i \in [N]$.*

The online learner obtaining the upper bound $\mathrm{L}(\mathcal{H})$ is the celebrated Standard Optimal Algorithm [Littlestone, 1987, Daniely et al., 2011], and thus we omit the details here. Our second and third learners are described in Sections 3.3.1 and 3.3.2 respectively. Finally, in Section 3.3.3, we give a lower bound showing that our upper bound in Theorem 3.3.1 can be tight.

3.3.1 Proof of upper bound (ii) in Theorem 3.3.1

Consider a lazy, consistent predictor $\mathcal{P}$. Given any sequence of examples $x_{1:T} \in \mathcal{X}^T$, the Predictor $\mathcal{P}$ makes $c \in \mathbb{N}$ mistakes at some timepoints $t_1, ..., t_c \in [T]$. Since $\mathcal{P}$ may be randomized, both c and $t_1, ..., t_c$ are random variables. Crucially, since $\mathcal{P}$ is lazy, for every $i \in \{0, ..., c+1\}$, the predictions made by $\mathcal{P}$ on timepoints strictly between t_i and t_{i+1} are correct and remain unchanged, where we take $t_0 = 0$ and $t_{c+1} = T + 1$. This means that on round t_i, we have that $\hat{x}^{t_i}_{t_i:t_{i+1}-1} = x_{t_i:t_{i+1}-1}$. Therefore, initializing a fresh copy of an offline learner $\mathcal{B}$ with the predictions $\hat{x}^{t_i}_{t_i:T}$ ensures that $\mathcal{B}$ makes at most $\mathrm{M}_{\mathcal{B}}(T - t_i + 1, \mathcal{H})$ mistakes on the stream $(x_{t_i}, y_{t_i}), ..., (x_{t_{i+1}-1}, y_{t_{i+1}-1})$. Repeating this argument for all adjacent pairs of timepoints in $\{t_1, ..., t_c\}$, gives the following strategy: initialize a new offline learner $\mathcal{B}$ every time $\mathcal{P}$ makes a mistakes, and use $\mathcal{B}$ to make predictions until the next time $\mathcal{P}$ makes a mistake. Algorithm 3 implements this idea.

Algorithm 3 Online Learner

Require: Hypothesis class $\mathcal{H}$, Offline learner $\mathcal{B}$, Time horizon T

Initialize: $i = 0$
for $t = 1, ..., T$ **do**
  Receive x_t from Nature.
  Receive predictions $\hat{x}^t_{1:T}$ from Predictor $\mathcal{P}$ such that $\hat{x}^t_{1:t} = x_{1:t}$.
  if $t = 1$ or $\hat{x}^{t-1}_t \neq x_t$ (i.e. $\mathcal{P}$ made a mistake) **then**
    Let $\mathcal{B}^i$ be a new copy of $\mathcal{B}$ initialized with the sequence $\hat{x}^t_{t:T}$ and set $i \leftarrow i + 1$.
  end if
  Query $\mathcal{B}^i$ on example x_t and play its returned prediction $\hat{y}_t$.
  Receive true label y_t from Nature and pass it to $\mathcal{B}^i$.
end for

Lemma 3.3.5. *For every $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, Predictor $\mathcal{P}$, no-regret offline learner $\mathcal{B}$, and realizable stream $(x_1, y_1), ..., (x_T, y_T)$, Algorithm 3 makes at most $(\mathrm{M}_{\mathcal{P}}(x_{1:T}) + 1)\,\mathrm{M}_{\mathcal{B}}(T, \mathcal{H})$ mistakes in expectation.*

Proof. Let $\mathcal{A}$ denote Algorithm 3, $(x_1, y_1), ..., (x_T, y_T)$ denote the realizable stream to be observed by $\mathcal{A}$, and $h^\star \in \mathcal{H}$ to be the labeling hypothesis. Let c be the random variable

denoting the number of mistakes made by Predictor $\mathcal{P}$ on the stream and $t_1, ..., t_c$ be the random variables denoting the time points where $\mathcal{P}$ makes these errors (e.g . $\hat{x}_{t_i}^{t_i-1} \neq x_{t_i}$). Note that $t_i \geq 2$ for all $i \in [c]$. We will show pointwise for every value of c and $t_1, ..., t_c$ that $\mathcal{A}$ makes at most $(c+1)\,\mathrm{M}_{\mathcal{B}}(T, \mathcal{H})$ mistakes in expectation over the randomness of $\mathcal{B}$. Taking an outer expectation with respect to the randomness of $\mathcal{P}$ and using the fact that $\mathbb{E}[c] = \mathrm{M}_{\mathcal{P}}(x_{1:T})$, completes the proof.

First, consider the case where $c = 0$ (i.e. $\mathcal{P}$ makes no mistakes). Then, since $\mathcal{P}$ is lazy, we have that $\hat{x}_{1:T}^{t} = x_{1:T}$ for every $t \in [T]$. Thus line 5 fires exactly once on round $t = 1$, $\mathcal{A}$ initializes an offline learner $\mathcal{B}^1$ with $x_{1:T}$, and $\mathcal{A}$ uses $\mathcal{B}^1$ to make its prediction on all rounds. Thus, $\mathcal{A}$ makes at most $\mathrm{M}_{\mathcal{B}}(T, \mathcal{H})$ mistakes in expectation.

Now, let $c > 0$ and $t_1, ..., t_c$ be the time points where $\mathcal{P}$ errs. Partition the sequence $1, ..., T$ into the disjoint intervals $(1, ..., t_1 - 1)$, $(t_1, ..., t_2 - 1), ..., (t_c, ..., T)$. Define $t_0 := 1$ and $t_{c+1} := T$. Fix an $i \in \{0, ..., c\}$. Observe that for every $j \in \{t_i, ..., t_{i+1} - 1\}$, we have that $\hat{x}_{1:t_{i+1}-1}^{j} = x_{t_{i+1}-1}$. This comes from the fact that $\mathcal{P}$ does not error on timepoints $t_i + 1, ..., t_{i+1} - 1$ and is both consistent and lazy (see Assumptions 1 and 2). Thus, line 5 fires on round t_i, $\mathcal{A}$ initializes an offline learner $\mathcal{B}^i$ with the sequence $\hat{x}_{t_i:T}^{t_i} = x_{t_i:t_{i+1}-1} \circ \hat{x}_{t_{i+1}:T}^{t_i}$, and $\mathcal{A}$ uses $\mathcal{B}^i$ it to make predictions for all remaining timepoints $t_i, ..., t_{i+1} - 1$. Note that line 5 does not fire on timepoints $t_i + 1, ..., t_{i+1} - 1$.

Consider the hypothetical labeled stream of examples $(\hat{x}_{t_i}^{t_i}, h^\star(\hat{x}_{t_i}^{t_i})), ..., (\hat{x}_{T}^{t_i}, h^\star(\hat{x}_{T}^{t_i}))$ equal to

$$(x_{t_i}, y_{t_i}), ..., (x_{t_{i+1}-1}, y_{t_{i+1}-1}), (\hat{x}_{t_{i+1}}^{t_i}, h^\star(\hat{x}_{t_{i+1}}^{t_i})), ..., (\hat{x}_{T}^{t_i}, h^\star(\hat{x}_{T}^{t_i})).$$

By definition, $\mathcal{B}^i$, after initialized with $\hat{x}_{t_i:T}^{t_i}$, makes at most $\mathrm{M}_{\mathcal{B}}(T - t_i + 1, \mathcal{H})$ mistakes in expectation when simulated on the stream $(\hat{x}_{t_i}^{t_i}, h^\star(\hat{x}_{t_i}^{t_i})), ..., (\hat{x}_{T}^{t_i}, h^\star(\hat{x}_{T}^{t_i}))$. Thus, $\mathcal{B}^i$ makes at most $\mathrm{M}_{\mathcal{B}}(T, \mathcal{H})$ mistakes in expectation on the <u>prefix</u> $(\hat{x}_{t_i}^{t_i}, h^\star(\hat{x}_{t_i}^{t_i})), ..., (\hat{x}_{t_{i+1}-1}^{t_i}, h^\star(\hat{x}_{t_{i+1}-1}^{t_i})) = (x_{t_i}, y_{t_i}), ..., (x_{t_{i+1}-1}, y_{t_{i+1}-1})$. Since on the interval timepoint t_i, $\mathcal{A}$ instantiates $\mathcal{B}^i$ with the sequence $\hat{x}_{t_i:T}^{t_i}$ and proceeds to simulate $\mathcal{B}^i$ on the sequence of labeled examples $(x_{t_i}, y_{t_i}), ..., (x_{t_{i+1}-1}, y_{t_{i+1}-1})$, $\mathcal{A}$ makes at most $\mathrm{M}_{\mathcal{B}}(T, \mathcal{H})$ mistakes in expectation on the sequence $(x_{t_i}, y_{t_i}), ..., (x_{t_{i+1}-1}, y_{t_{i+1}-1})$. Since the interval i was chosen arbitrarily, the above analysis is true for every $i \in \{0, ..., c\}$ and therefore $\mathcal{A}$ makes at most $(c+1)\,\mathrm{M}_{\mathcal{B}}(T, \mathcal{H})$ mistakes in expectation over the entire stream. ■

3.3.2 Proof sketch of upper bound (iii) in Theorem 3.3.1

When $\mathrm{M}_{\mathcal{B}}(T, \mathcal{H})$ is large (i.e. $\Omega(\sqrt{T})$), upper bound (ii) is sub-optimal. Indeed, if $t_1, ..., t_c$ denotes the timepoints where $\mathcal{P}$ makes mistakes on the stream $x_{1:T}$, then Algorithm 3 initializes offline learners with sequences of length $T - t_i + 1$. The resulting mistake-bound of

these offline learners are then in the order of $T - t_i + 1$, which can be large if $t_1, ..., t_c$ are evenly spaced across the time horizon. To overcome this, we construct a family $\mathcal{E}$ of online learners, each of which explicitly controls the length of the sequences offline learners can be initialized with. Finally, we run DWMA using $\mathcal{E}$ as its set of experts. Our family of online learners is parameterized by integers $c \in \{0, ..., T-1\}$. Given an input $c \in \{0, ..., T-1\}$, the online learner parameterized by c partitions the stream into $c+1$ roughly equally sized parts of size $\lceil \frac{T}{c+1} \rceil$ and runs a fresh copy of Algorithm 3 on each partition. In this way, the online learner parameterized by c ensures that offline learners are initialized with time horizons at most $\lceil \frac{T}{c+1} \rceil$. Algorithm 4 formalizes this online learner and Lemma 3.3.6, whose proof is in Appendix A.2.1, bounds its expected number of mistakes.

Algorithm 4 Expert(c)

Require: Copy of Algorithm 3 denoted $\mathcal{K}$, Offline Learner $\mathcal{B}$, Time horizon T
1: **Initialize:** $\tilde{t}_i = i \lceil \frac{T}{c+1} \rceil$ for $i \in \{1, ..., c\}$, $\tilde{t}_0 = 0$, and $\tilde{t}_{c+1} = T$.
2: **for** $t = 1, ..., T$ **do**
3: Let $i \in \{0, ..., c\}$ such that $t \in \{\tilde{t}_i + 1, ..., \tilde{t}_{i+1}\}$.
4: **if** $t = \tilde{t}_i + 1$ **then**
5: Let $\mathcal{K}_i$ be a new copy of $\mathcal{K}$ initialized with time horizon $\tilde{t}_{i+1} - \tilde{t}_i$ and a new copy of $\mathcal{B}$.
6: **end if**
7: Receive x_t from Nature.
8: Receive predictions $\hat{x}^t_{1:T}$ from Predictor $\mathcal{P}$ such that $\hat{x}^t_{1:t} = x_{1:t}$.
9: Forward x_t and $\hat{x}^t_{\tilde{t}_i+1:\tilde{t}_{i+1}}$ to $\mathcal{K}_i$ via Lines 3 and 4 of Algorithm 3 respectively.
10: Receive $\hat{y}_t$ from $\mathcal{K}_i$ via line 8 in Algorithm 3 and predict $\hat{y}_t$.
11: Receive true label y_t and forward it to $\mathcal{K}_i$ via line 9 in Algorithm 3.
12: **end for**

Lemma 3.3.6 (Expert guarantee)**.** *For any $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, Predictor $\mathcal{P}$, and no-regret offline learner $\mathcal{B}$, Algorithm 4, given as input $c \in \{0, ..., T-1\}$, makes at most*

$$(\mathrm{M}_{\mathcal{P}}(x_{1:T}) + c + 1)\overline{\mathrm{M}}_{\mathcal{B}}\Big(\frac{T}{c+1} + 1, \mathcal{H}\Big)$$

mistakes in expectation on any realizable stream $(x_1, y_1), ..., (x_T, y_T)$.

Note that when $c = 0$ and $\mathrm{M}_{\mathcal{B}}(T, \mathcal{H}) = \overline{\mathrm{M}}_{\mathcal{B}}(T, \mathcal{H})$, this bound reduces to the one in Lemma 3.3.5 up to a constant factor. On the other hand, using $c = \lceil \mathrm{M}_{\mathcal{P}}(x_{1:T}) \rceil$ gives the upper bound

$$2(\mathrm{M}_{\mathcal{P}}(x_{1:T}) + 1)\overline{\mathrm{M}}_{\mathcal{B}}\Big(\frac{T}{\mathrm{M}_{\mathcal{P}}(x_{1:T}) + 1} + 1, \mathcal{H}\Big).$$

Since $\mathcal{E}$ contains an Expert parameterized for every $c \in \{0, ..., T-1\}$, there always exists an expert $E_{\lceil \mathrm{M}_{\mathcal{P}}(x_{1:T}) \rceil} \in \mathcal{E}$ initialized with input $c = \lceil \mathrm{M}_{\mathcal{P}}(x_{1:T}) \rceil$. Running DWMA using these set of experts $\mathcal{E}$ on the data stream $(x_1, y_1), ..., (x_T, y_T)$ ensures that our learner does not perform too much worse than $E_{\lceil \mathrm{M}_{\mathcal{P}}(x_{1:T}) \rceil}$. Algorithm 5 formalizes this idea and Lemma 3.3.7 is proved in Appendix A.2.1.

Algorithm 5 Online learner

Require: Hypothesis class $\mathcal{H}$, Offline learner $\mathcal{B}$, Time horizon T

1: For every $b \in \{0, ..., T-1\}$ let E_b denote Algorithm 4 parameterized by input b.
2: Run the DWMA using $\{E_b\}_{b \in \{0,...,T-1\}}$ over the stream $(x_1, y_1), ..., (x_T, y_T)$.

Lemma 3.3.7. *For every $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, Predictor $\mathcal{P}$, and no-regret offline learner $\mathcal{B}$, Algorithm 5 makes at most*

$$6\left((\mathrm{M}_{\mathcal{P}}(x_{1:T}) + 1)\overline{\mathrm{M}}_{\mathcal{B}}\Big(\frac{T}{\mathrm{M}_{\mathcal{P}}(x_{1:T}) + 1} + 1, \mathcal{H}\Big) + \log_2 T\right).$$

mistakes in expectation on any realizable stream $(x_1, y_1), ..., (x_T, y_T)$.

3.3.3 Lower bounds

In light of Theorem 3.3.1, it is natural ask whether the upper bounds derived in Section 3.3 are tight. A notable feature in upper bounds (ii) and (iii) is the product of the two mistake bounds $\mathrm{M}_{\mathcal{P}}(x_{1:T})$ and $\mathrm{M}_{\mathcal{B}}(T, \mathcal{H})$. Can this product can be replaced by a sum? Unfortunately, Theorem 3.3.8 shows that the upper bound in Theorem 3.3.1 can be tight.

Theorem 3.3.8. *Let $\mathcal{X} = [0, 1] \cup \{\star\}, \mathcal{Y} = \{0, 1\}$, and $\mathcal{H} = \{x \mapsto \mathbb{1}\{x \leq a\}\mathbb{1}\{x \neq \star\}\}$. Let $T, n \in \mathbb{N}$ be such that $n+1$ divides T and $\frac{T}{n+1} + 1 = 2^k$ for some $k \in \mathbb{N}$. Then, there exists a Predictor $\mathcal{P}$ such that for every online learner $\mathcal{A}$ that uses $\mathcal{P}$ according to Protocol 2, there exists a realizable stream $(x_1, y_1), ..., (x_T, y_T)$ such that $\mathrm{M}_{\mathcal{P}}(x_{1:T}) = n$ but*

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}(x_t) \neq y_t\}\right] \geq \frac{(n+1)}{2} \log_2\Big(\frac{T}{n+1}\Big).$$

Theorem 3.3.8 shows that the upper bound in Theorem 3.3.1 is tight up to an additive factor in $\log_2 T$ because Lemma 3.2.1 gives that $\inf_{\mathcal{B}} \overline{\mathrm{M}}_{\mathcal{B}}(T, \mathcal{H}) = O(\mathrm{VC}(\mathcal{H}) \log_2 T)$ and $\mathrm{VC}(\mathcal{H}) = 1$. The proof of Theorem 3.3.8 is technical and provided in Appendix A.4.2. Our proof involves four steps. First, we construct a class of streams $\mathcal{Z}_n \subseteq \mathcal{X}^\star$. Then, using $\mathcal{Z}_n$, we construct a deterministic, lazy, consistent Predictor $\mathcal{P}$ such that $\mathcal{P}$ makes

mistakes exactly on timepoints $\{\frac{T}{n+1}+1, ..., \frac{nT}{n+1}+1\}$ for every stream $x_{1:T} \in \mathcal{Z}_n$. Next, whenever $x_{1:T} \in \mathcal{Z}_n$, we establish an equivalence between the game defined by Protocol 2 when given access to Predictor $\mathcal{P}$ and Online Classification with Peeks, a different game where there is no Predictor, but the learner observes the next $\frac{T}{n+1}$ examples at timepoints $t \in \{1, \frac{T}{n+1}+1, ..., \frac{nT}{n+1}+1\}$. Finally, for Online Classification with Peeks, we give a strategy for Nature such that it can force any online learner to make $\frac{(n+1)\log_2(\frac{T}{n+1})}{2}$ mistakes in expectation while ensuring that its selected stream satisfies $x_{1:T} \in \mathcal{Z}_n$ and $\inf_{h\in\mathcal{H}} \sum_{t=1}^{T} \mathbb{1}\{h(x_t) \neq y_t\} = 0$. A key component of the fourth step is the stream constructed by [Hanneke et al., 2024, Claim 3.4] to show that the minimax mistakes for classes with infinite Ldim is at least $\log_2 T$ in the offline setting.

Remark. Although Theorem 3.3.8 is stated using the class of one dimensional thresholds, it can be adapted to hold for any VC class with infinite Ldim as these classes embed thresholds [Alon et al., 2019, Theorem 3].

CHAPTER 4

The Complexity of Sequential Prediction in Dynamical Systems

In this chapter, we study the problem of learning to predict the next state of a dynamical system when the underlying evolution function is unknown. Unlike previous work, we place no parametric assumptions on the dynamical system, and study the problem from a learning theory perspective. We define new combinatorial measures and dimensions and show that they quantify the optimal mistake and regret bounds in the realizable and agnostic settings respectively. By doing so, we find that in the realizable setting, the total number of mistakes can grow according to any increasing function of the time horizon T. In contrast, we show that in the agnostic setting under the commonly studied notion of Markovian regret, the only possible rates are $\Theta(T)$ and $\tilde{\Theta}(\sqrt{T})$.

4.1 Introduction

A discrete-time dynamical system is a mathematical model that describes the evolution of a system over discrete time steps. Formally, a discrete-time dynamical system is a tuple $(\mathbb{N}, \mathcal{X}, f)$, where $\mathbb{N}$ is the set of natural numbers that denote the timesteps, $\mathcal{X}$ is a non-empty set called the state space, and $f : \mathcal{X} \to \mathcal{X}$ is a deterministic map that describes the evolution of the state. Dynamical systems have been widely used in practice due to their ability to accurately model natural phenomena. For instance, boolean networks are an important class of discrete-time, discrete-space dynamical systems with widespread applicability to genetic modeling [Kauffman, 1969, Shmulevich et al., 2002]. In a boolean network, the state space is $\mathcal{X} = \{0,1\}^n$ with $|\mathcal{X}|^{|\mathcal{X}|} = (2^n)^{2^n}$ possible evolution functions. For genetic modeling, n is taken to be the number of genes and $x \in \mathcal{X}$ indicates the expression of all n genes under consideration. As an example, "1" could represent the gene with a high concentration of a certain protein, and "0" could represent the gene with a low concentration. With such formulation, one can study how the state of these genes evolves over time under certain medical interventions. Beyond genetics, dynamical systems have been used in control [Li et al., 2019], computer vision [Doretto et al., 2003], and natural language processing [Sutskever et al., 2014, Belanger and Kakade, 2015].

In this work, we consider the problem of predicting the next state of a dynamical system when the underlying evolution function is unknown [Ghai et al., 2020]. To capture the sequential nature of dynamical systems, we consider the model where the learner plays a sequential game with nature over T rounds. At the beginning of the game, nature reveals the initial state $x_0 \in \mathcal{X}$. In each round $t \in [T]$, the learner makes its prediction of the next state $\hat{x}_t \in \mathcal{X}$, nature reveals the true next state $x_t \in \mathcal{X}$, and the learner suffers loss $\mathbb{1}\{\hat{x}_t \neq x_t\}$. Given an evolution class $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$, the goal of the learner is to output predictions of the next state such that its regret, the difference between its cumulative mistakes and the cumulative mistakes of the best-fixed evolution in hindsight (defined formally in Section 4.2.2), is small. The class $\mathcal{F}$ is said to be learnable if there exists a learning algorithm whose regret is a sublinear function of the time horizon T.

Although we allow the state space $\mathcal{X}$ to be arbitrary, we only consider the 0-1 loss, which may be more appropriate for *discrete-space* dynamical systems. However, even discrete-space dynamical systems can be very expressive, capturing complex processes like cellular automata [Hoekstra et al., 2010, Wolfram, 1986] and language modeling [Elman, 1995]. For example, let $\mathcal{V}$ be a countable token space and $\mathcal{X} = \mathcal{V}^\star$ to be the state space containing all finite sequences of elements from $\mathcal{V}$. Consider a function class $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ with the following property: for every $f \in \mathcal{F}$ and any input $x \in \mathcal{X}$, $f(x) = x \circ v$, where we use $\circ$ to represent the

concatenation operator. Then, $\mathcal{F}$ is a set of auto-regressive models which given a sequence of tokens $x \in \mathcal{X}$, predicts the next token $v \in \mathcal{V}$ in the sequence. In particular, $\mathcal{F}$ could be a class of language models which use deterministic decoding strategies (e.g. greedy decoding) to output the next token [Chorowski and Jaitly, 2016]. For such a function class $\mathcal{F}$, one might be interested in understanding the optimal number of mistakes for next-token prediction when the true sequence of tokens is generated by some unknown $f \in \mathcal{F}$.

Given any learning problem $(\mathcal{X}, \mathcal{F})$, we aim to find necessary and sufficient conditions for the learnability of $\mathcal{F}$ while also quantifying the minimax rates under two notions of expected regret: Markovian and Flow regret (Equations 4.1 and 4.2 respectively). To that end, our main contributions are summarized below.

(i) We provide a quantitative characterization of learnability in the realizable setting, when there exists an evolution function in the class $\mathcal{F}$ that generates the sequence of states. Our characterization is in terms of a new combinatorial complexity measure we call the Evolution complexity. Using this characterization, we show that all rates are possible for the minimax expected mistakes when learning dynamical systems in the realizable setting. This is in contrast to online multiclass classification where only two rates are possible. Finally, we compare realizable learnability of dynamical systems to realizable learnability in PAC and online classification.

(ii) In the agnostic setting, we lower and upper bound the minimax Markovian regret in terms of the Littlestone dimension. This result shows that the finiteness of the Littlestone dimension characterizes learnability under Markovian regret. As a corollary, we establish a separation between realizable and agnostic learnability under Markovian regret. We show this separation between realizable and agnostic learnability continues to hold when considering Flow regret. However, if the evolution class has uniformly bounded projections, we show that realizable and agnostic learnability under Flow regret are equivalent.

Our characterization of realizable learnability in terms of the Evolution complexity follows from standard techniques in online classification. However, our results showing all possible rates requires a careful construction of a family of classes which have not been studied in learning theory. Likewise, our comparisons of realizable learnability require a careful construction of non-trivial classes, and computing their combinatorial dimensions.

In the agnostic setting, our upper bound on the minimax Markovian regret in terms of the Littlestone dimension results from reducing learning dynamical systems to online multiclass classification. However, both of the lower bounds $\Omega(\sqrt{T})$ and $\Omega(\mathrm{L}(\mathcal{H}))$ in Theorem 4.4.1 are

not standard. The lower bound of $\Omega(\mathrm{L}(\mathcal{H}))$ requires constructing a hard stream by traversing down a Littlestone tree skipping certain levels. The lower bound of $\sqrt{T}$ requires constructing a hard stream using two different evolution functions f_1, f_2 by: (a) setting x_0 to be a state they differ on (b) generating their trajectories starting from x_0 and finally (c) picking states from each trajectory in an alternating fashion. These arguments are different from the typical lower bound construction for online classification. In addition, the construction in Theorem 4.4.4 showing the separation between realizable and agnostic learnability under Flow regret is non-trivial. It involves constructing an $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ so that on a large subset of states in $\mathcal{X}$, every function $f \in \mathcal{F}$ effectively reveals itself.

4.1.1 Related Works

There has been a long line of work studying prediction and regret minimization when learning unknown dynamical systems [Hazan et al., 2017, 2018, Ghai et al., 2020, Lee, 2022, Rashidinejad et al., 2020, Kozdoba et al., 2019, Tsiamis and Pappas, 2022, Lale et al., 2020]. However, these works focus on prediction for fully/partially-observed linear dynamical systems under various data-generating processes. Moreover, there is also a line of work on regret minimization for linear dynamical systems for online control problems [Abbasi-Yadkori and Szepesvári, 2011, Cohen et al., 2018, Agarwal et al., 2019]. For non-linear dynamical systems, there has been some applied work studying data-driven approaches to prediction [Wang et al., 2016, Korda and Mezić, 2018, Ghadami and Epureanu, 2022]. Regret minimization for non-linear dynamical systems has mainly been studied in the context of control [Kakade et al., 2020, Muthirayan and Khargonekar, 2021]. Another related line of work is online learning for time series forecasting, where autoregressive models are used to predict the next state [Anava et al., 2013, 2015, Liu et al., 2016, Yang et al., 2018].

Another important line of work is that of system identification and parameter estimation [Åström and Eykhoff, 1971, Ljung, 1999]. Here, the goal is to recover and estimate the parameters of the underlying evolution function from the observed sequence of states. There is a long history of work studying system identification in both the batch [Campi and Weyer, 2002, Vidyasagar and Karandikar, 2006, Foster et al., 2020, Sattar and Oymak, 2022, Bahmani and Romberg, 2020] and streaming settings [Kowshik et al., 2021a,b, Giannakis et al., 2023, Jain et al., 2021]. Several works have also considered the problem of learning the unknown evolution rule of dynamical systems defined over discrete state spaces. For example, Wulff and Hertz [1992] trains a neural network on a sequence of observed states to approximate the unknown evolution rule of a cellular automaton. Grattarola et al. [2021] extends this work to learning the unknown evolution rule of a graph cellular automaton,

a generalization of a regular cellular automaton, using graph neural networks. [Qiu et al., Rosenkrantz et al., 2022] also consider the problem of PAC learning a discrete-time, finite-space dynamical system defined over a (un)directed graph. We also note the related work of Berry and Das [2023, 2025], who study the problem of learning dynamics observed through a continuous embedding and derive learning guarantees based on various structural properties of the underlying system. Finally, our work builds on a rich tradition in learning theory that characterizes learnability through complexity measures and combinatorial dimensions [Vapnik and Chervonenkis, 1971, Littlestone, 1987, Bartlett and Mendelson, 2002, Daniely et al., 2011].

4.2 Preliminaries

4.2.1 Discrete-time Dynamical Systems

A discrete-time dynamical system is a tuple $(\mathbb{N}, \mathcal{X}, f)$, where $\mathbb{N}$ is the set of natural numbers denoting the time steps, $\mathcal{X}$ is a non-empty set called the state space. In this work, we make no assumption on the cardinality of $\mathcal{X}$, so it can be unbounded and perhaps even uncountable. The function $f : \mathcal{X} \to \mathcal{X}$ is a deterministic map that defines the evolution of the dynamical system. That is, the $(t+1)$-th iterate of the dynamics can be expressed in terms of t-th iterate using the relation $x_{t+1} = f(x_t)$. Define f^t to be the t-fold composition of f. That is, $f^2 = f \circ f$, $f^3 = f \circ f \circ f$, and so forth. Given an initial state $x_0 \in \mathcal{X}$, the sequence $\{f^t(x_0)\}_{t\in\mathbb{N}}$ is called the flow of the dynamical system through x_0. Finally, let $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ denote a class of evolution functions on the state space $\mathcal{X}$ and $\mathcal{F}(x) = \{f(x) \mid f \in \mathcal{F}\} \subseteq \mathcal{X}$ to be the projection of $\mathcal{F}$ onto $x \in \mathcal{X}$.

4.2.2 Learning-to-Predict in Dynamical Systems

When learning-to-predict in dynamical systems, nature plays a sequential game with the learner over T rounds. At the beginning of the game, nature reveals the initial state $x_0 \in \mathcal{X}$. In each round $t \in [T]$, the learner $\mathcal{A}$ uses the observed sequence of states $x_{<t} := (x_0, ..., x_{t-1})$ to predict the next state $\mathcal{A}(x_{<t}) \in \mathcal{X}$. Nature then reveals the true state $x_t \in \mathcal{X}$, and the learner suffers the loss $\mathbb{1}\{\mathcal{A}(x_{<t}) \neq x_t\}$. Given a class of evolution functions $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$, the goal of the learner is to make predictions such that its *regret*, defined as a difference between cumulative loss of the learner and the best possible cumulative loss over evolution functions in $\mathcal{F}$, is small.

Formally, given $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$, the expected Markovian regret of an algorithm $\mathcal{A}$ is defined as

$$\mathrm{MR}_{\mathcal{A}}(T, \mathcal{F}) := \sup_{(x_0, x_1, \ldots, x_T)} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}(x_{<t}) \neq x_t\}\right] - \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} \mathbb{1}\{f(x_{t-1}) \neq x_t\}, \tag{4.1}$$

where the expectation is taken with respect to the randomness of the learner $\mathcal{A}$. Given this definition of regret, we define agnostic learnability of an evolution class.

Definition 4.2.1 (Agnostic Learnability under Markovian Regret)**.** *An evolution class $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ is learnable in the agnostic setting if and only if* $\inf_{\mathcal{A}} \mathrm{MR}_{\mathcal{A}}(T, \mathcal{F}) = o(T)$[1].

Perhaps a more natural definition of expected regret in the agnostic setting is to compare the prediction of the learner to the prediction of the best-fixed *trajectory* generated by functions in our evolution class. To that end, define

$$\mathrm{FR}_{\mathcal{A}}(T, \mathcal{F}) := \sup_{(x_0, x_1, \ldots, x_T)} \Big(\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}(x_{<t}) \neq x_t\}\right] - \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} \mathbb{1}\{f^t(x_0) \neq x_t\}\Big). \tag{4.2}$$

as the expected Flow regret. An analogous definition of agnostic learnability follows.

Definition 4.2.2 (Agnostic Learnability under Flow Regret)**.** *An evolution class $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ is learnable in the agnostic setting under Flow regret if and only if* $\inf_{\mathcal{A}} \mathrm{FR}_{\mathcal{A}}(T, \mathcal{F}) = o(T)$.

A sequence of states $x_0, x_1, \ldots, x_T$ is said to be *realizable* by $\mathcal{F}$ if there exists an evolution function $f \in \mathcal{F}$ such that $f(x_{t-1}) = x_t$ for all $t \in [T]$. In the realizable setting, the cumulative loss of the best-fixed function is 0, and the goal of the learner is to minimize its expected cumulative mistakes

$$\mathrm{M}_{\mathcal{A}}(T, \mathcal{F}) := \sup_{x_0} \sup_{f \in \mathcal{F}} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}(x_{<t}) \neq f(x_{t-1})\}\right].$$

Analogously, we define the realizable learnability of $\mathcal{F}$.

Definition 4.2.3 (Realizable Learnability)**.** *An evolution class $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ is learnable in the realizable setting if and only if* $\inf_{\mathcal{A}} \mathrm{M}_{\mathcal{A}}(T, \mathcal{F}) = o(T)$.

4.2.3 Complexity Measures

In sequential learning tasks, complexity measures are often defined in terms of *trees*, a basic unit that captures temporal dependence. In this chapter, we use complete binary trees to

[1] $o(T)$ refers to any sublinear function of T.

define a new combinatorial object called a trajectory tree. In the remainder of this section and Section 4.2.4, we use trajectory trees to define complexity measures and combinatorial dimensions for evolution classes.

Definition 4.2.4 (Trajectory tree)**.** *A trajectory tree of depth d is a complete binary tree of depth d where internal nodes are labeled by states in $\mathcal{X}$.*

Given a trajectory tree $\mathcal{T}$ of depth d, a root-to-leaf path down $\mathcal{T}$ is defined by a string $\sigma \in \{-1,1\}^d$ indicating whether to go left ($\sigma_t = -1$) or to go right ($\sigma_i = +1$) at each depth $t \in [d]$. A path $\sigma \in \{-1,1\}^d$ down $\mathcal{T}$ gives a trajectory $\{x_t\}_{t=0}^d$, where x_0 denotes the instance labeling the root node and x_t is the instance labeling the edge following the prefix $(\sigma_1, ..., \sigma_t)$ down the tree. A path $\sigma \in \{-1,1\}^d$ down $\mathcal{T}$ is shattered by $\mathcal{F}$ if there exists a $f \in \mathcal{F}$ such that $f(x_{t-1}) = x_t$ for all $t \in [d]$, where $\{x_t\}_{t=0}^d$ is the corresponding trajectory obtained by traversing $\mathcal{T}$ according to σ. If every path down $\mathcal{T}$ is shattered by $\mathcal{F}$, we say that $\mathcal{T}$ is shattered by $\mathcal{F}$.

To make this more rigorous, we define a trajectory tree $\mathcal{T}$ of depth d as a sequence $(\mathcal{T}_0, \mathcal{T}_1, ..., \mathcal{T}_d)$ of node-labeling functions $\mathcal{T}_t : \{-1,1\}^t \to \mathcal{X}$, which provide the labels for each internal node. Then, $\mathcal{T}_t(\sigma_1, ..., \sigma_t)$ gives the label of the node by following the prefix $(\sigma_1, ..., \sigma_t)$ and $\mathcal{T}_0$ denotes the instance labeling the root node. For brevity, we define $\sigma_{\leq t} = (\sigma_1, ..., \sigma_t)$ and write $\mathcal{T}_t(\sigma_1, ..., \sigma_t) = \mathcal{T}_t(\sigma_{\leq t})$. Analogously, we let $\sigma_{<t} = (\sigma_1, ..., \sigma_{t-1})$. Using this notation, a trajectory tree $\mathcal{T}$ of depth d is shattered by the evolution function class $\mathcal{F}$ if $\forall \sigma \in \{-1,1\}^d$, there exists a $f_\sigma \in \mathcal{F}$ such that $f_\sigma(\mathcal{T}_{t-1}(\sigma_{<t})) = \mathcal{T}_t(\sigma_{\leq t})$ for all $t \in [d]$. Moreover, we use this notation to define the *Branching factor* of a trajectory tree.

Definition 4.2.5 (Branching factor)**.** *The Branching factor of a trajectory tree $\mathcal{T}$ of depth d is*

$$\mathrm{B}(\mathcal{T}) := \min_{\sigma \in \{-1,1\}^d} \sum_{t=1}^{d} \mathbb{1}\{\mathcal{T}_t((\sigma_{<t}, -1)) \neq \mathcal{T}_t\big((\sigma_{<t}, +1)\big)\}.$$

The branching factor of a path $\sigma \in \{-1,1\}^d$ captures the distinctness of states labeling the two children of internal nodes in this path. In particular, it counts the number of nodes in the path whose two children are labeled by distinct states. The branching factor of a trajectory tree is just the smallest branching factor across all paths. Using the notion of shattering and Definition 4.2.5, we define a new complexity measure, termed the Evolution complexity, of a function class $\mathcal{F}$.

Definition 4.2.6 (Evolution complexity)**.** *Let $\mathcal{S}(\mathcal{F}, d)$ be the set of all trajectory trees of depth $d \in \mathbb{N}$ shattered by $\mathcal{F}$. Then, the Evolution complexity of $\mathcal{F}$ at depth d is defined as* $\mathrm{C}_d(\mathcal{F}) := \sup_{\mathcal{T} \in \mathcal{S}(\mathcal{F}, d)} \mathrm{B}(\mathcal{T})$.

In Section 4.3, we show that the Evolution complexity exactly (up to a factor of 2) captures the minimax expected mistakes in the realizable setting. We provide some examples of classes $\mathcal{F}$ and their evolution complexities in Theorem 4.3.5. We note that there is an existing notion of complexity for dynamical systems, termed topological entropy, that quantifies the complexity of a particular evolution function $f \in \mathcal{F}$ [Adler et al., 1965]. However, topological entropy does not characterize learnability as $\mathcal{F} = \{f\}$ is trivially learnable when f has infinite topological entropy.

4.2.4 Combinatorial dimensions

In addition to complexity measures, combinatorial dimensions play an important role in providing crisp quantitative characterizations of learnability. For example, the Daniely Shalev-Shwartz dimension (DSdim), originally proposed by Daniely and Shalev-Shwartz [2014] and formally defined below, was recently shown by Brukhim et al. [2022] to provide a tight quantitative characterization of multiclass PAC learnability. In Section 4.3.2, we use the DSdim to relate the realizable learnability of dynamical systems to multiclass PAC learnability of $\mathcal{F}$.

Definition 4.2.7 (DS dimension [Daniely and Shalev-Shwartz, 2014])**.** *We say that $A \subseteq \mathcal{X}$ is DS-shattered by $\mathcal{F}$ if there exists a finite $\mathcal{H} \subset \mathcal{F}$ such that for every $x \in A$ and $h \in \mathcal{H}$, there exists a $g \in \mathcal{H}$ such that $g(x) \neq h(x)$ and $g(z) = h(z)$ for all $z \in A \setminus \{x\}$. The DS dimension of $\mathcal{F}$, denoted $\mathrm{DS}(\mathcal{F})$, is the largest $d \in \mathbb{N}$ such that there exists a shattered set $A \subset \mathcal{X}$ with cardinality d. If there are arbitrarily large sets $A \subseteq \mathcal{X}$ that are shattered by $\mathcal{F}$, then we say that $\mathrm{DS}(\mathcal{F}) = \infty$.*

Analogously, for online multiclass classification, the Littlestone dimension (Ldim), originally proposed by Littlestone [1987] for binary classification and later extended to multiclass classification by Daniely et al. [2011], provides a tight quantitative characterization of learnability [Hanneke et al., 2023].

Definition 4.2.8 (Littlestone dimension [Littlestone, 1987, Daniely et al., 2011])**.** *Let $\mathcal{T}$ be a complete binary tree of depth d whose internal nodes are labeled by a sequence $(\mathcal{T}_0, \ldots, \mathcal{T}_{d-1})$ of node-labeling functions $\mathcal{T}_{t-1} : \{-1, 1\}^{t-1} \to \mathcal{X}$. The tree $\mathcal{T}$ is shattered by $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ if there exists a sequence $(Y_1, ..., Y_d)$ of edge-labeling functions $Y_t : \{-1, 1\}^t \to \mathcal{X}$ such that for every path $\sigma = (\sigma_1, ..., \sigma_d) \in \{-1, 1\}^d$, there exists a function $f_\sigma \in \mathcal{F}$ such that for all $t \in [d]$, $f_\sigma(\mathcal{T}_{t-1}(\sigma_{<t})) = Y_t(\sigma_{\leq t})$ and $Y_t((\sigma_{<t}, -1)) \neq Y_t((\sigma_{<t}, +1))$. The Littlestone dimension of $\mathcal{F}$, denoted $\mathrm{L}(\mathcal{F})$, is the maximal depth of a tree $\mathcal{T}$ that is shattered by $\mathcal{F}$. If there exists shattered trees of arbitrarily large depth, we say $\mathrm{L}(\mathcal{F}) = \infty$.*

4.3 Warmup: Realizable Learnability

In this section, we provide qualitative and quantitative characterizations of realizable learnability in terms of the Evolution complexity. Our main result in this section is Theorem 4.3.1, which provides bounds on the minimax expected number of mistakes.

Theorem 4.3.1 (Minimax Expected Mistakes)**.** *For any $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$, we have $\frac{1}{2}\,\mathrm{C}_T(\mathcal{F}) \leq \inf_{\mathcal{A}} \mathrm{M}_{\mathcal{A}}(T, \mathcal{F}) \leq \mathrm{C}_T(\mathcal{F})$. Moreover, the upper bound is achieved constructively by a deterministic learner.*

The factor of $\frac{1}{2}$ in the lower bound is due to randomized learners, and the lower bound of $\mathrm{C}_T(\mathcal{F})$ can be obtained if the learner is restricted only to deterministic learning rules.

We now describe our high-level proof strategy and defer the full proof of Theorem 4.3.1 to Appendix B.1. Our lower bound involves picking the worst-case shattered tree with the largest branching factor and traversing down this tree uniformly at random to generate the sequence of states. For such a sequence, the learner can do no better than random guessing at nodes where the branching occurred, yielding the lower bound $\mathrm{C}_T(\mathcal{F})/2$. Next, for our upper bound, we first define a localized complexity measure $\mathrm{C}_T(\mathcal{F}, x_0)$, where we only consider shattered trees rooted at the revealed initial state x_0. Our minimax learner is then a version space algorithm that predicts the state that will result in the largest reduction in the complexity measure if a mistake occurs. This learner is a generalization of the celebrated Standard Optimal Algorithm due to Littlestone [1987].

4.3.1 Minimax Rates in the Realizable Setting

While Theorem 4.3.1 provides a quantitative and qualitative characterization of realizable learnability, it does not shed light on how $\inf_{\mathcal{A}} \mathrm{M}_{\mathcal{A}}(\mathcal{F}, T)$ may depend on the time horizon T. In online classification with the 0-1 loss, the seminal work by Littlestone [1987] and Daniely et al. [2011] show that only two rates are possible: $\Theta(T)$ and $\Theta(1)$. That is, if a hypothesis class is online learnable in the realizable setting, then it is learnable with a constant mistake bound (i.e. the Littlestone dimension). Perhaps surprisingly, this is not the case for learning dynamical systems in a strong sense: *every* rate is possible.

Theorem 4.3.2. *For every $S \subset \mathbb{N} \cup \{0\}$, there exists $\mathcal{F}_S \subseteq \mathbb{Z}^{\mathbb{Z}}$ such that $\mathrm{C}_T(\mathcal{F}_S) = \sup_{n \in \mathbb{N} \cup \{0\}} |S \cap \{n, n+1, ..., n+T-1\}|$.*

Theorem 4.3.2, proved in Appendix B.1.2, along with Theorem 4.3.1 implies that any minimax rate in the realizable setting is possible. As an example, suppose we would like to achieve the rate $\Theta(T^\alpha)$ for some $\alpha < 1$. Then, picking $S = \{\lfloor t^{\frac{1}{\alpha}} \rfloor : t \in \mathbb{N} \cup \{0\}\}$ suffices since

$\sup_{x\in\mathbb{N}\cup\{0\}} |S \cap \{n, n+1, ..., n+T-1\}| = |\{\lfloor t^{\frac{1}{\alpha}} \rfloor : t \in \mathbb{N}\cup\{0\}\} \cap \{0, 1, ..., T-1\}| = \Theta(T^{\alpha})$. Likewise, one can get logarithmic rates by picking $S = \{2^t : t \in \mathbb{N}\cup\{0\}\}$ and constant rates by picking $S \subset \mathbb{N}\cup\{0\}$ such that $|S| < \infty$. In light of Theorem 4.3.2 and the fact that only constant mistake bounds are possible for online multiclass classification, it is natural to ask *when* one can achieve constant mistake bounds for learning dynamical systems. To answer this question, we introduce a new combinatorial dimension termed the Branching dimension.

Definition 4.3.1 (Branching dimension)**.** *The Branching dimension, denoted* $\mathrm{Bd}(\mathcal{F})$*, is the smallest natural number* $d \in \mathbb{N}$ *such that for every shattered trajectory tree* $\mathcal{T}$*, we have* $\mathrm{B}(\mathcal{T}) \leq d$*. If for every* $d \in \mathbb{N}$*, there exists a shattered trajectory tree* $\mathcal{T}$ *with* $\mathrm{B}(\mathcal{T}) > d$*, we say* $\mathrm{Bd}(\mathcal{F}) = \infty$*.*

Theorem 4.3.3, proved via non-constructive arguments in Appendix B.1.3, shows that $\inf_{\mathcal{A}} \mathrm{M}_{\mathcal{A}}(T, \mathcal{F}) = \Theta(1)$ if and and only if $\mathrm{Bd}(\mathcal{F}) < \infty$.

Theorem 4.3.3 (Constant Minimax Expected Mistakes)**.** *For any* $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$*, we have (i)* $\inf_{\mathcal{A}} \mathrm{M}_{\mathcal{A}}(T, \mathcal{F}) \leq \mathrm{Bd}(\mathcal{F})$ *and (ii) if* $\mathrm{Bd}(\mathcal{F}) = \infty$*, then* $\inf_{\mathcal{A}} \mathrm{M}_{\mathcal{A}}(T, \mathcal{F}) = \omega(1)$[2].

4.3.2 Relations to PAC and Online Multiclass Classification

By studying abstract state spaces $\mathcal{X}$ and evolutions function classes $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$, we can also compare realizable learnability of dynamical systems to existing notions of realizable learnability in the well-known PAC and online classification settings. Our main result relates the evolution complexity to the DS and Littlestone dimensions, which characterize PAC and online classification respecitively.

Theorem 4.3.4 (Relations to the DS and Littlestone dimension)**.** *The following statements are true.*

(i) There exists $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ *such that* $\mathrm{DS}(\mathcal{F}) = \infty$ *but* $\mathrm{C}_T(\mathcal{F}) = \Theta(\log(T))$*.*

(ii) There exists $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ *such that* $\mathrm{DS}(\mathcal{F}) = 1$ *but* $\mathrm{C}_T(\mathcal{F}) = T$*.*

(iii) For any $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$*, we have that* $\mathrm{C}_T(\mathcal{F}) \leq \mathrm{L}(\mathcal{F})$*.*

(iv) There exists $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ *such that* $\mathrm{L}(\mathcal{F}) = \infty$ *but* $\mathrm{C}_T(\mathcal{F}) = 1$*.*

The proof of Theorem 4.3.4 is in Appendix B.1.4. Parts (i) and (ii) show that the finiteness of the DSdim is neither necessary nor sufficient for learning dynamical systems in the

[2]Recall that $f(n) = \omega(1)$ if for all $k > 0$, there exists n_k, such that for all $n > n_k$ we have $f(n) > k$.

realizable setting. On the other hand, parts (iii) and (iv) show that finite Ldim is sufficient but not necessary for learning dynamical systems in the realizable setting. Overall, learning dynamical systems is always easier than online multiclass classification, but can be both easier and harder than multiclass PAC classification. The proof of (i), (ii), and (iv) are combinatorial in nature, while the proof of (iii) involves reducing learning-to-predict in dynamical systems to online multiclass classification.

4.3.3 Examples

In this section, we establish the minimax rates for discrete linear systems and linear Boolean networks. The proof of Theorem 4.3.5 is in Appendix B.2.

Theorem 4.3.5 (Linear Systems). *(i) Let $\mathcal{X} = \mathbb{Z}^n$ and $r < n$. For $\mathcal{F} = \{x \mapsto Wx : W \in \mathbb{Z}^{n\times n}, \underline{rank}(W) \le r\}$ and $T > r$, we have $\mathrm{C}_T(\mathcal{F}) = r + 1$.*

(ii) Let $\mathcal{X} = \{0,1\}^n$ and $T \ge n$. For $\mathcal{F} = \{x \mapsto Wx \,(\underline{mod}\, 2) : W \in \mathbb{Z}^{n\times n}\}$, we have $\mathrm{C}_T(\mathcal{F}) = n$.

(iii) Let $\mathcal{X} = \{0,1\}^n$ and $T \ge n$. For $\mathcal{F} = \{x \mapsto \mathbb{1}\{Wx > 0\} : W \in \{0,1\}^{n\times n}\}$, we have $n \le \mathrm{C}_T(\mathcal{F}) \le n^2$.

Thresholded Boolean networks have been used to model genetic regulatory dynamics [Mendoza and Alvarez-Buylla, 1998] and social networks [Kempe et al., 2003]. Modulo Boolean networks have been studied by Chandrasekhar et al. [2023] in the context of stability.

4.4 Agnostic Learnability

4.4.1 Markovian Regret

In this section, we go beyond the realizable setting and consider the case where nature may reveal a trajectory that is not consistent with any evolution function in the class. Our main result in this section establishes bounds on the minimax expected Markovian regret.

Theorem 4.4.1 (Minimax Expected Markovian Regret). *For any $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$,*

$$\max\left\{\frac{\mathrm{L}(\mathcal{F})}{18}, \frac{\sqrt{T}}{16\sqrt{3}}\right\} \le \inf_{\mathcal{A}} \mathrm{MR}_{\mathcal{A}}(T, \mathcal{F}) \le \mathrm{L}(\mathcal{F}) + \sqrt{T\,\mathrm{L}(\mathcal{F})\log T}.$$

Theorem 4.4.1 shows that the finiteness of the Littlestone dimension of $\mathcal{F}$ is both necessary and sufficient for agnostic learnability. This is in contrast to Theorem 4.3.4, which shows that

the finiteness of the Littlestone dimension of $\mathcal{F}$ is sufficient but not necessary for realizable learnability. Thus, Theorem 4.4.1 and 4.3.4 imply that realizable and agnostic learnability are not equivalent.

Corollary 4.4.2 (Realizable Learnability $\not\equiv$ Agnostic Learnability under Markovian Regret)**.** *There exists a class $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ such that $\mathcal{F}$ is learnable in the realizable setting but not in the agnostic setting under Markovian regret.*

One such class exhibiting the separation is the thresholds $\mathcal{F} = \{x \mapsto \mathbb{1}\{x \geq a\} : a \in (0,1)\}$ used in the proof of part (iv) in Theorem 4.3.4. Beyond the qualitative separation of realizable and agnostic learnability under Markovian regret, we also observe a quantitative separation in terms of possible minimax rates. Recall that Theorem 4.3.2 shows that every rate is possible in the realizable setting. However, Theorem 4.4.1 shows that only two types of rates are possible in the agnostic setting: $\Theta(T)$ whenever $\mathrm{L}(\mathcal{F}) = \infty$ and $\tilde{\Theta}(\sqrt{T})$ whenever $\mathrm{L}(\mathcal{F}) < \infty$. More precisely, when $\sqrt{T} \geq \mathrm{L}(\mathcal{F})$, we have a lower bound of $\Omega(\sqrt{T})$ and an upper bound of $O(\sqrt{T\,\mathrm{L}(\mathcal{F})\log T})$. This raises the natural question of what the right minimax rate is. In Appendix B.3.2, we provide an evolution class and establish a lower bound of $\Omega(\sqrt{T\,\mathrm{L}(\mathcal{F})})$, showing that the upper bound is tight up to $\sqrt{\log T}$.

Our proof of the upper bound in Theorem 4.4.1 reduces learning dynamical systems to online multiclass classification and uses a result due to Hanneke et al. [2023]. To prove the lower bound $\frac{\mathrm{L}(\mathcal{F})}{18}$, we construct a hard stream by carefully sampling a random path down a Littlestone tree of depth $\mathrm{L}(\mathcal{F})$. To prove the lower bound of $\frac{\sqrt{T}}{16\sqrt{3}}$, we construct a hard randomized stream using just two different evolution functions in $\mathcal{F}$. The full proof is in Appendix B.3.1.

4.4.2 Flow Regret

The necessity of the Littlestone dimension for agnostic learnability under Markovian regret is quite restrictive. For example, a simple (but unnatural) class like one-dimensional thresholds $\mathcal{F} = \{x \mapsto \mathbb{1}\{x \leq a\} : a \in (0,1)\}$ has $\mathrm{L}(\mathcal{F}) = \infty$ but $\mathrm{C}_T(\mathcal{F}) = 1$. The key idea in the lower bound of Theorem 4.4.3 is that the adversary can simulate the online multiclass classification game, where finiteness of the Littlestone dimension is necessary, by "giving up" every other round. This is possible because of the definition of Markovian regret. In particular, the evaluation of the "best-fixed evolution function in hindsight" under Markovian regret is only penalized on one-step prediction error but not on long-term consistency of the generated dynamics starting from the initial state x_0.

This motivates the following natural question. Which evolution classes $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ are agnostic learnable under Flow regret? Is the finiteness of Ldim still necessary? Theorem

4.4.3, whose proof can be found in Appendix B.3.3, provides partial answers to these question by bounding the minimax expected Flow regret for classes where $\sup_{x\in\mathcal{X}} |\mathcal{F}(x)|$ is uniformly bounded.

Theorem 4.4.3 (Minimax Expected Flow Regret)**.** *For any ordered set $\mathcal{X}$ and $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$,*

$$\frac{\mathrm{C}_T(\mathcal{F})}{2} \leq \inf_{\mathcal{A}} \mathrm{FR}_{\mathcal{A}}(T, \mathcal{F}) \leq \mathrm{C}_T(\mathcal{F}) + \sqrt{\mathrm{C}_T(\mathcal{F})\, T \ln\Big(\frac{T\, K_{\mathcal{F}}}{\mathrm{C}_T(\mathcal{F})}\Big)}.$$

where $K_{\mathcal{F}} = \sup_{x\in\mathcal{X}} |\mathcal{F}(x)|$. Moreover, both the lower- and upper bound can be tight.

Note that the upper bound becomes vacuous when $\sup_{x\in\mathcal{X}} |\mathcal{F}(x)| = \infty$, and finding a general characterization of Flow regret learnability remains an open question. Unlike agnostic learnability under Markovian regret, the finiteness of the Ldim is not necessary for agnostic learnability under Flow regret. Indeed, while the class of one-dimensional thresholds is not agnostic learnable under Markovian regret, it is agnostic learnable under Flow regret. Although Markovian learnability is not necessary for Flow regret learnability, when $K_{\mathcal{F}} < \infty$, learnability under Markovian regret is sufficient learnability under flow regret. To see this, recall that Theorem 4.4.1 states that learnability under Markovian regret implies $\mathrm{L}(\mathcal{F}) < \infty$. Then, since part (iii) of Theorem 4.3.4 states $\mathrm{C}_T(\mathcal{F}) \leq \mathrm{L}(\mathcal{F})$, we can use Theorem 4.4.3 to infer that $\mathcal{F}$ is also learnable under flow regret with regret $\leq \mathrm{L}(\mathcal{F}) + \sqrt{\mathrm{L}(\mathcal{F}) T \ln(T K_{\mathcal{F}})}$.

Theorem 4.4.3 shows that realizable and agnostic learnability under Flow regret are equivalent as long as $\sup_{x\in\mathcal{X}} |\mathcal{F}(x)| < \infty$. But this equivalence breaks down when the projection sizes are unbounded, as shown by Theorem 4.4.4.

Theorem 4.4.4 (Realizable learnability $\not\equiv$ Agnostic Learnability under Flow Regret)**.** *There exists an ordered set $\mathcal{X}$ and $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ such that (i) $\inf_{\mathcal{A}} \mathrm{M}_{\mathcal{A}}(T, \mathcal{F}) \leq 3$ but (ii) $\inf_{\mathcal{A}} \mathrm{FR}_{\mathcal{A}}(T, \mathcal{F}) \geq \frac{T}{6}$.*

To prove Theorem 4.4.4 (see Appendix B.3.5), we construct a class $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ such that on large subset of states in $\mathcal{X}' \subset \mathcal{X}$, every function $f \in \mathcal{F}$ effectively reveals its identity.

4.5 Discussion and Future directions

In this work, we studied the problem of learning-to-predict in discrete-time dynamical systems under the 0-1 loss. A natural extension is to consider continuous state spaces with real-valued losses. For example, one can take $\mathcal{X}$ to be a bounded subset of a Hilbert space and consider the squared norm as the loss function. Another natural extension is to consider

learnability under partial observability, where the learner only observes some transformation $\phi(x_t)$ instead of the true state x_t. Such feedback model is standard in prediction for linear dynamical systems [Hazan et al., 2018]. It is also natural to study the learnability of function classes where the output of the evolution rules $f : \mathcal{X}^p \to \mathcal{X}$, depend on the previous $p > 1$ states (e.g. the p-th order VAR model). Lastly, the learning algorithms in this work are *improper*: they use evolution functions that may not lie in $\mathcal{F}$ to make predictions. This might be undesirable as improper learning algorithms may be incompatible with downstream system identification and control tasks. To this end, characterizing proper learnability of dynamical systems is an important future direction.

CHAPTER 5

Multiclass Online Learning Under Bandit Feedback

In this chapter, we study online multiclass classification under bandit feedback. We extend the results of Daniely and Helbertal [2013] by showing that the finiteness of the Bandit Littlestone dimension is necessary and sufficient for bandit online learnability even when the label space is unbounded. Moreover, we show that, unlike the full-information setting, sequential uniform convergence is necessary but not sufficient for bandit online learnability. Our result complements the work by Hanneke, Moran, Raman, Subedi, and Tewari [2023] who show that the Littlestone dimension characterizes online multiclass learnability in the full-information setting even when the label space is unbounded.

5.1 Introduction

In the standard online multiclass classification model, a learner plays a repeated game against an adversary. In each round $t \in [T]$, an adversary picks a labeled instance $(x_t, y_t) \in \mathcal{X} \times \mathcal{Y}$ and reveals x_t to the learner. Using access to a hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, the learner makes a possibly random prediction $\hat{y}_t \in \mathcal{Y}$. The adversary then reveals the true label y_t and the learner then suffers the loss $\mathbb{1}\{y_t \neq \hat{y}_t\}$. Overall, the goal of the learner is to output predictions such that its expected cumulative loss is not too much larger than the smallest cumulative loss amongst all fixed hypothesis in $\mathcal{H}$. This standard setting of online multiclass classification is commonly referred to as the *full-information* setting because the learner gets to observe the true label y_t at the end of each round. Perhaps a more practical setting is the *bandit* feedback setting, where the learner does not get to observe the true label at the end of each round, but only the indication $\mathbb{1}\{\hat{y}_t \neq y_t\}$ of whether its prediction was correct or not [Kakade, Shalev-Shwartz, and Tewari, 2008]. One application of this setting is online advertising where the advertiser recommends an ad (label) to a user (instance), but only gets to observe whether the user clicked on the ad or not.

Unlike the full-information setting, where online learnability of a hypothesis class $\mathcal{H}$ has been fully characterized in both the realizable and agnostic settings, less is known about online learnability under bandit feedback. Indeed, the first work on characterizing bandit online learnability is due to Daniely, Sabato, Ben-David, and Shalev-Shwartz [2011]. They introduce a dimension named the Bandit Littlestone dimension (BLdim), and show that it exactly characterizes the bandit online learnability of deterministic learners in the realizable setting. Even prior to that, Auer and Long [1999] related the Bandit Littlestone dimension (which is the optimal deterministic mistake bound with bandit feedback) to the multiclass extension of the Littlestone dimension (Ldim) [Littlestone, 1987, Daniely et al., 2011] and showed that $\mathrm{BL}(\mathcal{H}) = O(|\mathcal{Y}| \log(|\mathcal{Y}|)\mathrm{L}(\mathcal{H}))$, where $\mathrm{BL}(\mathcal{H})$ is the BLdim of $\mathcal{H}$, $\mathrm{L}(\mathcal{H})$ is the Ldim of $\mathcal{H}$, and $|\mathcal{Y}|$ denotes the size of the label space $\mathcal{Y}$. Following the work of Daniely et al. [2011], Daniely and Helbertal [2013] studies the price of bandit feedback by quantifying the ratio between optimal error rates of the two feedback models in the realizable and agnostic settings. Using the inequality $\mathrm{L}(\mathcal{H}) \leq \mathrm{BL}(\mathcal{H}) = O(|\mathcal{Y}| \log(|\mathcal{Y}|)\mathrm{L}(\mathcal{H}))$, they infer that BLdim characterizes realizable online learnability under bandit feedback when $|\mathcal{Y}|$ is finite. Later, Long [2017] and Geneson [2021] proved that this upperbound on BLdim is the best possible up to a leading constant. They also found the exact optimal leading constant.

Moving beyond the realizable setting, Daniely and Helbertal [2013] give an agnostic online learner whose expected regret, under bandit feedback, is at most $O\left(\sqrt{\mathrm{L}(\mathcal{H})|\mathcal{Y}|T\log(T|\mathcal{Y}|)}\right)$. As a corollary, when $|\mathcal{Y}|$ is finite, they infer that the BLdim qualitatively characterizes

agnostic bandit online learnability. In addition, Daniely and Helbertal [2013] note a gap of $\tilde{O}(\sqrt{|\mathcal{Y}|})$ between their upperbound in the bandit setting and the known lowerbound of $\Omega(\sqrt{\mathrm{L}(\mathcal{H})\,T})$ in the full-information setting [Ben-David, Pál, and Shalev-Shwartz, 2009]. Accordingly, they ask whether a tighter quantitative characterization of bandit learnability is possible in the agnostic setting. In fact, it is unclear whether BLdim characterizes bandit online learnability when $|\mathcal{Y}|$ is unbounded.

Along this direction, there has been a recent surge of interest in characterizing learnability when the label space is unbounded. For example, in a recent breakthrough result, Brukhim, Carmon, Dinur, Moran, and Yehudayoff [2022] show that the Daniely-Schwartz (DS) dimension, defined by Daniely and Shalev-Shwartz [2014], characterizes multiclass learnability in the PAC setting even when the label space in unbounded. Following this work, Hanneke, Moran, Raman, Subedi, and Tewari [2023] show that the multiclass extension of the Littlestone dimension, originally proposed by Daniely, Sabato, Ben-David, and Shalev-Shwartz [2011], continues to characterize online multiclass learnability under *full-information feedback* when the label space is unbounded. Motivated by these results, we ask whether the BLdim continues to characterize *bandit online learnability* even when the label space is unbounded. In particular, can the optimal expected regret in the realizable and agnostic settings, under bandit feedback, be expressed as a function of the BLdim without a dependence on $|\mathcal{Y}|$?

In this chapter, we resolve this question by showing that the finiteness of BLdim is necessary and sufficient for bandit online learnability, in both the realizable and agnostic settings, even when the label space is unbounded.

Theorem 5.1.1. *Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ and $C_{\mathcal{H}} := \sup_{x \in \mathcal{X}} |\{h(x) : h \in \mathcal{H}\}|$. The following statements are equivalent:*

1. *$\mathcal{H}$ is bandit online learnable.*
2. *$\mathrm{BL}(\mathcal{H}) < \infty$.*
3. *$C_{\mathcal{H}} < \infty$ and $\mathrm{L}(\mathcal{H}) < \infty$.*

We prove (2) $\implies$ (1), (3) $\implies$ (2) in Section 5.3, and (1) $\implies$ (3) in Section 5.4. The proof of (2) $\implies$ (1) is given by an agnostic online learner whose expected regret under bandit feedback can be expressed as a function of BLdim without any dependence on $|\mathcal{Y}|$.

Theorem 5.1.2. *For any $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, there exists an agnostic online learner whose expected regret, under bandit feedback, is at most*

$$8\sqrt{\mathrm{L}(\mathcal{H})\,\mathrm{BL}(\mathcal{H})T\log(T)}.$$

Theorem 5.1.2 provides an improvement over the upperbound given by Daniely and Helbertal [2013] when $|\mathcal{Y}| \gg \mathrm{BL}(\mathcal{H})$. In fact, the gap between $|\mathcal{Y}|$ and $\mathrm{BL}(\mathcal{H})$ can be arbitrary. Consider the case where $\mathcal{Y} = \mathbb{N}$ but $|\mathcal{H}| < \infty$. Then, its not hard to see that $\mathrm{BL}(\mathcal{H}) \leq |\mathcal{H}|$ but $|\mathcal{Y}| = \infty$.

In addition to characterizing learnability, there has been recent interest in showing a separation between uniform convergence and learnability. For example, Montasser, Hanneke, and Srebro [2019] show that while uniform convergence is sufficient for adverarsially robust PAC learnability, it is not necessary. Likewise, for online mutliclass learning under full-information feedback, Hanneke et al. [2023] give a class that is learnable, but the online analog of uniform convergence [Rakhlin, Sridharan, and Tewari, 2015b], termed Sequential Uniform Convergence (SUC), does not hold. Towards this end, we ask whether there is a separation between SUC and bandit online learnability. We answer this question affirmatively: while SUC is *necessary* for bandit learnability, it is not sufficient.

Theorem 5.1.3. *If a hypothesis class is online learnable under bandit feedback, then it enjoys the* SUC *property. However, there exists a class which satisfies the* SUC *property, but is not online learnable under bandit feedback.*

Theorem 5.1.3 is in contrast to the full information setting where SUC is sufficient, but not necessary for online learnability [Hanneke et al., 2023]. We note that Theorem 5.1.3 along with Example 1 from Hanneke et al. [2023] also shows a separation in online learnability between the full-information and bandit feedback settings. Figure 5.1 visualizes the landscape of learnability for online multiclass problems.

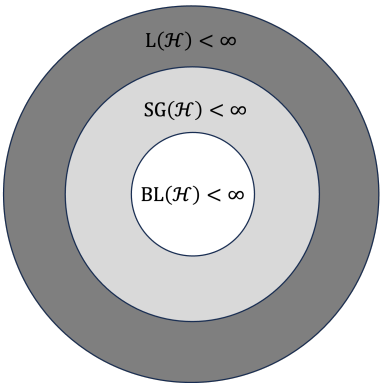

Figure 5.1: Landscape of multiclass online learnability. The Sequential Graph (SG) dimension (see Definition 5.2.4) characterizes SUC.

5.2 Preliminaries

Let $\mathcal{X}$ denote the instance space, $\mathcal{Y}$ be the label space, and $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ denote a hypothesis class. In this chapter, we place no assumptions on the size of the label space $\mathcal{Y}$. Given an instance $x \in \mathcal{X}$, we let $\mathcal{H}(x) := \{h(x) : h \in \mathcal{H}\}$ denote the projection of $\mathcal{H}$ onto x. As usual, $[N]$ is used to denote $\{1, 2, \ldots, N\}$.

5.2.1 Online Learning

In online multiclass classification with bandit feedback, an adversary plays a sequential game with the learner over T rounds. In each round $t \in [T]$, an adversary selects a labeled instance $(x_t, y_t) \in \mathcal{X} \times \mathcal{Y}$ and reveals x_t to the learner. The learner makes a (potentially randomized) prediction $\hat{y}_t \in \mathcal{Y}$. Finally, the adversary reveals to the learner its loss $\mathbb{1}\{\hat{y}_t \neq y_t\}$, but not the true label y_t. Given a hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, the goal of the learner is to output predictions $\hat{y}_t$ under *bandit feedback* such that its *expected regret*

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\hat{y}_t \neq y_t\} - \inf_{h \in \mathcal{H}} \sum_{t=1}^{T} \mathbb{1}\{h(x_t) \neq y_t\}\right]$$

is small. A hypothesis class $\mathcal{H}$ is said to be bandit online learnable if there exists an algorithm such that for any sequence of labeled examples $(x_1, y_1), ..., (x_T, y_T)$, its expected regret, under bandit feedback, is a sublinear function of T. In this chapter, we consider the oblivious setting where the adversary selects the entire sequence of labeled instances $(x_1, y_1), ..., (x_T, y_T)$ before the game begins. Thus, we treat the stream of labeled instances as a non-random, deterministic quantity.

Definition 5.2.1 (Bandit Online Learnability)**.** *A hypothesis class $\mathcal{H}$ is bandit online learnable, if there exists an (potentially randomized) algorithm $\mathcal{A}$ such that its expected regret,*

$$\mathrm{R}_{\mathcal{A}}(T, \mathcal{H}) := \sup_{(x_1,y_1),\ldots,(x_T,y_T)} \left(\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}(x_t) \neq y_t\}\right] - \inf_{h \in \mathcal{H}} \sum_{t=1}^{T} \mathbb{1}\{h(x_t) \neq y_t\}\right),$$

while only receiving bandit feedback, is a non-decreasing sub-linear function of T.

If it is guaranteed that the learner always observes a sequence of examples labeled by some hypothesis $h \in \mathcal{H}$, then we say that we are in the *realizable* setting.

Littlestone [1987] and Ben-David, Pál, and Shalev-Shwartz [2009] showed that a combinatorial parameter called the Littlestone dimension characterizes online learnability of

binary hypothesis classes under full-information feedback, in both the realizable and agnostic settings, respectively. Later, Daniely et al. [2011] defined a multiclass extension of the Littlestone dimension and showed that it tightly characterizes online learnability of multiclass hypothesis classes under full-information feedback in both the realizable and agnostic settings. The Littlestone dimension, in both the binary and multiclass case, is defined in terms of trees, a combinatorial object that captures the temporal dependence inherent in online learning.

Given an instance space $\mathcal{X}$ and a set of objects $\mathcal{M}$, an $\mathcal{X}$-valued, $\mathcal{M}$-ary tree $\mathcal{T}$ of depth T is a complete rooted tree such that (1) each internal node v is labeled by an instance $x \in \mathcal{X}$ and (2) for every internal node v and object $m \in \mathcal{M}$, there is an outgoing edge e_v^m indexed by m. Such a tree can be identified by a sequence $(\mathcal{T}_1, ..., \mathcal{T}_T)$ of labeling functions $\mathcal{T}_t : \mathcal{M}^{t-1} \to \mathcal{X}$ which provide the labels for each internal node. A path of length T is given by a sequence of objects $m = (m_1, ..., m_T) \in \mathcal{M}^T$. Then, $\mathcal{T}_t(m_1, ..., m_{t-1})$ gives the label of the node by following the path $(m_1, ..., m_{t-1})$ starting from the root node, going down the edges indexed by the m_t's. We let $\mathcal{T}_1 \in \mathcal{X}$ denote the instance labeling the root node. For brevity, we define $m_{<t} = (m_1, ..., m_{t-1})$ and therefore write $\mathcal{T}_t(m_1, ..., m_{t-1}) = \mathcal{T}_t(m_{<t})$. Analogously, we let $m_{\leq t} = (m_1, ..., m_t)$.

Using this notation, we define the extension of the Littlestone dimension to the multiclass setting proposed by Daniely et al. [2011].

Definition 5.2.2 (Littlestone dimension [Littlestone, 1987, Daniely et al., 2011])**.** *Let $\mathcal{T}$ be a complete, $\mathcal{X}$-valued, $\{\pm 1\}$-ary tree of depth d such that the edges from a single parent node to its child nodes are each labeled with a different element of $\mathcal{Y}$. The tree $\mathcal{T}$ is shattered by $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ if for every path $\sigma = (\sigma_1, ..., \sigma_d) \in \{\pm 1\}^d$, there exists a hypothesis $h_\sigma \in \mathcal{H}$ such that for all $t \in [d]$, $h_\sigma(\mathcal{T}_t(\sigma_{<t})) = y(\sigma_{\leq t})$, where $y(\sigma_{\leq t})$ is the label of the edge between the the nodes $(\mathcal{T}_t(\sigma_{<t}), (\mathcal{T}_{t+1}(\sigma_{\leq t})))$. The Littlestone dimension of $\mathcal{H}$, denoted $\mathrm{L}(\mathcal{H})$, is the maximal depth of a tree $\mathcal{T}$ that is shattered by $\mathcal{H}$. If there exist shattered trees of arbitrarily large depth, we say that $\mathrm{L}(\mathcal{H}) = \infty$.*

In the same work, Daniely et al. [2011] defined a combinatorial parameter called the Bandit Littlestone dimension (BLdim) and showed that it characterizes bandit online learnability of deterministic learners in the realizable setting.

Definition 5.2.3 (Bandit Littlestone dimension [Daniely et al., 2011])**.** *Let $\mathcal{T}$ be a complete, $\mathcal{X}$-valued, $\mathcal{Y}$-ary tree of depth d. The tree $\mathcal{T}$ is shattered by $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ if for every path $y = (y_1, ..., y_d) \in \mathcal{Y}^d$, there exists a hypothesis $h_y \in \mathcal{H}$ such that for all $t \in [d]$, $h_y(\mathcal{T}_t(y_{<t})) \neq y_t$. The Bandit Littlestone dimension of $\mathcal{H}$, denoted $\mathrm{BL}(\mathcal{H})$, is the maximal depth of a tree $\mathcal{T}$*

that is shattered by $\mathcal{H}$. If there exist shattered trees of arbitrarily large depth, we say that $\mathrm{BL}(\mathcal{H}) = \infty$.

In particular, Daniely et al. [2011] show a matching upper and lowerbound on the realizable error rate of deterministic learners in terms of the BLdim.

Theorem 5.2.1 (Realizable Learnability [Daniely et al., 2011])**.** *In the realizable setting, for any $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, there exists a deterministic online learner whose cumulative loss on the worst-case sequence, under bandit feedback, is at most* $\mathrm{BL}(\mathcal{H})$*. Also, the cumulative loss of any deterministic online learner on the worst-case sequence, under bandit feedback, is at least* $\mathrm{BL}(\mathcal{H})$.

In the agnostic setting, Daniely and Helbertal [2013] gave an upperbound on the expected regret under bandit feedback, in terms of $|\mathcal{Y}|$ and Ldim.

Theorem 5.2.2 (Agnostic Learnability [Daniely and Helbertal, 2013])**.** *For any $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, there exists an online learner $\mathcal{A}$ such that*

$$\mathrm{R}_{\mathcal{A}}(T, \mathcal{H}) \leq e\sqrt{\mathrm{L}(\mathcal{H})|\mathcal{Y}|T\log(T|\mathcal{Y}|)}.$$

In Section 5.3, we show that $|\mathcal{Y}|$ in Theorem 5.2.2 can be replaced with $\mathrm{BL}(\mathcal{H})$. Note that this both qualitatively and quantitatively improves over Theorem 5.2.2. Qualitatively, it shows that finite $\mathrm{BL}(\mathcal{H})$ suffices for bandit online learnability, without any requirements on $|\mathcal{Y}|$. Furthermore, there is no better qualitative characterization, since as we show in Section 5.4, finite $\mathrm{BL}(\mathcal{H})$ is also necessary for learnability. A quantitative improvement is achieved in cases where $|\mathcal{Y}| \gg \mathrm{BL}(\mathcal{H})$.

5.2.2 Online Learnability and Uniform Convergence

The relationship between learnability and uniform convergence has a rich history in learning theory. For binary classification in the PAC setting, the seminal work by Vapnik and Chervonenkis [1974b] shows that uniform convergence and PAC learnability are equivalent. Likewise, for online binary classification, an online analog of uniform convergence, termed Sequential Uniform Convergence (SUC), is equivalent to online learnability [Rakhlin, Sridharan, and Tewari, 2015b, Alon, Ben-Eliezer, Dagan, Moran, Naor, and Yogev, 2021b]. However, this equivalence between uniform convergence and learnability breaks down for multiclass classification. Indeed, in the PAC setting, it was shown that while uniform convergence suffices for multiclass learnability, it is not necessary [Natarajan, 1989]. Recently, Hanneke et al. [2023] extended this separation to the online, full-information feedback setting

by showing that SUC is sufficient but not necessary for multiclass learnability. Instead, they show that SUC is characterized by a different combinatorial parameter termed the Sequential Graph dimension (SGdim).

Definition 5.2.4 (Sequential Graph dimension [Hanneke et al., 2023]). *Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ and $\ell \circ \mathcal{H} = \{(x, y) \mapsto \mathbb{1}\{h(x) \neq y\} : h \in \mathcal{H}\}$ be its loss class. Then, the Sequential Graph dimension of $\mathcal{H}$, denoted $\mathrm{SG}(\mathcal{H})$, is defined as $\mathrm{SG}(\mathcal{H}) = \mathrm{L}(\ell \circ \mathcal{H})$.*

In particular, a hypothesis class $\mathcal{H}$ enjoys the SUC property if and only if its SGdim is finite.

Theorem 5.2.3 (Hanneke et al. [2023], Rakhlin et al. [2015a]). *For any hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, the SUC property holds for $\mathcal{H}$ if and only if $\mathrm{SG}(\mathcal{H}) < \infty$.*

In fact, there is a quantitative relation between SGdim and Ldim when the label space is bounded.

Theorem 5.2.4 (Hanneke et al. [2023]). *For any $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ such that $|\mathcal{Y}| < \infty$, we have $\mathrm{SG}(\mathcal{H}) = O(\mathrm{L}(\mathcal{H}) \log(|\mathcal{Y}|))$.*

A combination of Theorem 5.2.4 and a result due to Alon et al. [2021b], Hanneke et al. [2023] derives a new upperbound on the best achievable expected regret under full-information feedback.

Theorem 5.2.5 (Hanneke et al. [2023], Alon et al. [2021b]). *For any $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ such that $\mathrm{SG}(\mathcal{H}) < \infty$, there exists an online learner whose expected regret under full-information feedback is at most $O(\sqrt{\mathrm{SG}(\mathcal{H})\, T})$.*

In this work, we also investigate the relationship between bandit online learnablity and SUC. In Section 5.3, we show that finite BLdim implies finite SGdim, and more precisely that $\mathrm{SG}(\mathcal{H}) = O(\mathrm{L}(\mathcal{H}) \log(\mathrm{BL}(\mathcal{H})))$. On the other hand, in Section 5.4, we exhibit a class where SUC holds, but is not bandit online learnable. Together, these results imply that SUC is necessary, but not sufficient, for bandit online learnability.

5.3 BLdim is Sufficient for Bandit Online Learnability

In this section we prove Theorem 5.1.2, which implies direction (2) $\implies$ (1) in Theorem 5.1.1. We also prove (3) $\implies$ (2) towards the end of the section. The first ingredient of this proof is the following result which shows that the BLdim provides a uniform upperbound on the size of the projection of $\mathcal{H}$ on any instance $x \in \mathcal{X}$.

Lemma 5.3.1. *For any $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, we have* $\sup_{x \in \mathcal{X}} |\mathcal{H}(x)| \leq \mathrm{BL}(\mathcal{H}) + 1$.

Proof. Suppose that $|\mathcal{H}(x)| \geq \mathrm{BL}(\mathcal{H}) + 2$ for some $x \in \mathcal{X}$. We will prove the lemma by contradiction, constructing a BLdim tree of depth $\mathrm{BL}(\mathcal{H}) + 1$ that is shattered by $\mathcal{H}$. Let $\mathcal{T}$ be a BLdim tree of depth $\mathrm{BL}(\mathcal{H}) + 1$ with every internal node labeled by x. Without loss of generality, suppose that $\mathcal{H}(x) = \{1, 2, ..., \mathrm{BL}(\mathcal{H}) + 2\}$, and let there be $h_1, ..., h_{\mathrm{BL}(\mathcal{H})+2} \in \mathcal{H}$ such that $h_i(x) = i$ for all $1 \leq i \leq \mathrm{BL}(\mathcal{H})+2$. We now show that $\mathcal{H}$ shatters $\mathcal{T}$. Consider any path down $\mathcal{T}$. Since $\mathcal{T}$ has depth $\mathrm{BL}(\mathcal{H}) + 1$, there can only be $\mathrm{BL}(\mathcal{H}) + 1$ different labels on that path. Since there are at least $\mathrm{BL}(\mathcal{H}) + 2$ hypotheses in $\mathcal{H}$, there is a hypothesis $h_i \in \mathcal{H}$ such that $h_i(x)$ is not equal to any of the labels on the path. Since the path is arbitrary, the tree is shattered by $\mathcal{H}$ according to Definition 5.2.3. By contradiction, $|\mathcal{H}(x)| \leq \mathrm{BL}(\mathcal{H}) + 1$ for all $x \in \mathcal{X}$. ■

A uniform upperbound C on the projection size of $\mathcal{H}$ is a strong property: it allows us to effectively reduce the label space from $\mathcal{Y}$ to $[C]$. Lemma 5.3.2 makes this precise. For a bandit algorithm $\mathcal{A}$, let $\mathcal{A}(x)$ be its prediction on x, given the history of the game so far (for the sake of readability, we omit the information received prior to instance x from the notation).

Lemma 5.3.2. *Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ such that* $\sup_{x \in \mathcal{X}} |\mathcal{H}(x)| \leq C$. *Then, there exists a hypothesis class $\bar{\mathcal{H}} \subseteq [C]^{\mathcal{X}}$ such that*

(i) $\mathrm{L}(\bar{\mathcal{H}}) = \mathrm{L}(\mathcal{H})$.

(ii) $\mathrm{SG}(\bar{\mathcal{H}}) = \mathrm{SG}(\mathcal{H})$.

(iii) For every bandit algorithm $\bar{\mathcal{A}}$ for $\bar{\mathcal{H}}$ such that $\bar{\mathcal{A}}(x) \in \bar{\mathcal{H}}(x)$ at all times, there exists a bandit algorithm $\mathcal{A}$ for $\mathcal{H}$ such that $\mathrm{R}_{\mathcal{A}}(T, \mathcal{H}) = R_{\bar{\mathcal{A}}}(T, \bar{\mathcal{H}})$ for all T. Furthermore, if $\bar{\mathcal{A}}$ is deterministic, then so is $\mathcal{A}$.

(iv) $\mathrm{BL}(\bar{\mathcal{H}}) = \mathrm{BL}(\mathcal{H})$.

Proof. Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be such that $\sup_{x \in \mathcal{X}} |\mathcal{H}(x)| \leq C$. For every $x \in \mathcal{X}$, define a function $\phi_x : \mathcal{H}(x) \to [|\mathcal{H}(x)|]$ such that ϕ_x is one-to-one. Finally, consider the following hypothesis class $\bar{\mathcal{H}} = \{x \mapsto \phi_x(h(x)) : h \in \mathcal{H}\}$. Clearly, we have that $\bar{\mathcal{H}} \subseteq [C]^{\mathcal{X}}$ and we now show that $\bar{\mathcal{H}}$ also satisfies the four properties above.

Property (i) follows from observing that any non-empty shattered Ldim tree for $\bar{\mathcal{H}}$ can be transformed into a shattered Ldim tree for $\mathcal{H}$, since the the outgoing edges of any internal node labeled by x must be labeled using elements of $[|\mathcal{H}(x)|]$. Thus, the inverse mapping $\phi_x^{-1} : [|\mathcal{H}(x)|] \to \mathcal{H}(x)$ can be used to transform this tree into an Ldim tree of the same depth

for $\mathcal{H}$. Likewise, one can use the forward mapping ϕ_x to transform any non-empty shattered Ldim tree for $\mathcal{H}$ into a non-empty shattered Ldim tree for $\bar{\mathcal{H}}$. The equality trivially holds if no non-empty Ldim tree exists.

Property (ii) follows from the fact that every internal node in a non-empty shattered Ldim tree for the loss class $\{(x, y) \mapsto \mathbb{1}\{h(x) \neq y\} : h \in \mathcal{H}\}$ must be labeled using elements of $\{(x, y) : y \in \mathcal{H}(x)\}$. Thus the mapping function ϕ_x can be used to transform any non-empty shattered Ldim tree for the loss class $\{(x, y) \mapsto \mathbb{1}\{h(x) \neq y\} : h \in \mathcal{H}\}$ into a non-empty shattered Ldim tree for the loss class $\{(x, y) \mapsto \mathbb{1}\{\bar{h}(x) \neq y\} : \bar{h} \in \bar{\mathcal{H}}\}$. The reverse direction follows analogously by using the inverse mapping function ϕ_x^{-1}.

To prove property (iii), suppose that $\bar{\mathcal{A}}$ is a bandit algorithm for $\bar{\mathcal{H}}$ such that on any instance $x \in \mathcal{X}$, the prediction of $\bar{\mathcal{A}}$ always lies in $\bar{\mathcal{H}}(x)$. Algorithm 6 uses $\bar{\mathcal{A}}$ in a black-box fashion to construct a bandit learner $\mathcal{A}$ for $\mathcal{H}$.

Algorithm 6 Bandit algorithm $\mathcal{A}$

Require: Hypothesis class $\mathcal{H}$, bandit algorithm $\bar{\mathcal{A}}$ for $\bar{\mathcal{H}}$

for $t = 1, ..., T$ **do**
  Receive example x_t
  Query $\bar{y}_t = \bar{\mathcal{A}}(x_t)$
  Predict $\hat{y}_t = \phi_{x_t}^{-1}(\bar{y}_t)$
  Observe loss $\mathbb{1}\{y_t \neq \hat{y}_t\}$ and pass along the indication to $\bar{\mathcal{A}}$
end for

We claim that the expected regret of Algorithm $\mathcal{A}$ is $R_{\bar{\mathcal{A}}}(T, \bar{\mathcal{H}})$. To see this, fix $T \in \mathbb{N}$ and let $S = (x_1, y_1), \ldots, (x_T, y_T) \in (\mathcal{X} \times \mathcal{Y})^T$ be the sequence of examples to be passed to $\mathcal{A}$. We show that there exists a sequence of examples $S' \in (\mathcal{X} \times [C] \cup \{\star\})^T$ for $\bar{\mathcal{A}}$ such that

$$\min_{h \in \mathcal{H}} \sum_{(x_t, y_t) \in S} \mathbb{1}\{h(x_t) \neq y_t\} = \min_{\bar{h} \in \bar{\mathcal{H}}} \sum_{(x_t, y'_t) \in S'} \mathbb{1}\{\bar{h}(x_t) \neq y'_t\}, \tag{5.1}$$

$$\mathbb{E}\left[\sum_{(x_t, y_t) \in S} \mathbb{1}\{\mathcal{A}(x_t) \neq y_t\}\right] = \mathbb{E}\left[\sum_{(x_t, y'_t) \in S'} \mathbb{1}\{\bar{\mathcal{A}}(x_t) \neq y'_t\}\right], \tag{5.2}$$

and (3) the feedback that $\mathcal{A}$ provides to $\bar{\mathcal{A}}$ matches the feedback that $\bar{\mathcal{A}}$ would have received if it was executed on S'.

Combining (5.1), (5.2), and (3) and the regret guarantee $R_{\bar{\mathcal{A}}}(T, \bar{\mathcal{H}})$ for $\bar{\mathcal{A}}$ immediately implies property (iii). It remains to construct S' for which all three statements hold. For every $t \in [T]$, let $y'_t = \phi_{x_t}(y_t)\mathbb{1}\{y_t \in \mathcal{H}(x_t)\} + \star\mathbb{1}\{y_t \notin \mathcal{H}(x_t)\}$. Consider the following stream $S' = (x_1, y'_1), ..., (x_T, y'_T) \in (\mathcal{X} \times [C] \cup \{\star\})^T$. To see that (5.1) holds, observe that for every $h \in \mathcal{H}$ we have that

$$\begin{aligned}\sum_{t=1}^{T} \mathbb{1}\{h(x_t) \neq y_t\} &= \sum_{t:y_t \in \mathcal{H}(x_t)} \mathbb{1}\{h(x_t) \neq y_t\} + \sum_{t:y_t \notin \mathcal{H}(x_t)} \mathbb{1}\{h(x_t) \neq y_t\} \\ &= \sum_{t:y_t \in \mathcal{H}(x_t)} \mathbb{1}\{\phi_{x_t}(h(x_t)) \neq \phi_{x_t}(y_t)\} + \sum_{t:y_t \notin \mathcal{H}(x_t)} \mathbb{1}\{\phi_{x_t}(h(x_t)) \neq \star\} \\ &= \sum_{t=1}^{T} \mathbb{1}\{\bar{h}(x_t) \neq y'_t\}.\end{aligned}$$

To see (5.2), note that

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}(x_t) \neq y_t\}\right] &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\phi_{x_t}^{-1}(\bar{y}_t) \neq y_t\}\right] \\ &= \mathbb{E}\left[\sum_{t:y_t \in \mathcal{H}(x_t)} \mathbb{1}\{\phi_{x_t}^{-1}(\bar{y}_t) \neq y_t\} + \sum_{t:y_t \notin \mathcal{H}(x_t)} \mathbb{1}\{\phi_{x_t}^{-1}(\bar{y}_t) \neq y_t\}\right] \\ &= \mathbb{E}\left[\sum_{t:y_t \in \mathcal{H}(x_t)} \mathbb{1}\{\bar{\mathcal{A}}(x_t) \neq \phi_{x_t}(y_t)\} + \sum_{t:y_t \notin \mathcal{H}(x_t)} \mathbb{1}\{\bar{\mathcal{A}}(x_t) \neq \star\}\right] \\ &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\bar{\mathcal{A}}(x_t) \neq y'_t\}\right].\end{aligned}$$

Finally, to prove (3), it suffices to show that $\mathbb{1}\{y_t \neq \hat{y}_t\} = \mathbb{1}\{y'_t \neq \bar{\mathcal{A}}(x_t)\}$. If $\mathbb{1}\{y_t \neq \hat{y}_t\} = 0$, then $y_t = \phi_{x_t}^{-1}(\bar{y}_t)$ and $\bar{\mathcal{A}}(x_t) = \phi_{x_t}(y_t) = y'_t$ as needed. If $\mathbb{1}\{y_t \neq \hat{y}_t\} = 1$ and $y_t \in \mathcal{H}(x_t)$, then $\bar{y}_t \neq \phi_{x_t}(y_t) = y'_t$. Lastly, if $\mathbb{1}\{y_t \neq \hat{y}_t\} = 1$ and $y_t \notin \mathcal{H}(x_t)$, we get that $\bar{y}_t \neq y'_t = \star$ since $\bar{\mathcal{A}}(x_t) \in \bar{\mathcal{H}}(x_t)$. The "furthermore" part of the property is straightforward by the construction of $\mathcal{A}$.

Let us move on to Property (iv). The direction $\mathrm{BL}(\mathcal{H}) \leq \mathrm{BL}(\bar{\mathcal{H}})$ follows from Property (iii). Indeed, let $\bar{\mathcal{A}}$ be the optimal BSOA deterministic learner defined in [Daniely et al., 2011] for $\bar{\mathcal{H}}$ under the assumption of realizability. For every round t, the algorithm $\bar{\mathcal{A}}$ never predicts $y \notin \bar{\mathcal{H}}(x_t)$ by its definition. Therefore, by Property (iii) there exists a deterministic learner $\mathcal{A}$ for $\mathcal{H}$ having the same guarantees as of $\bar{\mathcal{A}}$. Therefore $\mathrm{BL}(\mathcal{H}) \leq \mathrm{BL}(\bar{\mathcal{H}})$. The reverse direction $\mathrm{BL}(\mathcal{H}) \geq \mathrm{BL}(\bar{\mathcal{H}})$ follows by considering the realizable setting and the bandit algorithm for $\bar{\mathcal{H}}$ that, given any instance x_t, passes x_t to the BSOA for $\mathcal{H}$, receives its prediction $\bar{y}_t \in \mathcal{Y}$, makes the prediction $\hat{y}_t = \phi_{x_t}(\bar{y}_t) \in [C]$, and upon receiving the feedback $\mathbb{1}\{\hat{y}_t \neq y_t\}$, passes the same feedback to the BSOA. The same analysis as in Property (iii)

can be used to show that this algorithm makes at most $\mathrm{BL}(\mathcal{H})$ mistakes on any realizable stream. ■

In order to use Property (iii) of Lemma 5.3.2, we need to construct a bandit learner $\bar{\mathcal{A}}$ which on any instance $x \in \mathcal{X}$, makes a prediction that lies in $\mathcal{H}(x)$ and achieves a sublinear regret bound whenever $\mathrm{BL}(\mathcal{H}) < \infty$. Unfortunately, the generic bandit learner witnessing the proof of Theorem 5.2.2 does not guarantee that its predictions always lie in the projection of $\mathcal{H}$. Fortunately, the following lemma, whose proof can be found in Appendix C.1.1, shows that a slight modification of the bandit learner used to prove Theorem 5.2.2 can achieve the same regret bound, while ensuring that the predictions always lie in the projection of $\mathcal{H}$.

Lemma 5.3.3. *For any $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, there exists an online learner $\mathcal{A}$ such that*

$$\mathrm{R}_{\mathcal{A}}(T, \mathcal{H}) \leq e\sqrt{\mathrm{L}(\mathcal{H})|\mathcal{Y}|T\log(T|\mathcal{Y}|)},$$

while ensuring that $\mathcal{A}(x_t) \in \mathcal{H}(x_t)$ almost surely.

We are now ready to prove Theorem 5.1.2, which implies that finitness of BLdim is sufficient for bandit online learnability even when the label space is unbounded. This proves direction (2) $\implies$ (1) in Theorem 5.1.1.

Proof. (of Theorem 5.1.2) We first prove a stronger result and then show that Theorem 5.1.2 follows. Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be such that $\mathrm{BL}(\mathcal{H}) < \infty$. Then, by Lemmas 5.3.2 and 5.3.3, there exists an online learner whose expected regret in the agnostic setting under bandit feedback is at most $e\sqrt{\mathrm{L}(\mathcal{H})CT\log(TC)}$ where $C = \sup_{x\in\mathcal{X}} |\mathcal{H}(x)|$. Since Lemma 5.3.1 states that $C \leq \mathrm{BL}(\mathcal{H}) + 1 \leq 2\mathrm{BL}(\mathcal{H})$, we can further upperbound the expected regret by $2e\sqrt{\mathrm{L}(\mathcal{H})\mathrm{BL}(\mathcal{H})T\log(T\,\mathrm{BL}(\mathcal{H}))}$. There are now two cases to consider. If $T \leq \mathrm{BL}(\mathcal{H})$, the expected regret of any bandit online learner can be trivially upperbounded by T. Noting that $8\sqrt{\mathrm{L}(\mathcal{H})\mathrm{BL}(\mathcal{H})T\log(T)} \geq T$ when $T \leq \mathrm{BL}(\mathcal{H})$ completes this case. If $T > \mathrm{BL}(\mathcal{H})$, then we can upperbound the expected regret of the bandit online learner by

$$2e\sqrt{\mathrm{L}(\mathcal{H})\mathrm{BL}(\mathcal{H})T\log(T\,\mathrm{BL}(\mathcal{H}))} \leq 2e\sqrt{2\mathrm{L}(\mathcal{H})\mathrm{BL}(\mathcal{H})T\log(T)} \leq 8\sqrt{\mathrm{L}(\mathcal{H})\mathrm{BL}(\mathcal{H})T\log(T)},$$

matching the upperbound given in the statement of Theorem 5.1.2. This completes the proof. ■

In online learning theory, upperbounds on the minimax expected regret are traditionally derived in terms of the single combinatorial dimension that characterizes learnability. However, our upperbound in Theorem 5.1.2 is in terms of both the Ldim and BLdim. To get

a bound depending only on the BLdim, one can trivially use the fact that $\mathrm{L}(\mathcal{H}) \leq \mathrm{BL}(\mathcal{H})$ to get a suboptimal upperbound of $8\,\mathrm{BL}(\mathcal{H})\sqrt{T\log(T)}$ on the minimax expected regret. However, as an intermediate step to our upperbound in Theorem 5.1.2, we show that the minimax expected regret can actually be upperbounded by $e\sqrt{\mathrm{L}(\mathcal{H})CT\log(TC)}$, and thus it is natural to ask whether there is an upperbound on $\sqrt{\mathrm{L}(\mathcal{H})C}$ that is significantly better than $\mathrm{BL}(\mathcal{H})$. Unfortunately, the following example shows that this is not the case.

Example 1. Fix $d, C \in \mathbb{N}$. Define the instance space $\mathcal{X} = \{x_0, ..., x_d\}$ and the label space $\mathcal{Y} = \{0, ..., C-1\}$. Let $\mathcal{H}_1 = \{0,1\}^{\{x_1,...,x_d\}}$ and $\mathcal{H}_2 = \{x \mapsto y\,\mathbb{1}\{x = x_0\} : y \in \mathcal{Y}\}$. Consider the hypothesis class $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2$. Clearly, $\mathrm{L}(\mathcal{H}) \geq \mathrm{L}(\mathcal{H}_1) = \mathrm{BL}(\mathcal{H}_1) = d$. Moreover, $\mathrm{BL}(\mathcal{H}_2) \leq C - 1$. We now give an upperbound on $\mathrm{BL}(\mathcal{H})$ by constructing a deterministic learner for $\mathcal{H}$. Consider the learning algorithm that predicts 0 until its first mistake, removes inconsistent hypotheses, and plays the Bandit Standard Optimal Algorithm (BSOA) from Daniely et al. [2011] on future rounds. We now show that this algorithm makes at most $1+\max\{d, C-1\}$ mistakes on any realizable stream. There are two cases to consider. Suppose the algorithm makes its first mistake on x_0. Then, by construction of $\mathcal{H}$, the true hypothesis must be in $\mathcal{H}_2$ and thus the BSOA makes no more than $\mathrm{BL}(\mathcal{H}_2) \leq C - 1$ mistakes in all future rounds. On the other hand, if the algorithm makes its first mistake on $x \in \{x_1, ..., x_d\}$, then the true hypothesis must be in $\mathcal{H}_1$ and thus the BSOA makes at most $\mathrm{BL}(\mathcal{H}_1) = d$ mistakes on all future rounds. Overall, the algorithm makes at most $1 + \max\{d, C-1\}$ mistakes. Since the BLdim lowerbounds the number of mistakes made by any deterministic learner under bandit feedback, we must have that $\mathrm{BL}(\mathcal{H}) \leq 1 + \max\{d, C-1\}$. Taking $C = d + 1$, we have that $\mathrm{BL}(\mathcal{H}) \leq 1 + d \leq 1 + \sqrt{\mathrm{L}(\mathcal{H})C}$, which completes the example.

We leave it as an interesting open question to derive optimal lower and upper bounds on the minimax expected regret in terms of only the BLdim (see Section 9.3). Lemma 5.3.2 can also be used to sharpen the relationship between BLdim and Ldim. In particular, due to [Auer and Long, 1999, Daniely and Helbertal, 2013, Long, 2017], there exists a *deterministic* online learner in the realizable setting whose number of mistakes, under bandit feedback, is at most $O(\mathrm{L}(\mathcal{H})|\mathcal{Y}|\log(|\mathcal{Y}|))$. Since the BLdim lowerbounds the number of mistakes made by any deterministic online learner in the realizable setting, Lemma 5.3.2 immediately implies that when $\sup_{x\in\mathcal{X}} |\mathcal{H}(x)| \leq C$, we have $\mathrm{BL}(\mathcal{H}) = O(\mathrm{L}(\mathcal{H})C\log C)$, proving direction (3) $\implies$ (2) in Theorem 5.1.1. In Section 5.4, we show that finiteness of both C and $\mathrm{L}(\mathcal{H})$ is also necessary for learnability (direction (1) $\implies$ (3)).

We end this section with Corollary 5.3.4, which shows that SUC is necessary for a hypothesis class to be bandit online learnable.

Corollary 5.3.4. *If* $\mathrm{BL}(\mathcal{H}) < \infty$*, then* $\mathrm{SG}(\mathcal{H}) = O(\mathrm{L}(\mathcal{H})\log(\mathrm{BL}(\mathcal{H})))$.

Proof. Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ such that $\mathrm{BL}(\mathcal{H}) < \infty$. Then by Lemmas 5.3.1 and 5.3.2, there exists a class $\bar{\mathcal{H}} \subseteq [\mathrm{BL}(\mathcal{H})+1]^{\mathcal{X}}$ such that $\mathrm{L}(\bar{\mathcal{H}}) = \mathrm{L}(\mathcal{H})$ and $\mathrm{SG}(\bar{\mathcal{H}}) = \mathrm{SG}(\mathcal{H})$. Since $\mathrm{BL}(\mathcal{H})+1 < \infty$, Theorem 5.2.4 implies that $\mathrm{SG}(\bar{\mathcal{H}}) = O(\mathrm{L}(\bar{\mathcal{H}}) \log(\mathrm{BL}(\bar{\mathcal{H}}))) = O(\mathrm{L}(\mathcal{H}) \log(\mathrm{BL}(\mathcal{H})))$. ■

Since $\mathrm{BL}(\mathcal{H}) < \infty$ implies that $\mathrm{L}(\mathcal{H}) < \infty$, Corollary 5.3.4 and Theorem 5.2.3 taken together prove the first half of Theorem 5.1.3, showing that $\mathcal{H}$ enjoys SUC when $\mathrm{BL}(\mathcal{H}) < \infty$. Moreover, when $\mathrm{BL}(\mathcal{H}) < \infty$, Corollary 5.3.4 along with Theorem 5.2.5 implies a slightly sharper upperbound on the optimal expected regret in the agnostic setting under *full-information* feedback.

Corollary 5.3.5. *Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ such that $\mathrm{BL}(\mathcal{H}) < \infty$. Then, there exists an agnostic online learner whose expected regret, under full-information feedback, is at most*

$$O\left(\sqrt{\mathrm{L}(\mathcal{H}) T \log(\mathrm{BL}(\mathcal{H}))}\right).$$

Proof. Let $\mathcal{H}$ be such that $\mathrm{BL}(\mathcal{H}) < \infty$. Then, by Corollary 5.3.4, $\mathrm{SG}(\mathcal{H}) = O(\mathrm{L}(\mathcal{H}) \log(\mathrm{BL}(\mathcal{H})))$. Also, by Theorem 5.2.5, we have that under full-information feedback, there exists a online learner whose expected regret is at most $O(\sqrt{T \,\mathrm{SG}(\mathcal{H})})$. Combining these two results gives the stated claim. ■

Namely, Corollary 5.3.5 improves upon the upperbound on expected regret given by [Hanneke et al., 2023, Theorem 1] by replacing the $\log(\frac{T}{\mathrm{L}(\mathcal{H})})$ factor with $\log(\mathrm{BL}(\mathcal{H}))$.

5.4 Finite BLdim is Necessary for Bandit Online Learnability

In this section, we complement the results of Section 5.3, and deduce that finiteness of BLdim is necessary for bandit online learnability in the realizable setting even when the label space is unbounded. Since agnostic learnability implies realizable learnability, this also implies that finiteness of the BLdim is necessary for agnostic learnability, completing the proof of the direction (1) $\implies$ (2) in Theorem 5.1.1. This will also imply (1) $\implies$ (3), which completes the proof of Theorem 5.1.1.

Lemma 5.4.1. *Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ and $C = \sup_{x \in \mathcal{X}} |\mathcal{H}(x)|$. Then, for every bandit online learner $\mathcal{A}$:*

1. *There exists a realizable stream with expected regret at least $\frac{\mathrm{BL}(\mathcal{H})}{4C \log C}$ if $T \geq \mathrm{L}(\mathcal{H})$ and at least $T/2$ otherwise.*

2. *There exists a realizable stream with expected regret at least* $\frac{C-1}{2}$ *if* $T \geq C$*, and at least* $\frac{T-1}{2}$ *otherwise.*

Proof. Let us start with the first item. A well-known result by Ben-David et al. [2009] states that there exists a realizable stream of length $T = \mathrm{L}(\mathcal{H})$ such that in expectation, $\mathcal{A}$ makes at least $\mathrm{L}(\mathcal{H})/2$ mistakes under full information feedback. On the other hand, by Long [2017] and Lemma 5.3.2 we have $\mathrm{BL}(\mathcal{H}) \leq 2\mathrm{L}(\mathcal{H})C \log C$, implying the item for the case $T \geq \mathrm{L}(\mathcal{H})$. If $T < \mathrm{L}(\mathcal{H})$, we employ the lower bound on T instead of on $\mathrm{L}(\mathcal{H})$, concluding this item. The second item follows immediately from [Daniely and Helbertal, 2013, Claim 2]. ■

Lemma 5.4.1 implies that finiteness of BLdim is necessary for bandit online learnability in the realizable setting. Recall that $\mathrm{BL}(\mathcal{H}) \geq C - 1$ due to Lemma 5.3.1. Now, if $C = \infty$ (where $C := \sup_{x \in \mathcal{X}} |\mathcal{H}(x)|$), then Lemma 5.4.1 implies that the expected regret of any online learner under bandit feedback and in the realizable setting, is at least $\frac{T-1}{2}$, a linear function of T. On the other hand, if $\mathrm{BL}(\mathcal{H}) = \infty$ and $C < \infty$, then the bound $\mathrm{BL}(\mathcal{H}) = O(\mathrm{L}(\mathcal{H})C \log C)$ implies that $\mathrm{L}(\mathcal{H}) = \infty$, and then Lemma 5.4.1 implies a lowerbound of $\frac{T}{2}$ on the expected regret. This proves the direction (1) $\implies$ (2) in Theorem 5.1.1. Using the fact that $\mathrm{BL}(\mathcal{H}) \geq \mathrm{L}(\mathcal{H})$ and Lemma 5.3.1 shows that (2) $\implies$ (3), completing the proof of Theorem 5.1.1.

Furthermore, if C is a constant, then taken together with Theorem 5.2.1, Lemma 5.4.1 implies that the BLdim characterizes the optimal expected mistake bound of randomized learners in the realizable setting up to constant factors. In the agnostic setting, the full-information lowerbound of $\sqrt{\frac{\mathrm{L}(\mathcal{H})T}{8}}$ on the expected regret can also be a tight lowerbound under bandit feedback up to logarithmic factors in T. For example, for every class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ such that $\sup_{x \in \mathcal{X}} |\mathcal{H}(x)| \leq 2$, Theorem 5.2.2 and Lemma 5.3.2 imply the existence of a bandit online learner whose expected regret is at most $8\sqrt{\mathrm{L}(\mathcal{H})T \log(T)}$.

Finally, Lemma 5.4.1 together with Lemma 5.4.2 shows that neither the finitness of Ldim nor the finiteness of SGdim is sufficient for bandit online learnability.

Lemma 5.4.2. *Let* $\mathcal{X} = \{0\}$*,* $\mathcal{Y} = \mathbb{N}$ *and* $\mathcal{H} = \{h_a : a \in \mathbb{N}\}$ *where* $h_a(0) = a$*. Then,* $\mathrm{L}(\mathcal{H}) = \mathrm{SG}(\mathcal{H}) = 1$ *but* $\mathrm{BL}(\mathcal{H}) = \infty$*.*

Proof. The equality $\mathrm{BL}(\mathcal{H}) = \infty$ follows from the fact that $|\mathcal{H}(0)| = \infty$ and Lemma 5.3.1. We have $\mathrm{L}(\mathcal{H}) = 1$ because for any labeled instance $(0, y) \in \mathcal{X} \times \mathcal{Y}$, there only exists one hypothesis $h \in \mathcal{H}$ such that $h(0) = y$ (namely h_y). Lastly, $\mathrm{SG}(\mathcal{H}) = 1$ because for any labeled instance $(0, y) \in \mathcal{X} \times \mathcal{Y}$ there exists only one function in the loss class $\{(0, y) \mapsto \mathbb{1}\{h(0) \neq y\} : h \in \mathcal{H}\}$ that achieves loss 0. ■

Lemma 5.4.2 completes the proof of Theorem 5.1.3, since we have exhibited a class for which SUC holds but is not bandit online learnable.

CHAPTER 6

Apple Tasting: Combinatorial Dimensions and Minimax Rates

In online binary classification under apple tasting feedback, the learner only observes the true label if it predicts "1". First studied by Helmbold et al. [2000a], in this chapter, we revisit this classical partial-feedback setting and study online learnability from a combinatorial perspective. We show that the Littlestone dimension continues to provide a tight quantitative characterization of apple tasting in the agnostic setting, closing an open question posed by Helmbold et al. [2000a]. In addition, we give a new combinatorial parameter, called the Effective width, that tightly quantifies the minimax expected number of mistakes in the realizable setting. As a corollary, we use the Effective width to establish a trichotomy of the minimax expected number of mistakes in the realizable setting. In particular, we show that in the realizable setting, the expected number of mistakes of any learner, under apple tasting feedback, can only be either $\Theta(1), \Theta(\sqrt{T})$, or $\Theta(T)$. This is in contrast to the full-information realizable setting where only $\Theta(1)$ and $\Theta(T)$ are possible.

6.1 Introduction

In the standard online binary classification setting, a learner plays a repeated game against an adversary. In each round, the adversary picks a labeled example $(x, y) \in \mathcal{X} \times \{0, 1\}$ and reveals the unlabeled example x to the learner. The learner observes x and then makes a prediction $\hat{y} \in \{0, 1\}$. Finally, the adversary reveals the true label y and the learner suffers the loss $\mathbb{1}\{\hat{y} \neq y\}$ [Littlestone, 1987]. In many situations, receiving feedback after every prediction may not be realistic. For example, in spam filtering, emails that are classified as spam are often not verified by the user. Accordingly, the learner only receives feedback when an email is classified as "not spam." In recidivism prediction, a person whose is predicted to re-commit a crime may not be released. Accordingly, we will not know whether this person would have re-committed a crime had they been released. Situations like these are known formally as "apple tasting" [Helmbold et al., 2000a]. In the generic model, a learner observes a sequence of apples, some of which may be rotten. For each apple, the learner must decide whether to discard or taste the apple. The learner suffers a loss if they discard a good apple or if they taste a rotten apple. Crucially, when the learner discards an apple, they do not receive any feedback on whether the apple was rotten or not.

Binary online classification under apple tasting feedback was first studied by Helmbold et al. [2000a] in the realizable setting. Here, they give a simple and generic conversion of a deterministic online learner in the full-information setting into a randomized online learner in the apple tasting setting. In particular, they show that if M_+ and M_- are upper bounds on the number of false positive and false negative mistakes of the deterministic online learner respectively, then the expected number of mistakes made by their conversion, under apple tasting feedback, is at most $M_+ + 2\sqrt{TM_-}$. Along with these upper bounds, they provide lower bounds on the expected number of mistakes for randomized apple tasting learners in terms of the number of false positive and false negative mistakes made by any deterministic online learner in the full-information setting. That is, if there exists $M_+, M_- \in \mathbb{N}$ such that every deterministic online learner in the full-information setting makes either at least M_+ false positive mistakes or M_- false negative mistakes, then every randomized online learner makes at least $\frac{1}{2} \min\left\{\frac{1}{2}\sqrt{TM_-}, M_+\right\}$ expected number of mistakes under apple tasting feedback. Finally, as an open question, they ask whether their results can be extended to the harder agnostic setting where the true labels can be noisy.

While Helmbold et al. [2000a] establish bounds on the minimax expected number of mistakes in the realizable setting, their bounds are in terms of the existence of an algorithm with certain properties. This is in contrast to much of online learning theory, where minimax regret is often quantified in terms of combinatorial dimensions that capture the complexity of

the hypothesis class [Littlestone, 1987, Ben-David et al., 2009, Daniely et al., 2011, Rakhlin et al., 2015a]. Accordingly, we revisit apple tasting and study online learnability from a combinatorial perspective. In particular, we are interested in identifying combinatorial dimensions that tightly quantify the minimax regret for apple tasting in both the realizable and agnostic settings. To that end, our main contributions are:

(1) We close the open question posed by Helmbold et al. [2000a] by showing that the minimax expected regret in the agnostic setting, under apple tasting feedback, is at most $3\sqrt{\mathrm{L}(\mathcal{H})T\log(T)}$ and at least $\sqrt{\frac{\mathrm{L}(\mathcal{H})T}{8}}$, where $\mathrm{L}(\mathcal{H})$ is the Littlestone dimension of $\mathcal{H}$.

(2) On the other hand, we show that the Littlestone dimension alone does not give a tight quantitative characterization in the realizable setting. Instead, we show that the minimax expected number of mistakes in the realizable setting, under apple tasting feedback is

$$\Theta\Big(\max\Big\{\sqrt{(\mathrm{W}(\mathcal{H})-1)T}, 1\Big\}\Big),$$

where $\mathrm{W}(\mathcal{H})$ is the Effective width of $\mathcal{H}$, a new combinatorial parameter that accounts for the asymmetric nature of the feedback.

(3) Using the bound above, we establish the following trichotomy on the minimax rates in the realizable setting: (i) $\Theta(1)$ when $\mathrm{W}(\mathcal{H}) = 1$, (ii) $\Theta(\sqrt{T})$ when $1 < \mathrm{W}(\mathcal{H}) < \infty$, and (iii) $\Theta(T)$ when $\mathrm{W}(\mathcal{H}) = \infty$.

To prove (1), we extend the EXP3.G algorithm from Alon et al. [2015] to binary prediction with expert advice. Then, we use the standard technique from Ben-David et al. [2009] to construct an agnostic learner using a realizable, mistake-bound learner in the full-information setting. To prove the upper bound in (2), we define a new combinatorial parameter, called the Effective width, and use it to construct a deterministic online learner in the realizable, full-information feedback setting with constraints on the number of false positive and false negative mistakes. We then use this online learner and a conversion technique from Helmbold et al. [2000a] to construct a randomized online learner in the realizable, apple tasting feedback setting with the stated guarantee in (2). For the lower bound in (2), we consider a new combinatorial object called an apple tree and use it to explicitly construct a hard, realizable stream for any randomized, apple tasting learner. This is in contrast to Helmbold et al. [2000a], who prove lower bounds on the minimax expected number of mistakes by converting randomized apple tasting learners into deterministic full-information feedback learners.

6.1.1 Related Works

Apple tasting is usually presented as an example of a more general partial feedback setting called partial monitoring games, where the player's feedback is specified by a feedback matrix [Cesa-Bianchi and Lugosi, 2006, Bartók et al., 2014]. Of particular interest is the work by Bartók [2012], who gives a beautiful result (Theorem 2) characterizing the minimax rates in different partial monitoring games (including apple tasting). However, this is done for a slightly different setting where there is no hypothesis class $\mathcal{H}$, but just a finite set of actions the learner can play. The goal here is to compete with the best fixed action in hindsight. In contrast, in our setting, there is a hypothesis class, often infinite in size, and the goal is compete against the best fixed hypothesis in hindsight. Related to partial monitoring games is sequential prediction with graph feedback, for which apple tasting feedback is also special case [Alon et al., 2015]. In this model, a learner plays a repeated game against an adversary. In each round, the learner selects one of K actions but observes the losses for a subset of the actions determined by a combinatorial structure called a feedback graph. Alon et al. [2015] classify feedback graphs into three types and establish a trichotomy on the rates of the minimax regret based on the type of graph. In this chapter, we extend the online learner presented in Alon et al. [2015] to the setting of binary prediction with expert advice to establish the minimax regret of apple tasting in the agnostic setting.

In a parallel direction, there has been an explosion of work using combinatorial dimension to give tight quantitative characterizations of online learnability. For example, Littlestone [1987] proposed the Littlestone dimension and showed that it exactly characterizes the optimal mistake bound of deterministic learners for online binary classification in the full-information, realizable setting. Later, Ben-David et al. [2009] show that the Littlestone dimension also provides a tight quantitative characterization of the optimal expected regret in the full-information, agnostic setting. Later, Daniely et al. [2011] define a multiclass extension of the Littlestone dimension and show that it provides a tight quantitative characterization of realizable and agnostic mutliclass online learnability under full-information feedback when the label space is finite. In their same work, Daniely et al. [2011] define the Bandit Littlestone dimension and show that it exactly characterizes the optimal mistake bound of deterministic learners in the realizable setting under partial feedback setting known as bandit feedback. Daniely and Helbertal [2013] and Raman et al. [2024a] later show that the Bandit Littlestone dimension also characterizes agnostic bandit online learnability. Beyond binary and multiclass classification, combinatorial dimensions have been used to characterize online learnability for regression [Rakhlin et al., 2015a], list classification [Moran et al., 2023], ranking [Raman et al., 2023c], and general supervised online learning models [Raman et al., 2023b].

6.2 Preliminaries

6.2.1 Notation

Let $\mathcal{X}$ denote the instance space and $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ denote a binary hypothesis class. Given an instance $x \in \mathcal{X}$, and any collection of hypothesis $V \subseteq \{0,1\}^{\mathcal{X}}$, we let $V(x) := \{h(x) : h \in V\}$ denote the projection of V onto x. As usual, $[N]$ is used to denote $\{1, 2, \ldots, N\}$.

6.2.2 Online Classification and Apple Tasting

In the standard binary online classification setting with full-information feedback a learner $\mathcal{A}$ plays a repeated game against an adversary over T rounds. In each round $t \in [T]$, the adversary picks a labeled instance $(x_t, y_t) \in \mathcal{X} \times \{0,1\}$ and reveals x_t to the learner. The learner makes a (possibly randomized) prediction $\mathcal{A}(x_t) \in \{0,1\}$. Finally, the adversary reveals the true label y_t and the learner suffers the 0-1 loss $\mathbb{1}\{\mathcal{A}(x_t) \neq y_t\}$. Given a hypothesis class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$, the goal of the learner is to output predictions such that its expected regret

$$\mathrm{R}_{\mathcal{A}}(T, \mathcal{H}) := \sup_{(x_1,y_1),\ldots,(x_T,y_T)} \left(\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}(x_t) \neq y_t\} \right] - \inf_{h \in \mathcal{H}} \sum_{t=1}^{T} \mathbb{1}\{h(x_t) \neq y_t\} \right)$$

is small, where the expectation is only over the randomness of the learner. A hypothesis class $\mathcal{H}$ is said to be online learnable under full-information feedback, if there exists an (potentially randomized) online learning algorithm $\mathcal{A}$ such that $\mathrm{R}_{\mathcal{A}}(T, \mathcal{H}) = o(T)$ while $\mathcal{A}$ receives the true label y_t at the end of each round. If it is guaranteed that the learner always observes a sequence of examples labeled by some hypothesis $h \in \mathcal{H}$, then we say we are in the realizable setting and the goal of the learner is to minimize its expected cumulative mistakes,

$$\mathrm{M}_{\mathcal{A}}(T, \mathcal{H}) := \sup_{h \in \mathcal{H}} \sup_{x_1,\ldots,x_T} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}(x_t) \neq h(x_t)\} \right],$$

where again the expectation is taken only with respect to the randomness of the learner. In the apple tasting feedback model, the adversary still picks a labeled instance $(x_t, y_t) \in \mathcal{X} \times \{0,1\}$ and reveals x_t to the learner. However, the learner only gets to observe the true label y_t if they predict $\hat{y}_t = 1$. Analogous to the full-information setting, a hypothesis class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ is online learnable under apple tasting feedback, if there exists an online learning algorithm whose expected regret, under apple tasting feedback, on any sequence of labeled instances is $o(T)$.

Definition 6.2.1 (Agnostic Online Learnability under Apple Tasting Feedback)**.** *A hypothesis class $\mathcal{H}$ is online learnable under apple tasting feedback, if there exists an algorithm $\mathcal{A}$ such that $\mathrm{R}_{\mathcal{A}}(T, \mathcal{H}) = o(T)$ while $\mathcal{A}$ only receives feedback when predicting* 1*.*

As in the full-information setting, if it is guaranteed that the sequence of examples is labeled by some hypothesis $h \in \mathcal{H}$, then we say we are in the realizable setting and an analogous definition of learnability under apple tasting feedback follows.

Definition 6.2.2 (Realizable Online Learnability under Apple Tasting Feedback)**.** *A hypothesis class $\mathcal{H}$ is online learnable under apple tasting feedback in the realizable setting, if there exists an algorithm $\mathcal{A}$ such that $\mathrm{M}_{\mathcal{A}}(T, \mathcal{H}) = o(T)$ while $\mathcal{A}$ only receives feedback when predicting* 1*.*

6.2.3 Trees and Combinatorial Dimensions

In online learning, combinatorial dimensions are defined in terms of trees, a basic unit that captures temporal dependence. A binary tree $\mathcal{T}$ of depth d is complete if it admits the following recursive structure. A depth one complete binary tree is a single root node with left and right outgoing edges. A complete binary tree $\mathcal{T}$ of depth d has a root node whose left and right subtrees are each complete binary trees of depth $d-1$. Given a complete binary tree $\mathcal{T}$, we can label its internal nodes and edges by elements of $\mathcal{X}$ and $\{0,1\}$ respectively to get a Littlestone tree.

Definition 6.2.3 (Littlestone tree)**.** *A Littlestone tree of depth d is a complete binary tree of depth d where the internal nodes are labeled by instances of $\mathcal{X}$ and the left and right outgoing edges from each internal node are labeled by* 0 *and* 1 *respectively.*

Given a Littlestone tree $\mathcal{T}$ of depth d, a root-to-leaf path down $\mathcal{T}$ is a bitstring $\sigma \in \{0,1\}^d$ indicating whether to go left ($\sigma_i = 0$) or to go right ($\sigma_i = 1$) at each depth $i \in [d]$. A path $\sigma \in \{0,1\}^d$ down $\mathcal{T}$ gives a sequence of labeled instances $\{(x_i, \sigma_i)\}_{i=1}^d$, where x_i is the instance labeling the internal node following the prefix $(\sigma_1, ..., \sigma_{i-1})$ down the tree. A hypothesis $h_\sigma \in \mathcal{H}$ shatters a path $\sigma \in \{0,1\}^d$, if for every $i \in [d]$, we have $h_\sigma(x_i) = \sigma_i$. In other words, h_σ is consistent with the labeled examples when following σ. A Littlestone tree $\mathcal{T}$ is shattered by $\mathcal{H}$ if for every root-to-leaf path σ down $\mathcal{T}$, there exists a hypothesis $h_\sigma \in \mathcal{H}$ that shatters it. Using this notion of shattering, we define the Littlestone dimension of a hypothesis class.

Definition 6.2.4 (Littlestone dimension)**.** *The Littlestone dimension of $\mathcal{H}$, denoted $\mathrm{L}(\mathcal{H})$, is the largest $d \in \mathbb{N}$ such that there exists a Littlestone tree $\mathcal{T}$ of depth d shattered by $\mathcal{H}$. If there exists shattered Littlestone trees $\mathcal{T}$ of arbitrary depth, then we say that $\mathrm{L}(\mathcal{H}) = \infty$.*

Remarkably, the Littlestone dimension gives a tight quantitative characterization of realizable learnability under full-information feedback. In particular, Littlestone [1987] gives a generic deterministic algorithm, termed the Standard Optimal Algorithm (SOA), and shows that it makes at most $\mathrm{L}(\mathcal{H})$ number of mistakes on any realizable stream. Moreover, they showed that for every deterministic learner, there exists a realizable stream that can force at least $\mathrm{L}(\mathcal{H})$ mistakes, proving that the Ldim exactly quantifies the mistake bound for deterministic realizable learnability under full-information feedback.

Under apple tasting feedback, one can use the lower and upper bounds derived by Helmbold et al. [2000a] to deduce that the Ldim also provides a qualitative characterization of realizable learnability. However, unlike the full-information feedback setting, the Ldim alone cannot provide matching lower and upper bounds on the minimax expected number of mistakes under apple tasting feedback. Indeed, for the simple class of singletons over the natural numbers, $\mathcal{H}_{\text{sing}} := \{x \mapsto \mathbb{1}\{x = a\} : a \in \mathbb{N}\}$ we have that $\mathrm{L}(\mathcal{H}_{\text{sing}}) = 1$ while the minimax expected number of mistakes scales with the time horizon T (see Section 6.3.2). On the other hand, for the "flip" of the singletons, $\mathcal{H} = \{x \mapsto \mathbb{1}\{x \neq a\} : a \in \mathbb{N}\}$, we also have that $\mathrm{L}(\mathcal{H}) = 1$, but $\mathcal{H}$ is trivially learnable in at most 1 mistake in the realizable setting. Accordingly, new ideas are needed to handle the asymmetric nature of apple tasting feedback.

As a first step, we go beyond the symmetric nature of complete binary trees and define a new asymmetric binary tree called an apple tree. In particular, a binary tree $\mathcal{T}$ of depth d and width w is an apple tree if it admits the following recursive structure. An apple tree of width $w \geq d$ is a complete binary tree with depth d. An apple tree with width $w = 1$ and depth d is a degenerate binary tree of depth d where every internal node has only a left child. An apple tree $\mathcal{T}(w, d)$ of depth d and width $w < d$ has a root node v whose left subtree is an apple tree $\mathcal{T}(w, d-1)$ and whose right subtree is an apple tree $\mathcal{T}(w-1, d-1)$. At a high-level, the width of an apple tree w controls the number of ones any path starting from the root can have before the path ends. The depth d of an apple tree controls the maximum number of zeros along any path starting from the root. From this perspective, one can alternatively construct an apple tree of width w and depth d by starting with a complete binary tree of depth d and then trimming each path starting from the root node such that it ends once it contains w ones or until a leaf node has been reached. See Figure 6.1 for some examples of apple trees.

Similar to Littlestone trees, we can label the internal nodes of an apple tree with instances in $\mathcal{X}$ and the edges with elements of $\{0, 1\}$. By doing so, we get an Apple Littlestone (AL) tree.

Definition 6.2.5 (Apple Littlestone tree)**.** *An Apple Littlestone tree of width w and depth*

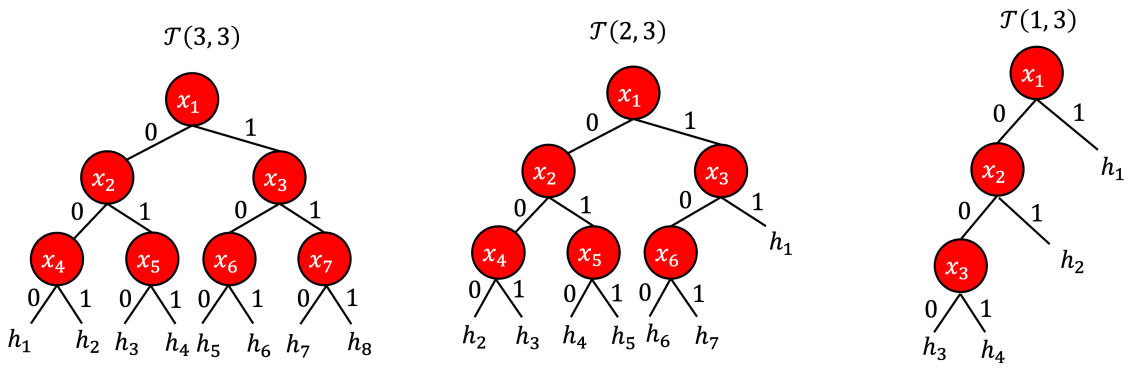

Figure 6.1: Shattered Apple Littlestone trees of (width, depth): $(3,3)$ (left), $(2,3)$ (middle), and $(1,3)$ (right).

d is an apple tree of width w and depth d where the internal nodes are labeled by instances of $\mathcal{X}$ and the left and right outgoing edges from each internal node are labeled by 0 and 1 respectively.

The notion of shattering for Littlestone trees extends exactly to AL trees. Formally, an AL tree $\mathcal{T}(w,d)$ of width w and depth d is shattered by $\mathcal{H}$, if for every path σ down the tree $\mathcal{T}$, there exists a hypothesis $h_\sigma \in \mathcal{H}$ consistent with $\{(x_i,\sigma_i)\}_{i=1}^{|\sigma|}$. Note that, unlike Littlestone trees, AL trees are imbalanced. In fact, for an AL tree $\mathcal{T}$ of width w and depth d, there can be at most w ones along any valid path σ down the tree before the path ends. Therefore, not all root-to-leaf paths are of the same length. Nevertheless, this notion of shattering is still well defined and naturally leads to a combinatorial dimension analogous to the Littlestone dimension.

Definition 6.2.6 (Apple Littlestone dimension)**.** *The Apple Littlestone dimension of $\mathcal{H}$ at width $w \in \mathbb{N}$, denoted $\mathrm{AL}_w(\mathcal{H})$, is the largest d such that there exists an apple tree $\mathcal{T}(w,d)$ of width w and depth d shattered by $\mathcal{H}$. If there exists shattered Apple Littlestone trees $\mathcal{T}$ with width w of arbitrarily large depth, then we say that $\mathrm{AL}_w(\mathcal{H}) = \infty$. If there are no shattered apple trees $\mathcal{T}$ of width w, then we say that $\mathrm{AL}_w(\mathcal{H}) = 0$.*

In general, the value of $\mathrm{AL}_w(\mathcal{H})$ for $w \leq \mathrm{L}(\mathcal{H})$ can be much larger than $\mathrm{L}(\mathcal{H})$. For example, even for the class of singletons defined over $\mathbb{N}$, we have that $\mathrm{AL}_1(\mathcal{H}_{\mathrm{sing}}) = \infty$ while $\mathrm{L}(\mathcal{H}_{\mathrm{sing}}) = 1$. Accordingly, unlike the Ldim, the Apple Littlestone dimension (ALdim), does not provide a qualitative characterization of learnability. Instead, using the ALdim, we define a new combinatorial parameter termed the Effective width. In Section 6.3, we show that the Effective width provides a tight quantitative characterization of realizable learnability under apple tasting feedback.

Definition 6.2.7 (Effective width)**.** *The Effective width of a hypothesis class $\mathcal{H}$, denoted $\mathrm{W}(\mathcal{H})$, is the smallest $w \in \mathbb{N}$ such that $\mathrm{AL}_w(\mathcal{H}) < \infty$. If there is no $w \in \mathbb{N}$ such that $\mathrm{AL}_w(\mathcal{H}) < \infty$, then we say that $\mathrm{W}(\mathcal{H}) = \infty$.*

The following lemma, whose proof is in Appendix D.2.1, establishes important properties of $\mathrm{AL}_w(\mathcal{H})$ and $\mathrm{W}(\mathcal{H})$ that we use to characterize learnability.

Lemma 6.2.1 (Structural Properties)**.** *For every $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$, the following statements are true.*

(i) $\mathrm{AL}_{w_1}(\mathcal{H}) \geq \mathrm{AL}_{w_2}(\mathcal{H})$ *for all* $w_1 < w_2$.

(ii) $\mathrm{AL}_w(\mathcal{H}) \geq \min\{w, \mathrm{L}(\mathcal{H})\}$.

(iii) $\mathrm{AL}_w(\mathcal{H}) = \mathrm{L}(\mathcal{H})$ *for all* $w \geq \mathrm{L}(\mathcal{H}) + 1$ *when* $\mathrm{L}(\mathcal{H}) < \infty$.

(iv) $\mathrm{W}(\mathcal{H}) \leq \mathrm{L}(\mathcal{H}) + 1$ *when* $\mathrm{L}(\mathcal{H}) < \infty$.

(v) $\mathrm{W}(\mathcal{H}) < \infty \iff \mathrm{L}(\mathcal{H}) < \infty$.

Property (iv) can be tight in the sense that for the class of singletons, $\mathrm{W}(\mathcal{H}_{\mathrm{sing}}) = 2$ while $\mathrm{L}(\mathcal{H}_{\mathrm{sing}}) = 1$. Moreover, one cannot hope to lower bound $\mathrm{W}(\mathcal{H})$ in terms of $\mathrm{L}(\mathcal{H})$. Indeed, for any finite hypothesis class $\mathcal{H}$, we have that $\mathrm{W}(\mathcal{H}) = 1$ while $\mathrm{L}(\mathcal{H})$ can be made arbitrarily large.

6.3 Realizable Learnability

In this section, we revisit the learnability of apple tasting in the realizable setting, first studied by Helmbold et al. [2000a]. Our main result is Theorem 6.3.1, which lower- and upper bounds the minimax expected number of mistakes in terms of the Littlestone dimension and the Effective width.

Theorem 6.3.1 (Realizable Learnability)**.** *For any hypothesis class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$,*

$$\frac{1}{8}\min\Big\{\max\Big\{\sqrt{(\mathrm{W}(\mathcal{H})-1)\,T}, \mathrm{L}(\mathcal{H})\Big\}, T\Big\} \leq \inf_{\mathcal{A}} \mathrm{M}_{\mathcal{A}}(T, \mathcal{H}) \leq \mathrm{AL}_{\mathrm{W}(\mathcal{H})}(\mathcal{H}) + 2\sqrt{(\mathrm{W}(\mathcal{H})-1)T}.$$

The lower and upper bounds of Theorem 6.3.1 can be tight up to constant factors. There are two cases to consider. When $\mathrm{W}(\mathcal{H}) = 1$, the lower and upper bounds in Theorem 6.3.1 reduce to $\frac{\mathrm{L}(\mathcal{H})}{8} \leq \inf_{\mathcal{A}} \mathrm{M}_{\mathcal{A}}(T, \mathcal{H}) \leq \mathrm{AL}_1(\mathcal{H})$ for $T \geq \mathrm{L}(\mathcal{H})$. Taking $|\mathcal{X}| = d < \infty$ and $\mathcal{H} = \{0,1\}^{\mathcal{X}}$ gives that $\mathrm{L}(\mathcal{H}) = \mathrm{AL}_1(\mathcal{H}) = d$, ultimately implying that the lower- and upper bounds can be off by only a constant factor of $\frac{1}{8}$. Secondly, consider the case

where $\mathrm{W}(\mathcal{H}) \geq 2$. Then, if $T \geq \max\{\mathrm{W}(\mathcal{H}) - 1, \mathrm{AL}^2_{\mathrm{W}(\mathcal{H})}(\mathcal{H})\}$, Theorem 6.3.1 implies that $\frac{1}{8}\sqrt{(\mathrm{W}(\mathcal{H}) - 1)\,T} \leq \inf_{\mathcal{A}} \mathrm{M}_{\mathcal{A}}(T, \mathcal{H}) \leq 3\sqrt{(\mathrm{W}(\mathcal{H}) - 1)T}$, showing that the upper- and lower bounds are off only by a constant factor.

Theorem 6.3.1 implies that when $\mathrm{W}(\mathcal{H}) = 1$, a constant upper bound on the expected regret is possible. In fact, when $\mathrm{AL}_1(\mathcal{H}) < \infty$, there exists a deterministic online learner which makes at most $\mathrm{AL}_1(\mathcal{H})$ mistakes in the realizable setting under apple tasting feedback (see Appendix D.1). On the other hand, Theorem 6.3.1 also shows that, in full generality, it is not possible to achieve a constant expected mistake bound under apple tasting feedback in the realizable setting. Indeed, if $\mathrm{W}(\mathcal{H}) > 1$, then the worst-case expected mistakes of any randomized learner, under apple tasting feedback, is at least $\Omega(\sqrt{T})$. This is in contrast to the full-information setting, where the minimax expected number of mistakes in the realizable setting is constant, and that too achieved by a deterministic learner (i.e SOA). Accordingly, Theorem 6.3.1 gives a trichotomy in the minimax expected number of mistakes for the realizable setting.

Corollary 6.3.2 (Trichotomy in minimax expected number of mistakes)**.** *For any hypothesis class* $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$*,*

$$\inf_{\mathcal{A}} \mathrm{M}_{\mathcal{A}}(T, \mathcal{H}) = \begin{cases} \Theta(1), & \text{if } \mathrm{W}(\mathcal{H}) = 1. \\ \Theta(\sqrt{T}) & \text{if } 2 \leq \mathrm{W}(\mathcal{H}) < \infty. \\ \Theta(T), & \mathrm{W}(\mathcal{H}) = \infty. \end{cases}$$

In Section 6.4, we will show that $\inf_{\mathcal{A}} \mathrm{R}_{\mathcal{A}}(T, \mathcal{H}) = \tilde{\Theta}(\sqrt{T})$, where $\tilde{\Theta}$ hides poly-logarithmic factors in T. With this in mind, Corollary 6.3.2 shows that when $\mathrm{W}(\mathcal{H}) \geq 2$, realizable learnability under apple tasting feedback can be as hard as agnostic learnability under apple tasting feedback. Unfortunately, for many simple classes, like the singletons over $\mathbb{N}$, we have $\mathrm{W}(\mathcal{H}) \geq 2$. On the other hand, for classes containing hypothesis that rarely output 0, like the "flip" of the class of singletons, realizable learnability under apple tasting feedback can be as easy as realizable learnability under full-information feedback.

6.3.1 Upper Bounds for Randomized Learners in the Realizable Setting

We prove a slightly stronger upper bound than the one stated in Theorem 6.3.1.

Lemma 6.3.3 (Randomized Realizable Upper Bound)**.** *For any hypothesis class* $\mathcal{H} \subseteq$

$\{0,1\}^{\mathcal{X}}$,

$$\inf_{\mathcal{A}} \mathrm{M}_{\mathcal{A}}(T,\mathcal{H}) \leq \inf_{w\in\mathbb{N}} \left\{ \mathrm{AL}_w(\mathcal{H}) + 2\sqrt{(w-1)T} \right\}.$$

The upper bound in Theorem 6.3.1 follows by picking $w = \mathrm{W}(\mathcal{H})$. If one picks $w = \mathrm{L}(\mathcal{H}) + 1$, then $\mathrm{AL}_w(\mathcal{H}) = \mathrm{L}(\mathcal{H})$ and we get an upper bound of $3\sqrt{\mathrm{L}(\mathcal{H})T}$ on the expected mistakes.

Lemma 6.3.3 follows from composing the next two lemmas. Lemma 6.3.4 shows that if $\mathrm{AL}_w(\mathcal{H}) < \infty$, then there exists a deterministic online learner, under full-information feedback, that makes at most $w-1$ false negative mistakes and at most $\mathrm{AL}_w(\mathcal{H})$ false positive mistakes. Lemma 6.3.5 is from Helmbold et al. [2000a] and shows how to convert any online learner under full-information feedback into an online learner under apple tasting feedback.

Lemma 6.3.4. *For any $\mathcal{H} \subset \{0,1\}^{\mathcal{X}}$ and $w \in \mathbb{N}$ such that $\mathrm{AL}_w(\mathcal{H}) < \infty$, there exists a deterministic online learner which, under full-information feedback, makes at most $w-1$ false negative mistakes and at most $\mathrm{AL}_w(\mathcal{H})$ false positive mistakes in the realizable setting.*

Proof. Suppose $w \in \mathbb{N}$ such that $\mathrm{AL}_w(\mathcal{H}) < \infty$ and denote $\mathcal{A}$ to be Algorithm 7.

Algorithm 7 Realizable Algorithm Under Full-Information Feedback

Require: $V_1 = \mathcal{H}$, pick $w_1 = w$ such that $\mathrm{AL}_w(\mathcal{H}) < \infty$

for $t = 1, ..., T$ **do**
  Receive x_t.
  For each $y \in \{0,1\}$, define $V_t^y = \{h \in V_t \mid h(x_t) = y\}$.
  if $V_t(x_t) = \{y\}$ **then**
    Predict $\hat{y}_t = y$.
  else
    If $|V_t^1| \geq 1$, and $\mathrm{AL}_{w_t}(V_t^0) < \mathrm{AL}_{w_t}(V_t)$, predict $\hat{y}_t = 1$.
    Otherwise, predict $\hat{y}_t = 0$.
  end if
  Receive y_t and update $V_t \leftarrow V_t^{y_t}$.
  If $\hat{y}_t = 0$ and $y_t = 1$, then update $w_{t+1} \leftarrow w_t - 1$. Else, set $w_{t+1} \leftarrow w_t$.
end for

Let $(x_1, y_1), ..., (x_T, y_T)$ be the stream to be observed by $\mathcal{A}$. We show that $\mathcal{A}$, initialized at $w_1 = w$, makes at most $\mathrm{AL}_w(\mathcal{H})$ false positive mistakes and at most $w-1$ false negative mistakes. Let $S_+ = \{t \in [T] \mid \hat{y}_t = 1 \text{ and } y_t = 0\}$ be the set of time points where $\mathcal{A}$ makes false positive mistakes, and $S_- = \{t \in [T] \mid \hat{y}_t = 0 \text{ and } y_t = 1\}$ be the set of time points where $\mathcal{A}$ makes false negative mistakes. We show $|S_+| \leq \mathrm{AL}_w(\mathcal{H})$ by first establishing

$$\mathrm{AL}_{w_{t+1}}(V_{t+1}) \leq \mathrm{AL}_{w_t}(V_t) - \mathbb{1}\{t \in S_+\}, \qquad \forall t \in [T]. \tag{6.1}$$

This inequality then implies that the number of false positive mistakes of $\mathcal{A}$ is

$$\begin{aligned}\sum_{t=1}^{T} \mathbb{1}\{t \in S_+\} &\leq \sum_{t=1}^{T} \left(\mathrm{AL}_{w_t}(V_t) - \mathrm{AL}_{w_{t+1}}(V_{t+1})\right) \\ &= \mathrm{AL}_{w_1}(V_1) - \mathrm{AL}_{w_{T+1}}(V_{T+1}) \\ &\leq \mathrm{AL}_{w_1}(V_1) = \mathrm{AL}_w(\mathcal{H}).\end{aligned}$$

To prove inequality (6.1), we consider the two cases: $t \in S_+$ and $t \notin S_+$. Suppose $t \in S_+$. Then, we know that $\hat{y}_t = 1$ and by the prediction rule of $\mathcal{A}$, we must have $\mathrm{AL}_{w_t}(V_t^0) < \mathrm{AL}_{w_t}(V_t)$. Since $y_t = 0$, we further obtain that $V_{t+1} = V_t^0$ and $w_{t+1} = w_t$ in this case. This yields $\mathrm{AL}_{w_{t+1}}(V_{t+1}) < \mathrm{AL}_{w_t}(V_t)$, which subsequently implies $\mathrm{AL}_{w_{t+1}}(V_{t+1}) \leq \mathrm{AL}_{w_t}(V_t) - \mathbb{1}\{t \in S_+\}$.

Now, let us consider the case when $t \notin S_+$. In the case when $t \notin S_+ \cup S_-$, we have $w_{t+1} = w_t$ and $\mathbb{1}\{t \in S_+\} = 0$. Thus, we trivially obtain $\mathrm{AL}_{w_{t+1}}(V_{t+1}) \leq \mathrm{AL}_{w_t}(V_t) - \mathbb{1}\{t \in S_+\}$ since $V_{t+1} \subseteq V_t$. Next, let us consider the case when $t \in S_-$. In this case, we have $w_{t+1} = w_t - 1$, $V_t = V_t^1$, and $\mathbb{1}\{t \in S_+\} = 0$. Thus, to establish inequality (6.1), it suffices to show that $\mathrm{AL}_{w_t-1}(V_t^1) \leq \mathrm{AL}_{w_t}(V_t)$. Suppose, for the sake of contradiction, this is not true and we instead have $\mathrm{AL}_{w_t-1}(V_t^1) > \mathrm{AL}_{w_t}(V_t)$. Let $d := \mathrm{AL}_{w_t}(V_t)$. Note that $d > 0$ because there must exist $h_1, h_2 \in V_t$ such that $h_1(x_t) \neq h_2(x_t)$ or otherwise $\mathcal{A}$ would not have made a false negative mistake. Since $\mathrm{AL}_{w_t-1}(V_t^1) > d$, we are guaranteed the existence of an AL tree $\mathcal{T}_1(w_t - 1, d)$ shattered by V_t^1. Furthermore, as $\hat{y}_t = 0$ and $|V_t^1| \geq 1$, the prediction rule implies that $\mathrm{AL}_{w_t}(V_t^0) \geq \mathrm{AL}_{w_t}(V_t) = d$. Accordingly, we are also guaranteed the existence of an AL tree $\mathcal{T}_0(w_t, d)$ shattered by V_t^0. Now consider an AL tree $\mathcal{T}$ that has x_t in its root-node, has a subtree $\mathcal{T}_0(w_t, d)$ attached to left-outgoing edge from the root-node and has a subtree $\mathcal{T}_1(w_t - 1, d)$ attached to right-outgoing edge from the root-node. Since all hypotheses in V_t^0 output 0 on x_t and all hypotheses in V_t^1 output 1 on x_t, the tree $\mathcal{T}$ shattered by V_t. Since $\mathcal{T}$ is a valid AL tree of width w_t and depth $d + 1$, we have that $\mathrm{AL}_{w_t}(V_t) \geq d + 1$, a contradiction to our assumption that $\mathrm{AL}_{w_t}(V_t) = d$. Therefore, we must have $\mathrm{AL}_{w_t-1}(V_t^1) \leq \mathrm{AL}_{w_t}(V_t)$ when $t \in S_-$.

Next, we show that $\mathcal{A}$ makes at most $w - 1$ false negative mistakes. Let $t^\star \in [T]$ be the time point where the algorithm makes its $(w-1)$-th false negative mistake. If such time point $t^\star$ does not exist, then we trivially have $|S_-| \leq w - 2 < w - 1$. We now consider the case when $t^\star \in [T]$ exists. It suffices to show that, $\forall t > t^\star$, we have $t \notin S_-$. Suppose, for the sake of contradiction, $\exists\, t > t^\star$ such that $t \in S_-$. Since $\hat{y}_t = 0$ and $y_t = 1$, we must have $|V_t^1| \geq 1$. Thus, the prediction strategy implies that $\mathrm{AL}_{w_t}(V_t^0) \geq \mathrm{AL}_{w_t}(V_t)$. Given that $t > t^\star$ and $\mathcal{A}$ has already made $w - 1$ false negative mistakes, we must have $w_t = 1$. Thus,

we have $\mathrm{AL}_1(V_t^0) \geq \mathrm{AL}_1(V_t) =: d$. Note that $d \geq 1$ because there must exist $h_1, h_2 \in V_t$ such that $h_1(x_t) \neq h_2(x_t)$. Since $\mathrm{AL}_1(V_t^0) \geq d$, we are guaranteed the existence of an AL tree $\mathcal{T}_0(1, d)$ of width 1 and depth d shattered by V_t^0. Next, consider a tree $\mathcal{T}$ with x_t on the root node and has a subtree $\mathcal{T}_0(1, d)$ attached to the left-outgoing edge from the root node. Let $h \in V_t$ any hypothesis such that $h(x_t) = 1$. The hypothesis h must exist because $|V_t^1| \geq 1$. By putting h in the leaf node following the right-outgoing edge from the root node in $\mathcal{T}$, it is clear that $\mathcal{T}$ is a valid AL tree of width 1 and depth $d+1$ shattered by V_t. The existence of $\mathcal{T}$ implies that $\mathrm{AL}_1(V_t) \geq d+1$, a contradiction to our assumption $\mathrm{AL}_1(V_t) = d$. Thus, $\forall t > t^\star$, we have $t \notin S_-$. Therefore, $\mathcal{A}$ makes no more than $w-1$ false negative mistakes. ■

We remark that Helmbold et al. [2000b] also give a deterministic online learner in the full-information setting under constraints on the number of false positive and false negative mistakes (see Algorithm SCS in [Helmbold et al., 2000b, Section 2]). However, similar to Helmbold et al. [2000a], their algorithm checks the existence of an online learning algorithm satisfying certain properties. We extend on this result by giving an SOA-type algorithm that only requires computing combinatorial dimensions.

Lemma 6.3.5 is the restatement of Corollary 2 in Helmbold et al. [2000a]. For completeness sake, we provide a proof in Appendix D.2.2. Lemma 6.3.3 follows by composing Lemma 6.3.4 and Lemma 6.3.5.

Lemma 6.3.5 (Helmbold et al. [2000a])**.** *For any $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$, if there exists a deterministic learner which, under full-information feedback, makes at most M_- false negative mistakes and at most M_+ false positive mistakes, then there exists a randomized learner, whose expected number of mistakes, under apple tasting feedback, is at most $M_+ + 2\sqrt{TM_-}$ in the realizable setting.*

6.3.2 Lower Bounds for Randomized Learners in the Realizable Setting

As in the upper bound, we prove a slightly stronger lower bound than the one stated in Theorem 6.3.1.

Lemma 6.3.6 (Realizable Lower Bound)**.** *For any hypothesis class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$,*

$$\inf_{\mathcal{A}} \mathrm{M}_{\mathcal{A}}(T, \mathcal{H}) \geq \frac{1}{8} \sup_{w \in \mathbb{N}} \sqrt{\min\{w, \mathrm{L}(\mathcal{H}), T\} \, \min\{\mathrm{AL}_w(\mathcal{H}), T\}}.$$

The lower bounds in Theorem 6.3.1 follows by picking $w = \mathrm{W}(\mathcal{H}) - 1$ and $w = \mathrm{L}(\mathcal{H}) + 1$ respectively. When $w = \mathrm{W}(\mathcal{H}) - 1$, we have that $\min\{w, \mathrm{L}(\mathcal{H}), T\} = \min\{\mathrm{W}(\mathcal{H}) - 1, T\}$

and $\min\{\mathrm{AL}_w(\mathcal{H}), T\} \geq \min\{\mathrm{W}(\mathcal{H}) - 1, T\}$ using Lemma 6.2.1 (ii) and (iv).

On the other hand, when $w = \mathrm{L}(\mathcal{H}) + 1$, we have that $\min\{\mathrm{AL}_w(\mathcal{H}), T\} = \min\{\mathrm{L}(\mathcal{H}), T\}$ using Lemma 6.2.1 (iii).

Proof. Let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$, $w \in \mathbb{N}$, and $T \in \mathbb{N}$ be the time horizon. Since learning under apple tasting feedback implies learning under full-information feedback, a lower bound of $\frac{\min\{T,\mathrm{L}(\mathcal{H})\}}{2}$ on the minimax expected number of mistakes follows trivially from the full-information feedback lower bound. Accordingly, for the remainder of the proof we suppose $w \leq \min\{\mathrm{L}(\mathcal{H}), T\}$, since if this condition is not met, the claimed lower bound is at most $\frac{\min\{T,\mathrm{L}(\mathcal{H})\}}{2}$.

Let $\mathcal{T}$ be any AL tree of width w of depth $d = \left\lfloor \sqrt{w \min\{T, \mathrm{AL}_w(\mathcal{H})\}} \right\rfloor$ shattered by $\mathcal{H}$. Such a tree must exist because $d \leq \mathrm{AL}_w(\mathcal{H})$. Let $\mathcal{A}$ be any randomized apple tasting online learner. Our goal will be to construct a hard, deterministic, realizable stream of instances $(x_1, y_1), ..., (x_T, y_T)$ such that $\mathcal{A}$'s expected regret is at least $\frac{d}{4}$.

We first construct a path $\sigma^\star$ down $\mathcal{T}$ recursively using $\mathcal{A}$. Starting with $\sigma_1^\star$, let A_1 be the event that $\mathcal{A}$, if presented with $\left\lfloor \frac{d}{w} \right\rfloor$ copies of the root node $x_1^\star$, predicts 1 on at least one of the copies. Then, set $\sigma_1^\star = 0$ if $\mathbb{P}(A_1) \geq \frac{1}{2}$ and set $\sigma_1^\star = 1$ otherwise. For $j \geq 2$, let $x_1^\star, ..., x_j^\star$ be the sequence of instances labeling the internal nodes along the prefix $(\sigma_1^\star, ..., \sigma_{j-1}^\star)$ down $\mathcal{T}$. Let A_j be the event that $\mathcal{A}$, if simulated with the sequence of $(j-1)\left\lfloor \frac{d}{w} \right\rfloor$ labeled instances consisting of $\left\lfloor \frac{d}{w} \right\rfloor$ copies of the labeled instance $(x_1^\star, \sigma_1^\star)$, followed by $\left\lfloor \frac{d}{w} \right\rfloor$ copies of the labeled instance $(x_2^\star, \sigma_2^\star)$,..., followed by $\left\lfloor \frac{d}{w} \right\rfloor$ copies of the labeled instance $(x_{j-1}^\star, \sigma_{j-1}^\star)$, predicts the label 1 at least once when presented with $\left\lfloor \frac{d}{w} \right\rfloor$ copies of the instance $x_j^\star$. Set $\sigma_j^\star = 0$ if $\mathbb{P}(A_j) \geq \frac{1}{2}$ and set $\sigma_j^\star = 1$ otherwise. Continue this process until $\sigma^\star$ is a valid path that reaches the end of tree $\mathcal{T}$.

We now construct our hard, labeled stream in blocks of size $\left\lfloor \frac{d}{w} \right\rfloor$. Each block only contains a single labeled instance, repeated $\left\lfloor \frac{d}{w} \right\rfloor$ times. For the first block B_1, repeat the labeled instance $(x_1^\star, \sigma_1^\star)$. Likewise, for block B_j for $2 \leq j \leq |\sigma^\star|$, repeat for $\left\lfloor \frac{d}{w} \right\rfloor$ times the labeled instance $(x_j^\star, \sigma_j^\star)$. Now, consider the stream $S = (B_1, ..., B_{|\sigma^\star|})$ obtained by concatenating the blocks $B_1, ..., B_{|\sigma^\star|}$ in that order. If $|\sigma^\star| \left\lfloor \frac{d}{w} \right\rfloor < T$, populate the rest of the stream S with the labeled instance $(x_{|\sigma^\star|}^\star, \sigma_{|\sigma^\star|}^\star)$.

We first claim that such a stream is realizable by $\mathcal{H}$. This follows trivially from the fact that (1) $\sigma^\star$ is a valid path down $\mathcal{T}$, (2) by the definition of shattering, there exists a hypothesis $h \in \mathcal{H}$ such that for all $j \in [|\sigma^\star|]$, we have $h(x_j^\star) = \sigma_j^\star$ and (3) our stream S only contains labeled instances from the set $\{(x_j^\star, \sigma_j^\star)\}_j$. We now claim that $\mathcal{A}$'s expected regret on the stream S is at least $\frac{d}{4}$. To see this, observe that whenever $\sigma_j^\star = 1$, $\mathcal{A}$'s expected mistakes on the block B_j is at least $\frac{1}{2}\left\lfloor \frac{d}{w} \right\rfloor$ since $\mathcal{A}$ gets passed the labeled instance $(x_j^\star, 1)$ for $\left\lfloor \frac{d}{w} \right\rfloor$ iterations, but the probability that it never predicts 1 on this batch after seeing

$B_1, ..., B_{j-1}$ is $\mathbb{P}(A_j^c) \geq \frac{1}{2}$. Likewise, whenever $\sigma_j^\star = 0$, $\mathcal{A}$'s expected mistakes on the block B_j is at least $\frac{1}{2}$ since it gets passed the labeled instance $(x_j^\star, 0)$ for $\left\lfloor \frac{d}{w} \right\rfloor$ time points but predicts 1 on at least one of them with probability $\mathbb{P}(A_j) \geq \frac{1}{2}$.

We now lower bound the expected mistakes of $\mathcal{A}$ on the entire stream S by considering the number of ones in $\sigma^\star$ on a case by case basis. Note that since $\sigma^\star$ is a valid path down $\mathcal{T}$, we have $w \leq |\sigma^\star| \leq d$. Consider the case where $\sigma^\star$ has w ones. Then, $\mathcal{A}$'s expected regret is at least its expected regret on those batches B_j where $\sigma_j^\star = 1$. Thus, its expected regret is at least $\frac{w}{2} \left\lfloor \frac{d}{w} \right\rfloor \geq \frac{w}{2} \frac{d}{2w} \geq \frac{d}{4}$. Consider the case where $\sigma^\star$ has $w - j$ ones for $w \geq j \geq 1$. Then, since $\sigma^\star$ is a valid path, it must be the case that there are $d - (w - j)$ zero's in $\sigma^\star$. Therefore, $\mathcal{A}$'s expected regret is at least

$$\frac{(w-j)}{2} \left\lfloor \frac{d}{w} \right\rfloor + \frac{d - w + j}{2} \geq \frac{d}{2} - \frac{w-j}{2} + \frac{w-j}{2} \left\lfloor \frac{d}{w} \right\rfloor \geq \frac{d}{2}.$$

where the last inequality follows from the fact that $d \geq w$. Thus, in all cases, $\mathcal{A}$'s expected regret is at least $\frac{d}{4}$. The claimed lower bound follows by using the fact that $d \geq \sqrt{w \min\{T, \mathrm{AL}_w(\mathcal{H})\}}/2$. ■

6.4 Agnostic Learnability

We show that the Ldim quantifies the minimax expected regret in the agnostic setting under apple tasting feedback, closing the open problem posed by [Helmbold et al., 2000a, Page 138].

Theorem 6.4.1 (Agnostic Learnability)**.** *For any hypothesis class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$,*

$$\sqrt{\frac{\mathrm{L}(\mathcal{H})T}{8}} \leq \inf_{\mathcal{A}} \mathrm{R}_{\mathcal{A}}(T, \mathcal{H}) \leq 3\sqrt{\mathrm{L}(\mathcal{H})T \ln T}.$$

The lower bound in Theorem 6.4.1 follows directly from the full-information lower bound in the agnostic setting [Ben-David et al., 2009]. Therefore, in this section, we only focus on proving the upper bound. Our strategy will be in two steps. First, we modify the celebrated Randomized Exponential Weights Algorithm (REWA) [Cesa-Bianchi and Lugosi, 2006] to handle apple tasting feedback by using the ideas from Alon et al. [2015]. In particular, our algorithm EXP4.AT is an adaptation of EXP3.G from Alon et al. [2015] to binary prediction with expert advice under apple tasting feedback. Second, we give an agnostic online learner which uses the SOA to construct a finite set of experts that exactly covers $\mathcal{H}$ and then runs EXP4.AT using these experts. The upper bound in Theorem 6.4.1 follows immediately from the composition of these two results.

6.4.1 The EXP4.AT Algorithm

In this subsection, we present EXP4.AT, an adaptation of REWA to handle apple tasting feedback.

Algorithm 8 EXP4.AT: online learning with apple tasting feedback

Require: Learning rate $\eta \in (0, \frac{1}{2})$

Let q_1 be the uniform distribution over $[N]$
for $t = 1, ..., T$ **do**
  Get advice $\mathcal{E}_t^1, ..., \mathcal{E}_t^N \in \{0,1\}^N$
  Compute $p_t^1 = (1-\eta)\sum_{i=1}^N q_t^i \mathcal{E}_t^i + \eta$
  Predict $\hat{y}_t = 1$ with probability p_t^1 and $\hat{y}_t = 0$ with probability $p_t^0 = 1 - p_t^1$
  Observe true label y_t if $\hat{y}_t = 1$ and let $\hat{\ell}_t(y) = \frac{\mathbb{1}\{y \neq y_t\}\mathbb{1}\{\hat{y}_t = 1\}}{p_t^1}$
  For $i = 1, ..., N$ update $q_{t+1}^i = \frac{q_t^i \exp(-\eta \hat{\ell}_t(\mathcal{E}_t^i))}{\sum_{j=1}^N q_t^j \exp(-\eta \hat{\ell}_t(\mathcal{E}_t^j))}$
end for

Theorem 6.4.2 (EXP4.AT Regret Bound)**.** *If $\eta = \sqrt{\frac{\ln N}{2T}}$, then for any sequence of true labels $y_1, ..., y_T$, the predictions $\hat{y}_1, ..., \hat{y}_T$, output by EXP4.AT satisfy:*

$$\mathbb{E}\left[\sum_{t=1}^T \mathbb{1}\{\hat{y}_t \neq y_t\}\right] \leq \inf_{j \in [N]} \sum_{t=1}^T \mathbb{1}\{\mathcal{E}_t^j \neq y_t\} + 3\sqrt{T \ln N}.$$

In order to prove Theorem 6.4.2, we need the following lemma which gives a second-order regret bound for the EXP4.AT algorithm. The proof of Lemma 6.4.3 follows a similar potential-function strategy as in the proof of Lemma 4 in Alon et al. [2015] and can be found in Appendix D.2.3.

Lemma 6.4.3 (EXP4.AT Second-order Regret Bound)**.** *For any $\eta \in (0, \frac{1}{2})$ and any sequence of true labels $y_1, ..., y_T$, the probabilities $p_1, ..., p_T$ output by EXP4.AT satisfy*

$$\sum_{t=1}^T \sum_{y \in \{0,1\}} p_t^y \hat{\ell}_t(y) - \inf_{j \in [N]} \sum_{t=1}^T \hat{\ell}_t(\mathcal{E}_t^j) \leq \frac{\ln N}{\eta} + \eta \sum_{t=1}^T \hat{\ell}_t(1) + \eta \sum_{t=1}^T p_t^1 (1 - p_t^1) \hat{\ell}_t(0)^2 + \eta \sum_{t=1}^T p_t^1 \hat{\ell}_t(1)^2.$$

Theorem 6.4.2 follows by taking expectations of both sides of the inequality in Lemma 6.4.3. The full proof can be found in Appendix D.2.4.

6.4.2 Proof Sketch of Theorem 6.4.1

Given any hypothesis class $\mathcal{H}$, we construct an agnostic online learner under apple tasting feedback with the claimed upper bound on expected regret. Similar to the generic agnostic

online learner in the full-information setting [Ben-David et al., 2009], the high-level strategy is to use the SOA to construct a small set of experts E such that $|E| \leq T^{\mathrm{L}(\mathcal{H})}$ and for every $h \in \mathcal{H}$, there exists an expert $\mathcal{E}_h \in E$ such that $\mathcal{E}_h(x_t) = h(x_t)$ for all $t \in [T]$. Then, our agnostic online learner will run EXP4.AT using this set of experts E. The upper bound in Theorem 6.4.1 immediately follows from the guarantee of EXP4.AT in Theorem 6.4.2 and the fact that we have constructed an exact cover of $\mathcal{H}$. The full proof of Theorem 6.4.1 can be found in Appendix D.2.5.

CHAPTER 7

On the Learnability of Multilabel Ranking

Multilabel ranking is a central task in machine learning. However, the most fundamental question of learnability in a multilabel ranking setting with relevance-score feedback remains unanswered. In this chapter, we characterize the learnability of multilabel ranking problems in both batch and online settings for a large family of ranking losses. Along the way, we give two equivalence classes of ranking losses based on learnability that capture most losses used in practice.

7.1 Introduction

Multilabel ranking is a supervised learning problem where a learner is presented with an instance $x \in \mathcal{X}$ and is required to output a ranking of K different labels in decreasing order of relevance to x. This is in contrast with *multilabel classification* where given an instance $x \in \mathcal{X}$, the learner is tasked with predicting a subset of the K labels without any explicit ordering. Multilabel ranking is a canonical learning problem with a wide range of applications to text categorization, genetics, medical imaging, social networks, and visual object recognition [Joachims, 2005, Schapire and Singer, 2000, McCallum, 1999, Clare and King, 2001, Baltruschat et al., 2019, Wang and Sukthankar, 2013, Bucak et al., 2009, Yang et al., 2016]. Recent years have seen a surge in the development of multilabel ranking methods with strong practical and theoretical guarantees [Schapire and Singer, 2000, Dembczynski et al., 2012, Gong et al., 2013, Bucak et al., 2009, Jung and Tewari, 2018, Gao and Zhou, 2011, Koyejo et al., 2015, Zhang and Zhou, 2013, Korba et al., 2018]. A related line of work has studied consistency for the convex surrogates of natural ranking losses [Duchi et al., 2010, Buffoni et al., 2011, Gao and Zhou, 2011, Ravikumar et al., 2011, Calauzenes et al., 2012, Dembczynski et al., 2012]. Despite this vast literature on multilabel ranking, the fundamental question of when a multilabel ranking problem is *learnable* remains unanswered.

Understanding when a hypothesis class is learnable is a fundamental question in Statistical Learning Theory. For binary classification, the finiteness of the Vapnik–Chervonenkis (VC) dimension is both sufficient and necessary for Probably Approximately Correct (PAC) learning [Vapnik and Chervonenkis, 1974a, Valiant, 1984]. Likewise, the finiteness of the Daniely-Shwartz (DS) dimension characterizes multiclass PAC learnability Daniely and Shalev-Shwartz [2014], Brukhim et al. [2022]. In the online setting, the Littlestone dimension [Littlestone, 1987] characterizes the online learnability of a binary hypothesis class and the multiclass Littlestone dimension [Daniely et al., 2011] characterizes online multiclass learnability. Unlike classification, a distinguishing property of multilabel ranking is the mismatch between the predictions the learner makes and the feedback it receives. In particular, a learner is required to produce a permutation that ranks the relevance of the labels but only receives a *relevance-score vector* as feedback. This feedback model is standard in multilabel ranking since obtaining full permutation feedback is generally costly [Liu et al., 2009]. As a result, unlike the 0-1 loss in classification, there is no canonical loss function in ranking. Together, these two issues create barriers for existing techniques used to prove learnability, such as the agnostic-to-realizable reductions from Hopkins et al. [2022] and Raman et al. [2023a], to readily extend to ranking.

In this chapter, we characterize the batch and online learnability of a ranking hypothesis

class $\mathcal{H} \subseteq \mathcal{S}_K^{\mathcal{X}}$ under relevance-score feedback, where $\mathcal{S}_K$ is the set of all permutations over $[K] = \{1, ..., K\}$. In doing so, we make the following contributions.

- We show that a ranking hypothesis class $\mathcal{H}$ embeds K^2 different *binary* hypothesis classes $\mathcal{H}_i^j$ for $i, j \in [K]$, where hypotheses in $\mathcal{H}_i^j$ answer whether the label i should be ranked in the top j. Our main result relates the learnability of $\mathcal{H}$ to the learnability of $\mathcal{H}_i^j$'s.
- We define two families of ranking loss functions that capture most if not all ranking losses used in practice. We show that these families are actually *equivalence* classes - the same characterization of batch and online learnability holds for every loss in that family.
- By relating the learnability of $\mathcal{H}$ to the learnability of *binary* hypothesis classes $\mathcal{H}_i^j$, we show that existing combinatorial dimensions, like the VC and Littlestone dimension, continue to characterize learnability in the multilabel ranking setting. This allows us to prove that linear ranking hypothesis classes are learnable in the batch setting.

A unifying theme throughout the chapter is our ability to *constructively* convert a learning algorithm $\mathcal{A}$ for $\mathcal{H}$ into a learning algorithm $\mathcal{A}_i^j$ for $\mathcal{H}_i^j$ for each $i, j \in [K]$ and vice versa. To do so, our proof techniques involve adapting the agnostic-to-realizable reduction for batch and online classification, proposed by Hopkins et al. [2022] and Raman et al. [2023a] respectively, to ranking.

7.2 Preliminaries and Notation

Let $\mathcal{X}$ denote the instance space, $\mathcal{S}_K$ the set of permutations over labels $[K] := \{1, ..., K\}$, and $\mathcal{Y} = \{0, 1, ..., B\}^K$ the target space for some $K, B \in \mathbb{N}$. We highlight that the set of labels $[K]$ is fixed beforehand and does not depend on the instance $x \in \mathcal{X}$. This is to be contrasted with subset ranking, the set of labels can change depending on the instance $x \in \mathcal{X}$.

We refer to an element $y \in \mathcal{Y}$ as a *relevance-score vector* that indicates the relevance of each of the K labels, where B indicates the highest relevance and 0 indicates the lowest relevance. Throughout the chapter, we treat a permutation $\pi \in \mathcal{S}_K$ as a vector in $\{1, ..., K\}^K$ that induces a *ranking* of the K labels in decreasing order of relevance. Accordingly, for an index $i \in [K]$, we let $\pi_i \in [K]$ denote the *rank* of label i. Likewise, given an index $i \in [K]$, we let y^i denote the relevance of label i. In addition, it will be useful to define a mapping from $\mathcal{S}_K$ to $\{0, 1\}^K$. In particular, we define $\text{BinRel}(\cdot, \cdot) : \mathcal{S}_K \times [K] \to \{0, 1\}^K$ as

an operator that given a permutation (ranking) $\pi \in \mathcal{S}_K$ and threshold $p \in [K]$, outputs a bit string $b \in \{0,1\}^K$ s.t. $b_i = \mathbb{1}\{\pi_i \leq p\}$.

Ranking Equivalences. Our construction of ranking loss families in Section 7.3 requires different notions of equivalence between permutations (rankings) in $\mathcal{S}_K$. To that end, we say that $\pi = \hat{\pi}$ if and only if for all $i \in [K]$, $\pi_i = \hat{\pi}_i$. On the other hand, we say $\pi \overset{p}{=} \hat{\pi}$ if and only if $\{i : \pi_i \leq p\} = \{i : \hat{\pi}_i \leq p\}$. That is, two rankings are p-equivalent if the *set* of labels they rank in the top-p are equal. Finally, we say $\pi \overset{[p]}{=} \hat{\pi}$ if and only if for all $j \in [p]$, $\{i : \pi_i \leq j\} = \{i : \hat{\pi}_i \leq j\}$. That is, two rankings are $[p]$-equivalent if not only the *set* but also the *order* of labels they rank in the top-p are equal.

Ranking Hypothesis. A ranking hypothesis $h \in \mathcal{H} \subseteq \mathcal{S}_K^{\mathcal{X}}$ maps instances in $\mathcal{X}$ to a ranking (permutation) in $\mathcal{S}_K$. Given an instance $x \in \mathcal{X}$, one can think of $h(x)$ as h's ranking of the K different labels in decreasing order of relevance. For any ranking hypothesis h, we let $h_i : \mathcal{X} \to [K]$ denote its restriction to the i'th coordinate output. Accordingly, for an instance $x \in \mathcal{X}$, $h_i(x)$ gives the rank that h assigns to label i. Given a ranking hypothesis class $\mathcal{H} \subseteq \mathcal{S}_K^{\mathcal{X}}$ and any $i, j \in [K]$, we define its binary threshold-restricted hypothesis class $\mathcal{H}_i^j = \{h_i^j : h \in \mathcal{H}\}$ where $h_i^j(x) = \mathbb{1}\{h_i(x) \leq j\}$. We can think of hypotheses in $\mathcal{H}_i^j$ as providing binary responses to queries of the form: "for instance x, should label i ranked in the top j?" These threshold-restricted classes are central to our characterization of learnability in both the batch and online learning settings.

Batch Learnability. In the batch setting, we are interested in characterizing the learnability of a ranking hypothesis class $\mathcal{H}$ under a model similar to the classical PAC model [Valiant, 1984].

Definition 7.2.1 (Agnostic Ranking PAC Learnability)**.** *A ranking hypothesis class $\mathcal{H} \subseteq \mathcal{S}_K^{\mathcal{X}}$ is agnostic PAC learnable w.r.t. loss $\ell : \mathcal{S}_K \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$, if there exists a function $m : (0,1)^2 \times \mathbb{N} \to \mathbb{N}$ and a learning algorithm $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^\star \to \mathcal{S}_K^{\mathcal{X}}$ with the following property: for every $\varepsilon, \delta \in (0,1)$ and for every distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$, running algorithm $\mathcal{A}$ on $n \geq m(\varepsilon, \delta, K)$ iid samples from $\mathcal{D}$ outputs a predictor $g = \mathcal{A}(S)$ such that with probability at least $1 - \delta$ over $S \sim \mathcal{D}^n$,*

$$\mathbb{E}_{\mathcal{D}}[\ell(g(x), y)] \leq \inf_{h \in \mathcal{H}} \mathbb{E}_{\mathcal{D}}[\ell(h(x), y)] + \varepsilon.$$

If $\mathcal{D}$ is restricted to the class of distributions such that $\inf_{h \in \mathcal{H}} \mathbb{E}_{\mathcal{D}}[\ell(h(x), y)] = 0$, then we say we are in the *realizable* setting. Note that unlike in classification, realizability in the

multilabel ranking setting is loss dependent.

Online Learnability. In the online setting, an adversary plays a sequential game with the learner over T rounds. In each round $t \in [T]$, an adversary selects a labeled instance $(x_t, y_t) \in \mathcal{X} \times \mathcal{Y}$ and reveals x_t to the learner. The learner makes a (potentially randomized) prediction $\hat{\pi}_t \in \mathcal{S}_K$. Finally, the adversary reveals the true relevance-score vector y_t, and the learner suffers the loss $\ell(\hat{\pi}_t, y_t)$, where ℓ is some pre-specified ranking loss function. Given a ranking hypothesis class $\mathcal{H} \subseteq \mathcal{S}_K^{\mathcal{X}}$, the goal of the learner is to output predictions $\hat{\pi}_t$ such that its cumulative loss is close to the best possible cumulative loss over hypotheses in $\mathcal{H}$. A hypothesis class is online learnable if there exists an algorithm such that for any sequence of labeled examples $(x_1, y_1), ..., (x_T, y_T)$, the difference in cumulative loss between its predictions and the predictions of the best possible function in $\mathcal{H}$ is small.

Definition 7.2.2 (Agnostic Online Ranking Learnability)**.** *A ranking hypothesis class $\mathcal{H} \subseteq \mathcal{S}_K^{\mathcal{X}}$ is agnostic online learnable w.r.t. loss ℓ, if there exists an (potentially randomized) algorithm $\mathcal{A}$ such that for any adaptively chosen sequence of labeled examples $(x_t, y_t) \in \mathcal{X} \times \mathcal{Y}$, the algorithm outputs $\mathcal{A}(x_t) \in \mathcal{S}_K$ at every iteration $t \in [T]$ such that its expected regret,*

$$R(T, K) := \mathbb{E}\left[\sum_{t=1}^{T} \ell(\mathcal{A}(x_t), y_t) - \inf_{h \in \mathcal{H}} \sum_{t=1}^{T} \ell(h(x_t), y_t)\right],$$

is a non-decreasing, sub-linear function of T. Here, the expectation is taken with respect to the randomness of the algorithm $\mathcal{A}$.

If it is further guaranteed that there exists a hypothesis $h^\star \in \mathcal{H}$ such that $\sum_{t=1}^{T} \ell(h^\star(x_t), y_t) = 0$, then we say we are in the *realizable* setting. Again, realizability is loss dependent.

7.3 Ranking Loss Families

In statistical learning theory, we often characterize learnability with respect to a loss function. Unlike the 0-1 loss in classification, there is no canonical loss function in multilabel ranking. Accordingly, we define two general families of ranking loss functions in this section and later characterize learnability with respect to all losses in these families. In Appendix E.1, we show that many of the ranking metrics used in practice (e.g. Pairwise Rank Loss, Discounted Cumulative Gain, Reciprocal Rank, Average Precision, Precision@p, etc.) fall into one of these two families.

On a high level, we can classify ranking losses into two main groups: (A) those losses that care about both the order and magnitude of the relevance scores within the top-p ranked

labels and (B) those losses that only care about the magnitude of the relevance scores within the top-p ranked labels. Our goal will be to define a loss family for both groups A and B. To do so, we start by identifying a canonical ranking loss that lies in each group. For group A, the normalized sum loss@p,

$$\ell_{\text{sum}}^{@p}(\pi, y) = \sum_{i=1}^{K} \min(\pi_i, p+1) y^i - Z_y^p$$

captures both the order and magnitude of the relevance scores only for the top-p ranked labels. Here, Z_y^p is an appropriately chosen normalization factor that only depends on p and y such that $\min_{\pi \in \mathcal{S}_K} \ell_{\text{sum}}^{@p}(\pi, y) = 0$. For Group B, the normalized precision loss@p,

$$\ell_{\text{prec}}^{@p}(\pi, y) = Z_y^p - \sum_{i=1}^{K} \mathbb{1}\{\pi_i \leq p\} y^i$$

cares only about the magnitude of relevance scores in the top-p ranked labels. Again, Z_y^p is an appropriately chosen normalization constant that only depends on p and y such that the minimum loss is 0. The form of $\ell_{\text{prec}}^{@p}$ differs from $\ell_{\text{sum}}^{@p}$ because $\sum_{i=1}^{K} \mathbb{1}\{\pi_i \leq p\} y^i$ is a gain whereas $\sum_{i=1}^{K} \min(\pi_i, p+1) y^i$ is a loss.

Next, we build loss families around $\ell_{\text{sum}}^{@p}$ and $\ell_{\text{prec}}^{@p}$. For $\ell_{\text{sum}}^{@p}$, consider the family:

$$\mathcal{L}(\ell_{\text{sum}}^{@p}) = \{\ell \in \mathbb{R}^{\mathcal{S}_K \times \mathcal{Y}} : \ell = 0 \text{ if and only if } \ell_{\text{sum}}^{@p} = 0\} \cap \{\ell \in \mathbb{R}^{\mathcal{S}_K \times \mathcal{Y}} : \pi \overset{[p]}{=} \hat{\pi} \implies \ell(\pi, y) = \ell(\hat{\pi}, y)\}.$$

By definition, $\mathcal{L}(\ell_{\text{sum}}^{@p})$ contains those ranking losses that are (1) zero-matched with $\ell_{\text{sum}}^{@p}$ and (2) remain unchanged for any two predicted rankings (permutations) that are $[p]$-equivalent. The second constraint is needed to ensure that losses in $\mathcal{L}(\ell_{\text{sum}}^{@p})$ only depend on the order and set of labels that π ranks in the top-p. Likewise, we can construct a similar loss family around $\ell_{\text{prec}}^{@p}$ as follows:

$$\mathcal{L}(\ell_{\text{prec}}^{@p}) = \{\ell \in \mathbb{R}^{\mathcal{S}_K \times \mathcal{Y}} : \ell = 0 \text{ if and only if } \ell_{\text{prec}}^{@p} = 0\} \cap \{\ell \in \mathbb{R}^{\mathcal{S}_K \times \mathcal{Y}} : \pi \overset{p}{=} \hat{\pi} \implies \ell(\pi, y) = \ell(\hat{\pi}, y)\}.$$

The set $\mathcal{L}(\ell_{\text{prec}}^{@p})$ contains those ranking losses that are (1) zero-matched with $\ell_{\text{prec}}^{@p}$ and (2) remain unchanged for any two predicted rankings (permutations) that are $\underline{p\text{-equivalent}}$. The second constraint is needed to ensure that losses in $\mathcal{L}(\ell_{\text{prec}}^{@p})$ only depend on the set of labels that π ranks in the top-p. A major contribution of this chapter is showing that both $\mathcal{L}(\ell_{\text{sum}}^{@p})$

and $\mathcal{L}(\ell^{@p}_{\text{prec}})$ are actually *equivalence* classes - the same characterization of learnability holds for every loss in that family.

7.4 Batch Multilabel Ranking

In this section, we characterize the agnostic PAC learnability of hypothesis classes $\mathcal{H} \subseteq \mathcal{S}_K^{\mathcal{X}}$ with respect to both $\mathcal{L}(\ell^{@p}_{\text{sum}})$ and $\mathcal{L}(\ell^{@p}_{\text{prec}})$. Our main results, stated below as two theorems, relate the learnability of $\mathcal{H}$ to the learnability of the threshold-restricted classes $\mathcal{H}_i^j$.

Theorem 7.4.1. *A hypothesis class $\mathcal{H} \subseteq \mathcal{S}_K^{\mathcal{X}}$ is agnostic PAC learnable w.r.t $\ell \in \mathcal{L}(\ell^{@p}_{sum})$ if and only if for all $i \in [K]$ and $j \in [p]$, $\mathcal{H}_i^j$ is agnostic PAC learnable w.r.t the 0-1 loss.*

Theorem 7.4.2. *A hypothesis class $\mathcal{H} \subseteq \mathcal{S}_K^{\mathcal{X}}$ is agnostic PAC learnable w.r.t $\ell \in \mathcal{L}(\ell^{@p}_{prec})$ if and only if for all $i \in [K]$, $\mathcal{H}_i^p$ is agnostic PAC learnable w.r.t the 0-1 loss.*

Since VC dimension characterizes the learnability of binary hypothesis classes under the 0-1 loss, an important corollary of Theorems 7.4.1 and 7.4.2 is that finiteness of VC($\mathcal{H}_i^j$)'s, for the appropriate $i, j \in [K] \times [p]$, is necessary and sufficient for agnostic ranking PAC learnability. Later on, we use this fact to prove that linear ranking hypothesis classes are agnostic ranking PAC learnable.

We start with the proof of Theorem 7.4.1, which follows in three steps. First, we show that if for all $(i, j) \in [K] \times [p]$, $\mathcal{H}_i^j$ is agnostic PAC learnable w.r.t 0-1 loss, then Empirical Risk Minimization (ERM) is an agnostic PAC learner for $\mathcal{H}$ w.r.t $\ell^{@p}_{\text{sum}}$. Next, we show that if $\mathcal{H}$ is agnostic PAC learnable w.r.t $\ell^{@p}_{\text{sum}}$, then $\mathcal{H}$ is agnostic PAC learnable w.r.t any loss $\ell \in \mathcal{L}(\ell^{@p}_{\text{sum}})$. Our proof of the latter uses the realizable to agnostic conversion from Hopkins et al. [2022]. Finally, we prove the necessity direction - if $\mathcal{H}$ is agnostic PAC learnable w.r.t an arbitrary $\ell \in \mathcal{L}(\ell^{@p}_{\text{sum}})$, then for all $(i, j) \in [K] \times [p]$, $\mathcal{H}_i^j$ is agnostic PAC learnable w.r.t 0-1 loss. The proof of necessity direction also uses the realizable to agnostic conversion from Hopkins et al. [2022]. The proof of Theorem 7.4.2 follows exactly the same way as Theorem 7.4.1 with some minor changes. Thus, we only focus on the proof of Theorem 7.4.1 in this section and defer all discussion of Theorem 7.4.2 to Appendix E.3.3.

We begin with Lemma 7.4.3, which asserts that if $\mathcal{H}_i^j$ is agnostic PAC learnable for all $(i, j) \in [K] \times [p]$, then ERM is an agnostic PAC learner for $\mathcal{H}$ w.r.t $\ell^{@p}_{\text{sum}}$.

Lemma 7.4.3. *If for all $i \in [K]$ and $j \in [p]$, $\mathcal{H}_i^j$ is agnostic PAC learnable w.r.t the 0-1 loss, then ERM is an agnostic PAC learner for $\mathcal{H} \subseteq \mathcal{S}_K^{\mathcal{X}}$ w.r.t $\ell^{@p}_{sum}$.*

The proof of Lemma 7.4.3 exploits the nice structure of $\ell^{@p}_{\text{sum}}$ by upperbounding the empirical Rademacher complexity of the loss class $\ell^{@p}_{\text{sum}} \circ \mathcal{H} = \{(x, y) \mapsto \ell^{@p}_{\text{sum}}(h(x), y) : h \in \mathcal{H})\}$

and showing that it vanishes as the sample size n becomes large. Then, standard uniform convergence arguments outlined in Proposition E.3.1 imply that ERM is an agnostic PAC learner for $\mathcal{H}$ w.r.t $\ell_{\text{sum}}^{@p}$. The full proof is in Appendix E.3.

Since arbitrary losses in $\mathcal{L}(\ell_{\text{sum}}^{@p})$ may not have nice analytical forms, Lemma 7.4.4 relates the learnability of an arbitrary loss $\ell \in \mathcal{L}(\ell_{\text{sum}}^{@p})$ to the learnability of $\ell_{\text{sum}}^{@p}$.

Lemma 7.4.4. *If $\mathcal{H} \subseteq \mathcal{S}_K^{\mathcal{X}}$ is agnostic PAC learnable w.r.t $\ell_{sum}^{@p}$, then $\mathcal{H}$ is agnostic PAC learnable w.r.t any $\ell \in \mathcal{L}(\ell_{sum}^{@p})$.*

Proof. (of Lemma 7.4.4) Fix $\ell \in \mathcal{L}(\ell_{\text{sum}}^{@p})$. Let $a = \min_{\pi,y}\{\ell(\pi, y) \mid \ell(\pi, y) \neq 0\}$ and $b = \max_{\pi,y} \ell(\pi, y)$. We need to show that if $\mathcal{H}$ is agnostic PAC learnable w.r.t $\ell_{\text{sum}}^{@p}$, then $\mathcal{H}$ is agnostic PAC learnable w.r.t ℓ. We will do so in two steps. First, we will show that if $\mathcal{A}$ is an agnostic PAC learner for $\ell_{\text{sum}}^{@p}$, then $\mathcal{A}$ is also a *realizable* PAC learner for ℓ. Next, we will show how to convert a realizable PAC learner for ℓ into an agnostic PAC learner for ℓ in a black-box fashion. The composition of these two pieces yields an agnostic PAC learner for $\mathcal{H}$ w.r.t ℓ.

Realizable PAC learnability of $\mathcal{H}$ w.r.t ℓ. If $\mathcal{H}$ is agnostic PAC learnable w.r.t $\ell_{\text{sum}}^{@p}$, then there exists a learning algorithm $\mathcal{A}$ with sample complexity $m(\varepsilon, \delta, K)$ s.t. for any distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, with probability $1-\delta$ over a sample $S \sim \mathcal{D}^n$ of size $n \geq m(\varepsilon, \delta, K)$, the output predictor $g = \mathcal{A}(S)$ achieves $\mathbb{E}_{\mathcal{D}}\left[\ell_{\text{sum}}^{@p}(g(x), y))\right] \leq \inf_{h\in\mathcal{H}} \mathbb{E}_{\mathcal{D}}\left[\ell_{\text{sum}}^{@p}(h(x), y))\right] + \varepsilon$. In the realizable setting, we are further guaranteed that there exists a hypothesis $h^\star \in \mathcal{H}$ s.t. $\mathbb{E}_{\mathcal{D}}\left[\ell(h^\star(x), y))\right] = 0$. Since $\ell \in \mathcal{L}(\ell_{\text{sum}}^{@p})$, this also implies that $\mathbb{E}_{\mathcal{D}}\left[\ell_{\text{sum}}^{@p}(h^\star(x), y))\right] = 0$. Therefore, under realizability and the fact that $\ell \leq b\, \ell_{\text{sum}}^{@p}$, we have $\mathbb{E}_{\mathcal{D}}\left[\ell(g(x), y))\right] \leq b\varepsilon$. This completes the first part of the proof as we have shown that $\mathcal{A}$ is also a realizable PAC learner for $\mathcal{H}$ w.r.t ℓ with sample complexity $m(\frac{\varepsilon}{b}, \delta, K)$.

Realizable-to-agnostic conversion. Now, we show how to convert the realizable PAC learner $\mathcal{A}$ for ℓ into an agnostic PAC learner for ℓ in a black-box fashion. For this step, we will extend the agnostic-to-realizable reduction proposed by Hopkins et al. [2022] to the ranking setting by accommodating the mismatch between the range space of $\mathcal{H}$ and the label space $\mathcal{Y}$. In particular, we will show that Algorithm 9 below converts a realizable PAC learner for ℓ into an agnostic PAC learner for ℓ. Note that although input $\mathcal{A}$ is a realizable learner, the distribution $\mathcal{D}$ may not be realizable.

Let $h^\star = \arg\min_{h\in\mathcal{H}} \mathbb{E}_{\mathcal{D}}\left[\ell(h(x), y)\right]$ denote the optimal predictor in $\mathcal{H}$ w.r.t $\mathcal{D}$. Consider the sample $S_U^{h^\star}$ and let $g = \mathcal{A}(S_U^{h^\star})$. We can think of g as the output of $\mathcal{A}$ run over an i.i.d sample S drawn from $\mathcal{D}^\star$, a joint distribution over $\mathcal{X} \times \mathcal{Y}$ defined procedurally by first sampling $x \sim \mathcal{D}_{\mathcal{X}}$, then independently sampling $j \sim \text{Unif}([p])$, and finally outputting the labeled sample $(x, \text{BinRel}(h^\star(x), j))$. Note that $\mathcal{D}^\star$ is indeed a realizable distribution

Algorithm 9 Agnostic PAC learner for $\mathcal{H}$ w.r.t. ℓ

Require: Realizable PAC learner $\mathcal{A}$ for $\mathcal{H}$, unlabeled and labeled samples $S_U \sim \mathcal{D}_{\mathcal{X}}^n$ and $S_L \sim \mathcal{D}^m$

1: For each $h \in \mathcal{H}_{|S_U}$, construct a dataset

$$S_U^h = \{(x_1, \tilde{y}_1), ..., (x_n, \tilde{y}_n)\} \text{ s.t. } \tilde{y}_i \sim \text{Unif}\{\text{BinRel}(h(x_i), 1), ..., \text{BinRel}(h(x_i), p)\}$$

2: Run $\mathcal{A}$ over all datasets to get $C(S_U) := \left\{\mathcal{A}\left(S_U^h\right) \mid h \in \mathcal{H}_{|S_U}\right\}$

3: Return $\hat{g} \in C(S_U)$ with the lowest empirical error over S_L w.r.t. ℓ.

(realized by $h^\star$) w.r.t both ℓ and $\ell_{\text{sum}}^{@p}$. Recall that $m_{\mathcal{A}}(\frac{\varepsilon}{b}, \delta, K)$ is the sample complexity of $\mathcal{A}$. Since $\mathcal{A}$ is a realizable learner for $\mathcal{H}$ w.r.t ℓ, we have that for $n \geq m_{\mathcal{A}}(\frac{a\varepsilon}{2b^2p}, \delta/2, K)$, with probability at least $1 - \frac{\delta}{2}$, $\mathbb{E}_{\mathcal{D}^\star}[\ell(g(x), y)] \leq \frac{a\varepsilon}{2bp}$.

Next, by Lemma E.5.1, we have $\ell(g(x), y) \leq \ell(h^\star(x), y) + \frac{bp}{a}\mathbb{E}_{j\sim\text{Unif}([p])}[\ell(g(x), \text{BinRel}(h^\star(x), j))]$ pointwise. Taking expectations on both sides of the inequality gives

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}[\ell(g(x), y)] &\leq \mathbb{E}_{\mathcal{D}}[\ell(h^\star(x), y)] + \frac{bp}{a}\mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{E}_{j\sim\text{Unif}([p])}[\ell(g(x), \text{BinRel}(h^\star(x), j))]\right] \\ &\leq \mathbb{E}_{\mathcal{D}}[\ell(h^\star(x), y)] + \frac{\varepsilon}{2}.\end{aligned}$$

The last inequality follows from the definition of $\mathcal{D}^\star$, namely $\mathbb{E}_{\mathcal{D}^\star}[\ell(g(x), y)] = \mathbb{E}_{x\sim\mathcal{D}_{\mathcal{X}}}\mathbb{E}_{j\sim\text{Unif}([p])}[\ell(g(x), \text{BinRel}(h^\star(x), j))]$. This shows that $C(S_U)$ contains a hypothesis g that generalizes well with respect to $\mathcal{D}$. Now we want to show that the predictor $\hat{g}$ returned in step 3 also has good generalization. Crucially, observe that $C(S_U)$ is a finite hypothesis class with cardinality at most 2^{nK}. By standard Chernoff and union bounds, with probability at least $1 - \delta/2$, the empirical risk of every hypothesis in $C(S_U)$ on a sample of size $\geq \frac{8}{\varepsilon^2}\log\frac{4|C(S_U)|}{\delta}$ is at most $\varepsilon/4$ away from its true error. So, if $m = |S_L| \geq \frac{8}{\varepsilon^2}\log\frac{4|C(S_U)|}{\delta}$, then with probability at least $1 - \delta/2$,

$$\frac{1}{|S_L|}\sum_{(x,y)\in S_L}\ell(g(x), y) \leq \mathbb{E}_{\mathcal{D}}[\ell(g(x), y)] + \frac{\varepsilon}{4} \leq \mathbb{E}_{\mathcal{D}}[\ell(h^\star(x), y)] + \frac{3\varepsilon}{4}.$$

Since $\hat{g}$ is the ERM on S_L over $C(S)$, its empirical risk can be at most $\mathbb{E}_{\mathcal{D}}[\ell(h^\star(x), y)] + \frac{3\varepsilon}{4}$. Given that the population risk of $\hat{g}$ can be at most $\varepsilon/4$ away from its empirical risk, we have that

$$\mathbb{E}_{\mathcal{D}}[\ell(\hat{g}(x), y)] \leq \mathbb{E}_{\mathcal{D}}[\ell(h^\star(x), y)] + \varepsilon.$$

Applying union bounds, the entire process succeeds with probability $1 - \delta$. We can upper

bound the sample complexity of Algorithm 9, denoted $n(\varepsilon, \delta, K)$, as

$$\begin{aligned} n(\varepsilon, \delta, K) &\leq m_{\mathcal{A}}\left(\frac{a\varepsilon}{2b^2p}, \delta/2, K\right) + O\left(\frac{1}{\varepsilon^2}\log\frac{|C(S_U)|}{\delta}\right) \\ &\leq m_{\mathcal{A}}\left(\frac{a\varepsilon}{2b^2p}, \delta/2, K\right) + O\left(\frac{Km_{\mathcal{A}}(\frac{a\varepsilon}{2b^2p}, \delta/2, K) \ + \log\frac{1}{\delta}}{\varepsilon^2}\right), \end{aligned}$$

where we use $|C(S_U)| \leq 2^{Km_{\mathcal{A}}(\frac{a\varepsilon}{2b^2p}, \delta/2, K)}$. This shows that Algorithm 9 is an agnostic PAC learner for $\mathcal{H}$ w.r.t ℓ. ■

Finally, Lemma 7.4.5 gives the necessity direction of Theorem 7.4.1.

Lemma 7.4.5. *If a hypothesis class $\mathcal{H} \subseteq \mathcal{S}_K^{\mathcal{X}}$ is agnostic PAC learnable w.r.t $\ell \in \mathcal{L}(\ell_{sum}^{@p})$, then $\mathcal{H}_i^j$ is agnostic PAC learnable w.r.t the 0-1 loss for all $(i, j) \in [K] \times [p]$.*

Like the sufficiency proofs, the proof of Lemma 7.4.5 is constructive. Given an agnostic PAC learner for $\mathcal{H}$ w.r.t ℓ, we construct an agnostic PAC learner for $\mathcal{H}_i^j$ w.r.t 0-1 loss using a slight modification of Algorithm 9. We defer the full proof to Appendix E.3 since the analysis is similar to that of Algorithm 9. Together, Lemmas 7.4.3, 7.4.4 and 7.4.5 imply Theorem 7.4.1.

We conclude this section by giving a concrete application of our characterization. Consider the class of ranking-hypotheses $\mathcal{H} = \{x \mapsto \text{argsort}(Wx) : W \in \mathbb{R}^{K\times d}\}$ that compute rankings by sorting scores, in descending order, obtained from a linear function of the input features. Lemma 7.4.6, whose proof is in Appendix E.2, computes the VC dimension of $\mathcal{H}_i^j$ for an arbitrary $i, j \in [K]$.

Lemma 7.4.6. *Let $\mathcal{H} = \{x \mapsto \underline{argsort}(Wx) : W \in \mathbb{R}^{K\times d}\}$ be a linear ranking hypothesis class. Then for all $i, j \in [K]$, $VC(\mathcal{H}_i^j) = \tilde{O}(Kd)$, where $\tilde{O}$ hides logarithmic factors of d and K.*

Combining Lemma 7.4.6 with Theorems 7.4.1 and 7.4.2 shows that linear ranking hypothesis classes are agnostic ranking PAC learnable w.r.t to all losses in $\mathcal{L}(\ell_{\text{sum}}^{@p}) \cup \mathcal{L}(\ell_{\text{prec}}^{@p})$. More generally, in Appendix E.2 we give a dimension-based sufficient condition under which *generic* score-based ranking hypothesis classes are agnostic ranking PAC learnable.

7.5 Online Multilabel Ranking

We now move to the online setting and characterize the online learnability of hypothesis classes $\mathcal{H} \subseteq \mathcal{S}_K^{\mathcal{X}}$ with respect to both $\mathcal{L}(\ell_{\text{sum}}^{@p})$ and $\mathcal{L}(\ell_{\text{prec}}^{@p})$. As in the batch setting, our

characterization relates the learnability of $\mathcal{H}$ to the learnability of the threshold-restricted classes $\mathcal{H}_i^j$.

Theorem 7.5.1. *A hypothesis class $\mathcal{H} \subseteq \mathcal{S}_K^{\mathcal{X}}$ is agnostic online learnable w.r.t $\ell \in \mathcal{L}(\ell_{sum}^{@p})$ if and only if for all $i \in [K]$ and $j \in [p]$, $\mathcal{H}_i^j$ is agnostic online learnable w.r.t the 0-1 loss.*

Theorem 7.5.2. *A hypothesis class $\mathcal{H} \subseteq \mathcal{S}_K^{\mathcal{X}}$ is agnostic online learnable w.r.t $\ell \in \mathcal{L}(\ell_{prec}^{@p})$ if and only if for all $i \in [K]$, $\mathcal{H}_i^p$ is agnostic online learnable w.r.t the 0-1 loss.*

Since the Littlestone dimension characterizes the online learnability of binary hypothesis classes under the 0-1 loss, an important corollary of Theorems 7.5.1 and 7.5.2 is is that finiteness of $\text{Ldim}(\mathcal{H}_i^j)$, for the appropriate $i, j \in [K] \times [p]$, is necessary and sufficient for agnostic online ranking learnability.

We now begin the proof of Theorem 7.5.1. Since the proof of Theorem 7.5.2 follows a similar trajectory, we defer all discussion of Theorem 7.5.2 to Appendix E.4.2. Unlike Theorem 7.4.1 in the batch setting, we prove the sufficiency and necessity directions of Theorem 7.5.1 directly. We chose this direct path because, unlike the batch setting, sequential Rademacher analysis does not yield a constructive algorithm [Rakhlin et al., 2015b]. On the other hand, our proofs are constructive and use the celebrated Randomized Exponential Weights Algorithm (REWA) [Cesa-Bianchi and Lugosi, 2006]. Moreover, a key ingredient of our proof is the realizable to agnostic conversion from Raman et al. [2023a].

Proof. (of sufficiency in Theorem 7.5.1) Fix $\ell \in \mathcal{L}(\ell_{\text{sum}}^{@p})$. Let $a = \min_{\pi,y}\{\ell(\pi, y) \mid \ell(\pi, y) \neq 0\}$ and $M = \max_{\pi,y} \ell(\pi, y)$. Given online learners for $\mathcal{H}_i^j$ for the 0-1 loss, our goal is to construct an online learner $\mathcal{Q}$ for $\mathcal{H}$ w.r.t ℓ that enjoys sub-linear regret in T. Our strategy will be to construct a set of experts $\mathcal{E}$ using the online learners for $\mathcal{H}_i^j$'s and run REWA using $\mathcal{E}$ and an appropriately scaled version of ℓ. Our proof borrows ideas from the realizable-to-agnostic online conversion from Raman et al. [2023a] and so we use the same notation whenever possible.

Let $(x_1, y_1), ..., (x_T, y_T) \in (\mathcal{X} \times \mathcal{Y})^T$ denote the stream of points to be observed by the online learner. We will assume an oblivious adversary and thus the stream is fixed before the game starts. A standard reduction (Chapter 4 in Cesa-Bianchi and Lugosi [2006]) allows us to convert oblivious regret bounds to adaptive regret bounds. Since $\mathcal{H}_i^j \subseteq \{0, 1\}^{\mathcal{X}}$ is online learnable w.r.t. 0-1 loss, we are guaranteed the existence of online learners $\mathcal{A}_i^j$ for $\mathcal{H}_i^j$.

Constructing Experts. For any bitstring $b \in \{0, 1\}^T$, let $\phi : \{t \in [T] : b_t = 1\} \to \mathcal{S}_K$ denote a function mapping time points where $b_t = 1$ to rankings (permutations). Let $\Phi_b = \mathcal{S}_K^{\{t\in[T]:b_t=1\}}$ denote all such functions ϕ. For every $h \in \mathcal{H}$, there exists a $\phi_b^h \in \Phi_b$ such that for all $t \in \{t : b_t = 1\}$, $\phi_b^h(t) = h(x_t)$. Let $|b| = |\{t \in [T] : b_t = 1\}|$. For every $b \in \{0, 1\}^T$

and $\phi \in \Phi_b$, we will define an Expert $E_{b,\phi}$. Expert $E_{b,\phi}$, formally presented in Algorithm 10, uses $\mathcal{A}_i^j$'s to make predictions in each round. However, $E_{b,\phi}$ only updates the $\mathcal{A}_i^j$'s on those rounds where $b_t = 1$, using ϕ to compute a labeled instance. For every $b \in \{0,1\}^T$, let $\mathcal{E}_b = \bigcup_{\phi \in \Phi_b} \{E_{b,\phi}\}$ denote the set of all Experts parameterized by functions $\phi \in \Phi_b$. If b is the bitstring with all zeros, then $\mathcal{E}_b$ will be empty. Therefore, we will actually define $\mathcal{E}_b = \{E_0\} \cup \bigcup_{\phi \in \Phi_b} \{E_{b,\phi}\}$, where E_0 is the expert that never updates $\mathcal{A}_i^j$'s and only uses them for predictions in all $t \in [T]$. Note that $1 \leq |\mathcal{E}_b| \leq (K!)^{|b|} \leq K^{K|b|}$.

Algorithm 10 Expert (b, ϕ)

Require: Independent copy of realizable learners $\mathcal{A}_i^j$ of $\mathcal{H}_i^j$ for each $(i,j) \in [K] \times [p]$

for $t = 1, ..., T$ **do**
  Receive example x_t
  Define a binary vote matrix $V_t \in \{0,1\}^{K \times p}$ such that $V_t[i,j] = \mathcal{A}_i^j(x_t)$
  Predict $\hat{\pi}_t \in \arg\min_{\pi \in \mathcal{S}_K} \langle \pi, V_t \mathbf{1}_p \rangle$
  if $b_t = 1$ **then**
    Let $\pi = \phi(t)$ and for all $(i,j) \in [K] \times [p]$, update $\mathcal{A}_i^j$ by passing (x_t, π_i^j)
  end if
end for

Algorithm 11 Agnostic Online Learner $\mathcal{Q}$ for $\mathcal{H}$ w.r.t. ℓ

Require: Parameter $0 < \beta < 1$

Let $B \in \{0,1\}^T$ s.t. $B_t \overset{\text{iid}}{\sim} \text{Bernoulli}(\frac{T^\beta}{T})$
Construct the set of experts $\mathcal{E}_B = \{E_0\} \cup \bigcup_{\phi \in \Phi_B} \{E_{B,\phi}\}$ according to Algorithm 10
Run REWA $\mathcal{P}$ using $\mathcal{E}_B$ and the loss function $\frac{\ell}{M}$ over the stream $(x_1, y_1), ..., (x_T, y_T)$

Using these experts, Algorithm 11 presents our agnostic online learner $\mathcal{Q}$ for $\mathcal{H}$ w.r.t $\ell \in \mathcal{L}(\ell_{\text{sum}}^{@p})$. We now show that $\mathcal{Q}$ enjoys sub-linear regret. We highlight that there are three sources of randomness in online learner $\mathcal{Q}$, namely the randomness of sampling B, the internal randomness of $\mathcal{A}_i^j$'s, and the internal randomness of $\mathcal{P}$. One may think of internal randomness as arising from the sampling step involved in the randomized predictions. Let A be the random variable associated with joint internal randomness of $\mathcal{A}_i^j$ for all $(i,j) \in [K] \times [p]$. Similarly, denote P to be the random variable associated with the internal randomness of $\mathcal{P}$. We begin by using the guarantee of REWA.

REWA Guarantee. Using Theorem 21.11 in Shalev-Shwartz and Ben-David [2014a] and the fact that B, A and P are mutually independent, REWA guarantees almost surely that

$$\sum_{t=1}^{T} \mathbb{E}\left[\ell(\mathcal{P}(x_t), y_t) | B, A\right] \leq \inf_{E \in \mathcal{E}_B} \sum_{t=1}^{T} \ell(E(x_t), y_t) + M\sqrt{2T \ln(|\mathcal{E}_B|)}.$$

Taking an outer expectation gives

$$\mathbb{E}\left[\sum_{t=1}^{T}\ell(\mathcal{P}(x_t),y_t)\right] \le \mathbb{E}\left[\inf_{E\in\mathcal{E}_B}\sum_{t=1}^{T}\ell(E(x_t),y_t)\right] + \mathbb{E}\left[M\sqrt{2T\ln(|\mathcal{E}_B|)}\right].$$

Noting that $\mathcal{Q}(x_t) = \mathcal{P}(x_t)$, we obtain

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{T}\ell(\mathcal{Q}(x_t),y_t)\right] &\le \mathbb{E}\left[\inf_{E\in\mathcal{E}_B}\sum_{t=1}^{T}\ell(E(x_t),y_t)\right] + \mathbb{E}\left[M\sqrt{2T\ln(|\mathcal{E}_B|)}\right] \\ &\le \mathbb{E}\left[\sum_{t=1}^{T}\ell(E_{B,\phi_B^{h^\star}}(x_t),y_t)\right] + M\mathbb{E}\left[\sqrt{2T\ln(|\mathcal{E}_B|)}\right].\end{aligned}$$

In the last step, we used the fact that for all $b \in \{0,1\}^T$ and $h \in \mathcal{H}$, $E_{b,\phi_b^h} \in \mathcal{E}_b$. Here, $h^\star = \inf_{h\in\mathcal{H}}\sum_{t=1}^{T}\ell(h(x_t),y_t)$ is the optimal function in hindsight. First, note that $\ln(|\mathcal{E}_B|) \le K|B|\ln(K)$. Using Jensen's inequality gives $\mathbb{E}\left[\sqrt{2T\ln(|\mathcal{E}_B|)}\right] \le \sqrt{2T^{1+\beta}K\ln K}$. Thus,

$$\mathbb{E}\left[\sum_{t=1}^{T}\ell(\mathcal{Q}(x_t),y_t)\right] \le \underbrace{\mathbb{E}\left[\sum_{t=1}^{T}\ell(E_{B,\phi_B^{h^\star}}(x_t),y_t)\right]}_{\text{(I)}} + M\sqrt{2T^{1+\beta}K\ln K}. \tag{7.1}$$

Upperbounding (I). It now suffices to upperbound $\mathbb{E}\left[\sum_{t=1}^{T}\ell(E_{B,\phi_B^{h^\star}}(x_t),y_t)\right]$. Recall that Lemma E.5.1 gives pointwise

$$\ell(E_{B,\phi_B^{h^\star}}(x_t),y_t) \le \ell(h^\star(x_t),y_t) + \frac{pM}{a}\,\mathbb{E}_{j\sim\mathrm{Unif}([\mathrm{p}])}[\ell(E_{B,\phi_B^{h^\star}}(x_t),\mathrm{BinRel}(h^\star(x_t),j))] \tag{7.2}$$

where $M = \max_{\pi,y}\ell(\pi,y)$ and $a = \min_{\pi,y}\{\ell(\pi,y) \mid \ell(\pi,y)\neq 0\}$. Note that, by definition of the constant M, we further get

$$\begin{aligned}\ell(E_{B,\phi_B^{h^\star}}(x_t),\mathrm{BinRel}(h^\star(x_t),j)) &\le M\,\mathbb{1}\{\ell(E_{B,\phi_B^{h^\star}}(x_t),\mathrm{BinRel}(h^\star(x_t),j)) > 0\} \\ &= M\,\mathbb{1}\{\ell_{\mathrm{sum}}^{@p}(E_{B,\phi_B^{h^\star}}(x_t),\mathrm{BinRel}(h^\star(x_t),j)) > 0\},\end{aligned}$$

where the equality follows from the fact that $\ell \in \mathcal{L}(\ell_{\mathrm{sum}}^{@p})$.

In order to upperbound the indicator above, we need to introduce some more notations. Given the realizable online learner $\mathcal{A}_i^m$ for $(i,m) \in [K]\times[p]$, an instance $x \in \mathcal{X}$, and an ordered finite sequence of labeled examples $L \in (\mathcal{X}\times\{0,1\})^*$, let $\mathcal{A}_i^m(x|L)$ be the random variable denoting the prediction of $\mathcal{A}_i^m$ on the instance x after running and updating on L. For any $b \in \{0,1\}^T$, $h \in \mathcal{H}$, and $t \in [T]$, let $L_{b_{<t}}^h(i,m) = \{(x_s, h_i^m(x_s)) : s < t \text{ and } b_s = 1\}$

denote the *subsequence* of the sequence of labeled instances $\{(x_s, h_i^m(x_s))\}_{s=1}^{t-1}$ where $b_s = 1$. Then, for any $j \in [p]$, we have

$$\mathbb{1}\{\ell_{\text{sum}}^{@p}(E_{B,\phi_B^{h^\star}}(x_t), \text{BinRel}(h^\star(x_t), j)) > 0\} \leq \sum_{i=1}^{K}\sum_{m=1}^{p} \mathbb{1}\{\mathcal{A}_i^m(x_t \mid L_{B_{<t}}^{h^\star}(i,m)) \neq h_i^{\star,m}(x_t)\}.$$

To prove the inequality above, consider the case when $\sum_{i=1}^{K}\sum_{m=1}^{p} \mathbb{1}\{\mathcal{A}_i^m(x_t \mid L_{B_{<t}}^{h^\star}(i,m)) \neq h_i^{\star,m}(x_t)\} = 0$ because the inequality is trivial otherwise. Then, we must have $\mathcal{A}_i^m(x_t \mid L_{B_{<t}}^{h^\star}(i,m)) = h_i^{\star,m}(x_t)$ for all $(i,m) \in [K] \times [p]$. Let $V_t \in \{0,1\}^{K \times p}$ be a binary vote matrix that $E_{B,\phi_B^{h^\star}}$ constructs in round t. Then, we have $V_t[i,m] = \mathcal{A}_i^m(x_t \mid L_{B_{<t}}^{h^\star}(i,m)) = h_i^{\star,m}(x_t)$ for all $(i,m) \in [K] \times [p]$. Since $h^\star(x_t)$ is a permutation, the vote vector $V_t \mathbf{1}_p$ must contain p labels with distinct number of non-zero votes, namely $p, p-1, p-2, \ldots, 2, 1$ votes. Similarly, there must be $K-p$ labels with exactly 0 votes. Thus, every $\hat{\pi}_t \in \arg\min_{\pi \in \mathcal{S}_K} \langle \pi, V_t \mathbf{1}_p \rangle$ must rank label that obtained p votes as 1, label with $p-1$ votes as 2, and so forth. In other words, we must have $\hat{\pi}_t \overset{[p]}{=} h^\star(x_t)$, and thus $\ell_{\text{sum}}^{@p}(\hat{\pi}_t, \text{BinRel}(h^\star(x_t), j)) = 0$ for any $j \in [p]$ by definition of $\ell_{\text{sum}}^{@p}$. Our claim now follows because $E_{B,\phi_B^{h^\star}}(x_t) \in \arg\min_{\pi \in \mathcal{S}_K} \langle \pi, V_t \mathbf{1}_p \rangle$. Using these two inequalities in equation (7.2), we obtain

$$\ell(E_{B,\phi_B^{h^\star}}(x_t), y_t) \leq \ell(h^\star(x_t), y_t) + \frac{pM^2}{a} \sum_{i=1}^{K}\sum_{m=1}^{p} \mathbb{1}\{\mathcal{A}_i^m(x_t \mid L_{B_{<t}}^{h^\star}(i,m)) \neq h_i^{\star,m}(x_t)\},$$

which further implies that

$$\mathbb{E}\left[\sum_{t=1}^{T} \ell(E_{B,\phi_B^{h^\star}}(x_t), y_t)\right] \leq \sum_{t=1}^{T} \ell(h^\star(x_t), y_t) + \frac{pM^2}{a} \sum_{i=1}^{K}\sum_{m=1}^{p} \underbrace{\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}_i^m(x_t \mid L_{B_{<t}}^{h^\star}(i,m)) \neq h_i^{\star,m}(x_t)\}\right]}_{\text{(II)}}.$$

The first term above is the cumulative loss of the best-fixed hypothesis in hindsight.

Upperbounding (II). It now suffices to show that $\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}_i^m(x_t \mid L_{B_{<t}}^{h^\star}(i,m)) \neq h_i^{\star,m}(x_t)\}\right]$ is sub-linear for every $(i,m) \in [K] \times [p]$.

Note that we can write that $\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}_i^m(x_t \mid L_{B_{<t}}^{h^\star}(i,m)) \neq h_i^{\star,m}(x_t)\}\right]$ is equal to

$$\sum_{t=1}^{T} \mathbb{E}\left[\mathbb{1}\{\mathcal{A}_i^m(x_t \mid L_{B_{<t}}^{h^\star}(i,m)) \neq h_i^{\star,m}(x_t)\}\right] \frac{\mathbb{E}\left[\mathbb{1}\{B_t = 1\}\right]}{\mathbb{E}\left[\mathbb{1}\{B_t = 1\}\right]}$$
$$= \frac{T}{T^\beta} \sum_{t=1}^{T} \mathbb{E}\left[\mathbb{1}\{\mathcal{A}_i^m(x_t \mid L_{B_{<t}}^{h^\star}(i,m)) \neq h_i^{\star,m}(x_t)\}\mathbb{1}\left\{B_t = 1\right\}\right],$$

where the last equality follows because $\mathbb{E}\left[\mathbb{1}\{B_t = 1\}\right] = \frac{T^\beta}{T}$ and the prediction of $\mathcal{A}_i^m(x_t \mid L_{B_{<t}}^{h^\star}(i,m))$ on round t only depends on bitstring $(B_1, \ldots, B_{t-1})$, but is independent of B_t. Next, we can use the regret guarantee of algorithm $\mathcal{A}_i^m$ on the rounds it was updated. That is,

$$\begin{aligned}
\sum_{t=1}^{T} \mathbb{E}\left[\mathcal{A}_i^m(x_t \mid L_{B_{<t}}^{h^\star}(i,m))\mathbb{1}\left\{B_t = 1\right\}\right] &= \mathbb{E}\left[\sum_{t:B_t=1} \mathcal{A}_i^m(x_t \mid L_{B_{<t}}^{h^\star}(i,m)) \neq h_i^{\star,m}(x_t)\}\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\sum_{t:B_t=1} \mathcal{A}_i^m(x_t \mid L_{B_{<t}}^{h^\star}(i,m)) \neq h_i^{\star,m}(x_t)\}\right] \middle| B\right] \\
&\leq \mathbb{E}\left[R_i^m(|B|)\right],
\end{aligned}$$

where $R_i^m(|B|)$ is the regret of $\mathcal{A}_i^m$, a sub-linear function of $|B|$. In the last step, we use the fact that $\mathcal{A}_i^m$ is a realizable algorithm for $\mathcal{H}_i^m$ and the feedback that the algorithm received was $(x_t, h_i^{\star,m}(x_t))$ in the rounds whenever $B_t = 1$. Next, Lemma 5.17 from Woess [2017] guarantees that there exists a concave sub-linear function $\tilde{R}_i^m(|B|)$ that upperbounds $R_i^m(|B|)$. Thus, by Jensen's inequality, $\mathbb{E}_B\left[R_i^m(|B|)\right] \leq \mathbb{E}_B\left[\tilde{R}_i^m(|B|)\right] \leq \tilde{R}_i^m(\mathbb{E}_B\left[|B|\right]) \leq \tilde{R}_i^m(T^\beta)$, a sub-linear function of T^β.

Combining (I) and (II) together, we obtain

$$\mathbb{E}\left[\sum_{t=1}^{T} \ell(\mathcal{Q}(x_t), y_t)\right] \leq \inf_{h\in\mathcal{H}} \sum_{t=1}^{T} \ell(h(x_t), y_t) + \frac{pM^2}{a} \sum_{i=1}^{K}\sum_{m=1}^{p} \frac{T}{T^\beta}\, \tilde{R}_i^m(T^\beta) + M\sqrt{2T^{1+\beta}K\ln K}.$$

Since $\tilde{R}_i^m(T^\beta)$ is a sublinear function of T^β, we have that $\frac{T}{T^\beta}\tilde{R}_i^m(T^\beta)$ is a sublinear function of T. As the sum of sublinear functions is sublinear, the second term above must be a sublinear function of T. The regret is sub-linear for any choice of $\beta \in (0,1)$. This completes our proof as we have shown that the algorithm $\mathcal{Q}$ achieves sub-linear regret in T. ■

The proof of the necessity direction of Theorem 7.5.1 also involves constructing experts and running the REWA algorithm. We defer details to Appendix E.4.1.

CHAPTER 8

Estimating the (Un)seen: Sample-dependent Mass Estimation

In this chapter, we study mass estimation for distributions over countably infinite domains, where the objective is to estimate the probability mass of sample-dependent sets. Classical results such as missing mass estimation and its k-heavy-hitters generalizations fit into this framework, but little is known beyond these examples. We introduce a systematic study of mass estimation tasks defined by functions $g : \mathcal{X}^\star \to 2^{\mathcal{X}}$, and identify general conditions under which simple estimators succeed. In particular, we show that the empirical-distribution-based estimator achieves vanishing error whenever the image size of g grows sublinearly with the sample size, and that the leave-one-out estimator works whenever g satisfies a natural stability property. These results unify and extend prior analyses, yielding new guarantees for functionals such as neighboring mass, pierced sets, and structured combinations via unions and intersections. We conclude by broadening our scope to understand the landscape of estimability. To that end, we give an example of a function g that is not estimable and leave open the question of finding matching necessary and sufficient conditions for g to be estimable.

8.1 Introduction

Estimating the probability mass of unseen or structured subsets of a distribution is a fundamental problem at the intersection of probability, statistics, and learning theory. Classical work on missing mass estimation, pioneered by Good [1953] and later refined in minimax and concentration analyses [McAllester and Schapire, 2000, McAllester and Ortiz, 2003], established this task as a statistical primitive, with applications ranging from ecology to language modeling. At its core, missing mass quantifies how much probability remains on outcomes not yet observed in a finite sample. Despite decades of progress, most results largely focus on a few specific functionals, such as the missing mass itself or its k-heavy-hitter variants, leaving open how broadly these ideas can be extended.

In this work, we introduce a unified framework for sample-dependent mass estimation, where the goal is to estimate $D(g(x_{1:n}))$ for general set-valued functions $g : \mathcal{X}^\star \to 2^{\mathcal{X}}$ that map a sample to a subset of the domain. This formulation subsumes classical missing mass and heavy-hitter problems but also captures richer and previously unexplored examples, including neighboring sets, pierced sets, and structured combinations defined by unions and intersections. Within this framework, we identify general conditions under which simple estimators achieve vanishing error, thereby unifying and extending existing positive results. We also exhibit a concrete function g for which consistent estimation is impossible, providing a sharp lower-bound construction that delineates the limits of learnability in this setting.

Beyond their theoretical appeal, these questions have growing practical relevance. Large language models (LLMs) are trained on massive but finite corpora, and assessing how much "unseen mass" remains in their knowledge is essential for evaluation and safety. Recent approaches such as KnowSum and conformal prediction for generative models [Noorani et al., 2025, Li et al., 2025] already exploit missing-mass perspectives to quantify uncertainty. By extending the study of missing mass estimation to general set-valued functions g, our framework enables practitioners to quantify the uncertainty of LLM outputs over structured subsets of their unseen or unobserved generations.

8.2 Related Works

The study of missing mass estimation dates back to the Good–Turing estimator [Good, 1953], which estimates the probability mass of yet-unseen elements in a sample. This problem has since become a central object of study across probability, statistics, and learning theory. Early theoretical work by McAllester and Schapire [2000] provided a PAC-style high-probability confidence interval for the missing mass and its generalization to the mass

of elements that appear exactly k times. In a related line, McAllester and Ortiz [2003] developed concentration inequalities for the missing mass and the error rate of histogram rules. These bounds on the expected missing mass and its concentration properties were further sharpened by Berend and Kontorovich [2012, 2013] and extend to metric spaces by Maurer [2022].

A parallel thread of research has investigated minimax risk for missing mass estimation. Rajaraman et al. [2017] derive near-tight upper and lower bounds on the minimax risk of the Good-Turing estimator as well as estimator-independent information-theoretic lower bounds on the minimax risk. Acharya et al. [2018] sharpened these with improved upper and lower bounds. Complementary work by Valiant and Valiant [2011, 2017] study the ability to estimate the shape of the unobserved portion of the distribution. By doing so, they derive near-optimal estimators for estimating other statistical properties of distributions like their entropy, support size, and distance metrics between pairs of distributions. Paninski [2003] also derived information-theoretic lower bounds for entropy and unseen-probability estimation, connecting these impossibility results to classical Fano and Le Cam arguments [Fano, 1966, LeCam, 1973, Yu, 1997].

Recent works have also explored generalizations and structured settings. Chandra and Thangaraj [2024] study the task of estimating the missing g-mass, which given a positive function g from [0,1] to the reals, asks to estimate the sum of $g(\mathbb{P}(x))$ over the missing letters x. Pananjady et al. [2024] studied missing mass estimation in Markovian sequences, showing that near-optimal rates can still be achieved beyond the i.i.d. setting. These works highlight the robustness of missing mass estimation as a statistical primitive across diverse data-generating processes.

Finally, there has been a resurgence of interest in missing mass through its connections to modern machine learning and generative modeling. Noorani et al. [2025] used a missing-mass perspective to develop conformal prediction methods for uncertainty quantification in generative models. In parallel, the KnowSum framework [Li et al., 2025] applies missing-mass ideas to the evaluation of large language models, quantifying unseen knowledge that standard benchmarks fail to capture. Kalai and Vempala [2024] further connected Good–Turing reasoning to the inevitability of hallucinations in calibrated language models, while the Why Language Models Hallucinate technical report [Kalai et al., 2025] extends this perspective and explicitly credits Good–Turing missing-mass ideas in its theoretical analysis. These applications demonstrate how missing mass, originally motivated by species estimation, continues to inspire methodological advances at the frontier of language modeling.

8.3 Preliminaries

8.3.1 Notation

Let $\mathcal{X}$ be a countably infinite space and $\Delta\mathcal{X}$ denote the set of all distributions over $\mathcal{X}$. For a set $A \subset \mathcal{X}$, we use A^c to denote its complement and $|A|$ to denote its cardinality. For a distribution $D \in \Delta\mathcal{X}$ and a subset $A \subseteq \mathcal{X}$, we use $D(A)$ to denote the cumulative mass that D places on A. That is, $D(A) := \mathbb{P}_{x\sim D}[x \in A]$. As usual, for $n \in \mathbb{N}$, we define $[n] := \{1, \ldots, n\}$. We define $\mathcal{X}^\star$ as the set of all finite sequence of examples in $\mathcal{X}$. Given a sequence $x_1, \ldots, x_n \in \mathcal{X}^\star$, we will often abbreviate it as $x_{1:n}$ or S_n interchangeably. For a sample $x_{1:n} \sim D^n$ and index $i \in [n]$, we will use $x_{1:n}^{-i}$ to denote the sample $x_1, \ldots, x_{i-1}, x_{i+1}, x_n$ resulting from removing the i'th entry and likewise for S_n^{-i}.

8.3.2 Mass Estimation

In mass estimation, one aims to estimate the mass placed by an unknown distribution $D \in \Delta\mathcal{X}$ on a known, potentially sample-dependent, region using i.i.d. samples from D. Throughout the chapter, we will use $g : \mathcal{X}^\star \to \mathcal{X}$ to denote the function which maps a sample $x_{1:n}$ to a subset of $\mathcal{X}$. Throughout the chapter, we will typically assume that both $\mathcal{X}$ and g are known. A mass estimator $\hat{f} : \mathcal{X}^\star \to 2^{\mathcal{X}}$, defined formally in Definition 8.3.1, maps a sample to a number in $[0, 1]$.

Definition 8.3.1 (Mass Estimator)**.** *A mass estimator $\hat{f} : \mathcal{X}^\star \to [0, 1]$ is a deterministic function that maps a finite training sample $x_{1:n} \in \mathcal{X}^\star$ to $[0, 1]$.*

Given knowledge of of g (but not D), our goal is to find an estimator $\hat{f}$ such that $\hat{f}(x_{1:n})$ and $D(g(x_{1:n}))$ are "close", when $x_{1:n} \sim D^n$ is an i.i.d. draw. Definition 8.3.2 makes precise our notion of closeness in terms of the Mean Squared Error (MSE).

Definition 8.3.2 (estimability)**.** *A function $g : \mathcal{X}^\star \to 2^{\mathcal{X}}$ is estimable if there exists an estimator $\hat{f}$ such that*

$$\sup_{D\in\Delta\mathcal{X}} \mathop{\mathbb{E}}_{x_{1:n}\sim D^n} \left[(\hat{f}(x_{1:n}) - D(g(x_{1:n})))^2 \right] = o(1).$$

For some of our results, we will consider the Mean Absolute Deviation (MAD)

$$\mathop{\mathbb{E}}_{x_{1:n}\sim D^n} \left[\left| \hat{f}(x_{1:n}) - D(g(x_{1:n})) \right| \right]$$

instead of MSE. An analogous definition of estimability follows. Note that estimability under the MSE implies estimability under the MAD, but not the other way around.

In this chapter, we are interested in understanding which functions $g : \mathcal{X}^\star \to 2^{\mathcal{X}}$ are estimable. We note that for specific choices of g, estimability is known. Perhaps the most prominent example is that of missing mass estimation, where one takes $g(x_{1:n}) = \{x_{1:n}\}^c$. For this g, it is well known estimation is possible at a rate of $\Theta(1/n)$ for the MSE [Good, 1953]. More generally, if one defines $g(x_{1:n}) = \{x \in \mathcal{X} : \sum_{i=1}^n \mathbf{1}\{x_i = x\} = k\}$, then it is also known that estimation is possible at a rate of $\Theta(k/n)$ for the MSE [McAllester and Schapire, 2000]. Surprisingly, to the best of our knowledge, little is known beyond these examples and this chapter serves to close this gap.

In Sections 8.4 and 8.5, we will focus on two very natural estimators $\hat{f}$: the empirical-distribution-based and the leave-one-out-based estimator. We show that these two estimators are actually sufficient for estimating a very large class of functions g. In Section 8.6, we complement our upper bounds with lower bounds.

8.3.3 Comparison with Classical Estimation

A key difference between the estimation task we consider and classical estimation is whether the parameter we are trying to estimate is random or not. In classical estimation tasks, like mean estimation, the parameter we wish to estimate is fixed and not dependent on the sample the estimator observes. However, it can depend on the sample size n as in the study of U-statistics [Clémenccon et al., 2008]. The following classical analog of our estimation task is obtained by replacing the random quantity $D(g(x_{1:n}))$ with its expectation $\mathbb{E}_{z_{1:n}\sim D^n}[D(g(z_{1:n}))]$ where $z_{1:n}$ are also iid draws from D (and independent of $x_{1:n}$).

Definition 8.3.3 (Classical estimability)**.** *A function $g : \mathcal{X}^\star \to 2^{\mathcal{X}}$ is classically estimable if there exists an estimator $\hat{f}$ such that*

$$\sup_{D\in\Delta\mathcal{X}} \mathop{\mathbb{E}}_{x_{1:n}\sim D^n}\left[\left(\hat{f}(x_{1:n}) - \mathbb{E}_{z_{1:n}\sim D^n}[D(g(z_{1:n}))]\right)^2\right] = o(1).$$

On the other hand, by inspecting Definitions 8.3.2, it becomes clear that for the estimation task we consider, the parameter we hope to estimate is a function of the realized sample, and hence a random variable itself. One might wonder how our notion of estimability compares to its classical counterpart. In particular, is mass estimation stronger than classical estimation? In Appendix F.3, we provide a partial answer by giving a function g that is mass estimable, but not classically estimable. In addition to establishing a separation, this result also highlights that traditional lower bounds techniques, like LeCam's two-point method [LeCam, 1973], no longer work for mass estimation.

8.4 Estimation via the Empirical Distribution

Perhaps the simplest estimator $\hat{f}$ is the one which constructs the empirical distribution $\hat{D}$ over its samples $x_1, \ldots, x_n$ and returns $\mathbb{P}_{\hat{x} \sim \hat{D}}[\hat{x} \in g(x_{1:n})]$ or equivalently,

$$\hat{f}^{\text{emp}}(x_{1:n}) := \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\{x_i \in g(x_{1:n})\}.$$

In this section, we aim to understand for which functions $g : \mathcal{X}^\star \to 2^{\mathcal{X}}$ is $\hat{f}^{\text{emp}}$ a valid estimator. Surprisingly, we show that the empirical-distribution-based estimator is remarkably robust; it estimates any $g : \mathcal{X}^\star \to 2^{\mathcal{X}}$ whose image space is sufficiently slow growing. To make this more precise, define

$$\tau_g(n) := \sup_{D \in \Delta\mathcal{X}} \mathop{\mathbb{E}}_{x_{1:n} \sim D^n} |g(x_{1:n})| \tag{8.1}$$

as the expected cardinality of g. Then, the following theorem shows that estimation is possible via the empirical distribution-based estimator as long as $\tau_g(n) = o(n)$.

Theorem 8.4.1 (Estimation via the Empirical Distribution-based Estimator)**.** *Let $g : \mathcal{X}^\star \to 2^{\mathcal{X}}$ be any function such that $\tau_g(n) < \infty$ for all $n \geq 1$. Then,*

$$\sup_{D \in \Delta\mathcal{X}} \mathop{\mathbb{E}}_{x_{1:n} \sim D^n} \left[\left| \hat{f}^{emp}(x_{1:n}) - D(g(x_{1:n})) \right| \right] \leq \sqrt{\frac{\tau_g(n)}{n}}.$$

Moreover, we have that

$$\inf_{\hat{f}} \sup_{D \in \Delta\mathcal{X}} \mathop{\mathbb{E}}_{x_{1:n} \sim D^n} \left[\left| \hat{f}(x_{1:n}) - D(g(x_{1:n})) \right| \right] \leq \sqrt{\frac{\min\{\tau_g(n), \tau_{g^c}(n)\}}{n}},$$

where g^c is defined as $g^c(x_{1:n}) = g(x_{1:n})^c$.

Proof. The second claim follows from that the fact that any estimator for g^c can be converted into an estimator for estimating g with an identical rate. Hence, we focus on proving the first claim. We can write

$$\mathbf{1}\{x_i \in g(x_{1:n})\} = \sum_{x \in \mathbb{N}} \mathbf{1}\{x = x_i\} \mathbf{1}\{x \in g(x_{1:n})\}.$$

Hence,

$$\hat{f}^{\text{emp}}(x_{1:n}) := \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{x_i \in g(x_{1:n})\} = \frac{1}{n}\sum_{i=1}^{n}\sum_{x\in\mathbb{N}} \mathbf{1}\{x = x_i\}\mathbf{1}\{x \in g(x_{1:n})\}.$$

By Tonelli's theorem we can swap the sums to get

$$\hat{f}^{\text{emp}}(x_{1:n}) = \sum_{x\in\mathbb{N}} \left(\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{x = x_i\}\right) \mathbf{1}\{x \in g(x_{1:n})\}.$$

Next, observe that

$$D(g(x_{1:n})) = \sum_{x\in g(x_{1:n})} D(x) = \sum_{x\in\mathbb{N}} D(x)\mathbf{1}\{x \in g(x_{1:n})\}.$$

Now,

$$\left|\hat{f}^{\text{emp}}(x_{1:n}) - D(g(x_{1:n})\right| = \left|\sum_{x\in\mathbb{N}} \left(\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{x = x_i\}\right) \mathbf{1}\{x \in g(x_{1:n})\} - D(x)\mathbf{1}\{x \in g(x_{1:n})\}\right|.$$

Factoring and rearranging gives

$$\begin{aligned}\left|\hat{f}^{\text{emp}}(x_{1:n}) - D(g(x_{1:n})\right| &= \left|\sum_{x\in\mathbb{N}} \left(\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{x = x_i\} - D(x)\right) \mathbf{1}\{x \in g(x_{1:n})\}\right| \\ &\leq \sum_{x\in\mathbb{N}} \left|\left(\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{x = x_i\} - D(x)\right)\right| \mathbf{1}\{x \in g(x_{1:n})\},\end{aligned}$$

where the last step is due to the Triangle inequality and the fact that

$$\sum_{x\in\mathbb{N}} \left|\left(\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{x = x_i\} - D(x)\right)\right| \mathbf{1}\{x \in g(x_{1:n})\} < \infty.$$

Taking expectation on both sides, we get that $\mathbb{E}_{x_{1:n}\sim D^n}\left[\left|\hat{f}^{\text{emp}}(x_{1:n}) - D(g(x_{1:n})\right|\right]$ is at most

$$\begin{aligned}\mathbb{E}_{x_{1:n}\sim D^n}\left[\sum_{x\in\mathbb{N}} \left|\left(\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{x = x_i\} - D(x)\right)\right| \mathbf{1}\{x \in g(x_{1:n})\}\right] = \\ \sum_{x\in\mathbb{N}} \mathbb{E}_{x_{1:n}\sim D^n}\left[\left|\left(\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{x = x_i\} - D(x)\right)\right| \mathbf{1}\{x \in g(x_{1:n})\}\right],\end{aligned}$$

where we swap the expectation and sum in the last line by invoking the Monotone Convergence Theorem. By Cauchy-Schwartz's inequality, we can upper bound

$$\mathbb{E}_{x_{1:n}\sim D^n}\left[\left|\left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\{x=x_i\}-D(x)\right)\right|\mathbf{1}\{x\in g(x_{1:n})\}\right]\leq \sqrt{\mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\{x=x_i\}-D(x)\right)^2\right]\mathbb{E}\left[\left(\mathbf{1}\{x\in g(x_{1:n})\}\right)^2\right]}.$$

Since $\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\{x=x_i\}$ is an unbiased estimator of $D(x)$, we have that

$$\mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\{x=x_i\}-D(x)\right)^2\right]=\mathrm{Var}_{x_{1:n}\sim D^n}\left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\{x=x_i\}\right)=\frac{D(x)(1-D(x))}{n}.$$

Moreover, $\mathbb{E}\left[\left(\mathbf{1}\{x\in g(x_{1:n})\}\right)^2\right]=\mathbb{P}\left[x\in g(x_{1:n})\right]$. Plugging this in gives that

$$\sqrt{\mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\{x=x_i\}-D(x)\right)^2\right]\mathbb{E}\left[\left(\mathbf{1}\{x\in g(x_{1:n})\}\right)^2\right]}\leq \sqrt{\frac{D(x)(1-D(x))\mathbb{P}\left[x\in g(x_{1:n})\right]}{n}}.$$

Thus, we have that $\mathbb{E}_{x_{1:n}\sim D^n}\left[\left|\hat{f}(x_{1:n})-D(g(x_{1:n})\right|\right]$ is at most

$$\frac{1}{\sqrt{n}}\sum_{x\in\mathbb{N}}\sqrt{D(x)(1-D(x))\mathbb{P}\left[x\in g(x_{1:n})\right]}\leq \frac{1}{\sqrt{n}}\sqrt{\sum_{x\in\mathbb{N}}D(x)(1-D(x))}\sqrt{\sum_{x\in\mathbb{N}}\mathbb{P}\left[x\in g(x_{1:n})\right]},$$

where the inequality follows by Cauchy-Schwartz's inequality again. Observing that

$$\sqrt{\sum_{x\in\mathbb{N}}D(x)(1-D(x))}\leq\sqrt{\sum_{x\in\mathbb{N}}D(x)}=1$$

and

$$\sqrt{\sum_{x\in\mathbb{N}}\mathbb{P}\left[x\in g(x_{1:n})\right]}=\sqrt{\mathbb{E}_{x_{1:n}\sim D^n}\left[|g(x_{1:n})|\right]}.$$

gives that

$$\mathop{\mathbb{E}}_{x_{1:n}\sim D^n}\left[\left|\hat{f}^{\text{emp}}(x_{1:n}) - D(g(x_{1:n}))\right|\right] \leq \sqrt{\frac{\mathbb{E}_{x_{1:n}\sim D^n}\left[|g(x_{1:n})|\right]}{n}}.$$

Noting that

$$\sup_{D\in\Delta\mathcal{X}} \mathop{\mathbb{E}}_{x_{1:n}\sim D^n}\left[\left|\hat{f}^{\text{emp}}(x_{1:n}) - D(g(x_{1:n}))\right|\right] \leq \sup_{D\in\Delta\mathcal{X}} \sqrt{\frac{\mathbb{E}_{x_{1:n}\sim D^n}\left[|g(x_{1:n})|\right]}{n}} \leq \sqrt{\frac{\tau_g(n)}{n}},$$

completes the proof. ■

The upper bound in Theorem 8.4.1 is in terms of the worst-case expected cardinality of g. If instead, we have that $\sup_{x_{1:n}\in\mathcal{X}^n} g(x_{1:n}) = o(n)$, then we can upgrade Theorem 8.4.1 to hold for the MSE loss after replacing $\tau_g(n)$ with $\sup_{x_{1:n}\in\mathcal{X}^n} g(x_{1:n})$. We provide the proof of this in Appendix F.2. Note that $\tau_g(n)$ can be much smaller than $\sup_{x_{1:n}\in\mathcal{X}^n} g(x_{1:n})$. As a simple example, let $\mathcal{X} = \mathbb{N}$, and consider a g which outputs $[n]$ if $x_{1:n}$ contains exactly half 1's and half 2's and otherwise always outputs $\{1\}$. Then, one can verify that $\sup_{x_{1:n}\in\mathcal{X}^n} |g(x_{1:n})| = n$ while $\tau_g(n) = O\left(\sqrt{n}\right)$, where the last inequality is due to Khintchine's inequality (see Lemma A.9 in Cesa-Bianchi and Lugosi [2006]). We leave proving an upper bound on MSE in terms of τ_g as an open question.

Beyond cardinality, empirical-distribution-based estimation allows us to use tools from generalization theory to provide alternate sufficient conditions. We make this connection concrete in Section 8.4.1.

8.4.1 Empirical-distribution Estimation Rates via Generalization

If one picks $\hat{f}^{\text{emp}}$, then estimation is equivalent to a special case of generalization. To see why, define $\mathcal{Z} = 2^{\mathcal{X}}$ and an algorithm $\mathcal{A} : \mathcal{X}^\star \to 2^{\mathcal{X}}$ such that on input $x_{1:n} \in \mathcal{X}^n$, we have that $\mathcal{A}(x_{1:n}) = g(x_{1:n})$. Define the loss function $\ell : \mathcal{Z} \to \mathcal{X}$ as $\ell(Z, x) := \mathbf{1}\{x \in Z\}$. Then, observe that the minimax MSE of the empirical-distribution-based estimator $\sup_{D\in\Delta\mathcal{X}} \mathbb{E}_{x_{1:n}\sim D^n}\left[\left(\hat{f}^{\text{emp}}(x_{1:n}) - D(g(x_{1:n}))\right)^2\right]$ can be written as

$$\sup_{D\in\Delta\mathcal{X}} \mathop{\mathbb{E}}_{x_{1:n}\sim D^n}\left[\left(\frac{1}{n}\sum_{i=1}^n \mathbf{1}\{x_i \in g(x_{1:n})\} - D(g(x_{1:n}))\right)^2\right] =$$

$$\sup_{D\in\Delta\mathcal{X}} \mathop{\mathbb{E}}_{x_{1:n}\sim D^n}\left[\left(\frac{1}{n}\sum_{i=1}^n \ell(\mathcal{A}(x_{1:n}), x_i) - \mathbb{E}_{x\sim D}\left[\ell(\mathcal{A}(x_{1:n}), x)\right]\right)^2\right]$$

where the last line is exactly the generalization error for algorithm $\mathcal{A}$ with respect to loss function ℓ. By viewing estimation through the lens of generalization, we can now use existing bounds on generalization error to obtain bounds on estimation MSE. We give one such example based on uniform convergence and the VC dimension below.

For a sample size $n \in \mathbb{N}$, define $\mathcal{H}_n := \{g(S_n) | S_n \in \mathcal{X}^n\}$ as the hypothesis class induced by the image space of g on $\mathcal{X}^n$. Corollary 8.4.2 bounds the MSE of the empirical-distribution-based estimator in terms of the VC dimension [Vapnik and Chervonenkis, 1971] of $\mathcal{H}_n$, denoted by $\mathrm{VC}(\mathcal{H}_n)$. See Appendix F.1 for the definition of the VC dimension.

Corollary 8.4.2. *Let $g : \mathcal{X}^\star \to 2^{\mathcal{X}}$ be any function such that $\mathrm{VC}(\mathcal{H}_n) < \infty$ for all $n \in \mathbb{N}$. Then,*

$$\sup_{D \in \Delta\mathcal{X}} \mathop{\mathbb{E}}_{x_{1:n} \sim D^n} \left[\left(\hat{f}^{\underline{emp}}(x_{1:n}) - D(g(x_{1:n})) \right)^2 \right] \leq O\left(\frac{\mathrm{VC}(\mathcal{H}_n)}{n} \right).$$

Proof. Fix a distribution $D \in \Delta\mathcal{X}$ and sample size $n \in \mathbb{N}$. Then, standard uniform convergence bounds (c.f. Theorem 6.8 in Shalev-Shwartz and Ben-David [2014b]) implies that for every $\varepsilon > 0$

$$\mathbb{P}_{x_{1:n} \sim D^n} \left[\underbrace{\sup_{h \in \mathcal{H}_n} \left| \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\{x_i \in h\} - D(h) \right|}_{=:Z} > \varepsilon \right] \leq \min\Big\{ \exp(\mathrm{VC}(\mathcal{H}_n) - \frac{\varepsilon^2 n}{C}), 1 \Big\},$$

where $C > 0$ is some universal constant. In other words, for any $z > 0$, we have

$$\mathbb{P}_{x_{1:n} \sim D^n} \left[Z^2 > z \right] \leq \min\Big\{ \exp(\mathrm{VC}(\mathcal{H}_n) - \frac{zn}{C}), 1 \Big\}.$$

Thus,

$$\mathbb{E}_{x_{1:n} \sim D^n}[Z^2] = \int_{z=0}^{\infty} \mathbb{P}_{x_{1:n} \sim D^n} \left[Z^2 > z \right] dz \leq \int_{z=0}^{\infty} \min\Big\{ \exp(\mathrm{VC}(\mathcal{H}_n) - \frac{zn}{C}), 1 \Big\} dz = \frac{C(\mathrm{VC}(\mathcal{H}_n) + 1)}{n},$$

which completes the proof. ■

As a corollary, we get that any g with $\mathrm{VC}(\mathcal{H}_n) = o(n)$ is estimable under the MSE via the empirical-distribution-based estimator. In the same way as Corollary 8.4.2, one can get other sufficient conditions for estimability by upper bounding the generalization error in terms of the Conditional Mutual Information [Steinke and Zakynthinou, 2020] and various notions of stability [Bousquet and Elisseeff, 2002, Shalev-Shwartz et al., 2010, Feldman and

Vondrak, 2018, 2019], all of which become properties of g since $\mathcal{A}(x_{1:n}) = g(x_{1:n})$. Since these arguments more or less just require plugging in existing upper bounds on generalization error, we omit them here.

8.5 Estimation via the Leave-One-Out Estimator

In Section 8.4, we identified generic conditions on g for which the empirical distribution-based estimator enjoys sublinear estimation error. One might wonder if such an empirical distribution-based estimator can work for all possible g's. Unfortunately, this is not the case even for the well-studied missing mass estimation problem $g(x_{1:n}) = \{x_{1:n}\}^c$. Motivated by this example, we study the leave-one-out estimator $\hat{f}^{\text{loo}}$ which, given a sample $x_{1:n}$, computes

$$\hat{f}^{\text{loo}}(x_{1:n}) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{x_i \in g(x_{1:n}^{-i})\}.$$

It turns out that while the missing mass g is not estimable via the empirical-distribution-based estimator, it is estimable using the leave-one-out estimator. In fact, we show that leave-one-out estimator is a valid estimator for any sufficiently "stable" function g. Our notion of stability involves the following quantity:

$$\gamma_g(n) := \sup_{D \in \Delta\mathcal{X}} \mathbb{E}_{x_{1:n}\sim D^n}\left[D(g(x_{1:n})\Delta g(x_{1:n-1}))\right]. \tag{8.2}$$

We say that g is $\gamma_g(n)$-stable if $\gamma_g(n) = o(1)$. Theorem 8.5.1 shows that when $\gamma_g(n) = o(1)$, estimation is possible via the leave-one-out-based estimator.

Theorem 8.5.1 (Estimation via the Leave-One-Out-based Estimator)**.** *Let $g : \mathcal{X}^\star \to 2^{\mathcal{X}}$ be $\gamma_g(n)$-stable. Then,*

$$\sup_{D\in\Delta\mathcal{X}} \mathop{\mathbb{E}}_{x_{1:n}\sim D^n}\left[\left(\hat{f}^{\text{loo}}(x_{1:n}) - D(g(x_{1:n})\right)^2\right] \le \frac{2}{n} + 6\gamma_g(n).$$

Proof. Given a sample $S_n := x_{1:n}$, we define a "classifier" for test points x which classifies x as positive if $x \notin g(S_n)$ and negative otherwise. Thus, the 0-1 loss of this classifier on test point x is $\ell_{S_n}(x) := \mathbf{1}[x \in g(S_n)]$. We now appeal to the results of Kumar et al. [2013], building on Kale et al. [2011], which bound the variance of the leave-one-out estimator in terms of the mean-square stability defined as

$$\mathbb{E}[(\ell_{S_n}(x) - \ell_{\tilde{S}_n}(x))^2],$$

where $S_n \sim D^n$, $x \sim D$ is a random test point, and $\tilde{S}_n$ is the sample where a random point in S_n is replaced by a fresh random point $x' \sim D$. Let $\hat{S}_{n-1}$ be the $n-1$ points in S_n that are not replaced to form $\tilde{S}_n$. For our classifier, the mean-square-stability equals

$$\begin{aligned}
\mathbb{E}[(\mathbf{1}[x \in g(S_n)] - \mathbf{1}[x \in g(\tilde{S}_n)])^2] &= \mathbb{E}[|\mathbf{1}[x \in g(S_n)] - \mathbf{1}[x \in g(\tilde{S}_n)]|] \\
&\leq \mathbb{E}[|\mathbf{1}[x \in g(S_n)] - \mathbf{1}[x \in g(\hat{S}_{n-1})]|] \\
&\quad + \mathbb{E}[|\mathbf{1}[x \in g(\hat{S}_{n-1})] - \mathbf{1}[x \in g(\tilde{S}_n)]|] \\
&= 2\mathbb{E}[|\mathbf{1}[x \in g(S_n)] - \mathbf{1}[x \in g(\hat{S}_{n-1})]|] \\
&= 2\mathbb{E}[D(g(S_n)\Delta g(\hat{S}_{n-1}))] \\
&\leq 2\gamma_g(n).
\end{aligned}$$

Given S_n, define

$$\bar{f}(S_n) := \frac{1}{n}\sum_{i=1}^{n} D(g(S_n^{-i}))$$

and note that

$$\mathbb{E}_{S_n \sim D^n}\left[\hat{f}^{\text{loo}}(S_n) - \bar{f}(S_n)\right] = 0.$$

Kumar et al. [2013] give the following bound on the variance of $\hat{f}^{\text{loo}}(S_n) - \bar{f}(S_n)$:

$$\text{Var}(\hat{f}^{\text{loo}}(S_n) - \bar{f}(S_n)) \leq \frac{1}{n}\,\text{Var}(\mathbf{1}[x_1 \in g(S_n^{-1})] - D(g(S_n^{-1})]) + \left(1 - \frac{1}{n}\right) \cdot 2\gamma_g(n) \leq \frac{1}{n} + 2\gamma_g(n),$$

where the last inequality follows since

$$\text{Var}(\mathbf{1}[x_1 \in g(S_n^{-1})] - D(g(S_n^{-1}))]) = \mathbb{E}[D(g(S_n^{-1})(1 - D(g(S_n^{-1}))))] \leq 1.$$

Now, we have

$$
\begin{aligned}
\mathbb{E}_{S_n\sim D^n}\left[\left(\hat{f}^{\text{loo}}(S_n) - D(g(S_n))\right)^2\right] &\le 2\Big(\mathbb{E}\left[\left(\hat{f}^{\text{loo}}(S_n) - \bar{f}(S_n)\right)^2\right] \\
&\quad + \mathbb{E}\left[\left(\bar{f}(S_n) - D(g(S_n))\right)^2\right]\Big) \\
&\le 2\Big(\operatorname{Var}\left(\hat{f}^{\text{loo}}(S_n) - \bar{f}(S_n)\right) \\
&\quad + \mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^{n} D(g(S_n^{-i})) - D(g(S_n))\right)^2\right]\Big) \\
&\le 2\Big(\operatorname{Var}\left(\hat{f}^{\text{loo}}(S_n) - \bar{f}(S_n)\right) \\
&\quad + \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\left(D(g(S_n^{-i})) - D(g(S_n))\right)^2\right]\Big) \\
&= 2\Big(\operatorname{Var}\left(\hat{f}^{\text{loo}}(S_n) - \bar{f}(S_n)\right) \\
&\quad + \mathbb{E}\left[\left(D(g(S_n^{-n})) - D(g(S_n))\right)^2\right]\Big) \\
&\le 2\Big(\operatorname{Var}\left(\hat{f}^{\text{loo}}(S_n) - \bar{f}(S_n)\right) \\
&\quad + \mathbb{E}\left[\!\left[D(g(S_n^{-n})) - D(g(S_n))\right]\!\right]\Big) \\
&\le \frac{2}{n} + 6\gamma_g(n).
\end{aligned}
$$

which completes the proof. ■

8.5.1 Applications

In this section, we upper bound $\gamma_g(n)$ for several natural choices of g and compute $\gamma_g(n)$. Our results recover well-known results, like that missing mass estimation is possible at a MSE rate of $\frac{1}{n}$, while also showing that estimation is possible for new choices of g.

8.5.1.1 Estimating Mass of Neighbors

In this section, we study the estimability of natural g functions whose output size can be much larger than n. These functions g are best described by considering a directed graph $G = (V, E)$, where the vertex set $V = \mathcal{X}$ and the edge set E is defined by a function $h : \mathcal{X} \to 2^{\mathcal{X}}$. In particular, given such a function h and a vertex $x \in \mathcal{X}$, we take $h(x)$ as the

set of outgoing edges from x. Given such a function h, we consider functions g of the form

$$g(x_{1:n}) = \bigcup_{i\in[n]} h(x_i).$$

On a sample $x_{1:n} \sim D^n$, g returns the the set of all outgoing neighbors across the examples in $x_{1:n}$ and thus, $D(g(x_{1:n}))$ measures the mass placed by D on all the neighbors of the sample. The following theorem shows that for any neighboring function h, its corresponding g function is estimable.

Theorem 8.5.2. *Let $h : \mathcal{X} \to 2^{\mathcal{X}}$ be any function and consider the function $g(x_{1:n}) = \bigcup_{i\in[n]} h(x_i)$. Then,*

$$\sup_{D\in\Delta\mathcal{X}} \mathop{\mathbb{E}}_{x_{1:n}\sim D^n}\left[\left(\hat{f}^{\text{loo}}(x_{1:n}) - D(g(x_{1:n}))\right)^2\right] = O\left(\frac{1}{n}\right).$$

Proof. By Theorem 8.5.1 it suffices to bound $\gamma_g(n)$. To that end, observe that

$$\begin{aligned}
\mathbb{E}_{x_{1:n}\sim D^n}\left[D(g(x_{1:n}))\Delta g(x_{1:n}^{-n})\right] &= \mathbb{E}_{x_{1:n}}\left[D\left(h(x_n)\setminus\bigcup_{i<n} h(x_i)\right)\right] \\
&= \mathbb{P}_{x\sim D,\, x_{1:n}\sim D^n}\left[x \in h(x_n)\setminus\bigcup_{i<n} h(x_i)\right] \\
&= \mathbb{P}_{x\sim D,\, x_{1:n}\sim D^n}\Big[x \in h(x_n) \\
&\quad \wedge\ x \notin h(x_1) \wedge x \notin h(x_2) \wedge \cdots \wedge x \notin h(x_{n-1})\Big] \\
&= \mathbb{E}_{x\sim D}\Big[\mathbb{P}_{x_{1:n}\sim D^n}\big[x \in h(x_n) \\
&\quad \wedge\ x \notin h(x_1) \wedge x \notin h(x_2) \wedge \cdots \wedge x \notin h(x_{n-1}) \ \big|\ x\big]\Big] \\
&= \mathbb{E}_{x\sim D}\Big[D(h^{-1}(x))\big(1 - D(h^{-1}(x))\big)^{n-1}\Big] \\
&\le \mathbb{E}_{x\sim D}\left[\frac{1}{n}\right] = \frac{1}{n}.
\end{aligned}$$

Thus, $\gamma_g(n) \le \frac{1}{n}$, and hence g is estimable with MSE that decays at a rate of $O(\frac{1}{n})$. ■

Since estimability is preserved under complements, we get the following corollary.

Corollary 8.5.3. *Let $h : \mathcal{X} \to 2^{\mathcal{X}}$ be any function and consider the function $g(x_{1:n}) =$*

$\bigcap_{i\in[n]} h(x_i)$. *Then,*

$$\inf_{\hat{f}} \sup_{D\in\Delta\mathcal{X}} \mathop{\mathbb{E}}_{x_{1:n}\sim D^n} \left[\left(\hat{f}(x_{1:n}) - D(g(x_{1:n})\right)^2\right] = O\left(\frac{1}{n}\right).$$

Note that a special case of Corollary 8.5.3 is the classical missing mass estimator $g(x_{1:n}) = \{x_{1:n}\}^c$. Hence, these results provide, to the best of our knowledge, an alternate proof of the fact that missing mass can be estimated at a MSE rate of $O(\frac{1}{n})$.

8.5.1.2 Estimating Mass of Pierced Sets

In many applications, one is interested in estimating the mass of examples in-sample that satisfy certain properties. For example, in ecology, one might be interested in estimating the mass of species with a predetermined anomaly [Good, 1953, Chao and Lee, 1992, Bunge et al., 2014]. Motivated by these applications, we consider functions g of the form

$$g(x_{1:n}) = \{x_{1:n}\} \cap A_n$$

where $A_n \subseteq \mathcal{X}$ is a prespecified "piercing" set. Our main theorem shows that when the sequence $\{A_n\}_{n\in\mathbb{N}}$ is fixed (i.e. does not depend on the realized samples $x_{1:n}$), then estimation is possible at a rate of $O(1/n)$.

Theorem 8.5.4 (In-Sample Piercing)**.** *Let $\{A_n\}_{n\in\mathbb{N}}$ be any sequence where $A_n \subseteq \mathcal{X}$. Let $g(x_{1:n}) = \{x_{1:n}\} \cap A_n$. Then,*

$$\sup_{D\in\Delta\mathcal{X}} \mathop{\mathbb{E}}_{x_{1:n}\sim D^n} \left[\left(\hat{f}^{\text{loo}}(x_{1:n}) - D(g(x_{1:n})\right)^2\right] \leq O\left(\frac{1}{n}\right).$$

Proof. By Theorem 8.5.1, it suffices to prove the upper bound $\gamma_g(n) \leq \frac{1}{n}$. Fix a $D \in \Delta\mathcal{X}$. Note that

$$\begin{aligned}
\mathbb{E}_{x_{1:n}\sim D^n}\left[D(g(x_{1:n})\Delta g(x_{1:n-1}))\right] &= \mathbb{E}_{x_{1:n}\sim D^n}\left[D((\{x_{1:n}\}\cap A_n)\Delta(\{x_{1:n-1}\}\cap A_n))\right] \\
&= \mathbb{E}_{x_{1:n}\sim D^n}\left[D((\{x_{1:n}\}\Delta\{x_{1:n-1}\})\cap A_n)\right] \\
&= \mathbb{E}_{x_{1:n}\sim D^n}\left[D(x_n)\mathbf{1}\{x_n \in A_n \setminus \{x_{1:n-1}\}\})\right] \\
&\leq \mathbb{E}_{x_{1:n}\sim D^n}\left[D(x_n)\mathbf{1}\{x_n \in \mathcal{X} \setminus \{x_{1:n-1}\}\})\right] \\
&= \mathbb{E}_{x_{1:n}\sim D^n}\left[D(x_n)(1-D(x_n))^{n-1}\right] \leq \frac{1}{n},
\end{aligned}$$

which completes the proof. ■

Like the case of Corollary 8.5.3, we get the following corollary by using DeMorgan's law.

Corollary 8.5.5. *Let $\{A_n\}_{n\in\mathbb{N}}$ be any sequence where $A_n \subseteq \mathcal{X}$. Let $g(x_{1:n}) = \{x_{1:n}\}^c \cup A_n$. Then,*

$$\inf_{\hat{f}} \sup_{D\in\Delta\mathcal{X}} \mathop{\mathbb{E}}_{x_{1:n}\sim D^n} \left[\left(\hat{f}(x_{1:n}) - D(g(x_{1:n}))\right)^2\right] \leq O\left(\frac{1}{n}\right).$$

By taking $A_n = \emptyset$ for all $n \in \mathbb{N}$, one gets that the classical setting of missing mass estimation is a special case of Corollary 8.5.5. It is also interesting to consider the task of estimating the mass of missing examples with certain properties. The following result shows that this too is possible.

Theorem 8.5.6 (Out-of-Sample Piercing)**.** *Let $\{A_n\}_{n\in\mathbb{N}}$ be any sequence where $A_n \subseteq \mathcal{X}$. Let $g(x_{1:n}) := \{x_{1:n}\}^c \cap A_n$. Then,*

$$\sup_{D\in\Delta\mathcal{X}} \mathop{\mathbb{E}}_{x_{1:n}\sim D^n} \left[\left|\hat{f}^{\text{loo}}(x_{1:n}) - D(g(x_{1:n}))\right|\right] \leq O\left(\frac{1}{n}\right).$$

Proof. By Theorem 8.5.1, it suffices to prove the upper bound $\gamma_g(n) \leq \frac{1}{n}$. Fix a $D \in \Delta\mathcal{X}$. Note that

$$\begin{aligned}
\mathbb{E}_{x_{1:n}\sim D^n}\left[D(g(x_{1:n})\Delta g(x_{1:n-1}))\right] &= \mathbb{E}_{x_{1:n}\sim D^n}\left[D((\{x_{1:n}\}^c \cap A_n)\Delta(\{x_{1:n-1}\}^c \cap A_n))\right] \\
&= \mathbb{E}_{x_{1:n}\sim D^n}\left[D((\{x_{1:n}\}^c\Delta\{x_{1:n-1}\}^c) \cap A_n)\right] \\
&= \mathbb{E}_{x_{1:n}\sim D^n}\left[D(x_n)\mathbf{1}\{x_n \in A_n \setminus \{x_{1:n-1}\}\})\right] \\
&\leq \mathbb{E}_{x_{1:n}\sim D^n}\left[D(x_n)\mathbf{1}\{x_n \in \mathcal{X} \setminus \{x_{1:n-1}\}\})\right] \\
&= \mathbb{E}_{x_{1:n}\sim D^n}\left[D(x_n)(1 - D(x_n))^{n-1}\right] \leq \frac{1}{n},
\end{aligned}$$

which completes the proof. ■

We can also get that unions are estimable by noting that $D(A_n \cup \{x_{1:n}\}) = D(\{x_{1:n}\}) + D(A_n) - D(A_n \cap \{x_{1:n}\})$. $D(\{x_{1:n}\})$ and $D(A_n \cap \{x_{1:n}\})$ can be estimated by a leave-one-out estimator, while $D(A_n)$ can be estimated via the empirical-distribution-based estimator.

8.5.2 Estimating Mass of Functions of Heavy-Hitters

One important generalization of missing mass estimation is estimating the mass of heavy hitters. Given a sample $x_{1:n}$ and $k \in \mathbb{N}$, define $c_k(x_{1:n}) \subseteq \{x_{1:n}\}$ to be the set of examples $x \in \mathcal{X}$ that occur exactly k times in $x_{1:n}$ (i.e. the k-heavy hitters). When $g(x_{1:n}) = c_k(x_{1:n})$, our estimation task reduces exactly to that of estimating the k-heavy hitters, for which estimability is known to be possible at a rate of $O(\frac{k}{n})$ for the MSE [McAllester and Schapire, 2000]. Lemma 8.5.7 shows that $\gamma_g(n) \leq \frac{k}{n}$ when $g(x_{1:n}) = c_k(x_{1:n})$. Hence, by Theorem 8.5.1, we recover this known result.

Lemma 8.5.7. *Fix $k \in \mathbb{N}$ and let $g(x_{1:n}) = c_k(x_{1:n})$ Then, $\sup_{D \in \Delta\mathcal{X}} \gamma_{g,D}(n) \leq \frac{k}{n}$.*

Proof. Fix $k \in \mathbb{N}$ and a distribution $D \in \Delta\mathcal{X}$. Then, observe that

$$\begin{aligned}
\mathbb{E}_{x_{1:n} \sim D^n}\left[D(g(x_{1:n}) \Delta g(x_{1:n-1}))\right] &= \mathbb{E}_{x_{1:n} \sim D^n}\left[D(x_n) \mathbf{1}\{x_n \in c_{k-1}(x_{1:n-1})\}\right] \\
&= \mathbb{E}_{x_{1:n} \sim D^n}\left[D(x_n)\binom{n-1}{k-1} D(x_n)^{k-1}(1 - D(x_n))^{n-k}\right] \\
&= \frac{k}{n}\mathbb{E}_{x_{1:n} \sim D^n}\left[\binom{n}{k} D(x_n)^{k}(1 - D(x_n))^{n-k}\right] \leq \frac{k}{n},
\end{aligned}$$

which completes the proof. ■

Next, we consider estimating the mass of the neighbors of the k-heavy hitters. Unlike Theorem 8.5.2, we will need the additional assumption that $\sup_{x \in \mathcal{X}} |h^{-1}(x)| < \infty$.

Theorem 8.5.8. *Let $h : \mathcal{X} \to 2^{\mathcal{X}}$ be any function such that $r := \sup_{x \in \mathcal{X}} |h^{-1}(x)| < \infty$. For $k \geq 1$, consider the function $g(x_{1:n}) = \bigcup_{x \in c_k(x_{1:n})} h(x)$. Then,*

$$\sup_{D \in \Delta\mathcal{X}} \mathop{\mathbb{E}}_{x_{1:n} \sim D^n}\left[\left(\hat{f}^{\text{loo}}(x_{1:n}) - D(g_k(x_{1:n}))\right)^2\right] = O\left(\frac{rk}{n}\right).$$

Proof. Fix $k \in \mathbb{N}$ and a distribution $D \in \Delta\mathcal{X}$. It suffices to show $\gamma_g(n) = O\left(\frac{rk}{n}\right)$. Fix a sample size $n \geq 1$ and a sample $x_{1:n} \in \mathcal{X}^n$. Observe that

$$D(g(x_{1:n}) \Delta g(x_{1:n-1})) \leq \mathbf{1}\{x_n \in c_{k-1}(x_{1:n-1}) \cup c_k(x_{1:n-1}\} D(h(x_n)).$$

Hence, $\mathbb{E}_{x_{1:n} \sim D^n}\left[D(g(x_{1:n}) \Delta g(x_{1:n-1}))\right]$ is at most

$$\sum_{x \in \mathcal{X}} \mathbb{E}\left[\mathbf{1}\{x_n \in c_{k-1}(x_{1:n-1}) \cup c_k(x_{1:n-1}\} D(h(x_n)) | x_n = x\right] D(x_n) = \\ \sum_{x \in \mathcal{X}} \mathbb{P}\left[x \in c_{k-1}(x_{1:n-1}) \cup c_k(x_{1:n-1}) | x_n = x\right] D(h(x)) D(x).$$

Because, we have that

$$\mathbb{P}\left[x \in c_{k-1}(x_{1:n-1}) \cup c_k(x_{1:n-1}) | x_n = x\right] = \\ \mathbb{P}\left[x \in c_{k-1}(x_{1:n-1}) | x_n = x\right] + \mathbb{P}\left[x \in c_k(x_{1:n-1}) | x_n = x\right].$$

Now, observe that

$$\mathbb{P}\left[x \in c_{k-1}(x_{1:n-1}) | x_n = x\right] = \binom{n-1}{k-1} D(x)^{k-1}(1-D(x))^{n-k} = \frac{k}{n}\binom{n}{k} D(x)^{k-1}(1-D(x))^{n-k}$$

and

$$\mathbb{P}\left[x \in c_{k}(x_{1:n-1}) | x_n = x\right] = \binom{n-1}{k} D(x)^{k}(1-D(x))^{n-1-k} = \frac{k+1}{n}\binom{n}{k+1} D(x)^{k}(1-D(x))^{n-1-k}.$$

Hence,

$$D(x) \cdot \mathbb{P}\left[x \in c_{k-1}(x_{1:n-1}) | x_n = x\right] \leq \frac{k}{n},$$

and

$$D(x) \cdot \mathbb{P}\left[x \in c_{k}(x_{1:n-1}) | x_n = x\right] \leq \frac{k+1}{n}.$$

Plugging these bounds in, gives that

$$\sum_{x \in \mathcal{X}} \mathbb{P}\left[x \in c_{k-1}(x_{1:n-1}) \cup c_k(x_{1:n-1}) | x_n = x\right] D(h(x)) D(x) \leq \frac{2k+1}{n} \sum_{x \in \mathcal{X}} D(h(x)).$$

Noting that

$$\sum_{x \in \mathcal{X}} D(h(x)) = \sum_{z \in \mathcal{X}} D(z)|h^{-1}(z)| \leq r \sum_{z \in \mathcal{X}} D(z) = r,$$

completes the proof. ■

Finally, we also extend these results to estimating the mass of arbitrary piercing's of the k-heavy hitters.

Theorem 8.5.9. *Let $\{A_n\}_{n \in \mathbb{N}}$ be any sequence where $A_n \subseteq \mathcal{X}$. For $k \geq 1$, consider the function $g(x_{1:n}) = A_n \cap c_k(x_{1:n})$. Then,*

$$\sup_{D \in \Delta\mathcal{X}} \mathop{\mathbb{E}}_{x_{1:n} \sim D^n} \left[\left(\hat{f}^{\text{loo}}(x_{1:n}) - D(g(x_{1:n})) \right)^2 \right] = O\left(\frac{k}{n}\right).$$

Proof. Fix $k \in \mathbb{N}$ and a distribution $D \in \Delta\mathcal{X}$. Then, observe that

$$\begin{aligned}\mathbb{E}_{x_{1:n}\sim D^n}\left[D(g(x_{1:n})\Delta g(x_{1:n-1}))\right] &= \mathbb{E}_{x_{1:n}\sim D^n}\left[D(x_n)\mathbf{1}\{x_n \in c_{k-1}(x_{1:n-1})\}\mathbf{1}\{x_n \in A_n\}\right] \\ &\leq \mathbb{E}_{x_{1:n}\sim D^n}\left[D(x_n)\mathbf{1}\{x_n \in c_{k-1}(x_{1:n-1})\}\right] \leq \frac{k}{n},\end{aligned}$$

where the last line follows from the proof of Lemma 8.5.7. Hence, we have that $\gamma_n(g) \leq \frac{k}{n}$, giving that the leave-one-out estimator has estimation error at most $O(\frac{k}{n})$. ■

8.6 Towards a Characterization of Estimability

So far, we have focused on two popular estimators and established upper bounds on the estimation MSE for a wide class of g functions. In this section, we zoom out and attempt to understand the general landscape of estimability. Perhaps the first question to ask is whether the class of all g functions are estimable. The following Theorem shows that this is not the case – there exist a hard g function.

Theorem 8.6.1. *There exists a $g : \mathcal{X}^\star \to 2^\star$ such that*

$$\inf_{\hat{f}} \sup_{D\in\Delta\mathcal{X}} \mathbb{E}_{x_{1:n}\sim D^n}\left[(\hat{f}(x_{1:n}) - D(g(x_{1:n})))^2\right] = \Omega(1).$$

Proof. Fix $n \in \mathbb{N}$ and let $\mathcal{X} = \mathbb{F}_2^{n+2}$. For any vector $w \in \mathbb{F}_2^{n+2}$, define $N_w := \{u \in \mathbb{F}_2^{n+2} \mid \langle u, w\rangle = 0\}$, i.e., the $(n+1)$-dimensional subspace orthogonal to w. Consider the following g function. Given a sample set $S = \{x_1, x_2, \ldots, x_n\} \subset \mathcal{X}$, set $g(S) = \emptyset$ if S is linearly dependent. Otherwise, by the rank-nullity theorem, the subspace of vectors in $\mathbb{F}_2^{n+2}$ orthogonal to all x_i is 2-dimensional, and thus has exactly 3 non-zero vectors. Let z_S be the lexicographically first vector among those three. Then, we define

$$g(S) \;=\; \{\, x \in \mathcal{X} \;:\; \langle x, z_S\rangle = 0 \,\}.^1$$

Next, consider the following distribution over input distributions (which may depend on n). Choose a non-zero vector $z \in \mathbb{F}_2^{n+2}$ u.a.r., and let D_z be the uniform distribution on N_z.

[1]Note that in this proof, both $\mathcal{X}$ and g are a function of the sample size n. One can make these independent of n by taking $\mathcal{X} = \{0,1\}^{\mathbb{N}}$ and using projections onto the first $n+2$ coordinates. We avoid this here for clarity of exposition.

Now, let $\hat{f} : \mathcal{X}^n \to [0,1]$ be any purported estimator. We will show that

$$\mathbb{E}_{z,S}[(\hat{f}(S) - D_z(g(S)))^2] \geq c,$$

which implies that $\hat{f}$ is not an estimator for g.

To prove this, first, note that under D_z, each x_i is uniform on the $(n+1)$-dimensional space N_z, so the probability that S is linearly independent is

$$p := \left(1 - \frac{1}{2^{n+1}}\right)\left(1 - \frac{2}{2^{n+1}}\right)\cdots\left(1 - \frac{2^{n-1}}{2^{n+1}}\right) \geq$$
$$1 - \frac{1}{2^{n+1}} - \frac{2}{2^{n+1}} - \cdots - \frac{2^{n-1}}{2^{n+1}} = 1 - \frac{2^n - 1}{2^{n+1}} \geq \frac{1}{2}.$$

Next, given a linearly independent sample set S, there are exactly 3 non-zero vectors orthogonal to all elements of S, one of which is z_S. If the other two non-zero vectors orthogonal to S are u and v, then by Bayes' rule, we have $\Pr[z = z_S \mid S] = \Pr[S \mid z = z_S] \cdot \Pr[z = z_S] / \Pr[S]$, and the same holds for u and v also. Now it's easy to see that $\Pr[S \mid z = z_S] = \Pr[S \mid z = u] = \Pr[S \mid z = v]$ by construction (each law makes the sample uniform on some $(n+1)$-dimensional hyperplane), and $\Pr[z = z_S] = \Pr[z = u] = \Pr[z = v]$ since z is uniformly chosen from all non-zero vectors. Hence, $\Pr[z = z_S \mid S] = \Pr[z = u \mid S] = \Pr[z = v \mid S] = \frac{1}{3}$. Furthermore, if $z = z_S$, then $g(S)$ contains all of N_{z_S}, so $D_z(g(S)) = 1$. If $z \neq z_S$ (i.e., $z \in \{u, v\}$), then $N_z \cap N_{z_S}$ has dimension n, so within N_z we have $D_z(g(S)) = \frac{2^n}{2^{n+1}} = \frac{1}{2}$.

Hence, we have

$$\begin{aligned}
\mathbb{E}_{z,S}[(\hat{f}(S) - D_z(g(S)))^2] &\geq \mathbb{E}_{z,S}[(\hat{f}(S) - D_z(g(S)))^2 \mid S \text{ is linearly ind.}] \cdot p \\
&= \mathbb{E}_S\big[\mathbb{E}_z[(\hat{f}(S) - D_z(g(S)))^2 \mid S,\ S \text{ is linearly ind.}]\big] \cdot p \\
&\geq \mathbb{E}_S\big[\mathrm{Var}_z[D_z(g(S)) \mid S,\ S \text{ is linearly ind.}]\big] \cdot \frac{1}{2}.
\end{aligned}$$

Now, given such an S, we have

$$\mathrm{Var}_z[D_z(g(S)) \mid S,\ S \text{ is linearly ind.}] = \left(1^2 \cdot \frac{1}{3} + \frac{1}{2^2} \cdot \frac{2}{3}\right) - \left(1 \cdot \frac{1}{3} + \frac{1}{2} \cdot \frac{2}{3}\right)^2 = \frac{1}{18}.$$

Hence,

$$\mathbb{E}_{z,S}[(\hat{f}(S) - D_z(g(S)))^2] \geq \frac{1}{36},$$

which completes the proof. ■

In light of Theorem 8.6.1 we leave open the following question which asks for a tighter

characterization of estimability:

Question 8.6.2. *What are necessary and sufficient conditions for g to be estimable?*

CHAPTER 9

Future Directions

We conclude this thesis by providing interesting directions of future work related to some of the preceding chapters.

9.1 Online Classification with Predictions

In chapter 3, we initiated the study of online classification when the learner has access to machine-learned predictions about future examples. There are many interesting directions for future research and we list two below. Firstly, we only considered the classification setting, and it would be interested to extend our results to online scalar-valued regression. Secondly, we measure the performance of a Predictor through its mistake-bounds. When $\mathcal{X}$ is continuous, this might be an unrealistic measure of performance. Thus, it would be interesting to see whether our results can be generalized to the case where $\mathcal{X}$ is continuous and the guarantee of Predictors is defined in terms of ℓ_p losses.

9.2 The Complexity of Sequential Prediction in Dynamical Systems

In chapter 4, we studied the problem of learning-to-predict in discrete-time dynamical systems under the 0-1 loss. A natural extension is to consider continuous state spaces with real-valued losses. For example, one can take $\mathcal{X}$ to be a bounded subset of a Hilbert space and consider the squared norm as the loss function. Another natural extension is to consider learnability under partial observability, where the learner only observes some transformation $\phi(x_t)$ instead of the true state x_t. Such feedback model is standard in prediction for linear dynamical systems [Hazan et al., 2018]. It is also natural to study the learnability of function classes where the output of the evolution rules $f : \mathcal{X}^p \to \mathcal{X}$, depend on the previous $p > 1$ states (e.g. the p-th order VAR model). Lastly, the learning algorithms in

this work are *improper*: they use evolution functions that may not lie in $\mathcal{F}$ to make predictions. This might be undesirable as improper learning algorithms may be incompatible with downstream system identification and control tasks. To this end, characterizing proper learnability of dynamical systems is an important future direction.

9.3 Multiclass Online Learnability under Bandit Feedback

In Chapter 5, we revisited multiclass online learnability under bandit feedback and showed that, when $\mathcal{Y}$ is unbounded: (1) the Bandit Littlestone dimension, originally proposed by Daniely et al. [2011], continues to characterize bandit online learnability, and (2) while SUC is necessary for bandit online learnability, it is not sufficient.

Moving forward, there are still many interesting open questions. By Theorem 5.1.2, in the agnostic setting there is a gap of $\sqrt{\mathrm{BL}(\mathcal{H})\log(T)}$ between the upper and lowerbounds on the optimal expected regret under bandit feedback. Is this gap between the upper and lowerbound unavoidable? Using the fact that $\mathrm{BL}(\mathcal{H}) \leq 4C\log(C)\mathrm{L}(\mathcal{H})$, one can get a lowerbound of $\Omega\left(\sqrt{\frac{\mathrm{BL}(\mathcal{H})\,T}{C\log(C)}}\right)$ on the expected regret in the agnostic setting, where $C = \sup_{x\in\mathcal{X}}|\mathcal{H}(x)|$. Is it possible to remove the dependence on C and improve this lowerbound to $\Omega(\sqrt{\mathrm{BL}(\mathcal{H})\,T})$?

While the BLdim provides a sharp quantitative characterization of deterministic learnability in the realizable setting, it is unclear whether it provides a tight *quantitative* characterization of randomized learnability in both the realizable and agnostic settings. Recently, Filmus, Hanneke, Mehalel, and Moran [2023] gave a combinatorial parameter called the Randomized Littlestone dimension and showed that it exactly quantifies the optimal expected mistake bound for randomized learners in the realizable setting under full-information feedback. Is there a modification of this dimension that can exactly quantify the optimal expected mistake bound for randomized learners in the realizable setting under *bandit feedback*? Can such a dimension also be used to give a sharper upperbound on the expected regret in the agnostic setting?

9.4 Apple Tasting: Combinatorial Dimensions and Minimax Rates

In Chapter 6, we revisited the classical setting of apple tasting and studied learnability from a combinatorial perspective. Our work makes an important step towards developing learning

theory for online classification under partial feedback. An important future direction is to extend this work to multiclass classification under various partial feedback models, such as those captured by feedback graphs [Alon et al., 2015].

With respect to apple tasting, there are still interesting open questions. Our lower and upper bounds in the agnostic setting are matching up to a factor logarithmic in T. Recently, Alon et al. [2021b] showed that in the full-information setting, this $\log(T)$ factor can be removed from the upper bound, meaning that the optimal expected regret in the agnostic setting under full-information feedback is $\Theta(\sqrt{\mathrm{L}(\mathcal{H})T})$. As an open question, we ask whether it is possible to also remove the factor of $\log(T)$ from our upper bound in Theorem 6.4.1.

9.5 On the Learnability of Multilabel Ranking

In Chapter 7, we characterize the learnability of a multilabel ranking hypothesis class in both the batch and online setting for a wide range of practical ranking losses. In all cases, we show that a ranking hypothesis class is learnable if and only if a sufficient number of its binary-valued threshold restrictions are learnable. This chapter studies two families of ranking loss functions and leaves it open to characterize the learnability of other natural ranking loss functions. One loss function not captured by our families is recall@p.

While we do establish quantitative bounds on the sample complexity and regret, our bounds are not optimal. It may be difficult to improve the sample complexity and regret bound at the highest level of generality for all losses in the families considered here. However, for natural losses such as sum loss, it is an interesting future direction to derive the optimal sample complexity and regret bounds in both the realizable and agnostic settings. In addition, our bounds depend on the number of labels K. Recently, K-free bounds have been achieved for multiclass classification problems in both batch and online settings [Brukhim et al., 2022, Hanneke et al., 2023]. An interesting future direction is to study whether K-free bounds are possible for multilabel ranking.

Finally, the focus of this chapter is on characterizing learnability, and thus our algorithms are not computationally efficient. A natural future direction is to construct computationally efficient algorithms for multilabel ranking. Along this direction, since ERM is the most common algorithm used in practice, it is an important future direction to tightly quantify the sample complexity of ERM in the batch setting. Moreover, in learning theory, combinatorial dimensions play an important role in providing a tight quantitative characterization of learnability. Thus, it is an interesting future direction to identify combinatorial dimensions that characterize multilabel ranking learnability for specific loss functions.

9.6 Estimating the (Un)seen: Sample-dependent Mass Estimation

In Chapter 8, we initiated the study of mass estimation beyond the missing mass functional. We demonstrated that empirical-distribution–based and leave-one-out–based estimators are surprisingly powerful, enabling consistent estimation for a broad class of functions $g : \mathcal{X}^\star \to 2^{\mathcal{X}}$. Beyond Question 8.6.2, our results open up several compelling directions for future research, which we briefly outline below.

Our analysis primarily focused on establishing upper bounds on MSE for a wide family of g functions. A natural next step is to determine matching lower bounds on the achievable error rates. The estimators we study assume oracle access for efficiently testing membership of elements $x \in g(S)$ for all $x \in \mathcal{X}$ and $S \subset \mathcal{X}$. In practice, this assumption may be computationally unrealistic. This motivates a complementary line of inquiry into computable mass estimation, where one seeks estimators that are both statistically accurate and computationally efficient under appropriate runtime or query-complexity constraints. Finally, the results of Chapter 8 deal exclusively with discrete distribution defined over a countably infinite domain. Extending these results to continuous distributions over metric spaces, similar to the work by Maurer [2022], is also of interest.

APPENDIX A

Online Classification with Predictions

A.1 Combinatorial dimensions

In this section, we review existing combinatorial dimensions in statistical learning theory. We start with the VC and Natarajan dimensions which characterize PAC learnability when $|\mathcal{Y}| = 2$ and $|\mathcal{Y}| < \infty$ respectively.

Definition A.1.1 (VC dimension)**.** *A set $\{x_1, ..., x_n\} \in \mathcal{X}$ is shattered by $\mathcal{H}$, if $\forall y_1, ..., y_n \in \{0, 1\}$, $\exists h \in \mathcal{H}$, such that $\forall i \in [n]$, $h(x_i) = y_i$. The VC dimension of $\mathcal{H}$, denoted $\underline{VC}(\mathcal{H})$, is defined as the largest natural number $n \in \mathbb{N}$ such that there exists a set $\{x_1, ..., x_n\} \in \mathcal{X}$ that is shattered by $\mathcal{H}$.*

Definition A.1.2 (Natarajan Dimension)**.** *A set $S = \{x_1, \ldots, x_d\}$ is shattered by a multiclass function class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ if there exist two witness functions $f, g : S \to \mathcal{Y}$ such that $f(x_i) \neq g(x_i)$ for all $i \in [d]$, and for every $\sigma \in \{0, 1\}^d$, there exists a function $h_\sigma \in \mathcal{H}$ such that for all $i \in [d]$, we have*

$$h_\sigma(x_i) = \begin{cases} f(x_i) & \text{if } \sigma_i = 1 \\ g(x_i) & \text{if } \sigma_i = 0 \end{cases} .$$

The Natarajan dimension of $\mathcal{H}$, denoted $\mathrm{N}(\mathcal{H})$, is the size of the largest shattered set $S \subseteq \mathcal{X}$. If the size of the shattered set can be arbitrarily large, we say that $\mathrm{N}(\mathcal{H}) = \infty$.

We note that $\mathrm{N}(\mathcal{H}) = \mathrm{VC}(\mathcal{H})$ whenever $|\mathcal{Y}| = 2$. Next, we move to the online setting, where the Littlestone dimension (Ldim) characterizes multiclass online learnability. To define the Ldim, we first define a Littlestone tree and a notion of shattering.

Definition A.1.3 (Littlestone tree)**.** *A Littlestone tree of depth d is a $\underline{\text{complete}}$ binary tree of depth d where the internal nodes are labeled by examples of $\mathcal{X}$ and the left and right outgoing edges from each internal node are labeled by 0 and 1 respectively.*

Given a Littlestone tree $\mathcal{T}$ of depth d, a root-to-leaf path down $\mathcal{T}$ is a bitstring $\sigma \in \{0,1\}^d$ indicating whether to go left ($\sigma_i = 0$) or to go right ($\sigma_i = 1$) at each depth $i \in [d]$. A path $\sigma \in \{0,1\}^d$ down $\mathcal{T}$ gives a sequence of labeled examples $\{(x_i, \sigma_i)\}_{i=1}^d$, where x_i is the example labeling the internal node following the prefix $(\sigma_1, ..., \sigma_{i-1})$ down the tree. A hypothesis $h_\sigma \in \mathcal{H}$ shatters a path $\sigma \in \{0,1\}^d$, if for every $i \in [d]$, we have $h_\sigma(x_i) = \sigma_i$. In other words, h_σ is consistent with the labeled examples when following σ. A Littlestone tree $\mathcal{T}$ is shattered by $\mathcal{H}$ if for every root-to-leaf path σ down $\mathcal{T}$, there exists a hypothesis $h_\sigma \in \mathcal{H}$ that shatters it. Using this notion of shattering, we define the Littlestone dimension of a hypothesis class.

Definition A.1.4 (Littlestone dimension)**.** *The Littlestone dimension of $\mathcal{H}$, denoted $\underline{L}(\mathcal{H})$, is the largest $d \in \mathbb{N}$ such that there exists a Littlestone tree $\mathcal{T}$ of depth d shattered by $\mathcal{H}$. If there exists shattered Littlestone trees $\mathcal{T}$ of arbitrary depth, then we say that $\underline{L}(\mathcal{H}) = \infty$.*

Finally, the following notion of shattering is useful when proving the lower bound in Appendix A.4.2.

Definition A.1.5 (Threshold shattering)**.** *A sequence $(x_1, ..., x_k) \in \mathcal{X}^k$ is threshold-shattered by $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ if there exists $(h_1, ..., h_k) \in \mathcal{H}^k$ such that $h_i(x_j) = \mathbb{1}\{j \leq i\}$ for all $i, j \in [k]$.*

A.2 Proof of Helper Lemmas

A.2.1 Proof of Lemmas 3.3.6 and 3.3.7

Proof. (of Lemma 3.3.6) Let $(x_1, y_1), ..., (x_T, y_T)$ be the realizable stream to be observed by the Expert. For every $i \in \{0, ..., c\}$, let m_i be the random variable denoting the number of mistakes made by $\mathcal{P}$ in rounds $\{\tilde{t}_i + 1, ..., \tilde{t}_{i+1}\}$. Recall that $\tilde{t}_0 = 0$ and $\tilde{t}_{c+1} = T$. Let $M = \sum_{i=0}^c m_i$ be the random variable denoting the total number of mistakes made by $\mathcal{P}$ on the realizable stream. Finally, let $\mathcal{A}$ denote Algorithm 4. Observe that,

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}(x_t) \neq y_t\}\right] &= \mathbb{E}\left[\sum_{i=0}^{c} \sum_{t=\tilde{t}_i+1}^{\tilde{t}_{i+1}} \mathbb{1}\{\mathcal{A}(x_t) \neq y_t\}\right] \\
&= \mathbb{E}\left[\sum_{i=0}^{c} \sum_{t=\tilde{t}_i+1}^{\tilde{t}_{i+1}} \mathbb{1}\{\mathcal{K}_i(x_t) \neq y_t\}\right] \\
&\leq \mathbb{E}\left[\sum_{i=0}^{c} (m_i + 1)\overline{\mathrm{M}}_{\mathcal{B}}(\tilde{t}_{i+1} - \tilde{t}_i, \mathcal{H})\right]
\end{aligned}$$

where the first inequality follows from the guarantee of $\mathcal{K}$ and Lemma 3.2.2. Using Jensen's inequality, we get that

$$\begin{aligned}
\mathbb{E}\left[\sum_{i=0}^{c}(m_i+1)\overline{\mathrm{M}}_{\mathcal{B}}(\tilde{t}_{i+1}-\tilde{t}_i,\mathcal{H})\right] &\leq \mathbb{E}\left[\left(\sum_{i=0}^{c}(m_i+1)\right)\overline{\mathrm{M}}_{\mathcal{B}}\left(\frac{\sum_{i=0}^{c}(m_i+1)(\tilde{t}_{i+1}-\tilde{t}_i)}{\sum_{i=0}^{c}(m_i+1)},\mathcal{H}\right)\right] \\
&= \mathbb{E}\left[\Big(M+c+1\Big)\overline{\mathrm{M}}_{\mathcal{B}}\left(\frac{\sum_{i=0}^{c}(m_i+1)(\tilde{t}_{i+1}-\tilde{t}_i)}{M+c+1},\mathcal{H}\right)\right] \\
&= \mathbb{E}\left[\Big(M+c+1\Big)\overline{\mathrm{M}}_{\mathcal{B}}\left(\frac{\sum_{i=0}^{c}m_i(\tilde{t}_{i+1}-\tilde{t}_i)+T}{M+c+1},\mathcal{H}\right)\right] \\
&= \mathbb{E}\left[\Big(M+c+1\Big)\overline{\mathrm{M}}_{\mathcal{B}}\left(\frac{\lceil\frac{T}{c+1}\rceil\sum_{i=0}^{c}m_i+T}{M+c+1},\mathcal{H}\right)\right] \\
&= \mathbb{E}\left[\Big(M+c+1\Big)\overline{\mathrm{M}}_{\mathcal{B}}\left(\frac{\lceil\frac{T}{c+1}\rceil\ M+T}{M+c+1},\mathcal{H}\right)\right].
\end{aligned}$$

Using the fact that $\lceil\frac{T}{c+1}\rceil \leq \frac{T}{c+1}+1$, we have

$$\begin{aligned}
\mathbb{E}\left[\sum_{i=0}^{c}\sum_{j=0}^{m_i}\overline{\mathrm{M}}_{\mathcal{B}}(\tilde{t}_{i+1}-\tilde{t}_i,\mathcal{H})\right] &\leq \mathbb{E}\left[\Big(M+c+1\Big)\overline{\mathrm{M}}_{\mathcal{B}}\left(\frac{\frac{MT}{c+1}+M+T}{M+c+1},\mathcal{H}\right)\right] \\
&\leq \mathbb{E}\left[\Big(M+c+1\Big)\overline{\mathrm{M}}_{\mathcal{B}}\left(\frac{T}{c+1}+1,\mathcal{H}\right)\right] \\
&= \Big(\mathrm{M}_{\mathcal{P}}(x_{1:T})+c+1\Big)\overline{\mathrm{M}}_{\mathcal{B}}\left(\frac{T}{c+1}+1,\mathcal{H}\right),
\end{aligned}$$

which completes the proof. ■

Proof. (of Lemma 3.3.7) Let $(x_1,y_1),...,(x_T,y_T)$ be the realizable stream to be observed by the learner. Let $\mathcal{A}$ denote the online learner in Algorithm 5. By the guarantees of the DWMA, we have

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\{\mathcal{A}(x_t)\neq y_t\}\right] &\leq 3\mathbb{E}\left[\inf_{b\in\{0,\dots,T-1\}}\sum_{t=1}^{T}\mathbb{1}\{E_b(x_t)\neq y_t\}\right]+3\log_2 T \\
&\leq 3\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\{E_{\lceil\mathrm{M}_{\mathcal{P}}(x_{1:T})\rceil}(x_t)\neq y_t\}\right]+3\log_2 T \\
&\leq 6(\mathrm{M}_{\mathcal{P}}(x_{1:T})+1)\overline{\mathrm{M}}_{\mathcal{B}}\Big(\frac{T}{\mathrm{M}_{\mathcal{P}}(x_{1:T})+1}+1,\mathcal{H}\Big)+6\log_2 T,
\end{aligned}$$

where the last inequality follows from Lemma 3.3.6 and the fact that $\mathrm{M}_{\mathcal{P}}(x_{1:T}) \leq \lceil \mathrm{M}_{\mathcal{P}}(x_{1:T}) \rceil \leq \mathrm{M}_{\mathcal{P}}(x_{1:T}) + 1$. ■

A.3 Proof of Corollaries

A.3.1 Proof of Corollary 3.3.2 and 3.3.3

Using Theorem 3.3.1, we first show that for every $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, Predictor $\mathcal{P}$, $\mathcal{Z} \subseteq \mathcal{X}^{\star}$, and no-regret offline learner $\mathcal{B}$, we have that

$$\inf_{\mathcal{A}} \mathrm{M}_{\mathcal{A}}(T, \mathcal{H}, \mathcal{Z}) = O\Bigg(\mathrm{L}(\mathcal{H}) \wedge (\mathrm{M}_{\mathcal{P}}(T, \mathcal{Z}) + 1)\, \mathrm{M}_{\mathcal{B}}(T, \mathcal{H}) \\ \wedge \left((\mathrm{M}_{\mathcal{P}}(T, \mathcal{Z}) + 1)\, \overline{\mathrm{M}}_{\mathcal{B}}\Big(\frac{T}{\mathrm{M}_{\mathcal{P}}(T, \mathcal{Z}) + 1}, \mathcal{H} \Big) + \log_2 T \right) \Bigg).$$

Proof. It suffices to show that Algorithms 3 and 5 have mistake bounds $O((\mathrm{M}_{\mathcal{P}}(T, \mathcal{Z}) + 1)\, \mathrm{M}_{\mathcal{B}}(T, \mathcal{H}))$ and $O\left((\mathrm{M}_{\mathcal{P}}(T, \mathcal{Z}) + 1)\, \overline{\mathrm{M}}_{\mathcal{B}}\Big(\frac{T}{\mathrm{M}_{\mathcal{P}}(T, \mathcal{Z})+1}, \mathcal{H} \Big) + \log_2 T \right)$ respectively. To see that Algorithm 3's mistake bounds is $O((\mathrm{M}_{\mathcal{P}}(T, \mathcal{Z}) + 1)\, \mathrm{M}_{\mathcal{B}}(T, \mathcal{H}))$, note that $\mathrm{M}_{\mathcal{P}}(x_{1:T}) \leq \mathrm{M}_{\mathcal{P}}(T, \mathcal{Z})$ for every $x_{1:T} \in \mathcal{Z}$. To see that Algorithm 5's expected mistake bound is $O\left((\mathrm{M}_{\mathcal{P}}(T, \mathcal{Z}) + 1)\, \overline{\mathrm{M}}_{\mathcal{B}}\Big(\frac{T}{\mathrm{M}_{\mathcal{P}}(T, \mathcal{Z})+1}, \mathcal{H} \Big) + \log_2 T \right)$, we follow the exact same proof strategy as in the proof of Lemma 3.3.7, but picking a different expert when upper bounding the expected number of mistakes. Namely, following the same steps as in the proof of Lemma 3.3.7, we have that

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}(x_t) \neq y_t\} \right] \leq 3\mathbb{E}\left[\inf_{b \in \{0, \ldots, T-1\}} \sum_{t=1}^{T} \mathbb{1}\{E_b(x_t) \neq y_t\} \right] + 3 \log_2 T$$

where $\mathcal{A}$ denotes Algorithm 5. Picking $b = \lceil \mathrm{M}_{\mathcal{P}}(T, \mathcal{Z}) \rceil$, using Lemma 3.3.6, and the fact that $\mathrm{M}_{\mathcal{P}}(x_{1:T}) \leq \mathrm{M}_{\mathcal{P}}(T, \mathcal{Z})$ gives the desired upper bound on $\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}(x_t) \neq y_t\} \right]$ of

$$O\left((\mathrm{M}_{\mathcal{P}}(T, \mathcal{Z}) + 1) \overline{\mathrm{M}}_{\mathcal{B}}\Big(\frac{T}{\mathrm{M}_{\mathcal{P}}(T, \mathcal{Z}) + 1}, \mathcal{H} \Big) + \log_2 T \right), \tag{A.1}$$

completing the proof. ■

Corollary 3.3.2 follows from the fact that the upper bound on $\inf_{\mathcal{A}} \mathrm{M}_{\mathcal{A}}(T, \mathcal{H}, \mathcal{Z})$ is sublinear whenever $\mathrm{M}_{\mathcal{P}}(T, \mathcal{Z}) = o(T)$ and $\mathrm{M}_{\mathcal{B}}(T, \mathcal{H}) = o(T)$. To get Corollary 3.3.3, recall that by Lemma 3.2.1, there exists an offline learner $\mathcal{B}$ such that

$$\overline{\mathrm{M}}_{\mathcal{B}}(T, \mathcal{H}) = O\Big(\mathrm{VC}(\mathcal{H}) \log_2 T\Big).$$

Plugging this bound into upper bound (A.1) completes the proof.

A.4 Proof of Theorems

A.4.1 Proof of Theorem 3.3.1

Let $\mathcal{A}$ denote the DWMA using the Standard Optimal Algorithm (SOA), Algorithm 3 and Algorithm 5 as experts. Then, for any realizable stream $(x_1, y_1), ..., (x_T, y_T)$, Lemma 3.3.4 gives that

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}(x_t) \neq y_t\}\right] \leq 3\mathbb{E}\left[\min_{i \in [3]} M_i + \log_2 3\right] \leq 3 \min_{i \in [3]} \mathbb{E}\left[M_i\right] + 5,$$

where we take M_1, M_2 and M_3 to be the number of mistakes made by the SOA, Algorithm 3, and Algorithm 5 respectively. Note that M_2 and M_3 are random variables since $\mathcal{B}$ and $\mathcal{P}$ may be randomized algorithms. Finally, using Lemma 3.3.5, Lemma 3.3.7 and the fact that the SOA makes at most $\mathrm{L}(\mathcal{H})$ mistakes on any realizable stream [Littlestone, 1987] completes the proof of Theorem 3.3.1.

A.4.2 Proof of Theorem 3.3.8

Let $\mathcal{X} = \mathbb{R} \cup \{\star\}$ and $\mathcal{H} = \{x \mapsto \mathbb{1}\{x \leq a\}\mathbb{1}\{x \neq \star\}\}$. Let $T, n \in \mathbb{N}$ be chosen such that T is a multiple of $n+1$ and $\frac{T}{n+1} + 1 = 2^k$ for some $k \in \mathbb{N}$. Our proof of Theorem 3.3.8 will be in four steps, as described below.

(1) We construct a class of streams $\mathcal{Z}_n \subseteq \mathcal{X}^\star$.

(2) Using $\mathcal{Z}_n$, we construct a deterministic, lazy, consistent Predictor $\mathcal{P}$ such that $\mathcal{P}$ makes mistakes exactly on timepoints $\{\frac{T}{n+1} + 1, ..., \frac{nT}{n+1} + 1\}$ for every stream $x_{1:T} \in \mathcal{Z}_n$.

(3) When $x_{1:T} \in \mathcal{Z}_n$, we establish an equivalence between the game defined by Protocol 2 when given access to Predictor $\mathcal{P}$ and Online Classification with Peeks, a different

game where there is no Predictor, but the learner observes the next $\frac{T}{n+1}$ examples at timepoints $t \in \{1, \frac{T}{n+1}+1, ..., \frac{nT}{n+1}+1\}$.

(4) For Online Classification with Peeks, we give a strategy for Nature such that it can force any online learner to make $\frac{(n+1)\log_2(\frac{T}{n+1})}{2}$ mistakes in expectation while ensuring that the stream of labeled examples it picks $(x_1, y_1), ..., (x_T, y_T)$ satisfies the constraint that $x_{1:T} \in \mathcal{Z}_n$ and $\inf_{h\in\mathcal{H}} \sum_{t=1}^{T} \mathbb{1}\{h(x_t) \neq y_t\} = 0$.

Composing steps 1-4 shows the existence of a Predictor $\mathcal{P}$ such that for any learner $\mathcal{A}$ playing Protocol 2 using $\mathcal{P}$, there exists a realizable stream where $\mathcal{A}$ makes at least $\frac{(n+1)}{2}\log_2(\frac{T}{n+1})$ mistakes in expectation.

Step 1: Construction of $\mathcal{Z}_n$

Let $\mathcal{S}$ be the set of all strictly increasing sequences of real numbers in $(0, 1)$ of size $\frac{T}{n+1}$. Fix a function $f : \mathbb{R}^2 \to \mathcal{S}$ which, given $a < b \in \mathbb{R}$, outputs an element of $\mathcal{S}$ that lies strictly in between a and b. For example, given $a < b \in \mathbb{R}$, the function f can output evenly spaced real numbers of size $\frac{T}{n+1}$. Let $\text{Dyd} : \mathcal{S} \to \mathcal{X}^{\frac{T}{n+1}}$ be a function that reorders the input $S \in \mathcal{S}$ in Dyadic order. Namely, if $S = (x_1, ..., x_N)$ where $N + 1 = 2^k$ for some $k \in \mathbb{N}$, then $\text{Dyd}(S)$ is

$$x_{\frac{N}{2}}, x_{\frac{N}{4}}, x_{\frac{3N}{4}}, x_{\frac{N}{8}}, x_{\frac{3N}{8}}, x_{\frac{5N}{8}}, x_{\frac{7N}{8}}, ..., x_{\frac{(2^k-1)N}{2^k}}.$$

See the Proof of Claim 3.4 in Hanneke et al. [2024] for a more detailed description of a Dyadic order. On the other hand, let $\text{Sort} : \mathcal{X}^{\frac{T}{n+1}} \to \mathcal{S}$ be a function that reorders its input in increasing order. Let $\mathcal{J} := \{1, ..., \frac{T}{n+1}+1\}^{\leq n}$ be the set of all sequences of indices of length at most n taking values in $\{1, ..., \frac{T}{n+1}+1\}$. For the remainder of this section, we will use S_i to denote the ith element in a sequence $S \in \mathcal{S}$. Moreover, for any two sequences $S^1, S^2 \in \mathcal{S}$, we say $S^1 < S^2$ if $S^1_{|S^1|} < S^2_1$. That is, $S^1 < S^2$, if S^1 lies strictly to the left of S^2.

We will construct a stream for every sequence $j_{1:m} \in \mathcal{J}$, $m \leq n$, algorithmically as follows. Fix $S^0 := f(0, 1) \in \mathcal{S}$ and let SG denote Algorithm 12. Let

$$\mathcal{Z}_n = \left\{\text{SG}(S^0, j_{1:m}) : j_{1:m} \in \mathcal{J}, m \in \{1, ..., n\}\right\}$$

denote the stream class generated by applying SG to inputs S^0 and $j_{1:m}$ for every $j_{1:m} \in \mathcal{J}$, $m \leq n$. We make four important observations about $\mathcal{Z}_n$, which we will use to construct a Predictor that can reconstruct S^i given S^0 and the first example of the block S^{i-1}.

Observation 1. *For every sequence $x_{1:T} \in \mathcal{Z}$, we have that $x_{1:\frac{T}{n+1}} = \text{Dyd}(S^0)$.*

Algorithm 12 Stream Generator (SG)

Require: $S^0 \in \mathcal{S}$, $j_{1:m} \in \mathcal{J}$
Initialize: $a_0 = 0, b_0 = 1$
for $i = 1, ..., m$ **do**
  if $j_i = 1$ **then**
    $S^i \leftarrow f(a_{i-1}, S_1^{i-1})$
    $a_i \leftarrow a_{i-1}$
    $b_i \leftarrow S_1^{i-1}$
  else if $j_i = \frac{T}{n+1} + 1$ **then**
    $S_i \leftarrow f(S^{i-1}_{\frac{T}{n+1}}, b_{i-1})$
    $a_i \leftarrow S^{i-1}_{\frac{T}{n+1}}$
    $b_i \leftarrow b_{i-1}$
  else
    $S_i \leftarrow f(S_{j-1}^{i-1}, S_j^{i-1})$
    $a_i \leftarrow S_{j-1}^{i-1}$
    $b_i \leftarrow S_j^{i-1}$
  end if
end for
Return: $\mathrm{Dyd}(S^0) \circ ... \circ \mathrm{Dyd}(S^m)$

The first observation follows from the fact that the same initial sequence S^0 is used to generate every stream in $\mathcal{Z}_n$.

Observation 2. *For any pair $j^1_{1:n}, j^2_{1:n} \in \mathcal{J}$ and $m \leq n$, if $j^1_{1:m} = j^2_{1:m}$, then $\mathrm{SG}(S^0, j^1_{1:m}) = \mathrm{SG}(S^0, j^2_{1:m})$.*

The second observation follows from the fact that SG is deterministic.

Observation 3. *For every $x_{1:T} \in \mathcal{Z}_n$ such that $x_{1:T} := \mathrm{SG}(S^0, j_{1:n}) = \mathrm{Dyd}(S^0) \circ ... \circ \mathrm{Dyd}(S^n)$, the index j_i can be computed exactly using only S^{i-1} and S_1^i for every $i \in [n]$.*

To see the third observation, fix some $x_{1:T} \in \mathcal{Z}_n$. Then, there exists a sequence $S^1, ..., S^n \in \mathcal{S}$ such that $x_{1:T} = \mathrm{Dyd}(S^0) \circ ... \circ \mathrm{Dyd}(S^n)$ as well as a sequence $(a_0, b_0), ..., (a_n, b_n)$. In addition, there exists a $j_{1:n} \in \mathcal{J}$ such that $x_{1:T} = \mathrm{SG}(S^0, j_{1;n})$. Fix $i \in [n]$ and consider S^{i-1} and S^i. By definition of Algorithm 12, there exists an index $q \in \{1, ..., \frac{T}{n+1} + 1\}$ such that $S^i = f(S^{i-1}_{q-1}, S^{i-1}_q)$ where we take $S^{i-1}_0 = a_{i-1}$ and $S^{i-1}_{\frac{T}{n+1}+1} = b_{i-1}$. We claim that the index q is unique. This follows from the fact that the collection $\{f(S^{i-1}_j, S^{i-1}_{j+1})\}_{j=0}^{\frac{T}{n+1}}$ is pairwise disjoint since $a_{i-1} = S^{i-1}_0 < S^{i-1}_1 < ... < S^{i-1}_{\frac{T}{n+1}} < S^{i-1}_{\frac{T}{n+1}+1} = b_{i-1}$. Finally, we claim that S^{i-1} and the element S^i_1 identifies the index q. This follows because $f(a_{i-1}, S^{i-1}_1) < f(S^{i-1}_1, S^{i-1}_2) < ... < f(S^{i-1}_{\frac{T}{n+1}-1}, S^{i-1}_{\frac{T}{n+1}}) < f(S^{i-1}_{\frac{T}{n+1}}, b_{i-1})$ and thus q is the smallest index $p \in \{1, ..., \frac{T}{n+1}\}$ such that $S^i_1 < S^{i-1}_p$ and $\frac{T}{n+1} + 1$ if such a p does not exist.

Observation 4. *Fix a sequence $j_{1:n} \in \mathcal{J}$ and let* $\mathrm{Dyd}(S^0) \circ ... \circ \mathrm{Dyd}(S^n) = \mathrm{SG}(S^0, j_{1:n})$. *For every $i, p \in [n]$ such that $i < p$, we have that:*

(i) $S^p < S^i$ *if* $j_i = 1$;

(ii) $S^i_{j_i - 1} < S^p < S^i_{j_i}$ *if* $2 \leq j_i \leq \frac{T}{n+1}$;

(iii) $S^i < S^p$ *if* $j_i = \frac{T}{n+1} + 1$.

The fourth observation follows from the fact that for every $i \in [n]$ and index $j_i \in \{1, ..., \frac{T}{n+1} + 1\}$, the remaining sequence of sets $S^{i+1}, ..., S^n$ all lie in the interval $(S^i_{j_i - 1}, S^i_{j_i})$ by design of Algorithm 12, where again we take $S^{i-1}_0 = a_{i-1} < S^{i-1}_1$ and $S^{i-1}_{\frac{T}{n+1}+1} = b_{i-1} > S^{i-1}_{\frac{T}{n+1}}$.

Step 2: Constructing a Predictor for $\mathcal{Z}_n$

We now show that Algorithm 13 is a lazy, consistent Predictor for $\mathcal{Z}_n$ that only makes mistakes at timepoints $\{\frac{T}{n+1} + 1, ..., \frac{nT}{n+1} + 1\}$.

Algorithm 13 Predictor for $\mathcal{Z}_n$

Require: $\mathcal{Z}_n$
Initialize: $J = ()$
for $t = 1, .., T$ **do**
  Receive x_t
  if $t = 1$ **then**
    Set $\hat{x}^t_{1:T} = \mathrm{Dyd}(S^0) \circ \hat{x}_{\frac{T}{n+1}+1:T}$ where $\hat{x}_{\frac{T}{n+1}+1:T} = (\star, ..., \star)$.
  else if $t = \frac{iT}{n+1} + 1$ for some $i \in \{1, ..., n\}$ **then**
    Let $S = \mathrm{Sort}(x_{t - \frac{T}{n+1}:t-1})$ be the last $\frac{T}{n+1}$ examples sorted in increasing order.
    Find the smallest $j \in \{1, ..., \frac{T}{n+1}\}$ such that $x_t < S_j$.
    If no such j exists, set $j = \frac{T}{n+1} + 1$.
    Update $J \leftarrow J \circ j$.
    Set $\hat{x}^t_{1:T} = \mathrm{SG}(S^0, J) \circ \hat{x}_{t + \frac{T}{n+1}:T}$ where $\hat{x}_{t+\frac{T}{n+1}:T} = (\star, ..., \star)$.
  else
    Set $\hat{x}^t_{1:T} \leftarrow \hat{x}^{t-1}_{1:T}$.
  end if
  Predict $\hat{x}^t_{1:T}$.
end for

Lemma A.4.1. *For any sequence $x_{1:T} \in \mathcal{Z}_n$, Algorithm 13 is a lazy, consistent Predictor for $\mathcal{Z}_n$ that only makes mistakes at timepoints $\{\frac{T}{n+1} + 1, ..., \frac{nT}{n+1} + 1\}$.*

Proof. Let $\mathcal{P}$ denote Algorithm 13 and $x_{1:T} \in \mathcal{Z}_n$. Then, there exists $S^1, ..., S^n \in \mathcal{S}$ and a sequence of indices $j_{1:n} \in \mathcal{J}$ such that $x_{1:T} = \text{Dyd}(S^0) \circ \text{Dyd}(S^1) \circ ... \circ \text{Dyd}(S^n) = \text{SG}(S^0, j_{1:n})$.

We now prove that $\mathcal{P}$ makes mistakes only on timepoints $\{\frac{T}{n+1} + 1, ..., \frac{nT}{n+1} + 1\}$ and no where else. Our proof is by induction using the following inductive hypothesis. For every $i \in \{1, ..., n\}$, we have that $\mathcal{P}$:

(i) sets $J_{1:i} = j_{1:i}$ on round $\frac{iT}{n+1} + 1$;

(ii) makes mistakes on rounds $\{\frac{T}{n+1} + 1, \frac{2T}{n+1} + 1, ..., \frac{iT}{n+1} + 1\}$ and no where in between.

For the base case, let $i = 1$. $\mathcal{P}$ does not make any mistakes in $\{1, 2, ..., \frac{T}{n+1}\}$ since it knows S^0 using $\mathcal{Z}_n$, computes $x_{1:\frac{T}{n+1}} = \text{Dyd}(S^0)$ in line 5, and does not change its prediction until round $\frac{iT}{n+1} + 1$ based on line 6. At time point $t_1 = \frac{T}{n+1} + 1$, $\mathcal{P}$ makes a mistake since $\hat{x}_{t_1}^{t_1-1} = \star \neq x_{t_1}$. Moreover, using Observation 3, the index $j \in \{1, ..., \frac{T}{n+1}\}$ computed in round t_1 on line 8 matches j_1. Thus, we have that $J_1 = j_1$. This completes the base case.

Now for the induction step, let $i \in \{2, ..., n\}$. Suppose that the induction step is true for $i - 1$. This means that $\mathcal{P}$:

(i) sets $J_{1:i-1} = j_{1:i-1}$ on round $\frac{(i-1)T}{n+1} + 1$;

(ii) makes mistakes on rounds $\{\frac{T}{n+1} + 1, \frac{2T}{n+1} + 1, ..., \frac{(i-1)T}{n+1} + 1\}$ and no where in between.

We need to show that $\mathcal{P}$ sets $J_i = j_i$ on round $\frac{iT}{n+1} + 1$, $\mathcal{P}$ makes no mistakes between $\frac{(i-1)T}{n+1} + 2$ and $\frac{iT}{n+1}$, but makes a mistake at $\frac{iT}{n+1} + 1$. At timepoint $t_{i-1} = \frac{(i-1)T}{n+1} + 1$, $\mathcal{P}$ computes $J_{i-1} = j_{i-1}$ (by assumption) and thus sets $\hat{x}_{1:T}^{t_{i-1}} = \text{SG}(S^0, J_{1:i-1}) = \text{SG}(S^0, (j_1, ..., j_{i-1})) = \text{Dyd}(S^0) \circ ... \circ \text{Dyd}(S^{i-1})$ using Observation 2. Therefore, $\mathcal{P}$ predicts on round t_{i-1} the sequence $\hat{x}_{1:T}^{t_{i-1}} = x_{1:\frac{iT}{n+1}} \circ (\star, ..., \star)$, implying that $\mathcal{P}$ makes no mistakes for rounds $\frac{(i-1)T}{n+1} + 2, ..., \frac{iT}{n+1}$ since it does not change its prediction until round $\frac{iT}{n+1} + 1$ by line 12. However, since $\hat{x}_{t_i}^{t_i-1} = \star$, $\mathcal{P}$ makes a mistake on round $t_i = \frac{iT}{n+1} + 1$. Finally, by Observation 3, the example x_{t_i} and the previously observed sequence $x_{t_{i-1}:t_i-1}$ gives away j_i, thus $\mathcal{P}$ sets $J_i = j_i$ on line 8 in round $t = \frac{iT}{n+1} + 1$. This completes the induction step and the proof of the claim that $\mathcal{P}$ only makes mistakes on timepoints $\{\frac{T}{n+1} + 1, ..., \frac{nT}{n+1} + 1\}$. To see that $\mathcal{P}$ is lazy, observe that by line 12, $\mathcal{P}$ does not update its prediction on rounds in between those in $\{\frac{T}{n+1} + 1, ..., \frac{nT}{n+1} + 1\}$. To see that $\mathcal{P}$ is consistent, note that $\mathcal{P}$ uses prefixes of $j_1, ..., j_n$, S^0, and SG to compute its predictions in line 10. Thus, consistency follows from Observation 2. ■

Step 3: Equivalence to Online Classification with Peeks

For any stream $x_{1:T} \in \mathcal{Z}_n$, having access to the Predictor specified by Algorithm 13 implies that at every $t \in \{1, \frac{T}{n+1} + 1, ..., \frac{nT}{n+1} + 1\}$, the learner observes predictions $\hat{x}_{1:T}^t$ where $\hat{x}_{1:t-1}^t =$

$x_{1:t-1}$, $\hat{x}^t_{t:t+\frac{T}{n+1}} = x_{t:t+\frac{T}{n+1}}$, and $\hat{x}^t_{t+\frac{T}{n+1}+1:T} = (\star, ..., \star)$. Accordingly, at the timepoints $t \in \{1, \frac{T}{n+1}+1, ..., \frac{nT}{n+1}+1\}$, the learner observes the next $\frac{T}{n+1}-1$ examples $x_{t:t+\frac{T}{n+1}}$ in the stream, but learns nothing about the future examples $x_{t+\frac{T}{n+1}+1:T}$. In addition, for timepoints in between those in $\{1, \frac{T}{n+1}+1, ..., \frac{nT}{n+1}+1\}$, the learner does not observe any new information from $\mathcal{P}$ since by line 12 in Algorithm 13, $\hat{x}^i_{1:T} = \hat{x}^{i+r}_{1:T}$ for every $i \in \{1, \frac{T}{n+1}+1, ..., \frac{nT}{n+1}+1\}$ and $r \in \{1, ..., \frac{T}{n+1}-1\}$. As a result, whenever $x_{1:T} \in \mathcal{Z}_n$, Protocol 2 with the Predictor specified by Algorithm 13 is equivalent to the setting we call Online Classification with Peeks where there is no Predictor, but the learner observes the next $\frac{T}{n+1}-1$ examples exactly at timepoints $t \in \{1, \frac{T}{n+1}+1, ..., \frac{nT}{n+1}+1\}$. Indeed, by having knowledge of the next $\frac{T}{n+1}-1$ examples exactly at timepoints $t \in \{1, \frac{T}{n+1}+1, ..., \frac{nT}{n+1}+1\}$, a learner for Online Classification with Peeks can simulate a Predictor that acts like Algorithm 13. Likewise, a learner for Online Classification with Predictions can use Algorithm 13 to simulate an adversary that reveals the next $\frac{T}{n+1}-1$ examples exactly at timepoints $t \in \{1, \frac{T}{n+1}+1, ..., \frac{nT}{n+1}+1\}$. Accordingly, we consider Online Classification with Peeks for the rest of the proof and show how Nature can force the lower bound in Theorem 3.3.8 under this new setting.

Step 4: Nature's Strategy for Online Classification with Peeks

Let $\mathcal{A}$ be any online learner and consider the game where the learner $\mathcal{A}$ observes the next $\frac{T}{n+1}-1$ examples at timepoints $\{1, \frac{T}{n+1}+1, ..., \frac{nT}{n+1}+1\}$. We construct a hard stream for $\mathcal{A}$ in this setting. We first describe a minimax optimal offline strategy for Nature when it is forced to play a sequence of examples $S \in \mathcal{S}$ sorted in Dyadic order.

Algorithm 14 Nature's Minimax Offline Strategy

Require: $\tilde{S} = \text{Dyd}(S)$ for some $S \in \mathcal{S}$, Version space $V \subseteq \{0,1\}^{\mathcal{X}}$
1: **Initialize:** $V_1 = V$
2: Reveal $\tilde{S}$ to the learner $\mathcal{A}$.
3: **for** $t = 1, ..., \frac{T}{n+1}$ **do**
4: Observe the probability $\hat{p}_t$ of $\mathcal{A}$ predicting label 1.
5: **if** $\hat{p}_t \geq 1/2$ **then**
6: If there exists $h \in V_t$ such that $h(x_t) = 0$, reveal true label $y_t = 0$. Else, reveal $y_t = 1$.
7: **else**
8: If there exists $h \in V_t$ such that $h(x_t) = 1$, reveal true label $y_t = 1$. Else, reveal $y_t = 0$.
9: **end if**
10: Update $V_{t+1} = \{h \in V_t : h(x_t) = y_t\}$.
11: **end for**
12: Return: True labels $y_1, ...y_{\frac{T}{n+1}}$, Version space $V_{\frac{T}{n+1}+1}$

Lemma A.4.2. *For any learner $\mathcal{A}$, $\tilde{S} = \mathrm{Dyd}(S)$, and Version space $V \subseteq \{0,1\}^{\mathcal{X}}$, Algorithm 14 forces $\mathcal{A}$ to make at least $\frac{1}{2}\log_2(\frac{T}{n+1})$ mistakes in expectation if S is threshold-shattered (Definition A.1.5) by V.*

Proof. The lemma follows directly from Theorem 3.4 in Hanneke et al. [2024]. ■

For the definition of threshold-shattering, see Appendix A.1. Note that for every input $\tilde{S} = \mathrm{Dyd}(S)$ and $V \subseteq \{0,1\}^{\mathcal{X}}$ to Algorithm 14, its output version space $V_{|\tilde{S}|+1}$ is non-empty and consistent with the sequence $(\tilde{S}_1, y_1), ..., (\tilde{S}_{\frac{T}{n+1}}, y_{\frac{T}{n+1}})$ as long as $|V| > 0$. This property will be crucial when proving Lemma A.4.4. We are now ready to describe Nature's strategy for Online Classification with Peeks. The pseudocode is provided in Algorithm 15.

Algorithm 15 Nature's Strategy for Online Classification with Peeks

Require: Learner $\mathcal{A}$, Hypothesis class $\mathcal{H}$

Initialize: $V_1 = \mathcal{H}$
for $i = 1, .., n+1$ **do**
  if $i = 1$ **then**
    Set $x_{1:\frac{T}{n+1}} = \mathrm{Dyd}(S^0)$ and reveal it to the learner $\mathcal{A}$.
  else
    Compute $S = \mathrm{SG}(S^0, (j_1, ..., j_{i-1}))$.
    Let $x_{\frac{(i-1)T}{n+1}+1:\frac{iT}{n+1}}$ be the last $\frac{T}{n+1}$ examples in S and reveal it to the learner $\mathcal{A}$.
  end if
  Play against $\mathcal{A}$ according to Algorithm 14 using $x_{\frac{(i-1)T}{n+1}+1:\frac{iT}{n+1}}$ and version space V_i.
  Let $y_{\frac{(i-1)T}{n+1}+1}, ..., y_{\frac{iT}{n+1}}$ be the returned labels and $V_{i+1} \subseteq V_i$ be the returned version space.
  Let $\tilde{y}_{\frac{(i-1)T}{n+1}+1}, ..., \tilde{y}_{\frac{iT}{n+1}}$ be the sequence of true labels after sorting

$$(x_{\frac{(i-1)T}{n+1}+1}, y_{\frac{(i-1)T}{n+1}+1}), ..., (x_{\frac{iT}{n+1}}, y_{\frac{iT}{n+1}})$$

  in increasing order with respect to the examples.
  if $\tilde{y}_{\frac{iT}{n+1}} = 1$ **then**
    Set $j_i = \frac{T}{n+1} + 1$.
  else
    Set j_i to be the smallest $p \in \{1, ..., \frac{T}{n+1}\}$ such that $\tilde{y}_{\frac{(i-1)T}{n+1}+p} = 0$.
  end if
end for
Return: Stream $(x_1, y_1), ..., (x_T, y_T)$, indices $j_{1:n}$, and version spaces $V_1, ..., V_{n+2}$.

We establish a series of important lemmas.

Lemma A.4.3. *For every learner $\mathcal{A}$, if $(x_1, y_1), ..., (x_T, y_T)$ is the stream the output of Algorithm 15 when playing against $\mathcal{A}$, then $x_{1:T} \in \mathcal{Z}_n$.*

Proof. Fix a learner $\mathcal{A}$ and let $(x_1, y_1), ..., (x_T, y_T)$ denote the output of Algorithm 15 playing against $\mathcal{A}$. Let $j_{1:n}$ denote the sequences of indices output by Algorithm 15. Then, since SG is deterministic, by line 6-7 in Algorithm 15, we have that $x_{1:T} = \text{SG}(S^0, (j_1, ..., j_n)) \in \mathcal{Z}_n$. ■

Lemma A.4.4. *For every learner $\mathcal{A}$, if $(x_1, y_1), ..., (x_T, y_T)$ is the stream the output of Algorithm 15 when playing against $\mathcal{A}$, then $(x_1, y_1), ..., (x_T, y_T)$ is realizable by $\mathcal{H}$.*

Proof. Fix a learner $\mathcal{A}$ and let $(x_1, y_1), ..., (x_T, y_T)$ be the output of Algorithm 15 when playing against $\mathcal{A}$. Let $V_2, ..., V_{n+2}$ be the sequence of version spaces output by Algorithm 15. It suffices to show that V_{n+2} is not empty and is consistent with $(x_1, y_1), ..., (x_T, y_T)$. Our proof will be by induction using the following hypothesis: V_{i+1} is non-empty and consistent with the sequence $(x_1, y_1), ..., (x_{\frac{iT}{n+1}}, y_{\frac{iT}{n+1}})$. For the base case, let $i = 1$. Then, by Algorithm 14, line 9 in Algorithm 15, and the fact that $|V_1| = |\mathcal{H}| > 0$, we have that $|V_2| > 0$ and V_2 is consistent with $(x_1, y_1), ..., (x_{\frac{T}{n+1}}, y_{\frac{T}{n+1}})$. Now consider some $i \geq 2$ and suppose the induction hypothesis is true for $i - 1$, Then, we know that $|V_i| > 0$ and V_i is consistent with $(x_1, y_1), ..., (x_{\frac{(i-1)T}{n+1}}, y_{\frac{(i-1)T}{n+1}})$. Again, by design of Algorithm 14 and line 10 in Algorithm 15, it follows that $|V_{i+1}| > 0$ and V_{i+1} is consistent with $(x_{\frac{(i-1)T}{n+1}+1}, y_{\frac{(i-1)T}{n+1}+1}), ..., (x_{\frac{iT}{n+1}}, y_{\frac{iT}{n+1}})$. Since $V_{i+1} \subseteq V_i$, and V_i is consistent with $(x_1, y_1), ..., (x_{\frac{(i-1)T}{n+1}}, y_{\frac{(i-1)T}{n+1}})$, we get that V_{i+1} is consistent with $(x_1, y_1), ..., (x_{\frac{iT}{n+1}}, y_{\frac{iT}{n+1}})$, completing the induction step. ■

Lemma A.4.5. *For every learner $\mathcal{A}$, if $(x_1, y_1), ..., (x_T, y_T)$ and $V_1, ..., V_{n+2}$ are stream and version spaces output by Algorithm 15 when playing against $\mathcal{A}$, then for every $i \in \{1, ..., n + 1\}$, the version space V_i threshold-shatters $x_{\frac{(i-1)T}{n+1}+1:\frac{iT}{n+1}}$.*

Proof. Fix a learner $\mathcal{A}$ and let $(x_1, y_1), ..., (x_T, y_T)$ denote the output of Algorithm 15 playing against $\mathcal{A}$. Let $j_{1:n}$ and $V_1, ..., V_{n+2}$ denote the sequences of indices and version spaces output by Algorithm 15 respectively. Note that $x_{1:T} = \text{SG}(S^0, j_{1:n})$. Moreover, for every $i \in \{2, ..., n\}$, we have that $x_{1:\frac{iT}{n+1}} = \text{SG}(S^0, (j_1, ..., j_{i-1}))$ by lines 6-7.

Fix an $i \in \{1, ..., n + 1\}$. It suffices to show that the hypotheses parameterized by $x_{\frac{(i-1)T}{n+1}+1:\frac{iT}{n+1}}$ belong in V_i. Our proof will be by induction. For the base case, since $V_1 = \mathcal{H}$, it trivially follows that the hypothesis parameterized by $x_{\frac{(i-1)T}{n+1}+1:\frac{iT}{n+1}}$ belong to V_1. Now, suppose that $x_{\frac{(i-1)T}{n+1}+1:\frac{iT}{n+1}}$ belong to V_m for some $m < i$. We show that V_{m+1} also contains the hypothesis parameterized by $x_{\frac{(i-1)T}{n+1}+1:\frac{iT}{n+1}}$. Recall that $V_{m+1} \subseteq V_m$ is the subset of V_m that is consistent with the labeled data

$$(x_{\frac{(m-1)T}{n+1}+1}, y_{\frac{(m-1)T}{n+1}+1}), ..., (x_{\frac{mT}{n+1}}, y_{\frac{mT}{n+1}})$$

and is the result of running Algorithm 14 with input version space V_m and sequence $x_{\frac{(m-1)T}{n+1}+1:\frac{mT}{n+1}}$. It suffices to show that the hypotheses parameterized by $x_{\frac{(i-1)T}{n+1}+1:\frac{iT}{n+1}}$ are

also consistent with

$$(x_{\frac{(m-1)T}{n+1}+1}, y_{\frac{(m-1)T}{n+1}+1}), ..., (x_{\frac{mT}{n+1}}, y_{\frac{mT}{n+1}}).$$

To show this, recall that j_m is the index computed in Lines 12-16 of Algorithm 15 on round m. Let

$$(\tilde{x}_{\frac{(m-1)T}{n+1}+1}, \tilde{y}_{\frac{(m-1)T}{n+1}+1}), ..., (\tilde{x}_{\frac{mT}{n+1}}, \tilde{y}_{\frac{mT}{n+1}}).$$

be the sample sorted in increasing order by examples. There are three cases to consider. Suppose $j_m = 1$, then $\tilde{y}_{\frac{(m-1)T}{n+1}+1} = 0$, and it must be the case that $\tilde{y}_{\frac{(m-1)T}{n+1}+p} = 0$ for all $p \in \{2, ..., \frac{T}{n+1}\}$. Since $x_{1:\frac{mT}{n+1}} = \text{SG}(S^0, (j_1, ..., j_{m-1}))$, by definition of Algorithm 12, we have that the last $\frac{T}{n+1}$ entries of $\text{SG}(S^0, (j_1, ..., j_m))$ all lie strictly to the left of $\tilde{x}_{\frac{(m-1)T}{n+1}+1}$. Moreover, by Observation 4, this is true of the last $\frac{T}{n+1}$ entries of $\text{SG}(S^0, (j_1, ..., j_m, q_{m+1}, ..., q_{i-1})$ for any $q_{m+1}, ..., q_{i-1} \in \{1, ..., \frac{T}{n+1} + 1\}$. Therefore, we must have that $x_{\frac{(i-1)T}{n+1}+1:\frac{iT}{n+1}}$, which are the last $\frac{T}{n+1}$ entries of $\text{SG}(S^0, (j_1, ..., j_{i-1}))$, lies strictly to the left of $\tilde{x}_{\frac{(m-1)T}{n+1}+1}$, implying that their associated hypotheses output 0 on all of $(x_{\frac{(m-1)T}{n+1}+1}, y_{\frac{(m-1)T}{n+1}+1}), ..., (x_{\frac{mT}{n+1}}, y_{\frac{mT}{n+1}})$ as needed. By symmetry, when $j_m = \frac{T}{n+1} + 1$, we have that $x_{\frac{(i-1)T}{n+1}+1:\frac{iT}{n+1}}$ lies strictly to the right of $\tilde{x}_{\frac{mT}{n+1}}$, implying that their associated hypotheses output 1 on all of $(x_{\frac{(m-1)T}{n+1}+1}, y_{\frac{(m-1)T}{n+1}+1}), ..., (x_{\frac{mT}{n+1}}, y_{\frac{mT}{n+1}})$ as needed. Now, consider the case where $j_m \in \{2, ..., \frac{T}{n+1}\}$. Then, by Algorithm 12 and Observation 4, for any $q_{m+1}, ..., q_i \in \{1, ..., \frac{T}{n+1} + 1\}$, the last $\frac{T}{n+1}$ entries of $\text{SG}(S^0, (j_1, ..., j_m, q_{m+1}, ..., q_{i-1})$ lie strictly in between $\tilde{x}_{\frac{(m-1)T}{n+1}+j_m-1}$ and $\tilde{x}_{\frac{(m-1)T}{n+1}+j_m}$. Thus, the hypotheses parameterized by $x_{\frac{(i-1)T}{n+1}+1:\frac{iT}{n+1}}$ output 1 on examples $\tilde{x}_{\frac{(m-1)T}{n+1}+1:\frac{(m-1)T}{n+1}+j_m-1}$ and 0 on examples $\tilde{x}_{\frac{(m-1)T}{n+1}+j_m:\frac{mT}{n+1}}$. Finally, note that by definition of j_m, it must be the case that $\tilde{y}_{\frac{(m-1)T}{n+1}+1:\frac{(m-1)T}{n+1}+j_m-1} = (1, ..., 1)$ and $\tilde{y}_{\frac{(m-1)T}{n+1}+j_m:\frac{mT}{n+1}} = (0, ..., 0)$. Thus, once again the hypotheses parameterized by $x_{\frac{(i-1)T}{n+1}+1:\frac{iT}{n+1}}$ are consistent with the sample

$$(x_{\frac{(m-1)T}{n+1}+1}, y_{\frac{(m-1)T}{n+1}+1}), ..., (x_{\frac{mT}{n+1}}, y_{\frac{mT}{n+1}}).$$

This shows that these hypotheses are contained in V_{m+1}, completing the induction step. ■

Step 5: Completing the proof of Theorem 3.3.8

We are now ready to complete the proof of Theorem 3.3.8, which follows from composing A.4.1, A.4.2, A.4.3, A.4.4, and A.4.5. Namely, Lemma A.4.1 and the discussion in Section A.4.2 show that there exists a Predictor $\mathcal{P}$ such that for any learner $\mathcal{A}$ playing according to Protocol 2, Online Classification with Predictions is equivalent to Online Classification with Peeks whenever the stream $(x_1, y_1), ..., (x_T, y_T)$ selected by the adversary satisfies the constraint that $x_{1:T} \in \mathcal{Z}_n$. Lemmas A.4.3 and A.4.4 show that for any learner $\mathcal{A}$, Nature playing

according to Algorithm 15 guarantees that the resulting sequence $(x_1, y_1), ..., (x_T, y_T)$ satisfies the constraint that $x_{1:T} \in \mathcal{Z}_n$ and realizability by $\mathcal{H}$. Thus, for the Predictor $\mathcal{P}$ specified by Algorithm 13 and Nature playing according to Algorithm 15, Online Classification with Predictions is equivalent to Online Classification with Peeks. Finally, for Online Classification with Peeks, combining Lemmas A.4.2 and A.4.5 shows that for any learner $\mathcal{A}$, Nature, by playing according to Algorithm 15, guarantees that $\mathcal{A}$ makes at least $\frac{\log_2(\frac{T}{n+1})}{2}$ mistakes in expectation every $\frac{T}{n+1}$ rounds. Thus, Nature forces $\mathcal{A}$ to make at least $\frac{(n+1)}{2}\log_2(\frac{T}{n+1})$ mistakes in expectation by the end of the game, completing the proof.

A.5 Adaptive Rates in the Agnostic Setting

In this section, we consider the harder agnostic setting and prove analogous results as in Section 6.3. Our main quantitative result is the agnostic analog of Theorem 3.3.1.

Theorem A.5.1 (Agnostic upper bound)**.** *For every $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, Predictor $\mathcal{P}$, and no-regret offline learner $\mathcal{B}$, there exists an online learner $\mathcal{A}$ such that for every stream $(x_1, y_1), ..., (x_T, y_T)$, $\mathcal{A}$'s expected regret is at most*

$$\left(\underbrace{\sqrt{\mathrm{L}(\mathcal{H})\,T\log_2(eT)}}_{(i)} \wedge \underbrace{\left(2(\mathrm{M}_{\mathcal{P}}(x_{1:T})+1)\,\overline{\mathrm{R}}_{\mathcal{B}}\Big(\frac{T}{\mathrm{M}_{\mathcal{P}}(x_{1:T})+1}+1, \mathcal{H}\Big)+\sqrt{T\log_2 T}\right)}_{(ii)}\right)+\sqrt{T}.$$

With respect to learnability, Corollary A.5.2 shows that offline learnability of $\mathcal{H}$ is sufficient for online learnability under predictable examples.

Corollary A.5.2 (Offline learnability $\implies$ Agnostic Online learnability with Predictable Examples)**.** *For every $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ and $\mathcal{Z} \subseteq \mathcal{X}^{\star}$,*

$$\mathcal{Z} \text{ is predictable } \underline{\text{and}}\ \mathcal{H} \text{ is offline learnable } \implies (\mathcal{H}, \mathcal{Z}) \text{ is agnostic online learnable.}$$

In addition, we can also establish a quantitative version of Corollary A.5.2 for VC classes.

Corollary A.5.3. *For every $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$, Predictor $\mathcal{P}$, $\mathcal{Z} \subseteq \mathcal{X}^{\star}$, and no-regret offline learner $\mathcal{B}$, there exists an online learner $\mathcal{A}$ such that*

$$\mathrm{R}_{\mathcal{A}}(T, \mathcal{H}, \mathcal{Z}) = O\left((\mathrm{M}_{\mathcal{P}}(T, \mathcal{Z})+1)\sqrt{\frac{\mathrm{VC}(\mathcal{H})\,T}{\mathrm{M}_{\mathcal{P}}(T, \mathcal{Z})+1}\log_2\Big(\frac{T}{\mathrm{M}_{\mathcal{P}}(T, \mathcal{Z})+1}\Big)}+\sqrt{T\log_2 T}\right).$$

The proof of Corollary A.5.3 is in Section A.5.4. The remainder of this section is dedicated to proving Theorem 3.3.1 and Corollary A.5.3. The proof is similar to the realizable case. It

involves constructing two online learners with expected regret bounds (i) and (ii) respectively, and then running the celebrated Randomized Exponential Weights Algorithm (REWA) using these learners as experts [Cesa-Bianchi and Lugosi, 2006]. The following guarantee of REWA along with upper bound (i) and (ii) gives the upper bound in Theorem A.5.1.

Lemma A.5.4 (REWA guarantee [Cesa-Bianchi and Lugosi, 2006])**.** *The expected regret of REWA when run with N experts and learning rate $\eta = \sqrt{\frac{8 \ln N}{T}}$ is at most $\min_{i \in [N]} M_i + \sqrt{T \log_2 N}$, where M_i is the number of mistakes made by expert $i \in [N]$.*

The online learner obtaining the regret bound $\sqrt{\mathrm{L}(\mathcal{H}) T \log_2(eT)}$ is the generic agnostic online learner from Hanneke et al. [2023], thus we omit the details here. Our second learner is described in Section A.5.2 and uses Algorithm 3 as a subroutine. The following lemma, bounding the expected regret of Algorithm 3 in the agnostic setting, will be crucial.

Lemma A.5.5. *For every $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, Predictor $\mathcal{P}$, no-regret offline learner $\mathcal{B}$, and stream $(x_1, y_1), ..., (x_T, y_T)$, the expected regret of Algorithm 3 is at most $(\mathrm{M}_{\mathcal{P}}(x_{1:T}) + 1)\, \mathrm{R}_{\mathcal{B}}(T, \mathcal{H})$.*

A.5.1 Proof of Lemma A.5.5

The proof closely follows that of Lemma 3.3.5.

Proof. Let $\mathcal{A}$ denote Algorithm 3 and $(x_1, y_1), ..., (x_T, y_T)$ denote the stream to be observed by $\mathcal{A}$. Let c be the random variable denoting the number of mistakes made by Predictor $\mathcal{P}$ on the stream and $t_1, ..., t_c$ be the random variables denoting the time points where $\mathcal{P}$ makes these errors (e.g . $\hat{x}_{t_i}^{t_i - 1} \neq x_{t_i}$). Note that $t_i \geq 2$ for all $i \in [c]$. We will show pointwise for every value of c and $t_1, ..., t_c$ that $\mathcal{A}$ makes at most $(c+1)\, \mathrm{R}_{\mathcal{B}}(T, \mathcal{H})$ mistakes in expectation over the randomness of $\mathcal{B}$. Taking an outer expectation with respect to the randomness of $\mathcal{P}$ and using the fact that $\mathbb{E}[c] = \mathrm{M}_{\mathcal{P}}(x_{1:T})$, completes the proof.

First, consider the case where $c = 0$ (i.e. $\mathcal{P}$ makes no mistakes). Then, since $\mathcal{P}$ is lazy, we have that $\hat{x}_{1:T}^{t} = x_{1:T}$ for every $t \in [T]$. Thus line 5 fires exactly once on round $t = 1$, $\mathcal{A}$ initializes an offline learner $\mathcal{B}^1$ with $x_{1:T}$, and $\mathcal{A}$ uses $\mathcal{B}^1$ to make its prediction on all rounds. Thus, $\mathcal{A}$ makes at most $\mathrm{R}_{\mathcal{B}}(T, \mathcal{H})$ mistakes in expectation.

Now, let $c > 0$ and $t_1, ..., t_c$ be the time points where $\mathcal{P}$ errs. Partition the sequence $1, ..., T$ into the disjoint intervals $(1, ..., t_1 - 1)$, $(t_1, ..., t_2 - 1), ..., (t_c, ..., T)$. Define $t_0 := 1$ and $t_{c+1} := T$. Fix an $i \in \{0, ..., c\}$. Then, for every $j \in \{t_i, ..., t_{i+1} - 1\}$, we have that $\hat{x}_{1:t_{i+1}-1}^{j} = x_{t_{i+1}-1}$. This comes from the fact that $\mathcal{P}$ does not error on timepoints $t_i + 1, ..., t_{i+1} - 1$ and is both consistent and lazy (see Assumptions 1 and 2). Thus, line 5 fires on round t_i, $\mathcal{A}$ initializes an offline learner $\mathcal{B}^i$ with the sequence $\hat{x}_{t_i:T}^{t_i} = x_{t_i:t_{i+1}-1} \circ \hat{x}_{t_{i+1}:T}^{t_i}$, and $\mathcal{A}$ uses $\mathcal{B}^i$ it

to make predictions for all remaining timepoints $t_i, ..., t_{i+1} - 1$. Note that line 5 does not fire on timepoints $t_i + 1, ..., t_{i+1} - 1$.

Let $h^i \in \arg\min_{h \in \mathcal{H}} \sum_{t=t_i}^{t_{i+1}-1} \mathbb{1}\{h(x_t) \neq y_t\}$ be an optimal hypothesis for the partition $(t_i, ..., t_{i+1} - 1)$. Let $y_t^i = y_t$ for $t_i \leq t \leq t_{i+1} - 1$ and $y_t^i = h^i(\hat{x}_t^{t_i})$ for all $t \geq t_{i+1}$. Then, note that

$$\inf_{h \in \mathcal{H}} \sum_{t=t_i}^{T} \mathbb{1}\{h(\hat{x}_t^{t_i}) \neq y_t^i\} = \inf_{h \in \mathcal{H}} \sum_{t=t_i}^{t_{i+1}-1} \mathbb{1}\{h(x_t) \neq y_t\}.$$

Now, consider the hypothetical labeled stream

$$(\hat{x}_{t_i}^{t_i}, y_{t_i}^i), ..., (\hat{x}_T^{t_i}, y_T^i) = (x_{t_i}, y_{t_i}), ..., (x_{t_{i+1}-1}, y_{t_{i+1}-1}), (\hat{x}_{t_{i+1}}^{t_i}, y_{t_{i+1}}^i), ..., (\hat{x}_T^{t_i}, y_T^i).$$

By definition, $\mathcal{B}^i$, after initialized with $\hat{x}_{t_i:T}^{t_i}$, makes at most

$$\inf_{h \in \mathcal{H}} \sum_{t=t_i}^{T} \mathbb{1}\{h(\hat{x}_t^{t_i}) \neq y_t^i\} + \mathrm{R}_{\mathcal{B}}(T - t_i, \mathcal{H}) = \inf_{h \in \mathcal{H}} \sum_{t=t_i}^{t_{i+1}-1} \mathbb{1}\{h(x_t) \neq y_t\} + \mathrm{R}_{\mathcal{B}}(T - t_i, \mathcal{H})$$

mistakes in expectation when simulated on the stream $(\hat{x}_{t_i}^{t_i}, y_{t_i}^i), ..., (\hat{x}_T^{t_i}, y_T^i)$. Thus, $\mathcal{B}^i$ makes at most $\inf_{h \in \mathcal{H}} \sum_{t=t_i}^{t_{i+1}-1} \mathbb{1}\{h(x_t) \neq y_t\} + \mathrm{R}_{\mathcal{B}}(T - t_i + 1, \mathcal{H})$ mistakes in expectation on the prefix $(\hat{x}_{t_i}^{t_i}, y_{t_i}^i), ..., (\hat{x}_{t_{i+1}-1}^{t_i}, y_{t_{i+1}-1}^i) = (x_{t_i}, y_{t_i}), ..., (x_{t_{i+1}-1}, y_{t_{i+1}-1})$. Since on timepoint t_i, $\mathcal{A}$ instantiates $\mathcal{B}^i$ with the sequence $\hat{x}_{t_i:T}^{t_i}$ and proceeds to simulate $\mathcal{B}^i$ on the sequences of labeled examples $(x_{t_i}, y_{t_i}), ..., (x_{t_{i+1}-1}, y_{t_{i+1}-1})$, $\mathcal{A}$ makes at most $\inf_{h \in \mathcal{H}} \sum_{t=t_i}^{t_{i+1}-1} \mathbb{1}\{h(x_t) \neq y_t\} + \mathrm{R}_{\mathcal{B}}(T - t_i + 1, \mathcal{H})$ mistakes in expectation on the sequence $(x_{t_i}, y_{t_i}), ..., (x_{t_{i+1}-1}, y_{t_{i+1}-1})$. Since the interval i was chosen arbitrarily, this is true for every $i \in \{0, ..., c\}$ and

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}(x_t) \neq y_t\}\right] &= \mathbb{E}\left[\sum_{i=0}^{c} \sum_{t=t_i}^{t_{i+1}-1} \mathbb{1}\{\mathcal{A}(x_t) \neq y_t\}\right] \\
&\leq \sum_{i=0}^{c} \Big(\inf_{h \in \mathcal{H}} \sum_{t=t_i}^{t_{i+1}-1} \mathbb{1}\{h(x_t) \neq y_t\} + \mathrm{R}_{\mathcal{B}}(T - t_i + 1, \mathcal{H}) \Big) \\
&\leq \inf_{h \in \mathcal{H}} \sum_{t=1}^{T} \mathbb{1}\{h(x_t) \neq y_t\} + (c+1)\, \mathrm{R}_{\mathcal{B}}(T, \mathcal{H}),
\end{aligned}$$

as needed. ■

A.5.2 Proof of upper bound (ii) in Theorem A.5.1

The proof of upper bound (ii) in Theorem A.5.1 closely follows the proof of upper bound (iii) in Theorem 3.3.1 from the realizable setting. The main idea is to run REWA using the same

experts defined in Algorithm 4 and bounding the expected regret in terms of the expected regret of $\mathcal{K}$ from Lemma A.5.5.

We show that Algorithm 5 using REWA in line 2 and the experts in Algorithm 4 with their guarantee in Lemma A.5.5 achieves upper bound (ii) in Theorem A.5.1. Let $(x_1, y_1), ..., (x_T, y_T)$ be the stream to be observed by the learner. Let $\mathcal{A}$ denote the online learner in Algorithm 5 using REWA in line 2 of Algorithm 5. By the guarantees of the REWA, we have that

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}(x_t) \neq y_t\}\right] &\leq \mathbb{E}\left[\inf_{b \in \{0,\dots,T-1\}} \sum_{t=1}^{T} \mathbb{1}\{E_b(x_t) \neq y_t\}\right] + \sqrt{T \log_2 T} \\
&\leq \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{E_{\lceil \mathrm{M}_{\mathcal{P}}(x_{1:T}) \rceil}(x_t) \neq y_t\}\right] + \sqrt{T \log_2 T} \\
&\leq \mathbb{E}\left[\sum_{i=0}^{\lceil \mathrm{M}_{\mathcal{P}}(x_{1:T}) \rceil} \sum_{t=\tilde{t}_i+1}^{\tilde{t}_{i+1}} \mathbb{1}\{\mathcal{K}_i(x_t) \neq y_t\}\right] + \sqrt{T \log_2 T}.
\end{aligned}$$

Note that by the guarantee of $\mathcal{K}$ from Lemma A.5.5, we have

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i=0}^{\lceil \mathrm{M}_{\mathcal{P}}(x_{1:T}) \rceil} \sum_{t=\tilde{t}_i+1}^{\tilde{t}_{i+1}} \mathbb{1}\{\mathcal{K}_i(x_t) \neq y_t\}\right] \leq \\
&\qquad \mathbb{E}\left[\sum_{i=0}^{\lceil \mathrm{M}_{\mathcal{P}}(x_{1:T}) \rceil} (m_i + 1) \overline{\mathrm{R}}_{\mathcal{B}}(\tilde{t}_{i+1} - \tilde{t}_i, \mathcal{H})\right] + \inf_{h \in \mathcal{H}} \sum_{t=1}^{T} \mathbb{1}\{h(x_t) \neq y_t\}.
\end{aligned}$$

Since $\overline{\mathrm{R}}_{\mathcal{B}}(T, \mathcal{H})$ is a concave, sublinear function of T, we can use an identical argument as in the proof of Lemma 3.3.6 to get that

$$\mathbb{E}\left[\sum_{i=0}^{\lceil \mathrm{M}_{\mathcal{P}}(x_{1:T}) \rceil} (m_i + 1) \overline{\mathrm{R}}_{\mathcal{B}}(\tilde{t}_{i+1} - \tilde{t}_i, \mathcal{H})\right] \leq 2(\mathrm{M}_{\mathcal{P}}(x_{1:T}) + 1) \overline{\mathrm{R}}_{\mathcal{B}}\Big(\frac{T}{\mathrm{M}_{\mathcal{P}}(x_{1:T}) + 1} + 1, \mathcal{H}\Big),$$

which completes the proof.

A.5.3 Proof of Theorem A.5.1

Let $\mathcal{A}$ denote the REWA using the generic agnostic online learner from Hanneke et al. [2023] and the algorithm described in Section A.5.2 as experts. Then, for any stream

$(x_1, y_1), ..., (x_T, y_T)$, Lemma A.5.4 gives that

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}(x_t) \neq y_t\}\right] \leq \mathbb{E}\left[\min_{i\in[2]} M_i\right] + \sqrt{T} \leq \min_{i\in[2]} \mathbb{E}\left[M_i\right] + \sqrt{T},$$

where we take M_1 and M_2 to be the number of mistakes made by the generic agnostic online learner from Hanneke et al. [2023] and the algorithm described in Section A.5.2 respectively. Note that M_1 and M_2 are random variables. Finally, using [Hanneke et al., 2023, Theorem 4] as well as upper bound (ii) completes the proof of Theorem A.5.1.

A.5.4 Proof of Corollaries A.5.2 and A.5.3

The proof of the generic upper bound on $\mathrm{M}_{\mathcal{A}}(T, \mathcal{H}, \mathcal{Z})$ follows by using the same learner $\mathcal{A}$ as in the proof of upper bound (ii) in Theorem A.5.1. However, this time we bound

$$\mathbb{E}\left[\inf_{b\in\{0,\ldots,T-1\}} \sum_{t=1}^{T} \mathbb{1}\{E_b(x_t) \neq y_t\}\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{E_{\lceil \mathrm{M}_{\mathcal{P}}(T,\mathcal{Z})\rceil}(x_t) \neq y_t\}\right]$$

and use an identical analysis as in the proof of upper bound (ii) and Lemma 3.3.6 to get

$$\mathrm{R}_{\mathcal{A}}(T, \mathcal{H}, \mathcal{Z}) = O\Bigg(\sqrt{\mathrm{L}(\mathcal{H})\, T \log_2 T} \wedge \Big((\mathrm{M}_{\mathcal{P}}(T, \mathcal{Z}) + 1)\, \overline{\mathrm{R}}_{\mathcal{B}}\Big(\frac{T}{\mathrm{M}_{\mathcal{P}}(T, \mathcal{Z}) + 1}, \mathcal{H}\Big) + \sqrt{T \log_2 T}\Big)\Bigg).$$

Corollary A.5.2 follows from the fact that $\mathrm{R}_{\mathcal{A}}(T, \mathcal{H}, \mathcal{Z}) = o(T)$ if $\mathrm{M}_{\mathcal{P}}(T, \mathcal{Z}) = o(T)$ and $\mathrm{R}_{\mathcal{B}}(T, \mathcal{H}) = o(T)$. To get the upper bound in Corollary A.5.3, it suffices to plug in the upper bound $\overline{\mathrm{R}}_{\mathcal{B}}(T, \mathcal{H}) = O\Big(\sqrt{\mathrm{VC}(\mathcal{H}) T \log_2 T}\Big)$, given by Theorem 6.1 from Hanneke et al. [2024], into the above upper bound on $\mathrm{R}_{\mathcal{A}}(T, \mathcal{H}, \mathcal{Z})$.

APPENDIX B

The Complexity of Sequential Prediction in Dynamical Systems

B.1 Proofs for Realizable learnability

B.1.1 Proof of Theorem 4.3.1

Proof. (of lower bound of Theorem 4.3.1). Let $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ be any evolution class and $T \in \mathbb{N}$ be the time horizon. Let $\mathcal{A}$ be any randomized learner. Our goal will be to construct a hard realizable trajectory $\{x_t\}_{t=0}^{T}$ such that $\mathcal{A}$'s expected number of mistakes is at least $\frac{\mathrm{C}_T(\mathcal{F})}{2}$. Without loss of generality, suppose $d := \mathrm{C}_T(\mathcal{F}) > 0$ as otherwise the lower bound holds trivially. Then, by definition of the Evolution complexity, there exists a trajectory tree $\mathcal{T}$ of depth T shattered by $\mathcal{F}$ with branching factor at least d. This means that for every path $\sigma \in \{-1,1\}^T$ down $\mathcal{T}$, we have $\sum_{t=1}^{T} \mathbb{1}\{\mathcal{T}_t((\sigma_{<t},-1)) \neq \mathcal{T}_t((\sigma_{<t},+1))\} \geq d$.

Let $\sigma \sim \{-1,1\}^T$ denote a random path down $\mathcal{T}$ and consider the trajectory $\mathcal{T}_0 \cup \{\mathcal{T}_t(\sigma_{\leq t})\}_{t=1}^{T}$. Define $\mathcal{T}_{<t}(\sigma_{<t}) := (\mathcal{T}_0, \mathcal{T}_1(\sigma_1), \ldots, \mathcal{T}_{t-1}(\sigma_{<t}))$. Then, observe that

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}(\mathcal{T}_{<t}(\sigma_{<t})) \neq \mathcal{T}_t(\sigma_{\leq t})\}\right] &= \sum_{t=1}^{T} \mathbb{E}\Big[\mathbb{E}\left[\mathbb{1}\left\{\mathcal{A}\big(\mathcal{T}_{<t}(\sigma_{<t})\big) \neq \mathcal{T}_t\big((\sigma_{<t},\sigma_t)\big)\right\} \,\middle|\, \sigma_{<t}\right]\Big] \\
&\geq \frac{1}{2}\sum_{t=1}^{T} \mathbb{E}\left[\mathbb{E}\Big[\mathbb{1}\left\{\mathcal{T}_t\big((\sigma_{<t},-1)\big) \neq \mathcal{T}_t\big((\sigma_{<t},+1)\big)\right\} \,\Big|\, \sigma_{<t}\Big]\right] \\
&= \frac{1}{2}\,\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left\{\mathcal{T}_t((\sigma_{<t},-1)) \neq \mathcal{T}_t\big((\sigma_{<t},+1)\big)\right\}\right] \geq \frac{d}{2}
\end{aligned}$$

where the first inequality follows from the fact that $\sigma_t \sim \{-1,1\}$. ■

Proof. (of upper bound of Theorem 4.3.1). Let $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ be any evolution function class and let $T \in \mathbb{N}$ be the time horizon. Our goal will be to construct a deterministic learner $\mathcal{A}$ such

that for any realizable trajectory $\{x_t\}_{t=0}^T$, $\mathcal{A}$ makes at most $\mathrm{C}_T(\mathcal{F})$ mistakes. To that end, it will be useful to define an instance-dependent version of the evolution complexity.

Given a state $x \in \mathcal{X}$, we say $\mathcal{T}$ is a trajectory tree rooted at x if the root node is labeled by x. For an initial state $x_0 \in \mathcal{X}$ and evolution class $V \subseteq \mathcal{F}$, let

$$\mathrm{C}_T(V, x_0) := \sup\{\mathrm{B}(\mathcal{T}) \mid \mathcal{T} \in \mathcal{S}(V, T) \text{ and } \mathcal{T} \text{ rooted at } x_0\}$$

denote an instance-dependent Evolution complexity of V, where $\mathcal{S}(V, T)$ is set of all trajectory trees of depth T shattered by V. Note that $\mathrm{C}_T(V) = \sup_{x_0 \in \mathcal{X}} \mathrm{C}_T(V, x_0)$. $\mathcal{A}$ is a version-space algorithm that uses the instance-dependent Evolution complexity to make its prediction such that whenever $\mathcal{A}$ errs, the instance-dependent Evolution complexity decreases. In this way, $\mathrm{C}_T(V, x_0)$ acts as a potential function. Algorithm 16 formalizes this idea.

Algorithm 16 Deterministic Realizable Algorithm

Require: Initial state $x_0 \in \mathcal{X}$

Let $V_1 = \mathcal{F}$
for $t = 1, ..., T$ **do**
    For $x \in \mathcal{X}$, define $V_t^x = \{f \in V_t : f(x_{t-1}) = x\}$.
    if $\{f(x_{t-1}) : f \in V_t\} = \{x\}$ **then**
        Predict $\hat{x}_t = x$.
    else
        Predict $\hat{x}_t \in \arg\max_{x \in \mathcal{X}} \mathrm{C}_{T-t}(V_t^x, x)$.
    end if
    Receive x_t and update $V_{t+1} \leftarrow V_t^{x_t}$.
end for

We now show that Algorithm 16 makes at most $\mathrm{C}_T(\mathcal{F})$ mistakes on any realizable trajectory. Let $\{x_t\}_{t=0}^T$ denote the realizable trajectory to be observed. It suffices to show that

$$\mathrm{C}_{T-t}(V_{t+1}, x_t) \leq \mathrm{C}_{T-t+1}(V_t, x_{t-1}) - \mathbb{1}\{x_t \neq \hat{x}_t\} \tag{B.1}$$

for all $t \in [T]$. To see why this is true, note that (B.1) implies

$$\sum_{t=1}^{T} \mathrm{C}_{T-t}(V_{t+1}, x_t) \leq \sum_{t=1}^{T} \mathrm{C}_{T-t+1}(V_t, x_{t-1}) - \sum_{t=1}^{T} \mathbb{1}\{x_t \neq \hat{x}_t\}.$$

Rearranging, we get that

$$\begin{aligned}\sum_{t=1}^{T} \mathbb{1}\{x_t \neq \hat{x}_t\} &\leq \sum_{t=1}^{T} \mathrm{C}_{T-t+1}(V_t, x_{t-1}) - \sum_{t=1}^{T} \mathrm{C}_{T-t}(V_{t+1}, x_t) \\ &= \sum_{t=1}^{T} \Big(\mathrm{C}_{T-t+1}(V_t, x_{t-1}) - \mathrm{C}_{T-t}(V_{t+1}, x_t) \Big) \\ &= \mathrm{C}_T(V_1, x_0) - \mathrm{C}_0(V_{T+1}, x_T) \leq \mathrm{C}_T(\mathcal{F}).\end{aligned}$$

as needed. To prove (B.1), we need to consider three cases: (a) $\mathrm{C}_{T-t+1}(V_t, x_{t-1}) > 0$ and $\hat{x}_t \neq x_t$, (b) $\mathrm{C}_{T-t+1}(V_t, x_{t-1}) > 0$ and $\hat{x}_t = x_t$, and (c) $\mathrm{C}_{T-t+1}(V_t, x_{t-1}) = 0$.

Starting with (a), let $t \in [T]$ such that $\mathrm{C}_{T-t+1}(V_t, x_{t-1}) > 0$ and $\hat{x}_t \neq x_t$. We need to show that $\mathrm{C}_{T-t}(V_{t+1}, x_t) < \mathrm{C}_{T-t+1}(V_t, x_{t-1})$. Suppose for the sake of contradiction that $\mathrm{C}_{T-t}(V_{t+1}, x_t) \geq \mathrm{C}_{T-t+1}(V_t, x_{t-1}) := d$. Then, by the prediction rule, we have that $\mathrm{C}_{T-t}(V_t^{\hat{x}_t}, \hat{x}_t) \geq \mathrm{C}_{T-t}(V_t^{x_t}, x_t) \geq d$. By definition of the instance-dependent Evolution complexity, we are guaranteed the existence of a trajectory tree $\mathcal{T}_{x_t}$ of depth $T-t$, rooted at x_t, shattered by $V_t^{x_t}$ with branching factor at least d and a tree $\mathcal{T}_{\hat{x}_t}$ of depth $T-t$, rooted at $\hat{x}_t$, shattered by $V_t^{\hat{x}_t}$ with branching factor at least d. Consider a binary tree $\mathcal{T}$ whose root node v is labeled by x_{t-1} and left and right subtrees are $\mathcal{T}_{x_t}$ and $\mathcal{T}_{\hat{x}_t}$ respectively. Then, observe that $\mathcal{T}$ is a trajectory tree of depth $T-t+1$, rooted at x_{t-1}, shattered by V_t with branching factor at least $d+1$ (because $x_t \neq \hat{x}_t$). This contradicts our assumption that $\mathrm{C}_{T-t+1}(V_t, x_{t-1}) = d$. Thus, we must have $\mathrm{C}_{T-t}(V_{t+1}, x_t) < \mathrm{C}_{T-t+1}(V_t, x_{t-1})$.

Moving to (b), let $t \in [T]$ be such that $\hat{x}_t = x_t$. Then, we need to show that $d := \mathrm{C}_{T-t}(V_{t+1}, x_t) \leq \mathrm{C}_{T-t+1}(V_t, x_{t-1})$. If $d = 0$, the inequality holds trivially. Thus, assume $d > 0$. Then, by definition, there exists a trajectory tree $\mathcal{T}$ of depth $T-t$, rooted at x_t, shattered by V_{t+1} with branching factor at least d. Consider a binary tree $\tilde{\mathcal{T}}$ whose root node v is labeled by x_{t-1} and left and right subtrees are both $\mathcal{T}$. Then since $V_{t+1} = V_t^{x_t} \subseteq V_t$, we have that $\tilde{\mathcal{T}}$ is a trajectory tree of depth $T-t+1$, rooted at x_{t-1}, shattered by V_t with branching factor at least d. Thus, we must have $\mathrm{C}_{T-t+1}(V_t, x_{t-1}) \geq d$.

Finally for (c), let $t \in [T]$ be such that $\mathrm{C}_{T-t+1}(V_t, x_{t-1}) = 0$. Using the same logic as (b), $\mathrm{C}_{T-t}(V_{t+1}, x_t) = 0$. Thus, to prove that (B.1) holds, it suffices to show that $x_t = \hat{x}_t$. To do so, we will show that the projection size $|\{f(x_{t-1}) : f \in V_t\}| = 1$. Thus, Algorithm 16 does not make a mistake. Suppose for the sake of contradiction that $|\{f(x_{t-1}) : f \in V_t\}| \geq 2$. Then, there exists two functions $f_1, f_2 \in V_t$ such that $f_1(x_{t-1}) \neq f_2(x_{t-1})$. Consider a binary tree $\mathcal{T}$ of depth $T-t+1$ where the root node is labeled by x_{t-1}, and its left and right child by $f_1(x_{t-1})$ and $f_2(x_{t-1})$ respectively. Label all remaining internal nodes in the left subtree of the root node such that every path is shattered by f_1. Likewise, for the right subtree of

the root node, label every internal node such that every path is shattered by f_2. Following this procedure, $\mathcal{T}$ is a trajectory tree of depth $T-t+1$, rooted at x_{t-1}, shattered by V_t with branching factor at least 1 (because $f_1(x_{t-1}) \neq f_2(x_{t-1})$). Thus, $\mathrm{C}_{T-t+1}(V_t, x_{t-1}) \geq 1$, which is a contradiction. ■

B.1.2 Proof of Theorem 4.3.2

Let $S \subset \mathbb{N} \cup \{0\}$ be an arbitrary subset of the extended natural numbers. For every $\sigma \in \{-1,1\}^{\mathbb{N}\cup\{0\}}$, define the evolution function

$$f_\sigma(x) = \Big(\sigma_{|x|}\mathbb{1}\{|x| \in S\} + \mathbb{1}\{|x| \notin S\}\Big)(|x|+1)$$

and consider the class $\mathcal{F}_S = \Big\{f_\sigma : \sigma \in \{-1,1\}^{\mathbb{N}\cup\{0\}}\Big\}$. By construction, functions in $\mathcal{F}_S$ only disagree on states in S and their negation. Moreover, given any initial state $x_0 \in \mathbb{Z}$ and a time horizon $T \in \mathbb{N}$, the trajectory of any evolution in $\mathcal{F}$ is $\varepsilon_1(|x_0|+1), ..., \varepsilon_T(|x_0|+T)$ for some $\varepsilon \in \{-1,1\}^T$.

We now show that $\mathrm{C}_T(\mathcal{F}) \leq \sup_{x_0\in\mathbb{Z}} |S \cap \{|x_0|, ..., |x_0|+T-1\}|$. Let $\mathcal{T}$ be any trajectory tree of depth T shattered by $\mathcal{F}$. It suffices to show that $\mathrm{B}(\mathcal{T}) \leq |S \cap \{|\mathcal{T}_0|, ..., |\mathcal{T}_0|+T-1\}|$. Fix any path $\varepsilon \in \{-1,1\}^T$ down $\mathcal{T}$ and consider the sequence of states $\mathcal{T}_0, ..., \mathcal{T}_T(\varepsilon_{\leq T})$. Note that $\mathcal{T}_t((\varepsilon_{<t}, -1)) \neq \mathcal{T}_t((\varepsilon_{<t}, +1))$ only if $|\mathcal{T}_{t-1}(\varepsilon_{<t})| \in S$. Thus,

$$\begin{aligned}
\sum_{t=1}^{T} \mathbb{1}\{\mathcal{T}_t((\varepsilon_{<t}, -1)) \neq \mathcal{T}_t\big((\varepsilon_{<t}, +1)\big)\} &\leq \sum_{t=1}^{T} \mathbb{1}\{|\mathcal{T}_{t-1}(\varepsilon_{<t})| \in S\} \\
&= \sum_{t=0}^{T-1} \mathbb{1}\{|\mathcal{T}_t(\varepsilon_{\leq t})| \in S\} \\
&= \mathbb{1}\{|\mathcal{T}_0| \in S\} + \sum_{t=1}^{T-1} \mathbb{1}\{|\mathcal{T}_t(\varepsilon_{\leq t})| \in S\}
\end{aligned}$$

Moreover, since the path ε is shattered by $\mathcal{F}_S$, it must be the case that for every $t \in [T]$, we have that $|\mathcal{T}_t(\varepsilon_{\leq t})| = |\mathcal{T}_0| + t$. Thus,

$$\sum_{t=1}^{T} \mathbb{1}\{\mathcal{T}_t((\varepsilon_{<t}, -1)) \neq \mathcal{T}_t\big((\varepsilon_{<t}, +1)\big)\} \leq \mathbb{1}\{|\mathcal{T}_0| \in S\} + \sum_{t=1}^{T-1} \mathbb{1}\{|\mathcal{T}_t(\varepsilon_{\leq t})| \in S\}$$
$$= \sum_{t=0}^{T-1} \mathbb{1}\{|\mathcal{T}_0| + t \in S\}$$
$$= |S \cap \{|\mathcal{T}_0|, ..., |\mathcal{T}_0| + T - 1\}|,$$

Taking the supremum of both sides with respect to $\mathcal{T}_0$ completes the proof of the upper bound.

To prove the lower bound, fix $x \in \mathbb{Z}$ and consider the following trajectory tree $\mathcal{T}^x$ of depth T. Let $\mathcal{T}_0^x = x$. For every path $\varepsilon \in \{-1, 1\}^T$ and $t \in [T]$, let

$$\mathcal{T}_t^x(\varepsilon_{\leq t}) = \begin{cases} \varepsilon_t(|\mathcal{T}_0^x| + t), & \text{if} |\mathcal{T}_0^x| + t - 1 \in S. \\ |\mathcal{T}_0^x| + t & \text{if } |\mathcal{T}_0^x| + t - 1 \notin S. \end{cases}$$

Note $|\mathcal{T}_t^x(\varepsilon_{\leq t})| = |\mathcal{T}_0^x| + t$ for all $t \in [T]$. Moreover, for every path $\varepsilon \in \{-1, 1\}^T$, there exists a function $\sigma \in \{-1, 1\}^{\mathbb{N} \cup \{0\}}$ such that $\sigma_{|\mathcal{T}_0^x|} = \varepsilon_1$ and $\sigma_{|\mathcal{T}_t^x(\varepsilon_{\leq t})|} = \varepsilon_{t+1}$ for all $t \in [T-1]$. Thus, the function $f_\sigma \in \mathcal{F}_S$ shatters the path ε and the tree $\mathcal{T}^x$ is shattered by $\mathcal{F}_S$. We claim that $\mathrm{B}(\mathcal{T}^x) = |S \cap \{|\mathcal{T}_0^x|, ..., |\mathcal{T}_0^x| + T - 1\}$. To see this, observe that for every path $\varepsilon \in \{-1, 1\}^T$, we have

$$\sum_{t=1}^{T} \mathbb{1}\{\mathcal{T}_t^x((\varepsilon_{<t}, -1)) \neq \mathcal{T}_t^x\big((\varepsilon_{<t}, +1)\big)\} = \sum_{t=1}^{T} \mathbb{1}\{|\mathcal{T}_0^x| + t - 1 \in S\}$$
$$= |S \cap \{|x_0|, ..., |x_0| + T - 1\}|.$$

Thus, $\mathrm{B}(\mathcal{T}^x) = |S \cap \{|x_0|, ..., |x_0| + T - 1\}|$ and $\mathrm{C}_T(\mathcal{F}) \geq \sup_{x \in \mathbb{Z}} \mathrm{B}(\mathcal{T}^x) = \sup_{x \in \mathbb{Z}} |S \cap \{|x|, ..., |x| + T - 1\}|$. This completes the proof.

B.1.3 Proof of Theorem 4.3.3

The proof of (i) follows from Theorem 4.3.1 and the fact that $\mathrm{C}_T(\mathcal{F}) \leq \mathrm{Bd}(\mathcal{F})$. To prove (ii), observe that by Theorem 4.3.1, it suffices to show that when $\mathrm{Bd}(\mathcal{F}) = \infty$, we have that $\mathrm{C}_T(\mathcal{F}) = \omega(1)$. If $\mathrm{Bd}(\mathcal{F}) = \infty$, then for every $d \in \mathbb{N}$, there exists a shattered trajectory tree $\mathcal{T}$ such that $\mathrm{B}(\mathcal{T}) > d$. Thus, for every $d \in \mathbb{N}$, there exists a depth $d' \in \mathbb{N}$, such that for every $T \geq d'$, there exists a shattered tree $\mathcal{T}$ of depth T with $\mathrm{B}(\mathcal{T}) > d$. In other words,

for every $d \in \mathbb{N}$, there exists a $d' \in \mathbb{N}$ such that for all $T \geq d'$ we have that $\mathrm{C}_T(\mathcal{F}) > d$. By definition of $\omega(\cdot)$, this means that $\mathrm{C}_T(\mathcal{F}) = \omega(1)$ as needed.

B.1.4 Proof of Theorem 4.3.4

Proof. (of (i) in Theorem 4.3.4) To prove (i), pick $S = \{2^t : t \in \mathbb{N} \cup \{0\}\}$ and consider the function class $\mathcal{F}_S$ from Theorem 4.3.2. It is not too hard to see that $\mathrm{C}_T(\mathcal{F}) = \Theta(\log(T))$. In addition, one can verify that every finite subset of S is a shattered set according to Definition 4.2.7. Indeed, consider any finite subset $A \subset S$. For every $i \in [|A|]$, let A_i denote the i'th element of A after sorting A in increasing order. Then, observe that by the construction of $\mathcal{F}_S$, for every sequence $\varepsilon \in \{-1, 1\}^{|A|}$, there exists a function $f_\varepsilon \in \mathcal{F}_S$ such that $f_\varepsilon(A_i) = \varepsilon_i(A_i + 1)$ for every $i \in [|A|]$. By letting $\mathcal{H} = \{f_\varepsilon : \varepsilon \in \{-1, 1\}^{|A|}\}$ in Definition 4.2.7, one can verify that $|\mathcal{H}| = 2^{|A|} < \infty$, and for every $x \in A$ and $h \in \mathcal{H}$, there exists a $g \in \mathcal{H}$ such that $h(x) = -g(x)$ and $h(z) = g(z)$ for all $z \in A \setminus \{x\}$. Thus, $\mathcal{F}_S$ shatters $A \subset S$. Since finite subsets of S can be arbitrary large, we have that $\mathrm{DS}(\mathcal{F}) = \infty$. ■

Proof. (of (ii) in in Theorem 4.3.4) To prove (ii), let $\mathcal{X} = \mathbb{N} \cup \{\star\}$ and consider the following evolution function class. For every $\sigma \in \{-1, 1\}^{\mathbb{N}}$, define a sequence $a_\sigma : \mathbb{N} \to \mathbb{N}$ recursively such that $a_\sigma(1) = 1$ and $a_\sigma(n) = 2a_\sigma(n-1) + \frac{1+\sigma_{n-1}}{2}$ for $n \geq 2$. Equivalently, we can define the sequence a_σ explicitly by $a_\sigma(1) = 1$ and $a_\sigma(n) = 2^{n-1} + \sum_{i=1}^{n-1} \left(\frac{1+\sigma_i}{2}\right) 2^{n-(i+1)}$ for $n \geq 2$. Let $S_n := \bigcup_\sigma \{a_\sigma(n)\}$ and note that $S_n = \{2^{n-1}, ..., 2^n - 1\}$ for every $n \geq 1$ and $S_n \cap S_r = \emptyset$ for all $n \neq r$. We establish some important properties about these sequences.

(1) The sequence a_σ is strictly monotonically increasing in its input, and hence invertible. Accordingly, given a sequence a_σ and an element $x \in \mathrm{im}(a_\sigma)$, let $a_\sigma^{-1}(x)$ denote the index $n \in \mathbb{N}$ such that $a_\sigma(n) = x$.

(2) For every σ_1, σ_2, if $a_{\sigma_1}(n) = a_{\sigma_2}(r)$, then $n = r$ since $S_n \cap S_r = \emptyset$ for all $n \neq r$.

(3) The value of $a_\sigma(n)$ depends only on the prefix $(\sigma_1, ..., \sigma_{n-1})$. Hence, two strings σ_1, σ_2 that share the same prefix up to and including index d will have the property that $a_{\sigma_1}(n) = a_{\sigma_2}(n)$ for all $n \leq d + 1$.

(4) If $a_{\sigma_1}(d) = a_{\sigma_2}(d)$, then $a_{\sigma_1}(i) = a_{\sigma_2}(i)$ for all $i \leq d$. This follows by induction. For the base case, consider $i = d - 1$. If $a_{\sigma_1}(i) \neq a_{\sigma_2}(i)$, then $a_{\sigma_1}(d) = 2a_{\sigma_1}(i) + \frac{1+\sigma_{1,i}}{2} \neq 2a_{\sigma_2}(i) + \frac{1+\sigma_{2,i}}{2} = a_{\sigma_2}(d)$ for any value of $\sigma_{1,i}$ and $\sigma_{2,i}$. For the induction step, suppose $a_{\sigma_1}(i) = a_{\sigma_2}(i)$ for some $2 \leq i < d$. Then, if $a_{\sigma_1}(i-1) \neq a_{\sigma_2}(i-1)$, we have that $a_{\sigma_1}(i) = 2a_{\sigma_1}(i-1) + \frac{1+\sigma_{1,i-1}}{2} \neq 2a_{\sigma_2}(i-1) + \frac{1+\sigma_{2,i-1}}{2} = a_{\sigma_2}(i)$ for any value of $\sigma_{1,i-1}$ and $\sigma_{2,i-1}$.

We now construct a function class. For every $\sigma \in \{-1,1\}^{\mathbb{N}}$, define the evolution function

$$f_\sigma(x) = \star \mathbb{1}\{x \notin \text{im}(a_\sigma)\} + a_\sigma\Big(a_\sigma^{-1}(x)+1\Big)\mathbb{1}\{x \in \text{im}(a_\sigma)\}.$$

At a high-level, the evolution function f_σ maps every state in $\mathcal{X} \setminus \text{im}(a_\sigma)$ to $\star$ and every state in $\text{im}(a_\sigma)$ to the next element in the sequence corresponding to a_σ.

Consider the function class $\mathcal{F} = \{f_\sigma : \sigma \in \{-1,1\}^{\mathbb{N}}\}$. We now claim that $\text{C}_T(\mathcal{F}) = T$. To see this, fix a depth $d \in \mathbb{N}$, and consider the following trajectory tree $\mathcal{T}$ of depth d. Let the root node be labeled by 1, that is $\mathcal{T}_0 = 1$. For all $t \in [d]$ and $\varepsilon \in \{-1,1\}^t$, let $\mathcal{T}_t(\varepsilon) = a_{\tilde{\varepsilon}}(t+1)$ where $\tilde{\varepsilon}$ denotes an arbitrary extension of ε over $\mathbb{N}$. Note that the completion can be arbitrary because the value of $a_\sigma(t+1)$ for any $\sigma \in \{-1,1\}^{\mathbb{N}}$ depends only on the prefix $(\sigma_1, ..., \sigma_t)$. One can verify that such a tree $\mathcal{T}$ is a complete binary tree of depth d where the root node is labeled with 1 and the internal nodes on depth $i \geq 1$ are labeled from left to right by $2^i, 2^i+1, ..., 2^{i+1}-1$. Thus, it is clear that $\text{B}(\mathcal{T}) = d$ since for every internal node including the root, its two children are labeled by differing states. In addition, observe that for every path $\varepsilon \in \{-1,1\}^d$ down $\mathcal{T}$, the function $f_{\tilde{\varepsilon}} \in \mathcal{F}$ shatters the associated sequence of states, where $\tilde{\varepsilon}$ again is an arbitrary completion of ε over $\mathbb{N}$. Indeed, fix a $\varepsilon \in \{-1,1\}^d$, a completion $\tilde{\varepsilon} \in \{-1,1\}^{\mathbb{N}}$, and consider any $t \in [d]$. Then, by definition of $\mathcal{T}$, we have that $\mathcal{T}_{t-1}(\varepsilon_{<t}) = a_{\tilde{\varepsilon}}(t)$ and $\mathcal{T}_t(\varepsilon_{\leq t}) = a_{\tilde{\varepsilon}}(t+1)$. Consider the function $f_{\tilde{\varepsilon}} \in \mathcal{F}$. By definition of $f_{\tilde{\varepsilon}}$, we have that $f_{\tilde{\varepsilon}}(a_{\tilde{\varepsilon}}(t)) = a_{\tilde{\varepsilon}}(t+1)$, which implies that $f_{\tilde{\varepsilon}}(\mathcal{T}_{t-1}(\varepsilon_{<t})) = \mathcal{T}_t(\varepsilon_{\leq t})$. Since $t \in [d]$ was arbitrary, this is true for every $t \in [d]$, and thus $f_{\tilde{\varepsilon}}$ shatters the path ε down $\mathcal{T}$ as claimed. Since $\varepsilon \in \{-1,1\}^d$ was also arbitrary, we have that the entire tree $\mathcal{T}$ is shattered by $\mathcal{F}$. Finally, since $d \in \mathbb{N}$ was arbitrary, this is true for arbitrarily large depths. Thus, $\text{C}_T(\mathcal{F}) = T$.

We now show that $\text{DS}(\mathcal{F}) = 1$ by proving that $\mathcal{F}$ cannot DS-shatter any two instances $x_1, x_2 \in \mathcal{X}$. Our proof will be in cases. First, observe that if either $x_1 = \star$ or $x_2 = \star$, then (x_1, x_2) cannot be shattered since all functions in $\mathcal{F}$ will output $\star$ on either x_1 or x_2. Thus, without loss of generality, suppose both $x_1, x_2 \in \mathbb{N}$ and $x_1 < x_2$. Consider any finite subset $\mathcal{H} \subset \mathcal{F}$ and suppose there exists a function $h \in \mathcal{H}$ such that $h(x_2) \neq \star$. Then, in order to shatter (x_1, x_2), there must exist a function $g \in \mathcal{H}$ such that $h(x_1) \neq g(x_1)$ but $h(x_2) = g(x_2)$. However, if $h(x_2) = g(x_2) \neq \star$, then it must be the case that $h(x_1) = g(x_1)$. To see why, fix an instance $x \in \mathbb{N}$, and suppose $f_{\sigma_1}(x) = f_{\sigma_2}(x) \neq \star$. Then, by properties (2) and (4) above, it must be the case that $a_{\sigma_1}^{-1}(x) = a_{\sigma_2}^{-1}(x) = c$ and $a_{\sigma_1}(i) = a_{\sigma_2}(i)$ for all $i \leq c$. Thus, if $x_1 < x_2$ and $f_{\sigma_1}(x_2) = f_{\sigma_2}(x_2) \neq \star$, we must have that $f_{\sigma_1}(x_1) = f_{\sigma_2}(x_1)$ because either $x_1 \notin \text{im}(a_{\sigma_1}) \cup \text{im}(a_{\sigma_2})$ or $a_{\sigma_1}^{-1}(x_1) < a_{\sigma_1}^{-1}(x_2) = a_{\sigma_2}^{-1}(x_2)$. Thus, if $\mathcal{H}$ were to satisfy the property in Definition 4.2.7, there cannot exist a hypothesis $h \in \mathcal{H}$ such that $h(x_2) \neq \star$. However, if for every $h \in \mathcal{H}$, we have that $h(x_2) = \star$, then for every $h \in \mathcal{H}$,

there cannot exist a $g \in \mathcal{H}$ such that $h(x_1) = g(x_1)$ and $h(x_2) \neq g(x_2)$. Therefore, the two points (x_1, x_2) cannot be shattered. Since (x_1, x_2) and $\mathcal{H} \subset \mathcal{F}$ were arbitrary, this is true for all such points, implying that $\mathrm{DS}(\mathcal{F}) \leq 1$. Since $|\mathcal{F}| \geq 2$, we have also that $\mathrm{DS}(\mathcal{F}) \geq 1$, completing the proof that $\mathrm{DS}(\mathcal{F}) = 1$. ■

Proof. (of (iii) in Theorem 4.3.4) To prove part (iii) of Theorem 4.3.4, we reduce learning dynamical systems to online multiclass classification. Let $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ and $\mathcal{B}$ be any (potentially randomized) online learner for $\mathcal{F}$ for online multiclass classification with expected regret bound R. We will construct a learner $\mathcal{A}$ that uses $\mathcal{B}$ as a subroutine such that $\mathrm{M}_{\mathcal{A}}(T, \mathcal{F}) \leq R$.

To that end, let $(x_0, x_1, ..., x_T)$ be the *realizable* stream to be observed by the learner and consider the following learning algorithm $\mathcal{A}$ which makes the same predictions as $\mathcal{B}$ while simulating the stream of labeled instance $(x_0, x_1), ..., (x_{T-1}, x_T)$ to $\mathcal{B}$.

Algorithm 17 Learning algorithm $\mathcal{A}$.

Require: Online multiclass learner $\mathcal{B}$, initial state x_0.

1: **for** $t = 1, ..., T$ **do**
2: Pass x_{t-1} to $\mathcal{B}$ and receive prediction $\hat{z}_t$.
3: Predict $\hat{x}_t = \hat{z}_t$.
4: Receive next state x_t and update $\mathcal{B}$ using labeled instance (x_{t-1}, x_t).
5: **end for**

Then, observe that

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\hat{x}_t \neq x_t\}\right] = \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\hat{z}_t \neq x_t\}\right] \overset{\text{(i)}}{\leq} \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} \mathbb{1}\{f(x_{t-1}) \neq x_t\} + R \overset{\text{(ii)}}{=} R \tag{B.2}$$

where (i) follows from the expected regret guarantee of $\mathcal{B}$ and (ii) follows from the fact that the stream of states is realizable. Thus, $\inf_{\mathcal{A}} \mathrm{M}_{\mathcal{A}}(T, \mathcal{F}) \leq R$. Since $\mathcal{B}$ is always guaranteed to observe a realizable sequence of labeled instances, picking the SOA [Littlestone, 1987, Daniely et al., 2011] gives that $R = \mathrm{L}(\mathcal{F})$. Since the SOA is deterministic, this choice of $\mathcal{B}$ implies that $\mathcal{A}$ is deterministic. Part (i) then follows from that the fact that for any deterministic learner $\mathcal{A}$ we have $\mathrm{M}_{\mathcal{A}}(T, \mathcal{F}) \geq \mathrm{C}_T(\mathcal{F})$. ■

Proof. (of (iv) in Theorem 4.3.4) To prove part (iv), let $\mathcal{X} = [0, 1]$ and consider the class of thresholds $\mathcal{F} = \{x \mapsto \mathbb{1}\{x \geq a\} : a \in (0, 1)\}$. It is well known that $\mathrm{L}(\mathcal{F}) = \infty$. On the other hand, note that $d(\mathcal{F}) = 1$ since $\mathcal{F}(1) = \{1\}$, $\mathcal{F}(0) = \{0\}$, and $\mathcal{F}(x) = \{0, 1\}$ for all $x \in (0, 1)$. ■

B.2 Proofs for Linear Systems

Proof. (of (i) in Theorem 4.3.5) We first prove the lower bound. It suffices to show that

$$\inf_{\text{Deterministic } \mathcal{A}} \mathrm{M}_{\mathcal{A}}(T, \mathcal{F}) \geq r + 1.$$

Let $\mathcal{A}$ be any deterministic algorithm and $\{e_0, e_1, \ldots, e_r\}$ be arbitrary $r+1$ standard basis on $\mathbb{R}^n$. This set exists because $n \geq r+1$. Consider a stream such that $x_0 = e_0$, $x_t = \{-e_t, e_t\} \backslash \mathcal{A}(x_{<t})$ for all $t \in [r]$, and $x_{r+1} = \{-e_1, e_1\} \backslash \mathcal{A}(x_{<r+1})$. Here, $\forall t \in [r]$, we choose x_t to be an element of the set $\{-e_t, e_t\}$ other than $\mathcal{A}$'s prediction on round t. Similarly, x_{r+1} is chosen among $\{-e_1, e_1\}$. We can define a stream using $\mathcal{A}$ because such a stream can be simulated before the game starts as $\mathcal{A}$ is deterministic. Let $\{\sigma_t\}_{t=0}^{r+1} \in \{-1, 1\}^{r+2}$ such that $(x_0, \ldots, x_{r+1}) = (\sigma_0 e_0, \sigma_1 e_1, \sigma_2 e_2, \ldots, \sigma_r e_r, \sigma_{r+1} e_1)$. Consider the integer-valued matrix $W^\star = \sum_{t=1}^{r} \sigma_{t-1}\sigma_t \, e_t \otimes e_{t-1} + \sigma_r \sigma_{r+1} \, e_1 \otimes e_r$. For $r+2 \leq t \leq T$, one can define the stream to be $x_t = W^\star x_{t-1}$. Note that the image$(W^\star) \subseteq$ span$(\{e_1, \ldots, e_r\})$. Thus, rank$(W^\star) \leq r$ and $W^\star \in \mathcal{F}$. Finally, since $W^\star$ satisfies $W^\star(\sigma_{t-1} e_{t-1}) = \sigma_t \, e_t$ for $t \in [r]$, $W^\star(\sigma_r e_r) = \sigma_{r+1} e_1$, and $W^\star x_{t-1} = x_t$ for $t \geq r+2$, the stream is realizable. By the definition of the stream, $\mathcal{A}$ makes mistakes on all rounds $t \in \{1, \ldots, r+1\}$. Thus, we have $\mathrm{M}_{\mathcal{A}}(T, \mathcal{F}) \geq r+1$.

To prove the upper bound, fix $T \in \mathbb{N}$ and suppose that $T \geq r+1$ (otherwise $\mathrm{C}_T(\mathcal{F}) \leq T \leq r+1$). Consider a trajectory tree $\mathcal{T}$ of depth T shattered by $\mathcal{F}$. We first show that, for any path $\sigma \in \{-1, 1\}^T$ down $\mathcal{T}$ and $t \in [T-1]$, if $\mathcal{T}_{t+1}((\sigma_{\leq t}, -1)) \neq \mathcal{T}_{t+1}((\sigma_{\leq t}, +1))$, then $\mathcal{T}_t(\sigma_{\leq t}) \notin$ span$(\{\mathcal{T}_0, \mathcal{T}_1(\sigma_{\leq 1}), ..., \mathcal{T}_{t-1}(\sigma_{\leq t-1}\})$. To prove the contraposition of this statement, suppose $\mathcal{T}_t(\sigma_{\leq t}) \in$ span$(\{\mathcal{T}_0, \mathcal{T}_1(\sigma_{\leq 1}), ..., \mathcal{T}_{t-1}(\sigma_{\leq t-1}\})$. Then, there exists constants $a_0, ..., a_{t-1} \in \mathbb{Z}$ such that $a_0\mathcal{T}_0 + ..., +a_{t-1}\mathcal{T}_{t-1}(\sigma_{\leq t-1}) = \mathcal{T}_t(\sigma_{\leq t})$. Thus, every matrix W consistent with the sequence $\mathcal{T}_0, \mathcal{T}_1(\sigma_{\leq 1}), ..., \mathcal{T}_{t-1}(\sigma_{\leq t-1})$ satisfies $W\mathcal{T}_t(\sigma_{\leq t}) = W(a_0\mathcal{T}_0 + ..., +a_{t-1}\mathcal{T}_{t-1}(\sigma_{\leq t-1})) = a_0\mathcal{T}_1(\sigma_{\leq 1}) + ..., +a_{t-1}\mathcal{T}_t(\sigma_{\leq t})$. This implies that the path $\sigma_{\leq t+1}$ is shattered only if $\mathcal{T}_{t+1}((\sigma_{\leq t}, -1)) = \mathcal{T}_{t+1}((\sigma_{\leq t}, +1))$.

Next, we show that the branching factor of every path in $\mathcal{T}$ can be at most $r+1$. Suppose, for the sake of contradiction, there exists a path $\sigma \in \{-1, 1\}^T$ whose branching factor is $k > r+1$. We are guaranteed an increasing sequence of time points $t_1, t_2, \ldots, t_k$ in $\{0, \ldots, T-1\}$ such that $\mathcal{T}_{t_i+1}((\sigma_{\leq t_i}, -1)) \neq \mathcal{T}_{t_i+1}((\sigma_{\leq t_i}, +1))$ for all $i \in [k]$. Since $t_2 \geq 1$, by our argument above, we are guaranteed that $\mathcal{T}_{t_i}(\sigma_{\leq t_i}) \notin$ span$(\{\mathcal{T}_0, \ldots, \mathcal{T}_{t_i-1}(\sigma_{\leq t_i-1})\})$ for all $i \in \{2, \ldots, k\}$. This further implies that the set $S := \{\mathcal{T}_{t_2}(\sigma_{\leq t_2}), \ldots, \mathcal{T}_{t_k}(\sigma_{\leq t_k})\}$ is a linearly independent set. Note that $|S| = k - 1 > r$. Let W_σ be the matrix such that the associated function $f_\sigma \in \mathcal{F}$ shatters this path σ. By the definition of the tree, we must have that $S \subseteq$ image(W_σ). Since rank$(W_\sigma) \leq r$, the set image(W_σ) must be a subset of a vectorspace of dimension $\leq r$. However, this contradicts the fact that S contains at least $r+1$ linearly

independent vectors in $\mathbb{R}^n$. Thus, the branching factor of every path in $\mathcal{T}$ is at most $r+1$ and $\mathrm{B}(\mathcal{T}) \leq r+1$. Since $\mathcal{T}$ is arbitrary, we have $\mathrm{C}_T(\mathcal{F}) \leq r+1$. ■

Proof. (of (ii) in Theorem 4.3.5) We claim that it is without loss of generality to consider the function class $\mathcal{G} = \{x \mapsto Mx \,(\mathrm{mod}\ 2) : M \in \{0,1\}^{n\times n}\} \subset \mathcal{F}$. Indeed, for every $f \in \mathcal{F}$, there exists a $g \in \mathcal{G}$ such that $f(x) = g(x)$ for all $x \in \mathcal{X}$. To see this, let $W \in \mathbb{Z}^{n\times n}$ be such that $f(x) = Wx(\mathrm{mod}\ 2)$ and fix some $x \in \{0,1\}^n$. Then, observe that $f(x) = Wx(\mathrm{mod}\ 2) = (W(\mathrm{mod}\ 2)x)(\mathrm{mod}\ 2) = Mx(\mathrm{mod}\ 2)$ where $M = W(\mathrm{mod}\ 2) \in \{0,1\}^{n\times n}$. Thus the function $g \in \mathcal{G}$ parameterized by M matches f everywhere on $\mathcal{X}$. We now prove that $\mathrm{C}_T(\mathcal{G}) = n$. In the lower bound, we will use the fact that for a system of n linearly independent equations with n free variables defined over the integers modulo 2, there exists a unique solution.

Starting with the upper bound, fix $T \in \mathbb{N}$ and suppose that $T \geq n$ (as otherwise $\mathrm{C}_T(\mathcal{G}) \leq T \leq n$). Consider a trajectory tree $\mathcal{T}$ of depth T shattered by $\mathcal{G}$. Let $\sigma \in \{-1,1\}^T$ denote an arbitrary path down $\mathcal{T}$. We first claim that for $t \geq 1$ if $\mathcal{T}_t(\sigma_{\leq t})$ is linearly dependent modulo 2 on the preceding sequence of states $\mathcal{T}_0, \mathcal{T}_1(\sigma_{\leq 1}), ..., \mathcal{T}_{t-1}(\sigma_{<t})$, then $\mathcal{T}_{t+1}((\sigma_{\leq t}, -1)) = \mathcal{T}_{t+1}((\sigma_{\leq t}, +1))$. To see this, suppose that $\mathcal{T}_t(\sigma_{\leq t})$ is linearly dependent modulo 2 on the preceding sequence of states $\mathcal{T}_0, \mathcal{T}_1(\sigma_{\leq 1}), ..., \mathcal{T}_{t-1}(\sigma_{<t})$. Then, there exists constants $a_0, ..., a_{t-1} \in \{0,1\}$ such that $a_0\mathcal{T}_0 + ... + a_{t-1}\mathcal{T}_{t-1}(\sigma_{<t})(\mathrm{mod}\ 2) = \mathcal{T}_t(\sigma_{\leq t})$. Thus, every function $g \in \mathcal{G}$ consistent with the sequence $\mathcal{T}_0, \mathcal{T}_1(\sigma_{\leq 1}), ..., \mathcal{T}_{t-1}(\sigma_{<t})$ outputs $a_0\mathcal{T}_1(\sigma_{\leq 1}) + ... + a_{t-1}\mathcal{T}_t(\sigma_{\leq t})(\mathrm{mod}\ 2)$ on $\mathcal{T}_t(\sigma_{\leq t})$, implying that the paths $(\sigma_{\leq t}, -1)$ and $(\sigma_{\leq t}, +1)$ are shattered only if $\mathcal{T}_{t+1}((\sigma_{\leq t}, -1)) = \mathcal{T}_{t+1}((\sigma_{\leq t}, +1)) = a_0\mathcal{T}_1(\sigma_{\leq 1}) + ... + a_{t-1}\mathcal{T}_t(\sigma_{\leq t})(\mathrm{mod}\ 2)$. As a consequence, for $t \geq 1$, if $\mathcal{T}_{t+1}((\sigma_{\leq t}, -1)) \neq \mathcal{T}_{t+1}((\sigma_{\leq t}, +1))$, then $\mathcal{T}_t(\sigma_{\leq t})$ is linearly independent modulo 2 of its preceding states. Next, we claim that there can be at most $n-1$ timepoints $t_1, ..., t_{n-1} \in [T-1]$ such that $\mathcal{T}_{t_i}(\sigma_{\leq t_i})$ is linearly independent of its preceeding sequence of states $\mathcal{T}_0, \mathcal{T}_1(\sigma_{\leq 1}), ..., \mathcal{T}_{t_i-1}(\sigma_{<t_i})$. Indeed, suppose for sake of contradiction there exists n timepoints $t_1, ..., t_n \in [T-1]$ such that $\mathcal{T}_{t_i}(\sigma_{\leq t_i})$ is linearly independent of its preceeding states. Then, note that the following set of $n+1$ states $\{\mathcal{T}_0, \mathcal{T}_{t_1}(\sigma_{\leq t_1}), ..., \mathcal{T}_{t_n}(\sigma_{\leq t_n})\}$ are linearly independent modulo 2. This is a contradiction since $\mathcal{X}$ is n-dimensional. Combining the two claims, we get that

$$\begin{aligned}
\sum_{t=1}^{T} \mathbb{1}\{\mathcal{T}_t((\sigma_{<t}, -1)) \neq \mathcal{T}_t\big((\sigma_{<t}, +1)\big)\} &\leq 1 + \sum_{t=2}^{T} \mathbb{1}\{\mathcal{T}_t((\sigma_{<t}, -1)) \neq \mathcal{T}_t\big((\sigma_{<t}, +1)\big)\} \\
&\leq 1 + \sum_{t=1}^{T-1} \mathbb{1}\{\mathcal{T}_t(\sigma_{\leq t}) \text{ linearly indep. of prev. states}\} \\
&\leq 1 + n - 1 = n.
\end{aligned}$$

Since $\sigma \in \{-1,1\}^T$ is arbitrary, we get that $\mathrm{B}(\mathcal{T}) \leq n$. Finally, since $\mathcal{T}$ is arbitrary, we have that $\mathrm{C}_T(\mathcal{F}) \leq n$ which completes the proof of the upper bound.

To prove the lower bound, let $\mathcal{A}$ be any deterministic algorithm. Let $\mathcal{E} = \{e_0, ..., e_{n-1}\} \subset \mathcal{X}$ be the standard basis over $\mathbb{R}^n$. Consider the stream where we pick $x_0 = e_0$, for all $t \in [n-2]$, we pick $x_t \in \mathcal{E} \setminus \{x_0, ..., x_{t-1}, \mathcal{A}(x_{<t})\}$. Let $e = \mathcal{E} \setminus \{x_0, ..., x_{n-2}\}$ be the remaining basis. Pick $x_{n-1} \in \{e, e + e_0\} \setminus \{\mathcal{A}(x_{<n-1})\}$. Finally, pick $x_n \neq \mathcal{A}(x_{<n})$. We can define a stream using $\mathcal{A}$ because it is deterministic and thus can be simulated before the game begins. Next, we show that $x_0, ..., x_n$ is a realizable stream. This follows from the fact that $x_0, ..., x_{n-1}$ are linearly independent by definition, and thus there exists a function $g \in \mathcal{G}$ such that $g(x_{t-1}) = x_t$ for all $t \in [n]$. Moreover, by definition of the stream, $\mathcal{A}$ makes a mistake in every round. Thus, $\mathrm{M}_{\mathcal{A}}(T, \mathcal{F}) \geq n$. Using the fact that $\mathrm{M}_{\mathcal{A}}(T, \mathcal{F}) \leq \mathrm{C}_T(\mathcal{F})$ completes the proof. ■

Proof. (of (iii) in Theorem 4.3.5) Starting with the lower bound, it suffices to show that

$$\inf_{\text{Deterministic } \mathcal{A}} \mathrm{M}_{\mathcal{A}}(T, \mathcal{F}) \geq n.$$

Let $\mathcal{A}$ be any deterministic algorithm and $\{e_1, \ldots, e_n\}$ be the standard basis on $\mathbb{R}^n$. Consider a stream such that $x_0 = e_1 + e_2 + \ldots + e_n$, then $x_1 = x_0 - e_{i_1}$ for some $i_1 \in [n]$ such that $x_1 \neq \mathcal{A}(x_0)$, and for all $t \in \{2, \ldots, n-1\}$,

$$x_t = x_0 - (e_{i_1} + \ldots + e_{i_t}) \text{ such that } e_{i_t} \notin \{e_{i_1}, \ldots, e_{i_{t-1}}\} \text{ and } x_t \neq \mathcal{A}(x_{<t}).$$

Define $e_{i_n} = x_0 - (e_{i_1} + \ldots + e_{i_{n-1}})$. Note that $x_{n-1} = e_{i_n}$. Finally, we choose $x_n \in \{e_{i_n}, \mathbf{0}\}$ such that $x_n \notin \mathcal{A}(x_{<n-1})$. Here, $\forall t \in [n-2]$, we choose x_t by subtracting a basis e_{i_t}, which has not been subtracted before, from x_{t-1} such that x_t is other than what $\mathcal{A}$ would have predicted on round t. Finally, for x_n, we either choose it to be equal to $x_{n-1} = e_{i_n}$ again or $\mathbf{0}$ while ensuring that $x_n \neq \mathcal{A}(x_{<n-1})$. We can define a stream using $\mathcal{A}$ because such a stream can be simulated before the game starts as $\mathcal{A}$ is deterministic.

Next, we show that $(x_0, \ldots, x_n)$ is a realizable stream. Indeed, the boolean matrix

$$W^\star = \sum_{k=2}^{n} e_{i_k} \otimes (e_{i_1} + \ldots + e_{i_{k-1}}) + x_n \otimes e_{i_n}.$$

satisfies $\mathbb{1}\{W^\star x_{t-1} > 0\} = x_t$ for all $t \in [n]$. For $t = 1$, we have

$$\begin{aligned} W^\star x_0 &= \left(\sum_{k=2}^{n} e_{i_k} \left\langle e_{i_1} + \ldots + e_{i_{k-1}}, x_0 \right\rangle \right) + x_n \\ &= \sum_{k=2}^{n} (k-1) e_{i_k} + x_n \\ &= (k-1)(x_0 - e_{i_1}) + x_n \end{aligned}$$

Thus, $\mathbb{1}\{W^\star x_0 > 0\} = x_0 - e_{i_1} = x_1$. For $t \in \{2, \ldots, n-1\}$, we have

$$\begin{aligned} W^\star x_{t-1} &= \left(\sum_{k=2}^{n} e_{i_k} \left\langle e_{i_1} + \ldots + e_{i_{k-1}}, x_0 - \left(e_{i_1} + \ldots + e_{i_{t-1}} \right) \right\rangle \right) + x_n \\ &= \sum_{k=2}^{n} e_{i_k} \left\langle e_{i_1} + \ldots + e_{i_{k-1}}, e_{i_t} + \ldots + e_{i_n} \right\rangle + x_n \\ &= \sum_{k=t+1}^{n} (k-t)\, e_{i_k} + x_n. \end{aligned}$$

So, we obtain $\mathbb{1}\{W^\star x_{t-1} > 0\} = x_0 - (e_{i_1} + \ldots + e_{i_t}) > 0 = x_t$. Finally, since $x_{n-1} = e_{i_n}$ and $W^\star e_{i_n} = x_n$, we have $\mathbb{1}\{W^\star x_{n-1} > 0\} = x_n$. By the definition of the stream, $\mathcal{A}$ makes mistakes in every round on this stream. Thus, $\mathrm{M}_{\mathcal{A}}(T, \mathcal{F}) \geq n$.

As for the upper bound, since $|\{0,1\}^{n \times n}| = 2^{n^2}$, we have $\mathrm{L}(\mathcal{F}) \leq \log(2^{n^2}) = n^2$. Thus, by Theorem 4.3.4 (iii), we must have $\mathrm{C}_T(\mathcal{F}) \leq n^2$. ■

B.3 Proofs for Agnostic Learnability

B.3.1 Proof of Theorem 4.4.1

Proof. (of upper bound in Theorem 4.4.1) Let $(x_0, x_1, ..., x_T)$ be the stream to be observed by the learner. Given a multiclass classification learner $\mathcal{B}$ for $\mathcal{F}$, consider the algorithm $\mathcal{A}$ as defined in Algorithm 17. Here, $\mathcal{A}$ makes the same predictions as $\mathcal{B}$ while simulating the stream of labeled instance $(x_0, x_1), ..., (x_{T-1}, x_T)$ to $\mathcal{B}$. Using the bound (i) in Equation (B.2), we obtain

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\hat{x}_t \neq x_t\} \right] \leq \inf_{f \in \mathcal{F}} \sum_{t=1}^{T} \mathbb{1}\{f(x_{t-1}) \neq x_t\} + R,$$

where expectation is taken with respect to the randomness of $\mathcal{B}$ and R is the expected regret of the $\mathcal{B}$. This shows that $\mathrm{MR}_{\mathcal{A}}(T, \mathcal{F}) \leq R$. Taking $\mathcal{B}$ to be $\mathbb{A}_{\mathrm{AG}}$ defined in [Hanneke et al., 2023, Section 3], Theorem 4 in [Hanneke et al., 2023] implies that $R \leq \mathrm{L}(\mathcal{F}) + \sqrt{T\ \mathrm{L}(\mathcal{F}) \log T}$, matching the claimed upper bound in Theorem 4.4.1. ■

Proof. (of lower bound of $\frac{\mathrm{L}(\mathcal{F})}{18}$ in Theorem 4.4.1) Define $d := \mathrm{L}(\mathcal{F})$ and let $\mathcal{T}$ be the Littlestone tree of depth d shattered by $\mathcal{F}$. Let $(\mathcal{T}_0, \ldots, \mathcal{T}_{d-1})$ be the sequence of node-labeling functions and $(Y_1, \ldots, Y_d)$ be the sequence of edge labeling functions of the shattered tree $\mathcal{T}$. To construct a stream, take $r = \left\lfloor \frac{d+2}{3} \right\rfloor$ and define a random sequence $\{n_i\}_{i=1}^{r}$ such that

$$n_1 = 1, \quad n_{i+1} \sim \mathrm{Unif}(\{3i - 1, 3i, 3i + 1\}) \text{ for all } i \in [r - 1].$$

Note that $n_r \leq 3(r-1) + 1 = 3r - 2 \leq d$. Next, pick a random path $(\sigma_1, \ldots, \sigma_d)$ down the tree such that $\sigma_i \sim \mathrm{Unif}\{-1, 1\}$. Let $x_0 = \mathcal{T}_0$ be the initial state and consider the stream

$$Y_{n_1}(\sigma_{\leq n_1}), \mathcal{T}_{n_2-1}(\sigma_{<n_2}), Y_{n_2}(\sigma_{\leq n_2}), \ldots, \mathcal{T}_{n_r-1}(\sigma_{<n_r}), Y_{n_r}(\sigma_{\leq n_r}).$$

For any randomized algorithm $\mathcal{A}$, let $\mathcal{A}_t$ denote its prediction on round t. Then, its expected cumulative loss on this stream is

$$\mathbb{E}\left[\sum_{t=1}^{r} \mathbb{1}\{\mathcal{A}_t \neq Y_{n_t}(\sigma_{\leq n_t})\} + \sum_{t=2}^{r} \mathbb{1}\{\mathcal{A}_t \neq \mathcal{T}_{n_t-1}(\sigma_{<n_t})\}\right],$$

where the expectation is taken with respect to $\mathcal{A}$, σ, and n_t's. Note that

$$\mathbb{E}[\mathbb{1}\{\mathcal{A}_t \neq Y_{n_t}(\sigma_{\leq n_t})\}] \geq \frac{1}{2}.$$

where the inequality holds because conditioned on the history up to time point $t-1$, the state $Y_{n_t}(\sigma_{\leq n_t})$ is chosen uniformly at random between $Y_{n_t}((\sigma_{<n_t}, -1))$ and $Y_{n_t}((\sigma_{<n_t}, +1))$. So, the algorithm cannot do better than random guessing. Similarly, given a path σ, the state $\mathcal{T}_{n_t-1}(\sigma_{<n_t})$ is selected uniformly from the set

$$\left\{\mathcal{T}_{m-1}(\sigma_{<m}) : m = 3(t-1) - 1, 3(t-1), 3(t-1) + 1\right\}.$$

Since each node along a path must have distinct elements, the aforementioned set must contain 3 elements. Thus, we have

$$\mathbb{E}[\mathbb{1}\{\mathcal{A}_t \neq \mathcal{T}_{n_t-1}(\sigma_{<n_t})\}] \geq \frac{2}{3}.$$

Overall, the total expected cumulative loss of $\mathcal{A}$ is

$$\geq \sum_{t=1}^{r} \frac{1}{2} + \sum_{t=2}^{r} \frac{2}{3} \geq \frac{r}{2} + \frac{2(r-1)}{3} = \frac{7r-4}{6}.$$

To upper bound the loss of the best-fixed competitor, let $f_\sigma \in \mathcal{F}$ denote the function that shatters the path σ in the tree $\mathcal{T}$. Then, by definition of shattering, we have $f_\sigma(\mathcal{T}_{n_t-1}(\sigma_{<n_t})) = Y_{n_t}(\sigma_{\leq n_t})$. Thus, the cumulative loss of f_σ is

$$\mathbb{E}\left[\sum_{t=1}^{r} \mathbb{1}\{f_\sigma(\mathcal{T}_{n_t-1}(\sigma_{<n_t})) \neq Y_{n_t}(\sigma_{\leq n_t})\} + \sum_{t=2}^{r} \mathbb{1}\{f_\sigma(Y_{n_{t-1}}(\sigma_{\leq n_{t-1}})) \neq \mathcal{T}_{n_t-1}(\sigma_{<n_t})\}\right] \leq 0 + \sum_{t=2}^{r} 1 = r-1.$$

Hence, combining everything and plugging the value of r, the regret of $\mathcal{A}$ is

$$\mathrm{MR}_{\mathcal{A}}(T, \mathcal{F}) \geq \frac{7r-4}{6} - (r-1) = \frac{7r-4-6r+6}{6} = \frac{r+2}{6} \geq \frac{(d+2)/3-1+2}{6} \geq \frac{d}{18}.$$

This completes our proof. ■

Proof. (of lower bound of $\frac{\sqrt{T}}{16\sqrt{3}}$ in Theorem 4.4.1) Let f_{-1} and f_{+1} be any two distinct functions in $\mathcal{F}$ and $\bar{x}_0 \in \mathcal{X}$ be the point where they differ. That is, $f_{-1}(\bar{x}_0) \neq f_{+1}(\bar{x}_0)$. Define $S \subset \mathcal{X}$ such that $f_{-1}(x) = f_{+1}(x)$ for all $x \in S$. Moreover, define $S_0 = \{x \in S \mid f_{-1}(x) = f_{+1}(x) = \bar{x}_0\}$. Let $(\sigma_1, \ldots, \sigma_T)$ be a sequence such that $\sigma_t \sim \mathrm{Uniform}\{-1, 1\}$. Consider a random stream with initial state $x_0 = \bar{x}_0$ and for all $t \in [T]$,

$$x_t = \bar{x}_0 \mathbb{1}\{x_{t-1} \in S_0\} + \mathrm{Uniform}(\{\bar{x}_0, f_{\sigma_t}(x_{t-1})\})\, \mathbb{1}\{x_{t-1} \in S \backslash S_0\} + f_{\sigma_t}(x_{t-1})\, \mathbb{1}\{x_{t-1} \notin S\}.$$

For any algorithm $\mathcal{A}$, its expected cumulative loss is

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}(x_{<t}) \neq x_t\}\right] \geq \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{E}\left[\mathbb{1}\{\mathcal{A}(x_{<t}) \neq x_t\} \mid x_{<t}, \mathcal{A}\right] \mathbb{1}\{x_{t-1} \notin S_0\}\right] \geq \frac{1}{2}\,\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{x_{t-1} \notin S_0\}\right],$$

where the final step follows because when $x_{t-1} \notin S_0$, the state x_t is sampled uniformly at random between two states. So, the expected loss of algorithm in this round is $\geq \frac{1}{2}$.

To upper bound the cumulative loss of the best-fixed function in hindsight, define $\sigma =$

$\text{sign}\left(\sum_{t=1}^{T}\sigma_t\mathbb{1}\{x_{t-1}\notin S\}\right)$. Note that

$$\begin{aligned}
&\mathbb{E}\left[\inf_{f\in\mathcal{F}}\sum_{t=1}^{T}\mathbb{1}\{f(x_{t-1})\neq x_t\}\right]\\
&\leq\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\{f_\sigma(x_{t-1})\neq x_t\}\right]\\
&=\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\{f_\sigma(x_{t-1})\neq x_t\}\mathbb{1}\{x_{t-1}\in S\backslash S_0\}\right]+\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\{f_\sigma(x_{t-1})\neq f_{\sigma_t}(x_{t-1})\}\mathbb{1}\{x_{t-1}\notin S\}\right]\\
&\leq\frac{1}{2}\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\{x_{t-1}\in S\backslash S_0\}\right]+\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\{f_\sigma(x_{t-1})\neq f_{\sigma_t}(x_{t-1})\}\mathbb{1}\{x_{t-1}\notin S\}\right],
\end{aligned}$$

where the final step follows because conditioned on the event that $x_{t-1}\in S\backslash S_0$, the state x_t is $\sim\text{Uniform}(\{\bar{x}_0,f_{\sigma_t}(x_{t-1})\})$. Since $f_\sigma(x_{t-1})=f_{\sigma_t}(x_{t-1})$ whenever $x_{t-1}\in S$, the expected loss of f_σ on round t is equal to the conditional probability that $x_t=x_0$, which is $\leq 1/2$. Moreover, using the fact that $\mathbb{1}\{f_\sigma(x_{t-1})\neq f_{\sigma_t}(x_{t-1})\}\leq\mathbb{1}\{\sigma\neq\sigma_t\}$, we have

$$\begin{aligned}
\mathbb{E}\left[\inf_{f\in\mathcal{F}}\sum_{t=1}^{T}\mathbb{1}\{f(x_{t-1})\neq x_t\}\right]&\leq\frac{1}{2}\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\{x_{t-1}\in S\backslash S_0\}\right]+\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\{\sigma\neq\sigma_t\}\mathbb{1}\{x_{t-1}\notin S\}\right]\\
&=\frac{1}{2}\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\{x_{t-1}\in S\backslash S_0\}\right]+\mathbb{E}\left[\sum_{t=1}^{T}\frac{1-\sigma\sigma_t}{2}\mathbb{1}\{x_{t-1}\notin S\}\right]\\
&=\frac{1}{2}\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\{x_{t-1}\notin S_0\}\right]-\frac{1}{2}\mathbb{E}\left[\sigma\sum_{t=1}^{T}\sigma_t\mathbb{1}\{x_{t-1}\notin S\}\right]\\
&=\frac{1}{2}\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\{x_{t-1}\notin S_0\}\right]-\frac{1}{2}\mathbb{E}\left[\left|\sum_{t=1}^{T}\sigma_t\mathbb{1}\{x_{t-1}\notin S\}\right|\right],
\end{aligned}$$

where the final equality follows from the definition of σ. Thus, the regret of $\mathcal{A}$ is

$$\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\{\mathcal{A}(x_{<t})\neq x_t\}\right]-\mathbb{E}\left[\inf_{f\in\mathcal{F}}\sum_{t=1}^{T}\mathbb{1}\{f(x_{t-1})\neq x_t\}\right]\geq\frac{1}{2}\mathbb{E}\left[\left|\sum_{t=1}^{T}\sigma_t\mathbb{1}\{x_{t-1}\notin S\}\right|\right].$$

We now lower bound the Rademacher sum above by closely following the proof of Khinchine's inequality [Cesa-Bianchi and Lugosi, 2006, Lemma 8.1].

For any random variable Y with the finite-fourth moment, a simple application of Holder's

inequality implies that

$$\mathbb{E}[|Y|] \geq \frac{(\mathbb{E}[Y^2])^{3/2}}{(\mathbb{E}[Y^4])^{1/2}}.$$

We apply this inequality to $Y := \sum_{t=1}^{T} \sigma_t \mathbb{1}\{x_{t-1} \notin S\}$. Since $Y^2 = \sum_{t=1}^{T} \mathbb{1}\{x_{t-1} \notin S\} + 2\sum_{i<j} \sigma_i \sigma_j \mathbb{1}\{x_{i-1}, x_{j-1} \notin S\}$, we have

$$\begin{aligned}
\mathbb{E}[Y^2] &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{x_{t-1} \notin S\}\right] + 2\,\mathbb{E}\left[\sum_{i<j} \mathbb{E}[\sigma_j \mid \sigma_{<j}, x_{<j}]\ \sigma_i \mathbb{1}\{x_{i-1}, x_{j-1} \notin S\}\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{x_{t-1} \notin S\}\right],
\end{aligned}$$

where the final equality follows because σ_j is still a Rademacher random variable conditioned on the past and $\mathbb{E}[\sigma_j \mid \sigma_{<j}, x_{<j}] = 0$. Furthermore, note that

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{x_{t-1} \notin S\}\right] &\geq \mathbb{E}\left[\mathbb{1}\{x_0 \notin S\}\right] + \sum_{r=1}^{\lfloor \frac{T}{2} \rfloor - 1} \mathbb{E}\left[\mathbb{1}\{x_{2r-1} \notin S\} + \mathbb{1}\{x_{2r} \notin S\}\right] \\
&\geq 1 + \sum_{r=1}^{\lfloor \frac{T}{2} \rfloor - 1} \frac{1}{2} \\
&= \frac{1}{2}\left(\lfloor * \rfloor \frac{T}{2} + 1\right) \\
&\geq \frac{T}{4}.
\end{aligned}$$

Here, the second inequality above uses (i) $x_0 = \bar{x}_0 \notin S$ and (ii) $\mathbb{E}\left[\mathbb{1}\{x_{2r-1} \notin S\} + \mathbb{1}\{x_{2r} \notin S\}\right] \geq \frac{1}{2}$ for all $r \geq 1$. To see (ii), one can consider two cases. First, if $x_{2r-1} \notin S$, then the inequality is trivially true. On the other hand, $x_{2r-1} \in S$, then $x_{2r} = \bar{x}_0 \notin S$ with probability $1/2$. Thus, we have $\mathbb{E}[Y^2] \geq \frac{T}{4}$.

Similarly, for the fourth moment of Y, all the cross-term vanishes and we are left with the terms with fourth power and symmetric second powers.

$$\begin{aligned}
\mathbb{E}[Y^4] &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{x_{t-1} \notin S\}\right] + \binom{4}{2} \mathbb{E}\left[\sum_{i<j} \mathbb{1}\{x_{i-1} \notin S\}\, \mathbb{1}\{x_{j-1} \notin S\}\right] \\
&\leq T + \binom{4}{2} \frac{T(T-1)}{2} \\
&\leq T + 3T(T-1) \\
&\leq 3T^2.
\end{aligned}$$

The lower bound on the second moment of Y and the upper bound on the fourth moment of Y collectively implies that

$$\mathbb{E}[|Y|] \geq \frac{(\mathbb{E}[Y^2])^{3/2}}{(\mathbb{E}[Y^4])^{1/2}} \geq \frac{(T/4)^{3/2}}{(3T^2)^{1/2}} = \frac{\sqrt{T}}{8\sqrt{3}}.$$

Finally, combining everything, we obtain

$$\mathrm{MR}_{\mathcal{A}}(T, \mathcal{F}) \geq \frac{1}{2}\mathbb{E}\left[\left|\sum_{t=1}^{T} \sigma_t \mathbb{1}\{x_{t-1} \notin S\}\right|\right] = \frac{1}{2}\mathbb{E}[|Y|] = \frac{\sqrt{T}}{16\sqrt{3}}.$$

This completes our proof. ■

B.3.2 Tightness of upper bound in Theorem 4.4.1

Fix $d \in \mathbb{N}$ and let $\mathcal{X} = \{1, 2, \ldots, d\} \cup \{\pm(d+1), \pm(d+2), \ldots, \pm 2d\}$. For every $\sigma \in \{-1, 1\}^d$, define a function $f_\sigma : \mathcal{X} \to \mathcal{X}$ such that

$$f_\sigma(x) = \sigma_x(d + x)\mathbb{1}\{x \in [d]\} + (|x| - d)\mathbb{1}\{x \notin [d]\}.$$

Consider $\mathcal{F} = \{f_\sigma : \sigma \in \{-1, 1\}^d\}$. It is not too hard to see that $\mathrm{L}(\mathcal{F}) = d$. The fact that $\mathrm{L}(\mathcal{F}) \leq d$ is trivial because any Littlesone tree can only have $\{1, 2, \ldots, d\}$ in the internal nodes. This is because all functions output the same state in the domain $\mathcal{X} \backslash [d]$. Since the internal nodes in any given path of the Littlestone tree need to be different, the depth of any shattered tree is $\leq d$. To see why $\mathrm{L}(\mathcal{F}) \geq d$, consider a complete binary tree $\mathcal{T}$ of depth d with all the internal nodes in level $i \in [d]$ containing the element i. Let $-(d+i)$ and $(d+i)$ label the left and right outgoing edges respectively of every node in level i. Note that $\mathcal{T}$ is shattered by $\mathcal{F}$ as for any path $\varepsilon \in \{-1, 1\}^d$ down the tree $\mathcal{T}$, there exists a $f_\varepsilon \in \mathcal{F}$ such that $f_\varepsilon(i) = \varepsilon_i(d + i)$ for all $i \in [d]$. Thus, we have shown that $\mathrm{L}(\mathcal{F}) = d$.

We now show that $\inf_{\mathcal{A}} \mathrm{MR}_{\mathcal{A}}(T, \mathcal{F}) = \Omega(\sqrt{T\,\mathrm{L}(\mathcal{F})})$, proving that the upper bound in Theorem 4.4.1 is tight up to $\sqrt{\log T}$. For a $k \in \mathbb{N}$, pick $T = 2kd - 1$. Draw $\varepsilon \in \{-1, +1\}^T$ where $\varepsilon_i \sim \mathrm{Uniform}(\{-1, 1\})$ and consider a stream where $x_{2k(i-1)} = i$ and $x_t = \varepsilon_t(d + i)\mathbb{1}\{x_{t-1} = i\} + i\mathbb{1}\{x_{t-1} \neq i\} \quad \forall i \in [d]$ and $t \in \{2k(i-1) + 1, \ldots, 2ki - 1\}$ for all $i \in [d]$ and $t \in \{2k(i-1) + 1, \ldots, 2ki - 1\}$. Note that $x_0 = 1$ is the initial state and there are $T = 2kd - 1$ more states in this stream. For any algorithm $\mathcal{A}$, its expected cumulative loss

is

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}(x_{<t}) \neq x_t\}\right] \geq \mathbb{E}\left[\sum_{i=1}^{d} \sum_{t=2k(i-1)+1}^{2ki-1} \mathbb{1}\{\mathcal{A}(x_{<t}) \neq x_t\}\right]$$
$$\geq \frac{1}{2}\sum_{i=1}^{d} \sum_{t=2k(i-1)+1}^{2ki-1} \mathbb{1}\{x_{t-1} = i\}.$$

Here, the first inequality is true because we just rewrite the sum without including the rounds $t = 2k, 4k, 6k, \ldots, 2(d-1)k$. The second inequality holds because x_t is sampled uniformly at random between $-(d+i)$ and $d+i$ whenever $x_{t-1} = i$, and the algorithm cannot do better than guessing on these rounds. Note that there is no expectation following the second inequality because the event $x_{t-1} = i$ is deterministic.

To upper bound the loss of the best function in hindsight, define $\bar{\varepsilon}_i = \operatorname{sign}\big(\sum_{t=2k(i-1)+1}^{2ki-1} \varepsilon_t \mathbb{1}\{x_{t-1} = i\}\big)$ and consider $f_{\bar{\varepsilon}}$ where $\bar{\varepsilon} = (\bar{\varepsilon}_1, \ldots, \bar{\varepsilon}_d)$. Then,

$$\mathbb{E}\left[\inf_{f\in\mathcal{F}} \sum_{t=1}^{T} \mathbb{1}\{f(x_{t-1}) \neq x_t\}\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{f_{\bar{\varepsilon}}(x_{t-1}) \neq x_t\}\right]$$
$$\leq (d-1) + \mathbb{E}\left[\sum_{i=1}^{d} \sum_{t=2k(i-1)+1}^{2ki-1} \mathbb{1}\{f_{\bar{\varepsilon}}(x_{t-1}) \neq x_t\}\right]$$
$$= (d-1) + \mathbb{E}\left[\sum_{i=1}^{d} \sum_{t=2k(i-1)+1}^{2ki-1} \mathbb{1}\{f_{\bar{\varepsilon}}(x_{t-1}) \neq x_t\}\mathbb{1}\{x_{t-1} = i\}\right].$$

Here, the second inequality holds because we trivially upper bound the losses on rounds $t = 2k, 4k, \ldots, 2(d-1)k$ by 1. And the final equality follows because $f_{\bar{\varepsilon}}(x_{t-1}) = i = x_t$ whenever $x_{t-1} \neq i$ for $2k(i-1)+1 \leq t \leq 2ki-1$. Moreover, when $x_{t-1} = i$, we have $f_{\bar{\varepsilon}}(x_{t-1}) = \bar{\varepsilon}_i(d+i)$ and $x_t = \varepsilon_t(d+i)$. This implies that $\mathbb{1}\{f_{\bar{\varepsilon}}(x_{t-1}) \neq x_t\}\mathbb{1}\{x_{t-1} = i\} =$

$\mathbb{1}\{\bar{\varepsilon}_i \neq \varepsilon_t\}\mathbb{1}\{x_{t-1} = i\}$. Thus, we can write

$$\begin{aligned}
&\mathbb{E}\left[\inf_{f\in\mathcal{F}} \sum_{t=1}^{T} \mathbb{1}\{f(x_{t-1}) \neq x_t\}\right] \\
&\qquad \leq (d-1) + \mathbb{E}\left[\sum_{i=1}^{d} \sum_{t=2k(i-1)+1}^{2ki-1} \mathbb{1}\{\bar{\varepsilon}_i \neq \varepsilon_t\}\mathbb{1}\{x_{t-1} = i\}\right] \\
&\qquad = (d-1) + \mathbb{E}\left[\sum_{i=1}^{d} \sum_{t=2k(i-1)+1}^{2ki-1} \frac{1 - \bar{\varepsilon}_i\varepsilon_t}{2}\,\mathbb{1}\{x_{t-1} = i\}\right] \\
&\qquad = (d-1) + \sum_{i=1}^{d} \sum_{t=2k(i-1)+1}^{2ki-1} \frac{1}{2}\,\mathbb{1}\{x_{t-1} = i\} - \frac{1}{2}\mathbb{E}\left[\sum_{i=1}^{d} \bar{\varepsilon}_i \sum_{t=2k(i-1)+1}^{2ki-1} \varepsilon_t \mathbb{1}\{x_{t-1} = i\}\right] \\
&\qquad = (d-1) + \sum_{i=1}^{d} \sum_{t=2k(i-1)+1}^{2ki-1} \frac{1}{2}\,\mathbb{1}\{x_{t-1} = i\} - \frac{1}{2}\mathbb{E}\left[\sum_{i=1}^{d} \left|\sum_{t=2k(i-1)+1}^{2ki-1} \varepsilon_t \mathbb{1}\{x_{t-1} = i\}\right|\right]
\end{aligned}$$

Thus, the expected regret of $\mathcal{A}$ is

$$\begin{aligned}
\mathrm{MR}_{\mathcal{A}}(T, \mathcal{F}) &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}(x_{<t}) \neq x_t\}\right] - \mathbb{E}\left[\inf_{f\in\mathcal{F}} \sum_{t=1}^{T} \mathbb{1}\{f(x_{t-1}) \neq x_t\}\right] \\
&\geq \frac{1}{2}\mathbb{E}\left[\sum_{i=1}^{d} \left|\sum_{t=2k(i-1)+1}^{2ki-1} \varepsilon_t \mathbb{1}\{x_{t-1} = i\}\right|\right] - (d-1) \\
&\geq \frac{1}{2}\sum_{i=1}^{d} \sqrt{\frac{|\{2k(i-1)+1 \leq t \leq 2ki-1 \,:\, x_{t-1} = i\}|}{2}} - (d-1) \\
&= \frac{1}{2}\sum_{i=1}^{d} \sqrt{\frac{k}{2}} - (d-1).
\end{aligned}$$

where the second inequality follows due to Khinchine's inequality [Cesa-Bianchi and Lugosi, 2006, Lemma 8.2]. The final equality holds because in each block of $2k$ rounds from $t = 2k(i-1)+1$ to $t = 2ki-1$, we have $x_{t-1} = i$ in exactly k of those rounds. Using $T = 2kd-1$, we further obtain $\mathrm{MR}_{\mathcal{A}}(T, \mathcal{F}) \geq \frac{1}{4}\sqrt{(T+1)d} - (d-1) = \Omega(\sqrt{Td})$ for $T \gg d$. This establishes our claim that the upper bound in Theorem 4.4.1 is tight up to $\sqrt{\log T}$.

B.3.3 Proof of Theorem 4.4.3

The lower bound in Theorem 4.4.3 follows directly from the lower bound in the realizable setting. Accordingly, we only prove the upper bound. Let $K := \sup_{x\in\mathcal{X}} |\mathcal{F}(x)|$.

Proof. (of the upper bound in Theorem 4.4.3) Let $\{x_t\}_{t=0}^T$ be the trajectory to be observed by the learner and $f^\star \in \arg\min_{f\in\mathcal{F}} \sum_{t=1}^T \mathbb{1}\{x_t \neq f^t(x_0)\}$ the optimal evolution function in hindsight. Given the time horizon T, let $L_T = \{L \subset [T] : |L| \leq \mathrm{C}_T(\mathcal{F})\}$ denote the set of all possible subsets of $[T]$ with size at most $\mathrm{C}_T(\mathcal{F})$. For every $L \in L_T$, let $\phi : L \to [K]$ denote a function mapping time points in L to an integer in $[K]$. On time point $t \in L$, we should think of $\phi(t)$ as an index into the set $\mathcal{F}(x_{t-1})$. To that end, for an index $i \in [|\mathcal{F}(x)|]$, let $\mathcal{F}_i(x)$ denote the i-th element of the list obtained after sorting $\mathcal{F}(x)$ in its natural order. Let $\Phi_L = [K]^L$ denote all such functions ϕ. For each $L \in L_T$ and $\phi \in \Phi_L$, define an expert $\mathcal{E}_{L,\phi}$. On time point $t \in [T]$, the prediction of expert $\mathcal{E}_{L,\phi}$ is defined recursively by

$$\mathcal{E}_{L,\phi}(x_0, t) = \begin{cases} \mathcal{A}\left(x_0, t | \{\mathcal{E}_{L,\phi}(x_0, i)\}_{i=1}^{t-1}\right), & \text{if } t \notin L \\ \mathcal{F}_{\phi(t)}(\mathcal{E}_{L,\phi}(x_0, t-1)), & \text{otherwise} \end{cases}$$

where $\mathcal{E}_{L,\phi}(x_0, 0) = x_0$ and $\mathcal{A}\left(x_0, t | \{\mathcal{E}_{L,\phi}(x_0, i)\}_{i=1}^{t-1}\right)$ denotes the prediction of Algorithm 16 on timepoint t after running and updating on the trajectory $\{\mathcal{E}_{L,\phi}(x_0, i)\}_{i=1}^{t-1}$. Let $E = \bigcup_{L\in L_T} \bigcup_{\phi\in\Phi_L} \mathcal{E}_{L,\phi}$ denote the set of all experts parameterized by $L \in L_T$ and $\phi \in \Phi_L$.

We claim that there exists an expert $\mathcal{E}_{L^\star,\phi^\star}$ such that $\mathcal{E}_{L^\star,\phi^\star}(x_0, t) = f^{\star,t}(x_0)$ for all $t \in [T]$. To see this, consider the hypothetical trajectory $S^\star = \{f^{\star,t}(x_0)\}_{t=1}^T$ generated by $f^\star$. Let $L^\star = \{t_1, t_2, ...\}$ be the indices on which Algorithm 16 would have made a mistake had it run and updated on $S^\star$. By the guarantee of Algorithm 16, we know that $|L^\star| \leq \mathrm{C}_T(\mathcal{F})$. Moreover, there exists a $\phi^\star \in \Phi_{L^\star}$ such that $\mathcal{F}_{\phi^\star(i)}(f^{\star,i-1}(x_0)) = f^{\star,i}(x_0)$ for all $i \in L^\star$. By construction of $\mathcal{E}$, there exists an expert $\mathcal{E}_{L^\star,\phi^\star}$ parameterized by $L^\star$ and $\phi^\star$. We claim that $\mathcal{E}_{L^\star,\phi^\star}(x_0, t) = f^{\star,t}(x_0)$ for all $t \in [T]$. This follows by strong induction on $t \in [T]$. For the base case $t = 1$, there are two subcases to consider. If $1 \in L^\star$, then we have that $\mathcal{E}_{L^\star,\phi^\star}(x_0, 1) = \mathcal{F}_{\phi^\star(1)}(\mathcal{E}_{L^\star,\phi^\star}(x_0, 0)) = \mathcal{F}_{\phi^\star(1)}(x_0) = f^{\star,1}(x_0)$. If $1 \notin L^\star$, then $\mathcal{E}_{L^\star,\phi^\star}(x_0, 1) = \mathcal{A}(x_0, 1|\{\}) = f^{\star,1}(x_0)$, where the last equality follows by definition of $L^\star$. Now for the induction step, suppose that $\mathcal{E}_{L^\star,\phi^\star}(x_0, i) = f^{\star,i}(x_0)$ for all $i \leq t$. Then, if $t + 1 \in L^\star$, we have that $\mathcal{E}_{L^\star,\phi^\star}(x_0, t+1) = \mathcal{F}_{\phi^\star(t+1)}(\mathcal{E}_{L^\star,\phi^\star}(x_0, t)) = \mathcal{F}_{\phi^\star(t+1)}(f^{\star,t}(x_0)) = f^{\star,t+1}(x_0)$. If $t+1 \notin L^\star$, then $\mathcal{E}_{L^\star,\phi^\star}(x_0, t+1) = \mathcal{A}(x_0, t+1|\{\mathcal{E}_{L^\star,\phi^\star}(x_0, i)\}_{i=1}^t) = \mathcal{A}(x_0, t+1|\{f^{\star,i}(x_0)\}_{i=1}^t) = f^{\star,t+1}(x_0)$, where the last equality again is due to the definition of $L^\star$.

Now, consider the learner that runs the celebrated Randomized Exponential Weights Algorithm, denoted hereinafter by $\mathcal{P}$, using the set of experts E with learning rate $\eta = \sqrt{\frac{2\ln(|E|)}{T}}$. By Theorem 21.11 of Shalev-Shwartz and Ben-David [2014a], we have that

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{P}(x_0,t) \neq x_t\}\right] &\leq \inf_{\mathcal{E}\in E} \sum_{t=1}^{T} \mathbb{1}\{\mathcal{E}(x_0,t) \neq x_t\} + \sqrt{2T\ln(|E|)} \\
&\leq \inf_{\mathcal{E}\in E} \sum_{t=1}^{T} \mathbb{1}\{\mathcal{E}(x_0,t) \neq x_t\} + \sqrt{2\,\mathrm{C}_T(\mathcal{F})\,T\ln\Big(\frac{KT}{\mathrm{C}_T(\mathcal{F})}\Big)} \\
&\leq \sum_{t=1}^{T} \mathbb{1}\{\mathcal{E}_{L^\star,\phi^\star}(x_0,t) \neq x_t\} + \sqrt{2\,\mathrm{C}_T(\mathcal{F})\,T\ln\Big(\frac{KT}{\mathrm{C}_T(\mathcal{F})}\Big)} \\
&= \sum_{t=1}^{T} \mathbb{1}\{f^{\star,t}(x_0) \neq x_t\} + \sqrt{2\,\mathrm{C}_T(\mathcal{F})\,T\ln\Big(\frac{KT}{\mathrm{C}_T(\mathcal{F})}\Big)}
\end{aligned}$$

where the second inequality follows because $|E| = \sum_{i=0}^{\mathrm{C}_T(\mathcal{F})} K^i \binom{T}{i} \leq \left(\frac{KT}{\mathrm{C}_T(\mathcal{F})}\right)^{\mathrm{C}_T(\mathcal{F})}$. This completes the proof as $\mathcal{P}$ achieves the stated upper bound on expected regret. ■

The next two examples show that both the lower- and upper bounds in Theorem 4.4.3 can be tight.

B.3.4 Tightness of lower bound in Theorem 4.4.3

Let $\mathcal{X} = \mathbb{Z}$ and fix $p \in \mathbb{N}$. For every $\sigma \in \{-1,1\}^{\mathbb{N}\cup\{0\}}$, define the evolution function

$$f_\sigma(x) = (\sigma_{|x|}\mathbb{1}\{|x| \leq p-1\} + \mathbb{1}\{|x| \geq p\})(|x|+1)$$

and consider the class $\mathcal{F} = \Big\{f_\sigma : \sigma \in \{-1,1\}^{\mathbb{N}\cup\{0\}}\Big\}$. By construction of $\mathcal{F}$, branching only occurs on states in $\{0,1,...,p-1\}$ and their negation. Moreover, given any initial state $x_0 \in \mathcal{X}$ and a time horizon $T \in \mathbb{N}$, the trajectory of any evolution in $\mathcal{F}$ is some signed version of the sequence $|x_0|+1,...,|x_0|+T$. From Theorem 4.3.2, its not too hard to see that $\mathrm{C}_T(\mathcal{F}) = \min\{p,T\}$. We now show that $\inf_{\mathcal{A}} \mathrm{FR}_{\mathcal{A}}(T,\mathcal{F}) \leq \frac{\mathrm{C}_T(\mathcal{F})}{2}$. Consider the learner $\mathcal{A}$ which, given initial state x_0, checks whether $|x_0| \leq p-1$. If $|x_0| \leq p-1$, the learner $\mathcal{A}$ samples a random sequence $\varepsilon \sim \{-1,1\}^{p-|x_0|}$ and plays $\varepsilon_t(|x_0|+t)$ for $t \leq p-|x_0|$ and $|x_0|+t$ in all future rounds $t > p-|x_0|$. If $|x_0| > p-1$, the learner $\mathcal{A}$ plays $|x_0|+t$ for all $t \geq 1$.

Let $\{x_t\}_{t=0}^T$ be the trajectory to be observed by the learner. If $|x_0| > p-1$, then observe that

$$\begin{aligned}\mathrm{FR}_{\mathcal{A}}(T,\mathcal{F}) &= \mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\{\mathcal{A}(x_{<t})\neq x_t\} - \inf_{f\in\mathcal{F}}\sum_{t=1}^{T}\mathbb{1}\{f^t(x_0)\neq x_t\}\right]\\ &= \mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\{|x_0|+t\neq x_t\} - \inf_{f\in\mathcal{F}}\sum_{t=1}^{T}\mathbb{1}\{|x_0|+t\neq x_t\}\right] = 0\end{aligned}$$

On the other hand, if $|x_0|\leq p-1$, we can write the learner's expected loss as

$$\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\{\mathcal{A}(x_{<t})\neq x_t\}\right] = \mathbb{E}\left[\sum_{t=1}^{p-|x_0|}\mathbb{1}\{\varepsilon_t(|x_0|+t)\neq x_t\}\right] + \sum_{t=p-|x_0|+1}^{T}\mathbb{1}\{|x_0|+t\neq x_t\}.$$

Similarly, we can write the cumulative loss of the competitor term as:

$$\inf_{f\in\mathcal{F}}\sum_{t=1}^{T}\mathbb{1}\{f^t(x_0)\neq x_t\} = \inf_{f\in\mathcal{F}}\sum_{t=1}^{p-|x_0|}\mathbb{1}\{f^t(x_0)\neq x_t\} + \sum_{t=p-|x_0|+1}^{T}\mathbb{1}\{|x_0|+t\neq x_t\}.$$

Let $f_\sigma = \arg\min_{f\in\mathcal{F}}\sum_{t=1}^{p-|x_0|}\mathbb{1}\{f^t(x_0)\neq x_t\}$. Combining both bounds, we get that:

$$\begin{aligned}\mathrm{FR}_{\mathcal{A}}(T,\mathcal{F}) &= \mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\{\mathcal{A}(x_{<t})\neq x_t\} - \inf_{f\in\mathcal{F}}\sum_{t=1}^{T}\mathbb{1}\{f^t(x_0)\neq x_t\}\right]\\ &\leq \mathbb{E}\left[\sum_{t=1}^{p-|x_0|}\mathbb{1}\{\varepsilon_t(|x_0|+t)\neq x_t\} - \mathbb{1}\{f_\sigma^t(x_0)\neq x_t\}\right]\\ &\leq \mathbb{E}\left[\sum_{t=1}^{p-|x_0|}\mathbb{1}\{\varepsilon_t(|x_0|+t)\neq f_\sigma^t(x_0)\}\right]\\ &= \mathbb{E}\left[\sum_{t=1}^{p-|x_0|}\mathbb{1}\{\varepsilon_t(|x_0|+t)\neq \sigma_{|x_0|+t-1}(|x_0|+t)\}\right] = \frac{p-|x_0|}{2},\end{aligned}$$

where in the last equality we used the fact that $\varepsilon_t\sim\mathrm{Unif}(-1,1)$. Thus, the expected regret of such an learner $\mathcal{A}$ is at most $\frac{\min\{\max\{p-|x_0|,0\},T\}}{2}$. Taking $x_0=0$, gives that $\inf_{\mathcal{A}}\mathrm{FR}_{\mathcal{A}}(T,\mathcal{F})\leq \frac{\mathrm{C}_T(\mathcal{F})}{2}$.

B.3.4.1 Tightness of upper bound in Theorem 4.4.3

Let $\mathcal{X} = \mathbb{Z}$ and fix $p \in \mathbb{N}$. Let $\mathcal{G} = \{(0, s_1, ..., s_p) : s_1 < s_2 < ... < s_p \in \mathbb{N}\}$ be the set of all ordered tuples of extended natural numbers with size $p+1$. For any $S \in \mathcal{G}$ and $x \in \mathbb{N} \cup \{0\}$, let $S_x = \max\{s \in S : s \leq x\}$ be the largest element in S smaller than x. Note that $S_0 = 0$ for all $S \in \mathcal{G}$. For every $\sigma \in \{-1, 1\}^{\mathbb{N}\cup\{0\}}$ and $S \in \mathcal{G}$, consider the evolution function:

$$f_{\sigma,S}(x) = \sigma_{S_{|x|}}(|x| + 1).$$

Given any initial state $x_0 \in \mathcal{X}$ and time horizon $T \in \mathbb{N}$, the trajectory $\{f^t_{\sigma,S}(x_0)\}_{t=1}^T$ is some signed version of the sequence $(|x_0| + 1, ..., |x_0| + T)$, where the signs depend on S and σ. More importantly, the trajectory $\{f^t_{\sigma,S}(x_0)\}_{t=1}^T$ switches signs at most $p+1$ times. Finally, consider the evolution class $\mathcal{F} = \{f_{\sigma,S} : \sigma \in \{-1, 1\}^{\mathbb{N}\cup\{0\}}, S \in \mathcal{G}\}$. Since the trajectory of any evolution $f \in \mathcal{F}$ can switch signs at most $p + 1$ times, it follows that $\mathrm{C}_T(\mathcal{F}) \leq p + 1$. Moreover, by considering a trajectory tree with the root node labeled by 0 and the set of evolutions parameterized by the tuple $(0, 1, ..., p) \in \mathcal{G}$, its not hard to see that $\mathrm{C}_T(\mathcal{F}) \geq p+1$. Thus, $\mathrm{C}_T(\mathcal{F}) = p + 1$.

We now claim that $\inf_{\mathcal{A}} \mathrm{FR}_{\mathcal{A}}(T, \mathcal{F}) \geq \sqrt{\frac{\mathrm{C}_T(\mathcal{F})\, T}{8}}$, which shows that the upper bound in Theorem 4.4.3 is tight up to a logarithmic factor in T since $\sup_{x\in\mathcal{X}} |\mathcal{F}(x)| = 2$. Consider $T = k(p + 1)$ for some odd $k \in \mathbb{N}$. For $\varepsilon \in \{-1, 1\}^T$, define $\tilde{\varepsilon}_i = \mathrm{sign}\Big(\sum_{t=(i-1)k+1}^{ik} \varepsilon_t\Big)$ for all $i \in \{1, 2, ..., p + 1\}$. The game proceeds as follows. The adversary samples a string $\varepsilon \in \{-1, 1\}^T$ uniformly at random and constructs the random trajectory $\{x_t\}_{t=0}^T$ where $x_0 = 0$ is the initial state and $x_t = \varepsilon_t t$ for all $t \geq 1$. The adversary then passes $\{x_t\}_{t=0}^T$ to the learner.

Let $\mathcal{A}$ be any randomized learner. Then, for each block $i \in [p + 1]$, we have that

$$\mathbb{E}\left[\sum_{t=(i-1)k+1}^{ik} \mathbb{1}\{\mathcal{A}(x_{<t}) \neq x_t\}\right] \geq \sum_{t=(i-1)k+1}^{ik} \frac{1}{2} = \frac{k}{2},$$

where the inequality follows from the fact that x_t is chosen uniformly at random between t and $-t$. Let $f_{\sigma,S} \in \mathcal{F}$ be the function in $\mathcal{F}$ such that $S = (0, k, 2k, ..., (p+1)k)$ and $\sigma_{S_{|x|}} = \tilde{\varepsilon}_i$ for all $(i - 1)k \leq |x| \leq ik - 1$ and $i \in \{1, ..., p + 1\}$. For each block $i \in [p + 1]$, we have

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=(i-1)k+1}^{ik} \mathbb{1}\{f^t_{\sigma,S}(0) \neq x_t\}\right] &= \mathbb{E}\left[\sum_{t=(i-1)k+1}^{ik} \mathbb{1}\{\sigma_{S_{|t-1|}} t \neq \varepsilon_t t\}\right] \\
&= \mathbb{E}\left[\sum_{t=(i-1)k+1}^{ik} \mathbb{1}\{\tilde{\varepsilon}_i \neq \varepsilon_t\}\right] \\
&= \frac{k}{2} - \frac{1}{2}\mathbb{E}\left[\sum_{t=(i-1)k+1}^{ik} \tilde{\varepsilon}_i \varepsilon_t\right] \\
&= \frac{k}{2} - \frac{1}{2}\mathbb{E}\left[\Big|\sum_{t=(i-1)k+1}^{ik} \varepsilon_t\Big|\right] \\
&\leq \frac{k}{2} - \sqrt{\frac{k}{8}},
\end{aligned}$$

where the final step follows upon using Khinchine's inequality [Cesa-Bianchi and Lugosi, 2006]. Combining these two bounds above, we obtain,

$$\mathbb{E}\left[\sum_{t=(i-1)k+1}^{ik} \mathbb{1}\{\mathcal{A}(x_{<t}) \neq x_t\} - \sum_{t=(i-1)k+1}^{ik} \mathbb{1}\{f^t_{\sigma,S}(0) \neq x_t\}\right] \geq \sqrt{\frac{k}{8}}.$$

Summing this inequality over $p+1$ blocks, we obtain

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}(x_{<t}) \neq x_t\} - \inf_{f\in\mathcal{F}} \sum_{t=1}^{T} \mathbb{1}\{f^t(x_0) \neq x_t\}\right] \geq (p+1)\sqrt{\frac{k}{8}} = \sqrt{\frac{\mathrm{C}_T(\mathcal{F})T}{8}},$$

which completes our proof.

B.3.5 Proof of Theorem 4.4.4

Our proof of Theorem 4.4.4 is constructive. That is, we provide an evolution function class and show that there exists a deterministic algorithm with a mistake bound of 3 in the realizable setting. As for the agnostic setting, we use the probabilistic method to argue the existence of a hard stream such that every algorithm incurs $T/6$ regret.

Let $\mathcal{X} = \{-1,1\}^{\mathbb{N}} \times \mathbb{Z}$. For every $\sigma \in \{-1,1\}^{\mathbb{N}}$, define an evolution function

$$f_\sigma((\theta, z)) = (\sigma, \sigma_{|z|+1}(|z|+1)) \;\; \mathbb{1}\{|z|+1 = 0 \,(\mathrm{mod}\, 3)\} + \\ (\mathbf{1}, \sigma_{|z|+1}(|z|+1)) \;\; \mathbb{1}\{|z|+1 \neq 0 \,(\mathrm{mod}\, 3)\},$$

where $\theta \in \{-1,1\}^{\mathbb{N}}, z \in \mathbb{Z}$, and $\mathbf{1} := (1,1,\ldots,1,1)$ is the string of all ones. Consider the evolution function class $\mathcal{F} = \left\{ f_\sigma : \sigma \in \{-1,1\}^{\mathbb{N}} \right\}$.

To prove (i), it suffices to note that for any initial state (θ_0, z_0) and $f_\sigma \in \mathcal{F}$, the realizable stream $\{f_\sigma^t((\theta_0, z_0))\}_{t=1}^T$ must reveal σ within the first three rounds $t \in \{1,2,3\}$. A deterministic algorithm $\mathcal{A}$ that plays arbitrarily in the beginning and $f_\sigma^t((\theta_0, z_0))$ for all $t \geq 4$ makes no more than 3 mistakes.

To prove (ii), consider a random trajectory such that $x_0 = (\mathbf{1}, 0)$ is the initial state and $\{x_t\}_{t=1}^T = \{(\mathbf{1}, \varepsilon_t\, t)\}_{t=1}^T$, where $\varepsilon_t \sim \mathrm{Uniform}(\{-1,1\})$. Since the stream is generated uniformly at random, for any algorithm $\mathcal{A}$, we must have

$$\mathbb{E}\left[\sum_{t=1}^T \mathbb{1}\{\mathcal{A}(x_{<t}) \neq (\mathbf{1}, \varepsilon_t t)\}\right] \geq \frac{T}{2}.$$

Next, let $\varepsilon \in \{-1,1\}^{\mathbb{N}}$ be any completion of $(\varepsilon_1, \ldots, \varepsilon_T)$ and consider the function $f_\varepsilon \in \mathcal{F}$. Note that, for every $t \in [T]$, we have $f_\varepsilon(\mathbf{1}, \varepsilon_{t-1}(t-1)) = (\mathbf{1}, \varepsilon_t t)$ if $t \neq 0 \,(\mathrm{mod}\, 3)$ and $f_\varepsilon(\mathbf{1}, \varepsilon_{t-1}(t-1)) = (\varepsilon, \varepsilon_t\, t) \neq (\mathbf{1}, \varepsilon_t t)$ if $t = 0 \;(\mathrm{mod})\, 3$. Moreover, it is not too hard to see that

$$f_\varepsilon^t((\mathbf{1}, 0)) = (\varepsilon, \varepsilon_t t) \;\; \mathbb{1}\{t = 0 \,(\mathrm{mod}\, 3)\} + (\mathbf{1}, \varepsilon_t t) \;\; \mathbb{1}\{t \neq 0 \,(\mathrm{mod}\, 3)\}.$$

Since the functions $f \in \mathcal{F}$ ignore the first argument of the tuple, we have $f_\varepsilon^t((\mathbf{1}, 0)) = f_\varepsilon(f_\varepsilon^{t-1}(\mathbf{1}, 0)) = f_\varepsilon((\mathbf{1}, \varepsilon_{t-1}(t-1)))$. So, using the equality established above, we obtain

$$\begin{aligned}
\inf_{f \in \mathcal{F}} \sum_{t=1}^T \mathbb{1}\{f^t(x_0) \neq x_t\} &\leq \sum_{t=1}^T \mathbb{1}\{f_\varepsilon^t(x_0) \neq x_t\} \\
&\leq \sum_{t=1}^T \mathbb{1}\{f_\varepsilon^t((\mathbf{1}, 0)) \neq (\mathbf{1}, \varepsilon_t t)\} \\
&= \sum_{t=1}^T \mathbb{1}\{f_\varepsilon((\mathbf{1}, \varepsilon_{t-1}(t-1))) \neq (\mathbf{1}, \varepsilon_t t)\} \\
&= \sum_{t=1}^T \mathbb{1}\{t = 0 \,(\mathrm{mod}\, 3)\} \leq \frac{T}{3}.
\end{aligned}$$

Therefore, we have $\mathrm{FR}_{\mathcal{A}}(T, \mathcal{F}) \geq \frac{T}{2} - \frac{T}{3} \geq \frac{T}{6}$.

APPENDIX C

Multiclass Online Learnability Under Bandit Feedback

C.1 Missing Proofs

C.1.1 Proof of Lemma 5.3.3

To prove Lemma 5.3.3, we slightly modify the generic agnostic learner witnessing the proof of Theorem 5.2.2. Recall that the agnostic learner in Theorem 5.2.2 first constructs a sufficiently small set of experts E such that for every hypothesis $h \in \mathcal{H}$, there exists an expert $\mathcal{E}_h \in E$ whose predictions exactly match h over the stream. Then, the learner runs the non-mixing version of EXP4 (see Figure 4.1 and Theorem 4.2 in Bubeck et al. [2012]) with this set of experts E on the stream, for an appropriately chosen learning rate. Unfortunately, in all rounds $t \in [T]$, some of the experts constructed by this learner output predictions lying outside of $\mathcal{H}(x_t)$. Thus, EXP4 with this set of experts does not satisfy the constraint imposed by Lemma 5.3.3, that its predictions on x_t must lie in $\mathcal{H}(x_t)$. To fix this issue, we modify each expert $\mathcal{E} \in E$ such that for every $t \in [T]$, we have that $\mathcal{E}(x_t) \in \mathcal{H}(x_t)$ while still maintaining the property of the expert set of Daniely and Helbertal [2013]: for every $h \in \mathcal{H}$, there exists an expert $\mathcal{E}_h \in E$ that predicts exactly like h over the stream. Our modification is simple: in contrast to the experts constructed by Daniely and Helbertal [2013], our experts may predict using the "covering function" ϕ (as defined in Daniely and Helbertal [2013]) only if its value lies in $\mathcal{H}(x_t)$. Running the EXP4 algorithm using this new set of experts gives the claimed regret guarantee. We now formalize this construction.

Let $(x_1, y_1), ..., (x_T, y_T) \in (\mathcal{X} \times \mathcal{Y})^T$ denote the stream of instances to be observed by the learner and $h^\star \in \arg\min_{h \in \mathcal{H}} \sum_{t=1}^{T} \mathbb{1}\{h(x_t) \neq y_t\}$ denote the optimal hypothesis in hindsight. As stated before, our high-level strategy will be to construct a set of experts E and then run EXP4 using E over the stream. Crucially, we will guarantee that $\mathcal{E}(x_t) \in \mathcal{H}(x_t)$ for every $\mathcal{E} \in E$.

Given the time horizon T, let $L_T = \{L \subset [T]; |L| \leq \mathrm{L}(\mathcal{H})\}$ denote the set of all possible subsets of $[T]$ of size at most $\mathrm{L}(\mathcal{H})$. For every $L \in L_T$, let $\phi : L \to \mathcal{Y}$ denote a function mapping time points in L to a label in $\mathcal{Y}$. Let $\Phi_L = \mathcal{Y}^L$ denote all such functions ϕ. For each $L \in L_T$ and $\phi \in \Phi_L$, we define an expert $\mathcal{E}_{L,\phi}$. As presented below in Algorithm 18, expert $\mathcal{E}_{L,\phi}$ uses the Standard Optimal Algorithm (SOA) [Littlestone, 1987] to make its prediction in rounds t where $t \notin L$. When $t \in L$, there are two cases. If $\phi(t) \in \mathcal{H}(x_t)$, the expert $\mathcal{E}_{L,\phi}$ uses the function ϕ to compute a labeled instance to predict and update the SOA with. Otherwise, the expert chooses an arbitrary label in $\mathcal{H}(x_t)$ to predict and update SOA with. Let $E = \bigcup_{L \in L_T} \bigcup_{\phi \in \Phi_L} \mathcal{E}_{L,\phi}$ denote the set of all Experts parameterized by subsets $L \in L_T$ and $\phi \in \Phi_L$. Crucially, observe that by definition of SOA, for every time point $t \in [T]$ and expert $\mathcal{E} \in E$, it holds that $\mathcal{E}(x_t) \in \mathcal{H}(x_t)$. Finally, note that $|E| \leq (T|\mathcal{Y}|)^{\mathrm{L}(\mathcal{H})}$.

Algorithm 18 Expert $\mathcal{E}_{L,\phi}$

Require: Independent copy of SOA
for $t = 1, ..., T$ **do**
  Receive example x_t
  Let $\tilde{y}_t = \mathrm{SOA}(x_t)$
  if $t \in L$ and $\phi(t) \in \mathcal{H}(x_t)$ **then**
    Predict $\hat{y}_t = \phi(t)$
  else if $t \in L$ and $\phi(t) \notin \mathcal{H}(x_t)$ **then**
    Predict arbitrary label $\hat{y}_t \in \mathcal{H}(x_t)$
  else
    Predict $\hat{y}_t = \tilde{y}_t$
  end if
  Update SOA by passing $(x_t, \hat{y}_t)$
end for

We claim that there exists an expert $\mathcal{E}_{L^\star,\phi^\star} \in E$ such that $h^\star(x_t) = \mathcal{E}_{L^\star,\phi^\star}(x_t)$ for all $t \in [T]$. To see this, consider the hypothetical stream of instances labeled by the optimal hypothesis $S^\star = (x_1, h^\star(x_1)), ..., (x_T, h^\star(x_T))$. Let $L^\star = \{t_1, t_2, ...\}$ be the indices on which the SOA algorithm would have made a mistake had it run on S^*. By the guarantees of the SOA [Littlestone, 1987], we have that $|L^\star| \leq \mathrm{L}(\mathcal{H})$. Consider the function $\phi^\star : L^\star \to \mathcal{Y}$ such that for all $t \in L^\star$, we have $\phi^\star(t) = h^\star(x_t)$. By construction of E, there exists an expert $\mathcal{E}_{L^\star,\phi^\star} \in E$ parameterized by $L^\star$ and $\phi^\star$. We claim that for all $t \in [T]$, we have $\mathcal{E}_{L^\star,\phi^\star}(x_t) = h^\star(x_t)$. This follows by observing that $\mathcal{E}_{L^\star,\phi^\star}$ predicts and updates its copy of SOA using exactly the stream of instances labeled by $h^\star$. Since by definition of $L^\star$ the predictions of SOA match that of $h^\star$ outside of $L^\star$, we have $\mathcal{E}_{L^\star,\phi^\star}(x_t) = \mathrm{SOA}(x_t) = h^\star(x_t)$ for all $t \notin L^\star$. Moreover, for those time points $t \in L^\star$, we have that $\mathcal{E}_{L^\star,\phi^\star}(x_t) = \phi^\star(t) = h^\star(x_t)$ by definition of $\phi^\star(t)$. Thus, for all $t \in [T]$, we have that $\mathcal{E}_{L^\star,\phi^\star}(x_t) = h^\star(x_t)$.

Consider the agnostic online learner $\mathcal{A}$ that runs the non-mixing version of EXP4 (see Fig. 4.1 and Theorem 4.2 in Bubeck et al. [2012]) using the set of experts E with learning rate $\eta = \sqrt{\frac{\ln|E|}{T|\mathcal{Y}|}}$. By the guarantees of the EXP4 algorithm, it follows that

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}(x_t) \neq y_t\}\right] &\leq \inf_{\mathcal{E}\in E} \sum_{t=1}^{T} \mathbb{1}\{\mathcal{E}(x_t) \neq y_t\} + e\sqrt{T|\mathcal{Y}|\ln|E|} \\
&\leq \sum_{t=1}^{T} \mathbb{1}\{\mathcal{E}_{L^\star,\phi^\star}(x_t) \neq y_t\} + e\sqrt{\mathrm{L}(\mathcal{H})|\mathcal{Y}|T\ln(T|\mathcal{Y}|)} \\
&= \sum_{t=1}^{T} \mathbb{1}\{h^\star(x_t) \neq y_t\} + e\sqrt{\mathrm{L}(\mathcal{H})|\mathcal{Y}|T\ln(T|\mathcal{Y}|)}.
\end{aligned}$$

Finally, observing that for all $t \in [T]$, $\cup_{\mathcal{E}\in E}\{\mathcal{E}(x_t)\} \subseteq \mathcal{H}(x_t)$ together with the fact that EXP4 algorithm in Figure 4.1 of Bubeck et al. [2012] samples a label using a distribution supported only over $\cup_{\mathcal{E}\in E}\{\mathcal{E}(x_t)\}$ ensures that $\mathcal{A}(x_t) \in \mathcal{H}(x_t)$ almost surely (equivalently, the EXP4 algorithm samples an expert $\mathcal{E} \in E$ and uses its prediction). This completes the proof of Lemma 5.3.3.

APPENDIX D

Apple Tasting: Combinatorial Dimensions and Minimax Rates

D.1 Upper bounds for Deterministic Learners in the Realizable Setting

In this section, we provide deterministic apple tasting learners for some special classes. Our first contribution shows that when $\mathrm{W}(\mathcal{H}) = 1$, there exists deterministic online learner which makes at most $\mathrm{AL}_1(\mathcal{H})$ mistakes under apple tasting feedback.

Theorem D.1.1 (Deterministic Realizable upper bound when $\mathrm{W}(\mathcal{H}) = 1$)**.** *For any $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$, there exists a deterministic online learner which, under apple tasting feedback, makes at most $\mathrm{AL}_1(\mathcal{H})$ mistakes in the realizable setting.*

Proof. We will show that Algorithm 19 makes at most $\mathrm{AL}_1(\mathcal{H})$ mistakes in the realizable setting.

Algorithm 19 Deterministic Realizable Algorithm For Apple Tasting

Require: $V_1 = \mathcal{H}$

1: **for** $t = 1, ..., T$ **do**
2: Receive x_t.
3: If there exists $h \in V_t$ such that $h(x_t) = 1$, predict $\hat{y}_t = 1$. Else, predict $\hat{y}_t = 0$.
4: If $\hat{y}_t = 1$, receive y_t and update $V_{t+1} \leftarrow \{h \in V_t : h(x_t) = y_t\}$
5: **end for**

Let $t \in [T]$ be any round such that $\hat{y}_t \neq y_t$. We will show $\mathrm{AL}_1(V_{t+1}) \leq \mathrm{AL}_1(V_t) - 1$. By the prediction strategy and the fact that we are in the realizable setting, if $\hat{y}_t \neq y_t$ then it must be the case that $\hat{y}_t = 1$ but $y_t = 0$. For the sake of contradiction, suppose that $\mathrm{AL}_1(V_{t+1}) = \mathrm{AL}_1(V_t) = d$. Then, there exists an AL tree $\mathcal{T}$ of width 1 and depth d shattered by V_{t+1}. Consider a new AL tree $\mathcal{T}'$ of width 1 where the root node labeled is x_t and the

left subtree of the root node is $\mathcal{T}$. Note that $\mathcal{T}'$ is a width 1 AL tree with depth $d+1$. Since $\hat{y}_t = 1$, there exists a hypothesis $h \in V_t$ such that $h(x_t) = 1$. Moreover, for every hypothesis in $h \in V_{t+1} \subset V_t$, we have that $h(x_t) = 0$. Since $\mathcal{T}$ is shattered by $V_{t+1} \subset V_t$ and $\mathcal{T}$ is the left subtree of the root node in $\mathcal{T}'$, we have that $\mathcal{T}'$ is an AL tree of width 1 and depth $d+1$ shattered by V_t. However, this contradicts our assumption that $\mathrm{AL}_1(V_t) = d$. Thus, it must be the case $\mathrm{AL}_1(V_{t+1}) \leq \mathrm{AL}_1(V_t) - 1$ whenever the algorithm errs, and the algorithm can err at most $\mathrm{AL}_1(\mathcal{H})$ times before $\mathrm{AL}_1(V_t) = 0$. ■

We extend the results of Theorem D.1.1 to hypothesis classes where $\mathrm{L}(\mathcal{H}) = 1$. Note that $\mathrm{AL}_1(\mathcal{H})$ can be much larger than $\mathrm{L}(\mathcal{H})$ even when $\mathrm{L}(\mathcal{H}) = 1$. For example, for the class of singletons $\mathcal{H} = \{x \mapsto \mathbb{1}\{x = a\} : a \in \mathbb{N}\}$, we have that $\mathrm{L}(\mathcal{H}) = 1$ but $\mathrm{AL}_1(\mathcal{H}) = \infty$.

Theorem D.1.2 (Deterministic realizable upper bound for $\mathrm{L}(\mathcal{H}) = 1$)**.** *For any $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ such that $\mathrm{L}(\mathcal{H}) = 1$, there exists a deterministic learner which, under apple tasting feedback, makes at most $1 + 2\sqrt{T}$ mistakes in the realizable setting.*

Proof. We will show that Algorithm 20 makes at most $1 + 2\sqrt{T}$ mistakes in the realizable setting under apple tasting feedback after tuning r.

Let $S = (x_1, h^\star(x_t)), ..., (x_T, h^\star(x_T))$ be the stream observed by the learner, where $h^\star \in \mathcal{H}$ is the optimal hypothesis. As in the proof of Lemma 6.3.5, consider splitting the stream into the following three parts. Let S_1 denote those rounds where $\mathrm{L}(V_t^0) = 0$ but $y_t = 0$. Let S_2 denote the rounds where $\mathrm{L}(V_t^0) = 1$, $\hat{y}_t = 1$, but $y_t = 0$. Finally, let S_3 denote the rounds where $\mathrm{L}(V_t^0) = 1$, $\hat{y}_t = 0$, but $y_t = 1$. The number of mistakes Algorithm 20 makes on the stream S is at most $|S_1| + |S_2| + |S_3|$. We now upper bound each of these terms separately.

Starting with S_1, observe that if $\mathrm{L}(V_t^0) = 0$, then $|V_t^0| \leq 1$. Thus, if $y_t = 0$, Algorithm 20 correctly identifies the hypothesis labeling the data stream and does not make any further mistakes. Accordingly, we have that $|S_1| \leq 1$.

Next, $|S_2|$ is at most the number of times that Algorithm 20 predicts 1 when $\mathrm{L}(V_t^0) = 1$. Note that if $\mathrm{L}(V_t^0) = 1$ then $|V_t^1| \leq 1$. Thus, by the end of the game, there can be at most $\frac{|\{t:\mathrm{L}(V_t^0)=1\}|}{r}$ hypothesis $h \in \mathcal{H}$ such that $C(h) \geq r$. Since Algorithm 20 only predicts 1 when there exists a hypothesis in V_t^1 with count at least r, we have that $|S_2| \leq \frac{|\{t:\mathrm{L}(V_t^0)=1\}|}{r} \leq \frac{T}{r}$.

Finally, we claim that $|S_3| \leq r$. Suppose for the sake of contradiction that $|S_3| \geq r + 1$. Then, by definition, there exists $r+1$ rounds where $\mathrm{L}(V_t^0) = 1$, $\hat{y}_t = 0$ but $y_t = 1$. However, if $\mathrm{L}(V_t^0) = 1$ and $y_t = 1$, then $V_t^1 = \{h^\star\}$. Therefore, on the $r+1$'th round where $\mathrm{L}(V_t^0) = 1$, $\hat{y}_t = 0$, and $y_t = 1$, it must be the case $C(h^\star) \geq r$. However, if this were true, then the Algorithm would have predicted $\hat{y}_t = 1$ on the $r+1$'th round, a contradiction. Thus, it must be the case that $|S_3| \leq r$.

Algorithm 20 Deterministic Realizable Algorithm For Apple Tasting

Require: $V_1 = \mathcal{H}$ and $r > 0$

Initialize: $C(h) = 0$ for all $h \in \mathcal{H}$
for $t = 1, ..., T$ **do**
    Receive example x_t
    For each $y \in \{0, 1\}$, define $V_t^y = \{h \in V_t \mid h(x_t) = y\}$.
    if $V_t(x_t) = \{y\}$ **then**
        Predict $\hat{y}_t = y$
    else if $\mathrm{L}(V_t^0) = 0$ **then**
        Predict $\hat{y}_t = 1$
        Observe true label y_t
        Update $V_{t+1} = V_t^{y_t}$
    else if $\exists h \in V_t^1$ such that $C(h) \geq r$ **then**
        Predict $\hat{y}_t = 1$
        Observe true label y_t
        Update $V_{t+1} = V_t^{y_t}$
    else
        Predict $\hat{y}_t = 0$
        for $h \in V_t^1$ **do**
            Update $C(h)$ += 1
        end for
      Set $V_{t+1} = V_t$
    end if
end for

Putting it all together, Algorithm 20 makes at most $1 + \frac{T}{r} + r$ mistakes. Picking $r = \sqrt{T}$, gives the mistake bound $1 + 2\sqrt{T}$, completing the proof. ■

We highlight that Theorem D.1.2 is tight up to constants factors. Indeed, for the class $\mathcal{H}$ of singletons over $\mathbb{N}$, we have that $\mathrm{W}(\mathcal{H}) = 2$. Therefore, Theorem 6.3.1 implies the lower bound of $\frac{\sqrt{T}}{8}$.

D.2 Missing Proofs

D.2.1 Proof of Lemma 6.2.1

To see (i), observe that given any shattered AL tree $\mathcal{T}$ of depth d and width $w_2 > w_1$, we can truncate paths with more than w_1 ones to get a shattered AL tree $\mathcal{T}'$ of the same depth where now every path has at most w_1 ones and the right most path has exactly w_1 ones.

To see (ii), consider the case where $w \leq \mathrm{L}(\mathcal{H})$. Then, by property (i), we have that $\mathrm{AL}_w(\mathcal{H}) \geq \mathrm{AL}_{\mathrm{L}(\mathcal{H})}(\mathcal{H}) \geq \mathrm{L}(\mathcal{H}) \geq w$. If $w > \mathrm{L}(\mathcal{H})$, then $\mathrm{AL}_w(\mathcal{H}) \geq \mathrm{L}(\mathcal{H})$ which follows from the fact that an AL tree $\mathcal{T}$ of width w and depth $\mathrm{L}(\mathcal{H}) < w$ is a complete binary tree of depth $\mathrm{L}(\mathcal{H})$.

To see (iii), fix $w \geq \mathrm{L}(\mathcal{H}) + 1$. Then, by property (ii), we have that $\mathrm{AL}_w(\mathcal{H}) \geq \mathrm{L}(\mathcal{H})$. Thus, it suffices to show that $\mathrm{AL}_w(\mathcal{H}) \leq \mathrm{L}(\mathcal{H})$. Suppose for the sake of contradiction that $\mathrm{AL}_w(\mathcal{H}) \geq \mathrm{L}(\mathcal{H}) + 1$. Then, using property (i) and the fact that $w \geq \mathrm{L}(\mathcal{H}) + 1$, we have that $\mathrm{AL}_{\mathrm{L}(\mathcal{H})+1}(\mathcal{H}) \geq \mathrm{AL}_w(\mathcal{H}) \geq \mathrm{L}(\mathcal{H}) + 1$. Thus, by definition of ALdim, there exists a Littlestone tree of depth $\mathrm{L}(\mathcal{H}) + 1$ shattered by $\mathcal{H}$, a contradiction.

To see (iv), note that when $\mathrm{L}(\mathcal{H}) < \infty$, we have that $\mathrm{AL}_{\mathrm{L}(\mathcal{H})+1}(\mathcal{H}) = \mathrm{L}(\mathcal{H})$ by property (iii). Thus, by definition of the Effective width, it must be the case that $\mathrm{W}(\mathcal{H}) \leq \mathrm{L}(\mathcal{H}) + 1$.

To see (v), it suffices to prove that $\mathrm{L}(\mathcal{H}) = \infty \implies \mathrm{W}(\mathcal{H}) = \infty$ since (iv) shows that $\mathrm{L}(\mathcal{H}) < \infty \implies \mathrm{W}(\mathcal{H}) < \infty$. This is true because if $\mathrm{L}(\mathcal{H}) = \infty$, then for any width $w \in \mathbb{N}$ and depth $d \in \mathbb{N}$, one can always prune a shattered Littlestone tree of depth d to get a shattered AL tree of depth d and width w.

D.2.2 Proof of Lemma 6.3.5

If $T \leq M_-$, then the claimed expected mistake bound is $\geq T$, which trivially holds for any algorithm. So, we only consider the case when $T > M_-$. Let $\mathcal{A}$ be a deterministic online learner, which makes at most M_- false negative mistakes and at most M_+ false positive mistakes under full-information feedback. We now show that Algorithm 21, a randomized

Algorithm 21 Conversion of Full-Information Algorithm to Apple Tasting Algorithm

Require: Full-Information Algorithm $\mathcal{A}$, false negative mistake bound M_- of $\mathcal{A}$

1: **for** $t = 1, ..., T$ **do**

2: Receive x_t and query $\mathcal{A}$ to get $\xi_t = \mathcal{A}(x_t)$.

3: Draw $r \sim \text{Unif}([0, 1])$ and predict

$$\hat{y}_t = \begin{cases} 1 & \text{if } \xi_t = 1. \\ 1 & \text{if } \xi_t = 0 \text{ and } r \leq \sqrt{M_-/T}. \\ 0 & \text{otherwise.} \end{cases}$$

4: If $\hat{y}_t = 1$, receive y_t and update $\mathcal{A}$ by passing (x_t, y_t).

5: **end for**

algorithm that uses $\mathcal{A}$ in a black-box fashion, has expected mistake bound at most $M_+ + 2\sqrt{TM_-}$ in the realizable setting under apple-tasting feedback.

For each bitstring $b \in \{0, 1\}^3$, define $S_b = \{t \in [T] \mid b_1 = \xi_t, b_2 = \hat{y}_t, \text{and } b_3 = y_t\}$. Here, b_1, b_2, b_3 are the first, second, and third bits of the bitstring b. Using this notation, we can write the expected mistake bound of Algorithm 21 as

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\hat{y}_t \neq y_t\}\right] = \mathbb{E}\left[\, |S_{101}| + |S_{001}| + |S_{110}| + |S_{010}| \,\right].$$

Since $\hat{y}_t = 1$ whenever $\xi_t = 1$, we have $|S_{101}| = 0$. Note that $|S_{001}| \leq N$, where N is the number of failures before M_- successes in independent Bernoulli trials with probability $\sqrt{M_-/T}$ of success. That is, N quantifies the number of rounds before ξ_t is flipped M_- number of times from 0 to 1 in rounds when $y_t = 1$. Recalling that $N \sim \text{Negative-Binomial}(M_-, \sqrt{M_-/T})$, we have

$$\mathbb{E}[|S_{001}|] \leq \mathbb{E}[N] \leq M_- \left(\sqrt{\frac{T}{M_-}} - 1\right) \leq \sqrt{M_- T} - M_-.$$

Moreover, using the fact that $\mathcal{A}$ makes at most M_+ false positive mistakes, we have $|S_{110}| \leq M_+$.

Finally, using the prediction rule in Algorithm 21, we have

$$\mathbb{E}[|S_{010}|] \leq \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\xi_t = 0 \text{ and } \hat{y}_t = 1\}\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\left\{r \leq \sqrt{\frac{M_-}{T}}\right\}\right] \leq T\sqrt{\frac{M_-}{T}} = \sqrt{M_- T}.$$

Putting everything together, we have

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\hat{y}_t \neq y_t\}\right] \leq \sqrt{M_- T} - M_- + M_+ + \sqrt{M_- T} \leq M_+ + 2\sqrt{M_- T}.$$

This completes our proof.

D.2.3 Proof of Lemma 6.4.3

Observe that $\hat{\ell}_t(y) \leq \frac{1}{\eta}$ for $y \in \{0,1\}$ since $p_t^1 \geq \eta$. Let $\bar{\ell}_t = \sum_{y\in\{0,1\}} p_t^y \hat{\ell}_t(y)$ and define ℓ'_t such that $\ell'_t(y) = \hat{\ell}_t(y) - \bar{\ell}_t$ for all $y \in \{0,1\}$. Notice that executing EXP4.AT on the loss vectors $\hat{\ell}_1, ..., \hat{\ell}_T$ is equivalent to executing EXP4.AT on the loss vectors $\ell'_1, ..., \ell'_T$. Indeed, since $\bar{\ell}_t$ is constant over the experts, the weights q_t^i remained unchanged regardless of whether ℓ'_t or $\hat{\ell}_t$ is used to update the experts. Moreover, we have that $\ell'_t(y) \geq -\frac{1}{\eta}$.

We start by following the standard analysis of exponential weighting schemes. Let $w_1^i = 1$, $w_{t+1}^i = w_t^i \exp(-\eta \ell'_t(\mathcal{E}_t^i))$, and $W_t = \sum_{i=1}^N w_t^i$. Then, $q_t^i = \frac{w_t^i}{W_t}$ and we have

$$\begin{aligned}
\frac{W_{t+1}}{W_t} &= \sum_{i=1}^{N} \frac{w_{t+1}^i}{W_t} \\
&= \sum_{i=1}^{N} \frac{w_t^i \exp(-\eta \ell'_t(\mathcal{E}_t^i))}{W_t} \\
&= \sum_{i=1}^{N} q_t^i \exp(-\eta \ell'_t(\mathcal{E}_t^i)) \\
&\leq \sum_{i=1}^{N} q_t^i \left(1 - \eta(\ell'_t(\mathcal{E}_t^i)) + \eta^2 (\ell'_t(\mathcal{E}_t^i))^2\right) \\
&= 1 - \eta \sum_{i=1}^{N} q_t^i \ell'_t(\mathcal{E}_t^i) + \eta^2 \sum_{i=1}^{N} q_t^i (\ell'_t(\mathcal{E}_t^i))^2,
\end{aligned}$$

where the inequality follows from the fact that $\ell'_t(\mathcal{E}_t^i) \geq -\frac{1}{\eta}$ and $e^x \leq 1 + x + x^2$ for all $x \leq 1$. Taking logarithms, summing over t, and using the fact that $\ln(1-x) \leq -x$ for all $x \geq 0$ we get

$$\ln \frac{W_{T+1}}{W_1} \leq -\eta \sum_{t=1}^{T} \sum_{i=1}^{N} q_t^i \ell'_t(\mathcal{E}_t^i) + \eta^2 \sum_{t=1}^{T} \sum_{i=1}^{N} q_t^i (\ell'_t(\mathcal{E}_t^i))^2.$$

Also, for any expert $j \in [N]$, we have

$$\ln \frac{W_{T+1}}{W_1} \geq \ln \frac{w_{T+1}^j}{W_1} = -\eta \sum_{t=1}^{T} \ell_t'(\mathcal{E}_t^j)) - \ln N.$$

Combining this with the upper bound on $\ln \frac{W_{T+1}}{W_1}$, rearranging, and dividing by η, we get

$$\sum_{t=1}^{T} \sum_{i=1}^{N} q_t^i \ell_t'(\mathcal{E}_t^i) \leq \sum_{t=1}^{T} \ell_t'(\mathcal{E}_t^j) + \frac{\ln N}{\eta} + \eta \sum_{t=1}^{T} \sum_{i=1}^{N} q_t^i (\ell_t'(\mathcal{E}_t^i))^2.$$

Using the definition of ℓ_t', we further have that

$$\sum_{t=1}^{T} \sum_{i=1}^{N} q_t^i \hat{\ell}_t(\mathcal{E}_t^i) \leq \sum_{t=1}^{T} \hat{\ell}_t(\mathcal{E}_t^j) + \frac{\ln N}{\eta} + \eta \sum_{t=1}^{T} \sum_{i=1}^{N} q_t^i (\ell_t'(\mathcal{E}_t^i))^2.$$

Next, observe that

$$\sum_{i=1}^{N} q_t^i \hat{\ell}_t(\mathcal{E}_t^i) = \left(\sum_{i=1}^{N} q_t^i \mathcal{E}_t^i \right) \hat{\ell}_t(1) + \left(1 - \sum_{i=1}^{N} q_t^i \mathcal{E}_t^i \right) \hat{\ell}_t(0) = \frac{1}{1-\eta} \sum_{y \in \{0,1\}} p_t^y \hat{\ell}_t(y) - \frac{\eta}{1-\eta} \hat{\ell}_t(1).$$

Moreover,

$$\begin{aligned}
\sum_{i=1}^{N} q_t^i (\ell_t'(\mathcal{E}_t^i))^2 &= \sum_{i=1}^{N} q_t^i \left(\sum_{y \in \{0,1\}} \mathbb{1}\{y = \mathcal{E}_t^i\} \ell_t'(y) \right)^2 \\
&= \sum_{i=1}^{N} q_t^i \left(\sum_{y \in \{0,1\}} \mathbb{1}\{y = \mathcal{E}_t^i\} \ell_t'(y)^2 \right) \\
&= \sum_{y \in \{0,1\}} \left(\sum_{i=1}^{N} q_t^i \mathbb{1}\{y = \mathcal{E}_t^i\} \right) \ell_t'(y)^2 \\
&= \left(\sum_{i=1}^{N} q_t^i \mathcal{E}_t^i \right) \ell_t'(1)^2 + \left(1 - \sum_{i=1}^{N} q_t^i \mathcal{E}_t^i \right) \ell_t'(0)^2 \\
&\leq \frac{1}{1-\eta} \sum_{y \in \{0,1\}} p_t^y \ell_t'(y)^2.
\end{aligned}$$

Therefore, for any fixed expert j,

$$\frac{1}{1-\eta} \sum_{t=1}^{T} \sum_{y \in \{0,1\}} p_t^y \hat{\ell}_t(y) - \frac{\eta}{(1-\eta)} \sum_{t=1}^{T} \hat{\ell}_t(1) \leq \sum_{t=1}^{T} \hat{\ell}_t(\mathcal{E}_t^j) + \frac{\ln N}{\eta} + \frac{\eta}{1-\eta} \sum_{t=1}^{T} \sum_{y \in \{0,1\}} p_t^y \ell_t'(y)^2.$$

Multiplying by $1 - \eta$ and rearranging, we have

$$\sum_{t=1}^{T}\sum_{y\in\{0,1\}} p_t^y \hat{\ell}_t(y) - (1-\eta)\sum_{t=1}^{T}\hat{\ell}_t(\mathcal{E}_t^j) \leq \frac{(1-\eta)\ln N}{\eta} + \eta\sum_{t=1}^{T}\hat{\ell}_t(1) + \eta\sum_{t=1}^{T}\sum_{y\in\{0,1\}} p_t^y \ell_t'(y)^2$$

which further implies the guarantee:

$$\begin{aligned}\sum_{t=1}^{T}\sum_{y\in\{0,1\}} p_t^y \hat{\ell}_t(y) - \sum_{t=1}^{T}\hat{\ell}_t(\mathcal{E}_t^j) &\leq \frac{\ln N}{\eta} + \eta\sum_{t=1}^{T}\hat{\ell}_t(1) + \eta\sum_{t=1}^{T}\sum_{y\in\{0,1\}} p_t^y \ell_t'(y)^2 \\ &= \frac{\ln N}{\eta} + \eta\sum_{t=1}^{T}\hat{\ell}_t(1) + \eta\sum_{t=1}^{T}\sum_{y\in\{0,1\}} p_t^y (\hat{\ell}_t(y) - \bar{\ell}_t)^2.\end{aligned}$$

Next, note that

$$\begin{aligned}\sum_{y\in\{0,1\}} p_t^y (\hat{\ell}_t(y) - \bar{\ell}_t)^2 &= \sum_{y\in\{0,1\}} p_t^y \hat{\ell}_t(y)^2 - \left(\sum_{y\in\{0,1\}} p_t^y \hat{\ell}_t(y)\right)^2 \\ &\leq \sum_{y\in\{0,1\}} p_t^y \hat{\ell}_t(y)^2 - \sum_{y\in\{0,1\}} (p_t^y)^2 \hat{\ell}_t(y)^2 \\ &= \sum_{y\in\{0,1\}} p_t^y (1-p_t^y)\hat{\ell}_t(y)^2 \\ &\leq p_t^1(1-p_t^1)\hat{\ell}_t(0)^2 + p_t^1\hat{\ell}_t(1)^2,\end{aligned}$$

where the first inequality is true because of the nonnegativity of the losses $\hat{\ell}_t$ and the last inequality is true because $0 \leq p_t^1 \leq 1$. Putting things together, we have that

$$\sum_{t=1}^{T}\sum_{y\in\{0,1\}} p_t^y \hat{\ell}_t(y) - \sum_{t=1}^{T}\hat{\ell}_t(\mathcal{E}_t^j) \leq \frac{\ln N}{\eta} + \eta\sum_{t=1}^{T}\hat{\ell}_t(1) + \eta\sum_{t=1}^{T} p_t^1(1-p_t^1)\hat{\ell}_t(0)^2 + \eta\sum_{t=1}^{T} p_t^1\hat{\ell}_t(1)^2.$$

Since expert $j \in [N]$ was arbitrary, this completes the proof.

D.2.4 Proof of Theorem 6.4.2

From Lemma 6.4.3, we have that for an fixed expert $j \in [N]$

$$\sum_{t=1}^{T} \sum_{y \in \{0,1\}} p_t^y \hat{\ell}_t(y) - \sum_{t=1}^{T} \hat{\ell}_t(\mathcal{E}_t^j) \leq \frac{\ln N}{\eta} + \eta \sum_{t=1}^{T} \hat{\ell}_t(1) + \eta \sum_{t=1}^{T} p_t^0 p_t^1 \hat{\ell}_t(0)^2 + \eta \sum_{t=1}^{T} p_t^1 \hat{\ell}_t(1)^2.$$

Taking expectations on both sides and using the fact that $\mathbb{E}_t\left[\hat{\ell}_t(y)\right] = \mathbb{1}\{y \neq y_t\}$, $\mathbb{E}_t\left[\hat{\ell}_t(y)^2\right] = \frac{\mathbb{1}\{y \neq y_t\}}{p_t^1}$ gives

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\hat{y}_t \neq y_t\}\right] - \sum_{t=1}^{T} \mathbb{1}\{\mathcal{E}_t^j \neq y_t\} &\leq \frac{\ln N}{\eta} \\
&\quad + \eta \sum_{t=1}^{T} \mathbb{1}\{1 \neq y_t\} \\
&\quad + \eta \sum_{t=1}^{T} \mathbb{E}\left[p_t^0 p_t^1 \frac{\mathbb{1}\{0 \neq y_t\}}{p_t^1} + p_t^1 \frac{\mathbb{1}\{1 \neq y_t\}}{p_t^1}\right] \\
&\leq \frac{\ln N}{\eta} + 2\eta T.
\end{aligned}$$

Substituting $\eta = \sqrt{\frac{\ln N}{2T}}$, we have

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\hat{y}_t \neq y_t\}\right] - \sum_{t=1}^{T} \mathbb{1}\{\mathcal{E}_t^j \neq y_t\} \leq 2\sqrt{2T \ln N} \leq 3\sqrt{T \ln N},$$

which completes the proof.

D.2.5 Proof of Theorem 6.4.1

Let $(x_1, y_1), ..., (x_T, y_T)$ denote the stream of labeled instances to be observed by the agnostic learner and let $h^\star = \arg\min_{h \in \mathcal{H}} \sum_{t=1}^{T} \mathbb{1}\{h(x_t) \neq y_t\}$ be the optimal hypothesis in hindsight. Given the time horizon T, let $L_T = \{L \subset [T] : |L| \leq \mathrm{L}(\mathcal{H})\}$ denote the set of all possible subsets of $[T]$ of size of $\mathrm{L}(\mathcal{H})$. For every $L \in L_T$, define an expert $\mathcal{E}_L$, whose prediction on time point $t \in [T]$ on instance x_t is defined by

$$\mathcal{E}_L(x_t) = \begin{cases} \mathrm{SOA}\left(x_t | \{(x_i, \mathcal{E}_L(x_i))\}_{i=1}^{t-1}\right), & \text{if } t \notin L \\ \neg\mathrm{SOA}\left(x_t | \{(x_i, \mathcal{E}_L(x_i))\}_{i=1}^{t-1}\right), & \text{otherwise} \end{cases}$$

where $\mathrm{SOA}\left(x_t | \{(x_i, \mathcal{E}_L(x_i))\}_{i=1}^{t-1}\right)$ denotes the prediction of the SOA on the instance x_t

after running and updating on the labeled stream $\{(x_i, \mathcal{E}_L(x_i))\}_{i=1}^{t-1}$. Let $E = \{\mathcal{E}_L : L \in L_T\}$ denote the set of all Experts parameterized by subsets $L \in L_T$. Observe that $|E| \leq T^{\mathrm{L}(\mathcal{H})}$.

We claim that there exists an expert $\mathcal{E}_{L^\star} \in E$ such that for all $t \in [T]$, we have that $\mathcal{E}_{L^\star}(x_t) = h^\star(x_t)$. To see this, consider the hypothetical stream of instances labeled by the optimal hypothesis $S^\star = \{(x_t, h^\star(x_t))\}_{t=1}^T$. Let $L^\star = \{t_1, t_2, ...\}$ be the indices on which the SOA would have made mistakes had it run and updated on S^*. By the guarantee of SOA, we know that $|L^\star| \leq \mathrm{L}(\mathcal{H})$. By construction of E, there exists an expert $\mathcal{E}_{L^\star}$ parameterized by $L^\star$. We claim that for all $t \in [T]$, we have that $\mathcal{E}_{L^\star}(x_t) = h^\star(x_t)$. This follows by strong induction on $t \in [T]$. For the base case $t = 1$, there are two subcases to consider. If $1 \in L^\star$, then we have that $\mathcal{E}_{L^\star}(x_1) = \neg\mathrm{SOA}\,(x_1|\{\}) = h^\star(x_1)$, by definition of $L^\star$. If $1 \notin L^\star$, then $\mathcal{E}_{L^\star}(x_1) = \mathrm{SOA}\,(x_1|\{\}) = h^\star(x_1)$ also by definition of $L^\star$. Now for the induction step, suppose that $\mathcal{E}_{L^\star}(x_i) = h^\star(x_i)$ for all $i \leq t$. Then, if $t+1 \in L^\star$, we have that $\mathcal{E}_{L^\star}(x_{t+1}) = \neg\mathrm{SOA}\,(x_{t+1}|\{(x_i, \mathcal{E}_{L^\star}(x_i))\}_{i=1}^t) = \neg\mathrm{SOA}\,(x_{t+1}|\{(x_i, h^\star(x_i))\}_{i=1}^t) = h^\star(x_{t+1})$. If $t+1 \notin L^\star$, then $\mathcal{E}_{L^\star}(x_{t+1}) = \mathrm{SOA}\,(x_{t+1}|\{(x_i, \mathcal{E}_{L^\star}(x_i))\}_{i=1}^t) = \mathrm{SOA}\,(x_{t+1}|\{(x_i, h^\star(x_i))\}_{i=1}^t) = h^\star(x_{t+1})$. The final equality in both cases are due to the definition of $L^\star$.

Now, consider the agnostic online learner $\mathcal{A}$ that runs EXP4.AT using E. By Theorem 6.4.2, we have that

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^T \mathbb{1}\{\mathcal{A}(x_t) \neq y_t\}\right] &\leq \inf_{\mathcal{E}\in E} \sum_{t=1}^T \mathbb{1}\{\mathcal{E}(x_t) \neq y_t\} + 3\sqrt{T \ln |E|} \\
&\leq \sum_{t=1}^T \mathbb{1}\{\mathcal{E}_{L^\star}(x_t) \neq y_t\} + 3\sqrt{\mathrm{L}(\mathcal{H}) T \ln T} \\
&= \sum_{t=1}^T \mathbb{1}\{h^\star(x_t) \neq y_t\} + 3\sqrt{\mathrm{L}(\mathcal{H}) T \ln T}.
\end{aligned}$$

Thus, $\mathcal{A}$ achieves the stated upper bound on expected regret under apple tasting feedback, which completes the proof.

APPENDIX E

On the Learnability of Multilabel Ranking

E.1 Categorizing Popular Ranking Losses

Table E.1: Categorizing Popular Ranking Losses.

Loss	Loss Family
Sum Loss@p	$\mathcal{L}(\ell_{\text{sum}}^{@p})$
Precision Loss@p	$\mathcal{L}(\ell_{\text{prec}}^{@p})$
Average Precision	$\mathcal{L}(\ell_{\text{sum}}^{@K})$
Area Under the Curve	$\mathcal{L}(\ell_{\text{sum}}^{@K})$
Reciprocal Rank	$\mathcal{L}(\ell_{\text{prec}}^{@1})$
Pairwise Rank Loss	$\mathcal{L}(\ell_{\text{sum}}^{@K})$
Discounted Cumulative Loss	$\mathcal{L}(\ell_{\text{sum}}^{@K})$
Discounted Cumulative Loss@p	$\mathcal{L}(\ell_{\text{sum}}^{@p})$

In this section, we show that our loss families $\mathcal{L}(\ell_{\text{sum}}^{@p})$ and $\mathcal{L}(\ell_{\text{prec}}^{@p})$ are general and capture many of the popular ranking loss functions used in practice. We summarize the results in Table E.1.

Recall that

$$\mathcal{L}(\ell_{\text{sum}}^{@p}) = \{\ell \in \mathbb{R}^{\mathcal{S}_K \times \mathcal{Y}} : \ell = 0 \text{ if and only if } \ell_{\text{sum}}^{@p} = 0\} \cap \{\ell \in \mathbb{R}^{\mathcal{S}_K \times \mathcal{Y}} : \pi \overset{[p]}{=} \hat{\pi} \implies \ell(\pi, y) = \ell(\hat{\pi}, y)\},$$

where

$$\ell_{\text{sum}}^{@p}(\pi, y) = \sum_{i=1}^{K} \min(\pi_i, p+1) y^i - Z_y^p.$$

Note that the normalization constant is defined as $Z_y^p := \min_{\pi \in \mathcal{S}_K} \sum_{i=1}^{K} \min(\pi_i, p+1) y^i$ and

thus only depends on y. Furthermore,

$$\mathcal{L}(\ell^{@p}_{\text{prec}}) = \{\ell \in \mathbb{R}^{\mathcal{S}_K \times \mathcal{Y}} : \ell = 0 \text{ if and only if } \ell^{@p}_{\text{prec}} = 0\} \cap \{\ell \in \mathbb{R}^{\mathcal{S}_K \times \mathcal{Y}} : \pi \overset{p}{=} \hat{\pi} \implies \ell(\pi, y) = \ell(\hat{\pi}, y)\},$$

where

$$\ell^{@p}_{\text{prec}}(\pi, y) = Z^p_y - \sum_{i=1}^{K} \mathbb{1}\{\pi_i \leq p\} y^i.$$

As before, the normalization constant $Z^p_y := \max_{\pi \in \mathcal{S}_K} \sum_{i=1}^{K} \mathbb{1}\{\pi_i \leq p\} y^i$ only depends on y.

In ranking literature, many evaluation metrics are often stated in terms of *gain* functions. However, these can be easily converted into loss functions by subtracting the gain from the maximum possible value of the gain. When relevance scores are restricted to be binary (i.e. $\mathcal{Y} = \{0, 1\}^K$), the **Average Precision** (AP) metric is a *gain* function defined as

$$\text{AP}(\pi, y) = \frac{1}{\|y\|_1} \sum_{i \in \{\pi_m : y^m = 1\}} \frac{\sum_{j=1}^{K} \mathbb{1}\{\pi_j \leq i\} y^j}{i}.$$

Since the maximum value AP can take is 1, we can define its loss function variant as:

$$\ell_{\text{AP}}(\pi, y) = 1 - \text{AP}(\pi, y).$$

Note that $\ell_{\text{AP}}(\pi, y) = 0$ if and only if π ranks all labels where $y_i = 1$ in the top $\|y\|_1$. Therefore, $\ell_{\text{AP}}(\pi, y) \in \mathcal{L}(\ell^{@K}_{\text{sum}})$.

Another useful metric for binary relevance feedback is the **Area Under the Curve** (AUC) loss function:

$$\ell_{\text{AUC}}(\pi, y) = \frac{1}{\|y\|_1 (K - \|y\|_1)} \sum_{i=1}^{K} \sum_{j=1}^{K} \mathbb{1}\{\pi_i < \pi_j\} \mathbb{1}\{y^i < y^j\}.$$

The AUC computes the fraction of "bad pairs" of labels (i.e those pairs of labels where i was more relevant than j, but i was ranked lower than j). Again, note that $\ell_{\text{AUC}}(\pi, y) = 0$ if and only if π ranks all labels where $y^i = 1$ in the top $\|y\|_1$. Therefore, $\ell_{\text{AP}}(\pi, y) \in \mathcal{L}(\ell^{@K}_{\text{sum}})$.

Lastly, the **Reciprocal Rank** (RR) metric is another important *gain* function for binary relevance score feedback,

$$\text{RR}(\pi, y) = \frac{1}{\min_{i : y^i = 1} \pi_i}.$$

Its loss equivalent can be written as:

$$\ell_{\mathrm{RR}}(\pi, y) = 1 - \mathrm{RR}(\pi, y).$$

Since $\ell_{\mathrm{RR}}(\pi, y)$ only cares about the relevance of the top-ranked label, we have that $\ell_{\mathrm{RR}}(\pi, y) \in \mathcal{L}(\ell_{\mathrm{prec}}^{@1})$.

Moving onto non-binary relevance scores, we start with the **Pairwise Rank Loss** (PL):

$$\ell_{\mathrm{PL}}(\pi, y) = \sum_{i=1}^{K} \sum_{j=1}^{K} \mathbb{1}\{\pi_i < \pi_j\} \mathbb{1}\{y^i < y^j\}.$$

The Pairwise Ranking loss is the analog of AUC for non-binary relevance scores and thus $\ell_{\mathrm{PL}}(\pi, y) \in \mathcal{L}(\ell_{\mathrm{sum}}^{@K})$.

Finally, we have the **Discounted Cumulative Gain** (DCG) metric, defined as:

$$\mathrm{DCG}(\pi, y) = \sum_{i=1}^{K} \frac{2^{y^i} - 1}{\log_2(1 + \pi_i)}.$$

For an appropriately chosen normalizing constant Z_y, we can define its associated loss:

$$\ell_{\mathrm{DCG}}(\pi, y) = Z_y - \mathrm{DCG}(\pi, y).$$

Like $\ell_{\mathrm{sum}}^{@K}$, $\ell_{\mathrm{DCG}}(\pi, y)$ is 0 if and only if π ranks the K labels in increasing order of relevance, breaking ties arbitrarily. Thus, $\ell_{\mathrm{DCG}}(\pi, y) \in \mathcal{L}(\ell_{\mathrm{sum}}^{@K})$. If one only cares about the top-p ranked results, then the DCG@p loss function evaluates only the top-p ranked labels:

$$\ell_{\mathrm{DCG}}^{@p}(\pi, y) = Z_y^p - \sum_{i=1}^{K} \frac{2^{y^i} - 1}{\log_2(1 + \pi_i)} \mathbb{1}\{\pi_i \leq p\} = Z_y^p - \mathrm{DCG}^{@p}(\pi, y).$$

Analogously, we have that $\ell_{\mathrm{DCG}}^{@p}(\pi, y) \in \mathcal{L}(\ell_{\mathrm{sum}}^{@p})$.

E.2 Agnostic PAC Learnability of Score-based Rankers

In this section, we apply our results in Chapter 7 to give sufficient conditions for the agnostic PAC learnability of score-based ranking hypothesis classes. A score-based ranking hypothesis $h : \mathcal{X} \to \mathcal{S}_K$ first maps an input $x \in \mathcal{X}$ to a vector in $\mathbb{R}^K$ representing the "score" for each

label. Then, it outputs a ranking (permutation) over the labels in $[K]$ by sorting the real-valued vector in decreasing order of score.

More formally, let $\mathcal{F} \subseteq (\mathbb{R}^K)^{\mathcal{X}}$ denote a set of functions mapping elements from the input space $\mathcal{X}$ to score-vectors in $\mathbb{R}^K$. For each $f \in \mathcal{F}$, define the score-based ranking hypothesis $h_f(x) = \text{argsort}(f(x))$ which first computes the score-vector $f(x) \in \mathbb{R}^K$, and then outputs a ranking by sorting $f(x)$ in decreasing order, breaking ties by giving the smaller label the higher rank. That is, if $f_1(x) = f_2(x)$, then label 1 will be ranked higher than label 2. Given $\mathcal{F}$, define its induced score-based ranking hypothesis class as $\mathcal{H} = \{h_f : f \in \mathcal{F}\}$. Since our characterization of ranking learnability relates the learnability of $\mathcal{H}$ to the learnability of the *binary* threshold-restricted classes $\mathcal{H}_i^j = \{h_i^j : h \in \mathcal{H}\}$, it suffices to consider an arbitrary threshold-restricted class $\mathcal{H}_i^j$ and bound its VC dimension. Before we do so, we need some more notation regarding $\mathcal{F}$.

For each $k \in [K]$, define the scalar-valued function class $\mathcal{F}_k = \{f_k \mid (f_1, \ldots, f_K) \in \mathcal{F}\}$ by restricting each function in $\mathcal{F}$ to its k^{th} coordinate output. Here, each $\mathcal{F}_k \subseteq \mathbb{R}^{\mathcal{X}}$ and we can write $\mathcal{F} = (\mathcal{F}_1, \ldots, \mathcal{F}_K)$. For a function $f \in \mathcal{F}$, we will use $f_k(x)$ to denote the k^{th} coordinate output of $f(x)$. For every $(i, j) \in [K] \times [K]$, define the function class $\mathcal{F}_i - \mathcal{F}_j = \{f_i - f_j : f \in \mathcal{F}\}$ where we let $f_i - f_j : \mathcal{X} \to \mathbb{R}$ denote a function such that $(f_i - f_j)(x) = f_i(x) - f_j(x)$. Subsequently, for any $(i, j) \in [K] \times [K]$, define the *binary* hypothesis classes $\mathcal{G}_{i,j} = \{\mathbb{1}\{(f_i - f_j)(x) < 0\} : f_i - f_j \in \mathcal{F}_i - \mathcal{F}_j\}$ and $\tilde{\mathcal{G}}_{i,j} = \{\mathbb{1}\{(f_i - f_j)(x) \leq 0\} : f_i - f_j \in \mathcal{F}_i - \mathcal{F}_j\}$. Finally, let $C_j : \{0,1\}^K \to \{0,1\}$ be the K-wise composition s.t. $C_j(b) = \mathbb{1}\{\sum_{i=1}^K b_i \leq j\}$ and define $C_j(\mathcal{G}_1, ..., \mathcal{G}_K) = \{C_j(g_1, ..., g_K) : (g_1, ..., g_K) \in \mathcal{G}_1 \times ... \times \mathcal{G}_K\}$. In other words, $C_j(\mathcal{G}_1, ..., \mathcal{G}_K)$ is the *binary* hypothesis class constructed by taking all combinations of binary classifiers from $\mathcal{G}_1, ..., \mathcal{G}_K$, summing them up, and thresholding the sum at j. We are now ready to bound the VC dimension of an arbitrary threshold-restricted class $\mathcal{H}_i^j$.

Consider an arbitrary threshold-restricted class $\mathcal{H}_i^j$ and hypothesis $h \in \mathcal{H}$. By definition, $h_i^j \in \mathcal{H}_i^j$. Let $f \in \mathcal{F}$ denote the function associated with h. Given an instance $x \in \mathcal{X}$, recall that $h_i^j(x) = \mathbb{1}\{h_i(x) \leq j\}$ where $h_i(x)$ is the rank that h gives to the label i for instance x. Since $h(x) = \text{argsort}(f(x))$, we have

$$
\begin{aligned}
h_i(x) &= \text{argsort}(f(x))[i] \\
&= \sum_{m=1}^{i} \mathbb{1}\{f_i(x) \leq f_m(x)\} + \sum_{m=i+1}^{K} \mathbb{1}\{f_i(x) < f_m(x)\} \\
&= \sum_{m=1}^{i} \mathbb{1}\{(f_i - f_m)(x) \leq 0\} + \sum_{m=i+1}^{K} \mathbb{1}\{(f_i - f_m)(x) < 0\}
\end{aligned}
$$

Thus, we can write:

$$h_i^j(x) = \mathbb{1}\left\{\left(\sum_{m=1}^{i} \mathbb{1}\{(f_i - f_m)(x) \leq 0\} + \sum_{m=i+1}^{K} \mathbb{1}\{(f_i - f_m)(x) < 0\}\right) \leq j\right\}.$$

Note that $h_i^j \in C_j(\tilde{\mathcal{G}}_{i,1}, ..., \tilde{\mathcal{G}}_{i,i}, \mathcal{G}_{i,i+1}, ..., \mathcal{G}_{i,K})$ by construction. Since h, and therefore h_i^j, was arbitrary, it further follows that $\mathcal{H}_i^j \subseteq C_j(\tilde{\mathcal{G}}_{i,1}, ..., \tilde{\mathcal{G}}_{i,i}, \mathcal{G}_{i,i+1}, ..., \mathcal{G}_{i,K})$. Therefore,

$$\text{VC}(\mathcal{H}_i^j) \leq \text{VC}(C_j(\tilde{\mathcal{G}}_{i,1}, ..., \tilde{\mathcal{G}}_{i,i}, \mathcal{G}_{i,i+1}, ..., \mathcal{G}_{i,K})).$$

Since $C_j(\tilde{\mathcal{G}}_{i,1}, ..., \tilde{\mathcal{G}}_{i,i}, \mathcal{G}_{i,i+1}, ..., \mathcal{G}_{i,K})$ is some K-wise composition of binary classes $\tilde{\mathcal{G}}_{i,1}, ..., \tilde{\mathcal{G}}_{i,i}, \mathcal{G}_{i,i+1}, ..., \mathcal{G}_{i,K}$, standard VC composition guarantees that $\text{VC}(C_j(\tilde{\mathcal{G}}_{i,1}, ..., \tilde{\mathcal{G}}_{i,i}, \mathcal{G}_{i,i+1}, ..., \mathcal{G}_{i,K})) = \tilde{O}(\text{VC}(\tilde{\mathcal{G}}_{i,1}) + ... + \text{VC}(\tilde{\mathcal{G}}_{i,i}) + \text{VC}(\mathcal{G}_{i,i+1}) + ... + \text{VC}(\mathcal{G}_{i,K}))$, where we hide log factors of K and the VC dimensions [Dudley, 1978, Alon et al., 2020]. Putting things together, we have that

$$\text{VC}(\mathcal{H}_i^j) \leq \tilde{O}(\text{VC}(\tilde{\mathcal{G}}_{i,1}) + ... + \text{VC}(\tilde{\mathcal{G}}_{i,i}) + \text{VC}(\mathcal{G}_{i,i+1}) + ... + \text{VC}(\mathcal{G}_{i,K})).$$

An identical analysis can also be used to give sufficient conditions for the *online* learnability of score-based rankers in terms of the Littlestone dimensions of $\mathcal{H}_j^i$.

Now, we consider the special class of *linear* score-based ranker and prove Lemma 7.4.6.

Proof. (of Lemma 7.4.6) Let $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{F} = \{f_W : W \in \mathbb{R}^{K \times d}\}$ s.t. $f_W(x) = Wx$. Consider the class of linear score-based rankers $\mathcal{H} = \{h_{f_W} : f_W \in \mathcal{F}\}$ where $h_{f_W}(x) = \text{argsort}(f_W(x)) = \text{argsort}(Wx)$ breaking ties in the same way mentioned above. Note for all $i \in [K]$, $\mathcal{F}_i = \{f_w : w \in \mathbb{R}^d\}$ where $f_w(x) = w^T x$. Furthermore, $\mathcal{F}_i - \mathcal{F}_j = \mathcal{F}_i = \mathcal{F}_j$. Therefore, for any $(i, j) \in [K] \times [K]$,

$$\mathcal{G}_{i,j} = \{\mathbb{1}\{(f_i - f_j)(x) < 0\} : f_i - f_j \in \mathcal{F}_i - \mathcal{F}_j\} = \{\mathbb{1}\{f_w(x) < 0\} : w \in \mathbb{R}^d\}$$

and

$$\tilde{\mathcal{G}}_{i,j} = \{\mathbb{1}\{(f_i - f_j)(x) \leq 0\} : f_i - f_j \in \mathcal{F}_i - \mathcal{F}_j\} = \{\mathbb{1}\{f_w(x) \leq 0\} : w \in \mathbb{R}^d\}$$

are the set of half-space classifiers passing through the origin with dimension d. Since for all $(i, j) \in [K] \times [K]$, $\text{VC}(\tilde{\mathcal{G}}_{i,j}) = \text{VC}(\mathcal{G}_{i,j}) = d$, we get that $\text{VC}(\mathcal{H}_i^j) \leq \tilde{O}(Kd)$. ■

E.3 Proofs for Batch Multilabel Ranking

Since many of the ranking losses we consider map to values in $\mathbb{R}$, the *empirical* Rademacher complexity will be a useful tool for proving learnability in the batch setting.

Definition E.3.1 (Empirical Rademacher Complexity of Loss Class)**.** *Let $\ell(\cdot,\cdot)$ be a loss function, $S = \{(x_1, y_1), ..., (x_n, y_n)\} \in (\mathcal{X} \times \mathcal{Y})^*$ be a set of examples, and $\ell \circ \mathcal{H} = \{(x, y) \mapsto \ell(h(x), y) : h \in \mathcal{H}\}$ be a loss class. The empirical Rademacher complexity of $\ell \circ \mathcal{H}$ is defined as*

$$\hat{\mathfrak{R}}_n(\ell \circ \mathcal{H}) = \mathbb{E}_\sigma \left[\sup_{h \in \mathcal{H}} \left(\frac{1}{n} \sum_{i=1}^{n} \sigma_i \ell(h(x_i), y_i) \right) \right]$$

where $\sigma_1, ..., \sigma_n$ are independent Rademacher random variables.

In particular, a standard result relates the empirical Rademacher complexity to the generalization error of hypotheses in $\mathcal{H}$ with respect to a real-valued bounded loss function $\ell(h(x), y)$ [Bartlett and Mendelson, 2002].

Proposition E.3.1 (Rademacher-based Uniform Convergence)**.** *Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$ and $\ell(\cdot,\cdot) \leq c$ be a bounded loss function. With probability at least $1 - \delta$ over the sample $S \sim \mathcal{D}^n$, for all $h \in \mathcal{H}$ simultaneously,*

$$\left| \mathbb{E}_{\mathcal{D}}[\ell(h(x), y)] - \hat{\mathbb{E}}_S[\ell(h(x), y)] \right| \leq 2\hat{\mathfrak{R}}_n(\mathcal{F}) + O\left(c\sqrt{\frac{\ln(\frac{1}{\delta})}{n}} \right)$$

where $\hat{\mathbb{E}}_S[\ell(h(x), y)] = \frac{1}{|S|} \sum_{(x,y) \in S} \ell(h(x), y)$ is the empirical average of the loss over S.

When the empirical Rademacher complexity of the loss class $\ell \circ \mathcal{H} = \{(x, y) \mapsto \ell(h(x), y) : h \in \mathcal{H}\}$ is $o(1)$, we state that $\mathcal{H}$ enjoys the uniform convergence property w.r.t ℓ. If $\mathcal{H}$ enjoys the uniform convergence property w.r.t. a loss ℓ, a standard result shows that $\mathcal{H}$ is learnable according to Definition 7.2.1 via Empirical Risk Minimization (ERM) (Theorem 26.5 in Shalev-Shwartz and Ben-David [2014a]).

E.3.1 Proof of Lemma 7.4.3

Proof. Let $\mathcal{H} \subseteq \mathcal{S}_K^{\mathcal{X}}$ be an arbitrary ranking hypothesis class. We need to show that if $\mathcal{H}_i^j$ is agnostic PAC learnable w.r.t to 0-1 loss for all $(i, j) \in [K] \times [p]$, then ERM is an agnostic PAC learnable w.r.t $\ell_{\text{sum}}^{@p}$. By Proposition E.3.1, it suffices to show that the empirical Rademacher complexity of the loss class $\ell_{\text{sum}}^{@p} \circ \mathcal{H}$ vanishes as n increases. This will imply that $\ell_{\text{sum}}^{@p}$ enjoys

the uniform convergence property, and therefore ERM is an agnostic PAC learner for $\mathcal{H}$ w.r.t $\ell_{\text{sum}}^{@p}$. By definition, we have that

$$\begin{aligned}
\hat{\mathfrak{R}}_n(\ell_{\text{sum}}^{@p} \circ \mathcal{H}) &= \mathbb{E}_{\sigma\sim\{\pm1\}^n}\left[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \ell_{\text{sum}}^{@p}(h(x_i), y_i))\right] \\
&= \mathbb{E}_{\sigma\sim\{\pm1\}^n}\left[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \left(\sum_{m=1}^{K} \sigma_i \min(h_m(x_i), p+1) y_i^m - \sigma_i Z_{y_i}^p\right)\right] \\
&= \mathbb{E}_{\sigma\sim\{\pm1\}^n}\left[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n}\sum_{m=1}^{K} \sigma_i \min(h_m(x_i), p+1) y_i^m\right] \\
&\leq \sum_{m=1}^{K} \mathbb{E}_{\sigma\sim\{\pm1\}^n}\left[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \min(h_m(x_i), p+1) y_i^m\right] \\
&\leq B\sum_{m=1}^{K} \mathbb{E}_{\sigma\sim\{\pm1\}^n}\left[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \min(h_m(x_i), p+1)\right]
\end{aligned}$$

where the second inequality follows from the fact that $y_i^m \leq B$ and Talagrand's Contraction Lemma Ledoux and Talagrand [1991].

Next note that $\min(h_m(x_i), p+1) = (p+1) - \sum_{j=1}^{p} \mathbb{1}\{h_m(x_i) \leq j\} = (p+1) - \sum_{j=1}^{p} h_m^j(x_i)$. Substituting and getting rid of constant factors, we have that

$$\begin{aligned}
\hat{\mathfrak{R}}_n(\ell_{\text{sum}}^{@p} \circ \mathcal{H}) &\leq B\sum_{m=1}^{K} \mathbb{E}_{\sigma\sim\{\pm1\}^n}\left[\sup_{h_m\in\mathcal{H}_m} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \sum_{j=1}^{p} h_m^j(x_i)\right] \\
&\leq B\sum_{m=1}^{K}\sum_{j=1}^{p} \mathbb{E}_{\sigma\sim\{\pm1\}^n}\left[\sup_{h_m\in\mathcal{H}_m} \frac{1}{n}\sum_{i=1}^{n} \sigma_i h_m^j(x_i)\right] \\
&= B\sum_{m=1}^{K}\sum_{j=1}^{p} \hat{\mathfrak{R}}_n(\mathcal{H}_m^j).
\end{aligned}$$

Since for $\mathcal{H}_m^j$ is agnostic PAC learnable w.r.t 0-1 loss, by Theorem 6.5 in Shalev-Shwartz and Ben-David [2014a], $\lim_{n\to\infty} \hat{\mathfrak{R}}_n(\mathcal{H}_m^j) = 0$. Since p, K and B are finite,

$$\lim_{n\to\infty} \hat{\mathfrak{R}}_n(\ell_{\text{sum}}^{@p} \circ \mathcal{H}) = \lim_{n\to\infty} B\sum_{m=1}^{K}\sum_{j=1}^{p} \hat{\mathfrak{R}}_n(\mathcal{H}_m^j) = 0$$

.

By Proposition E.3.1, this implies that $\ell_{\text{sum}}^{@p}$ enjoys the uniform convergence property, and therefore ERM using $\ell_{\text{sum}}^{@p}$ is an agnostic PAC learner for $\mathcal{H}$. ■

E.3.2 Proof of Lemma 7.4.5

Proof. Fix $\ell \in \mathcal{L}(\ell_{\text{sum}}^{@p})$ and $(i,j) \in [K] \times [p]$. Let $a = \min_{\pi,y}\{\ell(\pi,y) \mid \ell(\pi,y) \neq 0\}$. Let $\mathcal{H}$ be an arbitrary ranking hypothesis class and $\mathcal{A}$ be an agnostic PAC learner for $\mathcal{H}$ w.r.t ℓ. Our goal will be to use $\mathcal{A}$ to construct an agnostic PAC learner for $\mathcal{H}_i^j$.

Let $\mathcal{D}$ be distribution over $\mathcal{X} \times \{0,1\}$ and $h_i^{\star,j} = \arg\min_{h_i^j \in \mathcal{H}_i^j} \mathbb{E}_{\mathcal{D}}\left[\mathbb{1}\{h_i^j(x) \neq y\}\right]$ be the optimal hypothesis. Let $h^\star \in \mathcal{H}$ be any valid completion of $h_i^{\star,j}$. Our goal will be to show that Algorithm 22 is an agnostic PAC learner for $\mathcal{H}_i^j$ w.r.t 0-1 loss.

Algorithm 22 Agnostic PAC learner for $\mathcal{H}_i^j$ w.r.t. 0-1 loss

Require: Agnostic PAC learner $\mathcal{A}$ for $\mathcal{H}$ w.r.t ℓ, unlabeled samples $S_U \sim \mathcal{D}_{\mathcal{X}}^n$, and labeled samples $S_L \sim \mathcal{D}^m$

1: For each $h \in \mathcal{H}_{|S_U}$, construct a dataset

$$S_U^h = \{(x_1, \tilde{y}_1), ..., (x_n, \tilde{y}_n)\} \text{ s.t. } \tilde{y}_i = \text{BinRel}(h(x_i), j)$$

2: Run $\mathcal{A}$ over all datasets to get $C(S_U) := \left\{\mathcal{A}\big(S_U^h\big) \mid h \in \mathcal{H}_{|S_U}\right\}$

3: Define $C_i^j(S_U) = \{g_i^j | g \in C(S_U)\}$

4: Return $\hat{g}_i^j \in C_i^j(S_U)$ with the lowest empirical error over S_L w.r.t. 0-1 loss.

Consider the sample $S_U^{h^\star}$ and let $g = \mathcal{A}(S_U^{h^\star})$. We can think of g as the output of $\mathcal{A}$ run over an i.i.d sample S drawn from $\mathcal{D}^\star$, a joint distribution over $\mathcal{X} \times \mathcal{Y}$ defined procedurally by first sampling $x \sim \mathcal{D}_{\mathcal{X}}$ and then outputting the labeled sample $(x, \text{BinRel}(h^\star(x), j))$. Note that $\mathcal{D}^\star$ is a realizable distribution (realized by $h^\star$) w.r.t $\ell_{\text{sum}}^{@p}$ and therefore also ℓ. Let $m_{\mathcal{A}}(\varepsilon, \delta, K)$ be the sample complexity of $\mathcal{A}$. Since $\mathcal{A}$ is an agnostic PAC learner for $\mathcal{H}$ w.r.t ℓ, we have that for sample size $n \geq m_{\mathcal{A}}(\frac{a\varepsilon}{2}, \delta/2, K)$, with probability at least $1 - \frac{\delta}{2}$,

$$\mathbb{E}_{\mathcal{D}^\star}\left[\ell(g(x), y)\right] \leq \inf_{h \in \mathcal{H}} \mathbb{E}_{\mathcal{D}^\star}\left[\ell(h(x), y)\right] + \frac{a\varepsilon}{2} = \frac{a\varepsilon}{2}.$$

Furthermore, by definition of $\mathcal{D}^\star$, $\mathbb{E}_{\mathcal{D}^\star}\left[\ell(g(x), y)\right] = \mathbb{E}_{x \sim \mathcal{D}_{\mathcal{X}}}\left[\ell(g(x), \text{BinRel}(h^\star(x), j))\right]$. Therefore, $\mathbb{E}_{x \sim \mathcal{D}_{\mathcal{X}}}\left[\ell(g(x), \text{BinRel}(h^\star(x), j))\right] \leq \frac{a\varepsilon}{2}$. Next, using Lemma E.5.3, we have pointwise that

$$\begin{aligned}\mathbb{1}\{g_i^j(x) \neq h_i^{\star,j}(x)\} &\leq \mathbb{1}\{\ell_{\text{sum}}^{@p}(g(x), \text{BinRel}(h^\star(x), j)) > 0\} \\ &= \mathbb{1}\{\ell(g(x), \text{BinRel}(h^\star(x), j)) > 0\} \\ &\leq \frac{1}{a}\,\ell(g(x), \text{BinRel}(h^\star(x), j)).\end{aligned}$$

Taking expectations on both sides gives,

$$\mathbb{E}_{\mathcal{D}}\left[\mathbb{1}\{g_i^j(x) \neq h_i^{\star,j}(x)\}\right] \leq \frac{1}{a}\,\mathbb{E}_{\mathcal{D}}\left[\ell(g(x), \text{BinRel}(h^\star(x), j))\right] \leq \frac{\varepsilon}{2},$$

where in the last inequality we use the fact that $\mathbb{E}_{x\sim\mathcal{D}_{\mathcal{X}}}\left[\ell(g(x), \text{BinRel}(h^\star(x), j))\right] \leq \frac{a\varepsilon}{2}$. Finally, using the triangle inequality, we have that

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}\left[\mathbb{1}\{g_i^j(x) \neq y\}\right] &\leq \mathbb{E}_{\mathcal{D}}\left[\mathbb{1}\{h_i^{\star,j}(x) \neq y\}\right] + \mathbb{E}_{\mathcal{D}}\left[\mathbb{1}\{g_i^j(x) \neq h_i^{\star,j}(x\}\right] \\ &\leq \mathbb{E}_{\mathcal{D}}\left[\mathbb{1}\{h_i^{\star,j}(x) \neq y\}\right] + \frac{\varepsilon}{2} \\ &= \arg\min_{h_i^j \in \mathcal{H}_i^j} \mathbb{E}_{\mathcal{D}}\left[\mathbb{1}\{h_i^j(x) \neq y\}\right] + \frac{\varepsilon}{2}.\end{aligned}$$

Since $g_i^j \in C_i^j(S_U)$, we have shown that $C_i^j(S_U)$ contains a hypothesis that generalizes well w.r.t $\mathcal{D}$. Now we want to show that the predictor $\hat{g}_i^j$ returned in step 4 also generalizes well. Crucially, observe that $C_i^j(S_U)$ is a finite hypothesis class with cardinality at most K^{jn}. Therefore, by standard Chernoff and union bounds, with probability at least $1 - \delta/2$, the empirical risk of every hypothesis in $C_i^j(S_U)$ on a sample of size $\geq \frac{8}{\varepsilon^2}\log\frac{4|C_i^j(S_U)|}{\delta}$ is at most $\varepsilon/4$ away from its true error. So, if $m = |S_L| \geq \frac{8}{\varepsilon^2}\log\frac{4|C_i^j(S_U)|}{\delta}$, then with probability at least $1 - \delta/2$, we have

$$\frac{1}{|S_L|}\sum_{(x,y)\in S_L} \mathbb{1}\{g_i^j(x) \neq y\} \leq \mathbb{E}_{\mathcal{D}}\left[\mathbb{1}\{g_i^j(x) \neq y\}\right] + \frac{\varepsilon}{4} \leq \frac{3\varepsilon}{4}.$$

Since $\hat{g}_i^j$ is the ERM on S_L over $C_i^j(S_U)$, its empirical risk can be at most $\frac{3\varepsilon}{4}$. Given that the population risk of $\hat{g}_i^j$ can be at most $\varepsilon/4$ away from its empirical risk, we have that

$$\mathbb{E}_{\mathcal{D}}[\mathbb{1}\{\hat{g}_i^j(x) \neq y\}] \leq \arg\min_{h_i^j \in \mathcal{H}_i^j} \mathbb{E}_{\mathcal{D}}\left[\mathbb{1}\{h_i^j(x) \neq y\}\right] + \varepsilon.$$

Applying union bounds, the entire process succeeds with probability $1 - \delta$. We can

compute the upper bound on the sample complexity of Algorithm 22, denoted $n(\varepsilon, \delta, K)$, as

$$\begin{aligned} n(\varepsilon, \delta, K) &\leq m_{\mathcal{A}}(\frac{a\varepsilon}{2}, \delta/2, K) + O\left(\frac{1}{\varepsilon^2}\log\frac{|C(S_U)|}{\delta}\right) \\ &\leq m_{\mathcal{A}}(\frac{a\varepsilon}{2}, \delta/2, K) + O\left(\frac{Km_{\mathcal{A}}(\frac{a\varepsilon}{2}, \delta/2, K) + \log\frac{1}{\delta}}{\varepsilon^2}\right), \end{aligned}$$

where we use $|C(S_U)| \leq 2^{Km_{\mathcal{A}}(\frac{a\varepsilon}{2}, \delta/2, K)}$. This shows that Algorithm 22 is an agnostic PAC learner for $\mathcal{H}_i^j$ w.r.t 0-1 loss. Since our choice of loss $\ell \in \mathcal{L}(\ell_{\text{sum}}^{@p})$ and indices (i, j) were arbitrary, agnostic PAC learnability of $\mathcal{H}$ w.r.t ℓ implies agnostic PAC learnability of $\mathcal{H}_i^j$ w.r.t the 0-1 loss for all $(i, j) \in [K] \times [p]$. ■

E.3.3 Characterizing Batch Learnability of $\mathcal{L}(\ell_{\textbf{prec}}^{@p})$

In this section, we prove Theorem 7.4.2 which characterizes the agnostic PAC learnability of an arbitrary hypothesis class $\mathcal{H} \subseteq \mathcal{S}_K^{\mathcal{X}}$ w.r.t losses in $\mathcal{L}(\ell_{\text{prec}}^{@p})$. Our proof will again be in three parts. First, we will show that if for all $i \in [K]$, $\mathcal{H}_i^p$ is agnostic PAC learnable w.r.t the 0-1 loss, then ERM is an agnostic PAC learnable w.r.t $\ell_{\text{prec}}^{@p}$. Next, we show that if $\mathcal{H}$ is agnostic PAC learnable w.r.t $\ell_{\text{prec}}^{@p}$, then $\mathcal{H}$ is agnostic PAC learnable w.r.t any loss $\ell \in \mathcal{L}(\ell_{\text{prec}}^{@p})$. Finally, we prove the necessity direction - if $\mathcal{H}$ is agnostic PAC learnable w.r.t an arbitrary $\ell \in \mathcal{L}(\ell_{\text{prec}}^{@p})$, then for all $i \in [K]$, $\mathcal{H}_i^p$ is agnostic PAC learnable w.r.t the 0-1 loss.

We begin with Lemma E.3.2 which asserts that if for all $i \in [K]$, $\mathcal{H}_i^p$ is agnostic PAC learnable, then ERM is an agnostic PAC learner for $\mathcal{H}$ w.r.t $\ell_{\text{prec}}^{@p}$.

Lemma E.3.2. *If for all $i \in [K]$, $\mathcal{H}_i^p$ is agnostic PAC learnable w.r.t the 0-1 loss, then ERM is an agnostic PAC learner for $\mathcal{H} \subseteq \mathcal{S}_K^{\mathcal{X}}$ w.r.t $\ell_{prec}^{@p}$*

The proof of Lemma E.3.2 is similar to the proof of Lemma 7.4.3 and involves bounding the empirical Rademacher complexity of the loss class $\ell_{\text{prec}}^{@p} \circ \mathcal{H}$. This will imply that $\ell_{\text{prec}}^{@p}$ enjoys the uniform convergence property, and therefore ERM is an agnostic PAC learner for $\mathcal{H}$ w.r.t $\ell_{\text{prec}}^{@p}$. The key insight is that we can write $\ell_{\text{prec}}^{@p}(h(x), y) = Z_y^p - \sum_{i=1}^K \mathbb{1}\{h_i(x) \leq p\}y^i = Z_y^p - \sum_{i=1}^K h_i^p(x)y^i$. Since Z_y^p does not depend on $h(x)$ and $y^i \leq B$, we can upperbound the empirical Rademacher complexity in terms of the empirical Rademacher complexities of $\mathcal{H}_i^p$ using Talagrand's contraction.

Proof. Let $\mathcal{H} \subseteq \mathcal{S}_K^{\mathcal{X}}$ be an arbitrary ranking hypothesis class. Similar to the proof of Lemma 7.4.3, it suffices to show that the empirical Rademacher complexity of the loss class $\ell_{\text{prec}}^{@p} \circ \mathcal{H}$ vanishes. By Proposition E.3.1, this will imply that $\ell_{\text{prec}}^{@p}$ enjoys the uniform convergence

property, and therefore ERM is an agnostic PAC learner for $\mathcal{H}$ w.r.t $\ell^{@p}_{\text{prec}}$. By definition, we have that

$$\begin{aligned}
\hat{\mathfrak{R}}_n(\ell^{@p}_{\text{prec}} \circ \mathcal{H}) &= \mathbb{E}_{\sigma\sim\{\pm1\}^n}\left[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \ell^{@p}_{\text{prec}}(h(x_i), y_i))\right] \\
&= \mathbb{E}_{\sigma\sim\{\pm1\}^n}\left[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \left(\sigma_i Z^p_{y_i} - \sum_{m=1}^{K} \sigma_i \mathbb{1}\{h_m(x_i) \leq p\} y_i^m\right)\right] \\
&= \mathbb{E}_{\sigma\sim\{\pm1\}^n}\left[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n}\sum_{m=1}^{K} \sigma_i h^p_m(x_i) y_i^m\right] \\
&\leq \sum_{m=1}^{K} \mathbb{E}_{\sigma\sim\{\pm1\}^n}\left[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i h^p_m(x_i) y_i^m\right] \\
&\leq B\sum_{m=1}^{K} \mathbb{E}_{\sigma\sim\{\pm1\}^n}\left[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i h^p_m(x_i)\right] \\
&= B\sum_{m=1}^{K} \hat{\mathfrak{R}}_n(\mathcal{H}^p_m),
\end{aligned}$$

where the second inequality follows from Talagrand's Contraction Lemma and the fact that $y_i^m \leq B$ for all i, m. Since for all $m \in [K]$, $\mathcal{H}^p_m$ is agnostic PAC learnable w.r.t 0-1 loss, by Theorem 6.7 in Shalev-Shwartz and Ben-David [2014a], $\lim_{n\to\infty} \hat{\mathfrak{R}}_n(\mathcal{H}^p_m) = 0$. Since K and B are finite,

$$\lim_{n\to\infty} \hat{\mathfrak{R}}_n(\ell^{@p}_{\text{prec}} \circ \mathcal{H}) = \lim_{n\to\infty} B\sum_{m=1}^{K} \hat{\mathfrak{R}}_n(\mathcal{H}^p_m) = 0$$

.

By Proposition E.3.1, this implies that $\ell^{@p}_{\text{prec}}$ enjoys the uniform convergence property, and therefore ERM using $\ell^{@p}_{\text{prec}}$ is an agnostic PAC learner for $\mathcal{H}$. ■

Next, Lemma E.3.3 extends the learnability of $\ell^{@p}_{\text{prec}}$ to the learnability of any loss $\ell \in \mathcal{L}(\ell^{@p}_{\text{prec}})$. In particular, Lemma E.3.3 asserts that if $\mathcal{H}$ is agnostic PAC learnable w.r.t $\ell^{@p}_{\text{prec}}$ then $\mathcal{H}$ is also agnostic PAC learnable w.r.t any $\ell \in \mathcal{L}(\ell^{@p}_{\text{prec}})$.

Lemma E.3.3. *If a hypothesis class $\mathcal{H} \subseteq \mathcal{S}^{\mathcal{X}}_K$ is agnostic PAC learnable w.r.t $\ell^{@p}_{prec}$, then $\mathcal{H}$ is agnostic PAC learnable w.r.t any $\ell \in \mathcal{L}(\ell^{@p}_{prec})$.*

The proof of Lemma E.3.3 follows the same the exact same strategy used in proving Lemma 7.4.4. More specifically, given an agnostic PAC learner $\mathcal{A}$ for $\mathcal{H}$ w.r.t. $\ell^{@p}_{\text{prec}}$, we first

create a *realizable* PAC learner for $\mathcal{H}$ w.r.t $\ell \in \mathcal{L}(\ell_{\text{prec}}^{@p})$. Then, we use a similar realizable-to-agnostic conversion technique as in the proof of Lemma 7.4.4 to convert the realizable PAC learner into an agnostic PAC learner for $\mathcal{H}$ w.r.t ℓ.

Proof. Fix $\ell \in \mathcal{L}(\ell_{\text{prec}}^{@p})$. Let $a = \min_{\pi,y}\{\ell(\pi, y) \mid \ell(\pi, y) \neq 0\}$ and $b = \max_{\pi,y} \ell(\pi, y)$. We need to show that if $\mathcal{H}$ is agnostic PAC learnable w.r.t $\ell_{\text{prec}}^{@p}$, then $\mathcal{H}$ is agnostic PAC learnable w.r.t ℓ. We will do so in two steps. First, we will show that if $\mathcal{A}$ is an agnostic PAC learner for $\mathcal{H}$ w.r.t. $\ell_{\text{prec}}^{@p}$, then $\mathcal{A}$ is also a *realizable* PAC learner for $\mathcal{H}$ w.r.t ℓ. Next, we will show how to convert the realizable PAC learner w.r.t ℓ into an agnostic PAC learner w.r.t ℓ in a black-box fashion. The composition of these two pieces yields an agnostic PAC learner for $\mathcal{H}$ w.r.t ℓ.

If $\mathcal{H}$ is agnostic PAC learnable w.r.t $\ell_{\text{prec}}^{@p}$, then there exists a learning algorithm $\mathcal{A}$ with sample complexity $m(\varepsilon, \delta, K)$ s.t. for any distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, with probability $1 - \delta$ over a sample $S \sim \mathcal{D}^n$ of size $n \geq m(\varepsilon, \delta, K)$, the output $g = \mathcal{A}(S)$ achieves

$$\mathbb{E}_{\mathcal{D}}\left[\ell_{\text{prec}}^{@p}(g(x), y))\right] \leq \inf_{h \in \mathcal{H}} \mathbb{E}_{\mathcal{D}}\left[\ell_{\text{prec}}^{@p}(h(x), y))\right] + \varepsilon.$$

If $\mathcal{D}$ is realizable w.r.t ℓ, then we are guaranteed that there exists a hypothesis $h^\star \in \mathcal{H}$ s.t. $\mathbb{E}_{\mathcal{D}}\left[\ell(h^\star(x), y)\right] = 0$. Since $\ell \in \mathcal{L}(\ell_{\text{prec}}^{@p})$, this also means that $\mathbb{E}_{\mathcal{D}}\left[\ell_{\text{prec}}^{@p}(h^\star(x), y)\right] = 0$. Furthermore, since $\ell \in \mathcal{L}(\ell_{\text{prec}}^{@p})$, $\ell \leq b\ell_{\text{prec}}^{@p}$. Together, this means we have $\mathbb{E}_{\mathcal{D}}\left[\ell(g(x), y)\right] \leq b\varepsilon$ showing have that $\mathcal{A}$ is also a realizable PAC learner for $\mathcal{H}$ w.r.t ℓ with sample complexity $m(\frac{\varepsilon}{b}, \delta, K)$. This completes the first part of the proof.

Now, we show how to convert the realizable PAC learner $\mathcal{A}$ for ℓ into an agnostic PAC learner for ℓ in a black-box fashion. For this step, we will use a similar algorithm as in the proof of Lemma 7.4.4. That is, we will show that Algorithm 23 below is an agnostic PAC learner for $\mathcal{H}$ w.r.t ℓ.

Algorithm 23 Agnostic PAC learner for $\mathcal{H}$ w.r.t. ℓ

Require: Realizable PAC learner $\mathcal{A}$ for $\mathcal{H}$ w.r.t ℓ, unlabeled samples $S_U \sim \mathcal{D}_{\mathcal{X}}^n$, and labeled samples $S_L \sim \mathcal{D}^m$

1: For each $h \in \mathcal{H}_{|S_U}$, construct a dataset

$$S_U^h = \{(x_1, \tilde{y}_1), ..., (x_n, \tilde{y}_n)\} \text{ s.t. } \tilde{y}_i = \text{BinRel}(h(x_i), p)$$

2: Run $\mathcal{A}$ over all datasets to get $C(S_U) := \left\{\mathcal{A}\left(S_U^h\right) \mid h \in \mathcal{H}_{|S_U}\right\}$

3: Return $\hat{g} \in C(S_U)$ with the lowest empirical error over S_L w.r.t. ℓ.

Let $\mathcal{D}$ be any (not necessarily realizable) distribution over $\mathcal{X} \times \mathcal{Y}$. Let $h^\star = \arg\min_{h \in \mathcal{H}} \mathbb{E}_{\mathcal{D}}\left[\ell(h(x), y))\right]$ denote the optimal predictor in $\mathcal{H}$ w.r.t $\mathcal{D}$. Consider the sample

$S_U^{h^\star}$ and let $g = \mathcal{A}(S_U^{h^\star})$. We can think of g as the output of $\mathcal{A}$ run over an i.i.d sample S drawn from $\mathcal{D}^\star$, a joint distribution over $\mathcal{X} \times \mathcal{Y}$ defined procedurally by first sampling $x \sim \mathcal{D}_{\mathcal{X}}$, and then outputting the labeled sample $(x, \text{BinRel}(h^\star(x), p))$. Note that $\mathcal{D}^\star$ is indeed a realizable distribution (realized by $h^\star$) w.r.t both ℓ and $\ell_{\text{prec}}^{@p}$. Recall that $m_{\mathcal{A}}(\frac{\varepsilon}{b}, \delta, K)$ is the sample complexity of $\mathcal{A}$. Since $\mathcal{A}$ is a realizable learner for $\mathcal{H}$ w.r.t ℓ, we have that for $n \geq m_{\mathcal{A}}(\frac{a\varepsilon}{2b^2}, \delta/2, K)$, with probability at least $1 - \frac{\delta}{2}$,

$$\mathbb{E}_{\mathcal{D}^\star}\left[\ell(g(x), y)\right] \leq \frac{a\varepsilon}{2b}.$$

By definition of $\mathcal{D}^\star$, it further follows that $\mathbb{E}_{\mathcal{D}^\star}\left[\ell(g(x), y)\right] = \mathbb{E}_{x \sim \mathcal{D}_{\mathcal{X}}}\left[\ell(g(x), \text{BinRel}(h^\star(x), p))\right]$. Therefore,

$$\mathbb{E}_{x \sim \mathcal{D}_{\mathcal{X}}}\left[\ell(g(x), \text{BinRel}(h^\star(x), p))\right] \leq \frac{a\varepsilon}{2b}.$$

Next, by Lemma E.5.2, we have pointwise that:

$$\ell(g(x), y) \leq \ell(h^\star(x), y) + \frac{b}{a}\ell(g(x), \text{BinRel}(h^\star(x), p)).$$

Taking expectations on both sides of the inequality gives:

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}}\left[\ell(g(x), y)\right] &\leq \mathbb{E}_{\mathcal{D}}\left[\ell(h^\star(x), y)\right] + \mathbb{E}_{\mathcal{D}}\left[\frac{b}{a}\ell(g(x), \text{BinRel}(h^\star(x), p))\right] \\
&= \mathbb{E}_{\mathcal{D}}\left[\ell(h^\star(x), y)\right] + \frac{b}{a}\mathbb{E}_{x \sim \mathcal{D}_{\mathcal{X}}}\left[\ell(g(x), \text{BinRel}(h^\star(x), p))\right] \\
&\leq \mathbb{E}_{\mathcal{D}}\left[\ell(h^\star(x), y)\right] + \frac{\varepsilon}{2}.
\end{aligned}$$

Therefore, we have shown that $C(S_U)$ contains a hypothesis g that generalizes well with respect to $\mathcal{D}$. The remaining proof follows exactly as in the proof of Lemma 7.4.4. We include them here for the sake of completeness.

Now we want to show that the predictor $\hat{g}$ returned in step 4 also has good generalization. Crucially, observe that $C(S_U)$ is a finite hypothesis class with cardinality at most K^{pn}. Therefore, by standard Chernoff and union bounds, with probability at least $1 - \delta/2$, the empirical risk of every hypothesis in $C(S_U)$ on a sample of size $\geq \frac{8}{\varepsilon^2}\log\frac{4|C(S_U)|}{\delta}$ is at most $\varepsilon/4$ away from its true error. So, if $m = |S_L| \geq \frac{8}{\varepsilon^2}\log\frac{4|C(S_U)|}{\delta}$, then with probability at least $1 - \delta/2$, we have

$$\frac{1}{|S_L|}\sum_{(x,y) \in S_L} \ell(g(x), y) \leq \mathbb{E}_{\mathcal{D}}\left[\ell(g(x), y)\right] + \frac{\varepsilon}{4} \leq \mathbb{E}_{\mathcal{D}}\left[\ell(h^\star(x), y)\right] + \frac{3\varepsilon}{4}.$$

Since $\hat{g}$ is the ERM on S_L over $C(S)$, its empirical risk can be at most $\mathbb{E}_{\mathcal{D}}\left[\ell(h^\star(x), y)\right] + \frac{3\varepsilon}{4}$. Given that the population risk of $\hat{g}$ can be at most $\varepsilon/4$ away from its empirical risk, we have that

$$\mathbb{E}_{\mathcal{D}}[\ell(\hat{g}(x), y)] \leq \mathbb{E}_{\mathcal{D}}\left[\ell(h^\star(x), y)\right] + \varepsilon.$$

Applying union bounds, the entire process succeeds with probability $1 - \delta$. We can upper bound the sample complexity of Algorithm 9, denoted $n(\varepsilon, \delta, K)$, as

$$\begin{aligned} n(\varepsilon, \delta, K) &\leq m_{\mathcal{A}}(\frac{a\varepsilon}{2b^2}, \delta/2, K) + O\left(\frac{1}{\varepsilon^2}\log\frac{|C(S_U)|}{\delta}\right) \\ &\leq m_{\mathcal{A}}(\frac{a\varepsilon}{2b^2}, \delta/2, K) + O\left(\frac{p\, m_{\mathcal{A}}(\frac{a\varepsilon}{2b^2}, \delta/2, K)\log(K) \ + \log\frac{1}{\delta}}{\varepsilon^2}\right), \end{aligned}$$

where we use $|C(S_U)| \leq K^{p m_{\mathcal{A}}(\frac{a\varepsilon}{2b^2}, \delta/2, K)}$. This shows that Algorithm 9, given as input an realizable PAC learner for $\mathcal{H}$ w.r.t ℓ, is an agnostic PAC learner for $\mathcal{H}$ w.r.t ℓ. Using the realizable learner we constructed before this step as the input completes this proof as we have constructively converted an agnostic PAC learner for $\ell^{@p}_{\text{prec}}$ into an agnostic PAC learner for ℓ. ■

Lemma E.3.2 and E.3.3 together complete the proof of sufficiency in Theorem 7.4.2. Finally, Lemma E.3.4 below shows that the agnostic PAC learnability of $\mathcal{H}^p_i$ for all $i \in [K]$ is necessary for the agnostic PAC learnability of $\mathcal{H}$ w.r.t any $\ell \in \mathcal{L}(\ell^{@p}_{\text{prec}})$. Like before, the proof of Lemma E.3.4 is constructive and follows exactly the same strategy as Lemma 7.4.5. That is, given as input a learner for ℓ, we will convert it into an agnostic learner for $\mathcal{H}^p_i$. In fact, the conversion is exactly the same as in the proof of Lemma 7.4.5 and just requires running Algorithm 22 with an input learner for $\ell \in \mathcal{L}(\ell^{@p}_{\text{prec}})$ and setting $j = p$.

Lemma E.3.4. *If a function class $\mathcal{H} \subseteq \mathcal{S}^{\mathcal{X}}_K$ is agnostic PAC learnable w.r.t $\ell \in \mathcal{L}(\ell^{@p}_{prec})$, then $\mathcal{H}^p_i$ is agnostic PAC learnable w.r.t the 0-1 loss for all $i \in [K]$.*

Proof. Fix $\ell \in \mathcal{L}(\ell^{@p}_{\text{prec}})$ and $i \in [K]$. Let $a = \min_{\pi,y}\{\ell(\pi, y) \mid \ell(\pi, y) \neq 0\}$. Let $\mathcal{H}$ be an arbitrary ranking hypothesis class and $\mathcal{A}$ be an agnostic PAC learner for $\mathcal{H}$ w.r.t ℓ. Our goal will to be to use $\mathcal{A}$ to construct an agnostic PAC learner for $\mathcal{H}^p_i$.

Let $\mathcal{D}$ be any distribution over $\mathcal{X} \times \{0, 1\}$, $h^{\star,p}_i = \arg\min_{h \in \mathcal{H}^p_i} \mathbb{E}_{\mathcal{D}}\left[\mathbb{1}\{h(x) \neq y\}\right]$ the optimal hypothesis, and $h^\star \in \mathcal{H}$ be any valid completion of $h^{\star,p}_i$. We will now show that Algorithm 22 from the proof of Lemma 7.4.5 is an agnostic PAC learner for $\mathcal{H}^p_i$ if we set $j = p$ and give it as input an agnostic PAC learner $\mathcal{A}$ for $\mathcal{H}$ w.r.t. $\ell \in \mathcal{L}(\ell^{@p}_{\text{prec}})$.

Consider the sample $S^{h^\star}_U$ and let $g = \mathcal{A}(S^{h^\star}_U)$. We can think of g as the output of $\mathcal{A}$ run over an i.i.d sample S drawn from $\mathcal{D}^\star$, a joint distribution over $\mathcal{X} \times \mathcal{Y}$ defined procedurally

by first sampling $x \sim \mathcal{D}_{\mathcal{X}}$ and then outputting the labeled sample $(x, \mathrm{BinRel}(h^\star(x), p))$. Note that $\mathcal{D}^\star$ is a realizable distribution (realized by $h^\star$) w.r.t $\ell_{\mathrm{prec}}^{@p}$ and therefore also ℓ. Let $m_{\mathcal{A}}(\varepsilon, \delta, K)$ be the sample complexity of $\mathcal{A}$.

Since $\mathcal{A}$ is an agnostic PAC learner for $\mathcal{H}$ w.r.t ℓ, we have that for sample size $n \geq m_{\mathcal{A}}(\frac{a\varepsilon}{2}, \delta/2, K)$, with probability at least $1 - \frac{\delta}{2}$,

$$\mathbb{E}_{\mathcal{D}^\star}\left[\ell(g(x), y)\right] \leq \inf_{h \in \mathcal{H}} \mathbb{E}_{\mathcal{D}^\star}\left[\ell(h(x), y)\right] + \frac{a\varepsilon}{2} = \frac{a\varepsilon}{2}.$$

Furthermore, by definition of $\mathcal{D}^\star$, $\mathbb{E}_{\mathcal{D}^\star}\left[\ell(g(x), y)\right] = \mathbb{E}_{x \sim \mathcal{D}_{\mathcal{X}}}\left[\ell(g(x), \mathrm{BinRel}(h^\star(x), p))\right]$. Therefore, $\mathbb{E}_{x \sim \mathcal{D}_{\mathcal{X}}}\left[\ell(g(x), \mathrm{BinRel}(h^\star(x), p))\right] \leq \frac{a\varepsilon}{2}$. Next, using Lemma E.5.4, we have pointwise that

$$\begin{aligned}
\mathbb{1}\{g_i^p(x) \neq h_i^{\star,p}(x)\} &\leq \mathbb{1}\{\ell_{\mathrm{prec}}^{@p}(g(x), \mathrm{BinRel}(h^\star(x), p)) > 0\} \\
&= \mathbb{1}\{\ell(g(x), \mathrm{BinRel}(h^\star(x), p)) > 0\} \\
&\leq \frac{1}{a}\,\ell(g(x), \mathrm{BinRel}(h^\star(x), p)).
\end{aligned}$$

Taking expectations on both sides gives,

$$\mathbb{E}_{\mathcal{D}}\left[\mathbb{1}\{g_i^p(x) \neq h_i^{\star,p}(x)\}\right] \leq \frac{1}{a}\,\mathbb{E}_{\mathcal{D}}\left[\ell(g(x), \mathrm{BinRel}(h^\star(x), p))\right] \leq \frac{\varepsilon}{2},$$

where in the last inequality we use the fact that $\mathbb{E}_{x \sim \mathcal{D}_{\mathcal{X}}}\left[\ell(g(x), \mathrm{BinRel}(h^\star(x), p))\right] \leq \frac{a\varepsilon}{2}$. Finally, using the triangle inequality, we have that

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}}\left[\mathbb{1}\{g_i^p(x) \neq y\}\right] &\leq \mathbb{E}_{\mathcal{D}}\left[\mathbb{1}\{h_i^{\star,p}(x) \neq y\}\right] + \mathbb{E}_{\mathcal{D}}\left[\mathbb{1}\{g_i^p(x) \neq h_i^{\star,p}(x)\}\right] \\
&\leq \mathbb{E}_{\mathcal{D}}\left[\mathbb{1}\{h_i^{\star,p}(x) \neq y\}\right] + \frac{\varepsilon}{2} \\
&= \arg\min_{h_i^p \in \mathcal{H}_i^p} \mathbb{E}_{\mathcal{D}}\left[\mathbb{1}\{h_i^p(x) \neq y\}\right] + \frac{\varepsilon}{2}.
\end{aligned}$$

Since $g_i^p \in C_i^p(S_U)$, we have shown that $C_i^p(S_U)$ contains a hypothesis that generalizes well w.r.t $\mathcal{D}$. Now we want to show that the predictor $\hat{g}_i^p$ returned in step 4 also generalizes well. Crucially, observe that $C_i^p(S_U)$ is a finite hypothesis class with cardinality at most K^{pn}. Therefore, by standard Chernoff and union bounds, with probability at least $1 - \delta/2$, the empirical risk of every hypothesis in $C_i^p(S_U)$ on a sample of size $\geq \frac{8}{\varepsilon^2} \log \frac{4|C_i^j(S_U)|}{\delta}$ is at most $\varepsilon/4$ away from its true error. So, if $m = |S_L| \geq \frac{8}{\varepsilon^2} \log \frac{4|C_i^j(S_U)|}{\delta}$, then with probability at least

$1-\delta/2$, we have

$$\frac{1}{|S_L|}\sum_{(x,y)\in S_L}\mathbb{1}\{g_i^p(x)\neq y\}\leq \mathbb{E}_{\mathcal{D}}\left[\mathbb{1}\{g_i^p(x)\neq y\}\right]+\frac{\varepsilon}{4}\leq\frac{3\varepsilon}{4}.$$

Since $\hat{g}_i^p$ is the ERM on S_L over $C_i^p(S_U)$, its empirical risk can be at most $\frac{3\varepsilon}{4}$. Given that the population risk of $\hat{g}_i^p$ can be at most $\varepsilon/4$ away from its empirical risk, we have that

$$\mathbb{E}_{\mathcal{D}}[\mathbb{1}\{\hat{g}_i^p(x)\neq y\}]\leq\min_{h_i^p\in\mathcal{H}_i^p}\mathbb{E}_{\mathcal{D}}\left[\mathbb{1}\{h_i^p(x)\neq y\}\right]+\varepsilon.$$

Applying union bounds, the entire process succeeds with probability $1-\delta$. We can compute the upper bound on the sample complexity of Algorithm 22, denoted $n(\varepsilon,\delta,K)$, as

$$\begin{aligned}n(\varepsilon,\delta,K)&\leq m_{\mathcal{A}}(\frac{a\varepsilon}{2},\delta/2,K)+O\left(\frac{1}{\varepsilon^2}\log\frac{|C(S_U)|}{\delta}\right)\\&\leq m_{\mathcal{A}}(\frac{a\varepsilon}{2},\delta/2,K)+O\left(\frac{p\,m_{\mathcal{A}}(\frac{a\varepsilon}{2},\delta/2,K)\log(K)\;+\log\frac{1}{\delta}}{\varepsilon^2}\right),\end{aligned}$$

where we use $|C(S_U)|\leq K^{pm_{\mathcal{A}}(\frac{a\varepsilon}{2},\delta/2,K)}$. This shows that Algorithm 22 is an agnostic PAC learner for $\mathcal{H}_i^p$ w.r.t 0-1 loss. Since our choice of loss $\ell\in\mathcal{L}(\ell_{\text{prec}}^{@p})$ and index i were arbitrary, agnostic PAC learnability of $\mathcal{H}$ w.r.t ℓ implies agnostic PAC learnability of $\mathcal{H}_i^p$ w.r.t the 0-1 loss for all $i\in[K]$. ■

Combining Lemma E.3.2, E.3.3 and E.3.4 gives Theorem 7.4.2.

E.4 Proofs for Online Multilabel Ranking

E.4.1 Proof of necessity in Theorem 7.5.1

Proof. Fix $\ell\in\mathcal{L}(\ell_{\text{sum}}^{@p})$ and $(i,j)\in[K]\times[p]$. Given an online learner $\mathcal{A}$ for $\mathcal{H}$ w.r.t ℓ, our goal is to construct an agnostic online learner $\mathcal{A}_i^j$ for $\mathcal{H}_i^j$. To that end, let $(x_1,y_1),...,(x_T,y_T)\in(\mathcal{X}\times\{0,1\})^T$ denote a stream of labeled instances. Define $h_i^{\star,j}=\arg\min_{h_i^j\in\mathcal{H}_i^j}\sum_{t=1}^T\mathbb{1}\{h_i^j(x_t)\neq y_t\}$ to be the optimal function in $\mathcal{H}_i^j$ and $h^\star$ be an arbitrary completion of $h_i^{\star,j}$. As in the sufficiency proof, our construction of the online learner for $\mathcal{H}_i^j$ will run REWA over a set of experts we construct below.

For any bitstring $b\in\{0,1\}^T$, let $\phi:\{t\in[T]:b_t=1\}\to\mathcal{S}_K$ denote a function mapping time points where $b_t=1$ to permutations. Let $\Phi_b=\mathcal{S}_K^{\{t\in[T]:b_t=1\}}$ denote all such functions ϕ. For every $h\in\mathcal{H}$, there exists a $\phi_b^h\in\Phi_b$ such that for all $t\in\{t:b_t=1\}$, $\phi_b^h(t)=h(x_t)$.

Let $|b| = |\{t \in [T] : b_t = 1\}|$. For every $b \in \{0,1\}^T$ and $\phi \in \Phi_b$, define an Expert $E_{b,\phi}$. Expert $E_{b,\phi}$, formally presented in Algorithm 24, uses $\mathcal{A}$ to make predictions in each round. For every $b \in \{0,1\}^T$, let $\mathcal{E}_b = \bigcup_{\phi \in \Phi_b} \{E_{b,\phi}\}$ denote the set of all Experts parameterized by functions $\phi \in \Phi_b$. As before, we will actually define $\mathcal{E}_b = \{E_0\} \cup \bigcup_{\phi \in \Phi_b} \{E_{b,\phi}\}$, where E_0 is the expert that never updates $\mathcal{A}$ and only uses it to make predictions in each round. Note that $1 \leq |\mathcal{E}_b| \leq (K!)^{|b|} \leq K^{K|b|}$.

Algorithm 24 Expert (b, ϕ)

Require: Independent copy of online learner $\mathcal{A}$ for $\mathcal{H}$
1: **for** $t = 1, ..., T$ **do**
2: Receive example x_t
3: Predict $\mathbb{1}\{\hat{\pi}_i \leq j\}$ where $\hat{\pi} = \mathcal{A}(x_t)$
4: **if** $b_t = 1$ **then**
5: Update $\mathcal{A}$ by passing $(x_t, \text{BinRel}(\phi(t), j))$
6: **end if**
7: **end for**

We are now ready to give the agnostic online learner for $\mathcal{H}_i^j$, henceforth denoted by $\mathcal{Q}$. Our online learner $\mathcal{Q}$ is very similar to Algorithm 11. First, it will sample a $B \in \{0,1\}^T$ s.t. $B_t \sim \text{Bernoulli}(T^\beta/T)$. Then, it will construct a set of experts $\mathcal{E}_B$ using Algorithm 24. Finally, it will run REWA, denoted by $\mathcal{P}$, on the 0-1 loss over the stream $(x_1, y_1), ..., (x_T, y_T)$. As before, let A and P be the random variables denoting internal randomness of the algorithm $\mathcal{A}$ and $\mathcal{P}$. Using REWA guarantees and following exactly the same calculation as in the sufficiency proof, we arrive at

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{Q}(x_t) \neq y_t\}\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{E_{B,\phi_B^{h^\star}}(x_t) \neq y_t\}\right] + \sqrt{2T^{1+\beta} K \ln K}.$$

The inequality above is the adaptation of Equation (7.1) for this proof. Recall that $h_i^{\star,j}$ is the optimal function in hindsight for the stream and $h^\star$ is a completion of $h_i^{\star,j}$. Since $\mathbb{1}\{E_{B,\phi_B^{h^\star}}(x_t) \neq y_t\} \leq \mathbb{1}\{h_i^{\star,j}(x_t) \neq y_t\} + \mathbb{1}\{E_{B,\phi_B^{h^\star}}(x_t) \neq h_i^{\star,j}(x_t)\}$, the inequality above reduces to

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{Q}(x_t) \neq y_t\}\right] \leq \sum_{t=1}^{T} \mathbb{1}\{h_i^{\star,j}(x_t) \neq y_t\} + \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{E_{B,\phi_B^{h^\star}}(x_t) \neq h_i^{\star,j}(x_t)\}\right] + \sqrt{2T^{1+\beta} K \ln K}.$$

It now suffices to show that $\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{E_{B,\phi_B^{h^\star}}(x_t) \neq h_i^{\star,j}(x_t)\}\right]$ is sub-linear function of T.

Given an online learner $\mathcal{A}$ for $\mathcal{H}$, an instance $x \in \mathcal{X}$, and an ordered finite sequence of labeled examples $L \in (\mathcal{X} \times \mathcal{Y})^*$, let $\mathcal{A}(x|L)$ be the random variable denoting the prediction of $\mathcal{A}$ on the instance x after running and updating on L. For any $b \in \{0,1\}^T$, $h \in \mathcal{H}$, and $t \in [T]$, let $L^h_{b_{<t}} = \{(x_i, \text{BinRel}(h(x_s), j)) : s < t \text{ and } b_s = 1\}$ denote the *subsequence* of the sequence of labeled instances $\{(x_s, \text{BinRel}(h(x_s), j))\}_{s=1}^{t-1}$ where $b_s = 1$. Thus, using Lemma E.5.3, we have

$$\begin{aligned}
\mathbb{1}\{E_{B,\phi_B^{h^\star}}(x_t) \neq h_i^{\star,j}(x_t)\} &\leq \mathbb{1}\{\ell_{\text{sum}}^{@p}(\mathcal{A}(x_t \mid L^{h^\star}_{B_{<t}}), \text{BinRel}(h^\star(x_t), j)) > 0\} \\
&= \mathbb{1}\{\ell(\mathcal{A}(x_t \mid L^{h^\star}_{B_{<t}}), \text{BinRel}(h^\star(x_t), j)) > 0\} \\
&\leq \frac{1}{a}\,\ell(\mathcal{A}(x_t \mid L^{h^\star}_{B_{<t}}), \text{BinRel}(h^\star(x_t), j), \text{BinRel}(h^\star(x_t), j)),
\end{aligned}$$

where equality follows from the fact that $\ell \in \mathcal{L}(\ell_{\text{sum}}^{@p})$. Here, a is the lower bound whenever it is non-zero. Taking expectations of both sides and summing over $t \in [T]$ gives

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{E_{B,\phi_B^{h^\star}}(x_t) \neq h_i^{\star,j}(x_t)\}\right] \leq \frac{1}{a}\mathbb{E}\left[\sum_{t=1}^{T} \ell(\mathcal{A}(x_t \mid L^{h^\star}_{B_{<t}}), \text{BinRel}(h^\star(x_t), j))\right].$$

To upperbound the right-hand side, we will again use the fact that the prediction $\mathcal{A}(x_t \mid L^{h^\star}_{B_{<t}})$ only depends on $(B_1, \ldots, B_{t-1})$, but is independent of B_t. The details of this calculation are omitted because they are identical to that of the sufficiency proof. Using independence of $\mathcal{A}(x_t \mid L^{h^\star}_{B_{<t}})$ and B_t, we obtain

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T} \ell(\mathcal{A}(x_t \mid L^{h^\star}_{B_{<t}}), \text{BinRel}(h^\star(x_t), j))\right] \\
&\qquad = \frac{T}{T^\beta}\,\mathbb{E}\left[\mathbb{E}\left[\sum_{t:B_t=1} \ell(\mathcal{A}(x_t \mid L^{h^\star}_{B_{<t}}), \text{BinRel}(h^\star(x_t), j)) \,\middle|\, B\right]\right] \leq \frac{T}{T^\beta}\,\mathbb{E}\left[R(|B|, K)\right],
\end{aligned}$$

where $R(|B|, K)$ is the regret of the algorithm $\mathcal{A}$, a sub-linear function of $|B|$. In the last step, we use the fact that $\mathcal{A}$ is a (realizable) online learner for $\mathcal{H}$ w.r.t. ℓ and the feedback that the algorithm received was $(x_t, \text{BinRel}(h^\star(x_t), j))$ in the rounds whenever $B_t = 1$. Again, Lemma 5.17 from Woess [2017] guarantees an existence of a concave sublinear upperbound $\tilde{R}(|B|, K)$ of $R(|B|, K)$. Then, applying Jensen's inequality yields $\mathbb{E}\left[R(|B|, K)\right] \leq \mathbb{E}\left[\tilde{R}(|B|, K)\right] \leq$

$\tilde{R}(T^\beta, K)$, a concave sub-linear function of T^β. Combining everything, we get

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{Q}(x_t) \neq y_t\}\right] \leq \sum_{t=1}^{T} \mathbb{1}\{h_i^{\star,j}(x_t) \neq y_t\} + \frac{T}{aT^\beta}\tilde{R}(T^\beta, K) + \sqrt{2T^{1+\beta}K\ln K}$$
$$= \arg\min_{h_i^j \in \mathcal{H}_i^j} \sum_{t=1}^{T} \mathbb{1}\{h_i^j(x_t) \neq y_t\} + \frac{T}{aT^\beta}\tilde{R}(T^\beta, K) + \sqrt{2T^{1+\beta}K\ln K}$$

For any choice of $\beta \in (0,1)$, the regret above is a sub-linear function of T. Therefore, we have shown that $\mathcal{Q}$ is an agnostic learner for $\mathcal{H}_i^j$ w.r.t. 0-1 loss. ■

E.4.2 Proof of Theorem 7.5.2

Proof. (of sufficiency in Theorem 7.5.2) Fix $\ell \in \mathcal{L}(\ell_{\text{prec}}^{@p})$ and let $M = \max_{\pi,y} \ell(\pi, y)$. This proof is virtually identical to the proof of sufficiency in Theorem 7.4.1. However, we provide the full details here for completion. Our proof is also based on reduction. That is, given realizable learners $\mathcal{A}_i^p$ of $\mathcal{H}_i^p$'s for $i \in [K]$ w.r.t. 0-1 loss, we will construct an agnostic learner $\mathcal{Q}$ for $\mathcal{H}$ w.r.t. ℓ. We will construct a set of experts $\mathcal{E}$ that uses $\mathcal{A}_i^p$ to make predictions and run the REWA algorithm using these experts.

Let $(x_1, y_1), ..., (x_T, y_T) \in (\mathcal{X} \times \mathcal{Y})^T$ denote the stream of points to be observed by the online learner. As before, we will assume an oblivious adversary. Define $h^\star = \arg\min_{h\in\mathcal{H}} \sum_{t=1}^{T} \ell(h(x_t), y_t)$ to be the optimal hypothesis in hindsight.

For any bitstring $b \in \{0,1\}^T$, let $\phi : \{t \in [T] : b_t = 1\} \to \mathcal{S}_K$ denote a function mapping time points where $b_t = 1$ to permutations. Let $\Phi_b = \mathcal{S}_K^{\{t\in[T]:b_t=1\}}$ denote all such functions ϕ. For every $h \in \mathcal{H}$, there exists a $\phi_b^h \in \Phi_b$ such that for all $t \in \{t : b_t = 1\}$, $\phi_b^h(t) = h(x_t)$. Let $|b| = |\{t \in [T] : b_t = 1\}|$. For every $b \in \{0,1\}^T$ and $\phi \in \Phi_b$, we will define an Expert $E_{b,\phi}$. Expert $E_{b,\phi}$, formally presented in Algorithm 11, uses $\mathcal{A}_i^p$'s to make predictions in each round. However, $E_{b,\phi}$ only updates the $\mathcal{A}_i^p$'s on those rounds where $b_t = 1$, using ϕ to compute a labeled instance. For every $b \in \{0,1\}^T$, let $\mathcal{E}_b = \bigcup_{\phi\in\Phi_b}\{E_{b,\phi}\}$ denote the set of all Experts parameterized by functions $\phi \in \Phi_b$. If b is the bitstring with all zeros, then $\mathcal{E}_b$ will be empty. Therefore, we will actually define $\mathcal{E}_b = \{E_0\} \cup \bigcup_{\phi\in\Phi_b}\{E_{b,\phi}\}$, where E_0 is the expert that never updates $\mathcal{A}_i^j$'s and only uses them for predictions in all $t \in [T]$. Note that $1 \leq |\mathcal{E}_b| \leq (K!)^{|b|} \leq K^{K|b|}$. Using these experts, Algorithm 11 is our agnostic online learner $\mathcal{Q}$ for $\mathcal{H}$ w.r.t $\ell \in \mathcal{L}(\ell_{\text{prec}}^{@p})$.

Using REWA guarantees and following exactly the same calculation as in the proof of

Algorithm 25 Expert (b, ϕ)

Require: Independent copy of realizable learners $\mathcal{A}_i^p$ of $\mathcal{H}_i^p$ for $i \in [K]$
1: **for** $t = 1, ..., T$ **do**
2: Receive example x_t
3: Define a binary vote vector $v_t \in \{0,1\}^K$ such that $v_t[i] = \mathcal{A}_i^p(x_t)$
4: Predict $\hat{\pi}_t \in \arg\min_{\pi \in \mathcal{S}_K} \langle \pi, v_t \rangle$
5: **if** $b_t = 1$ **then**
6: Let $\pi = \phi(t)$ and for each $i \in [K]$, update $\mathcal{A}_i^p$ by passing (x_t, π_i^p)
7: **end if**
8: **end for**

Theorem 7.5.1 we immediately arrive at

$$\mathbb{E}\left[\sum_{t=1}^{T} \ell(\mathcal{Q}(x_t), y_t)\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} \ell(E_{B,\phi_B^{h^\star}}(x_t), y_t)\right] + M\sqrt{2T^{1+\beta}K \ln K},$$

the analog of Equation (7.1) for this setting. Using Lemma E.5.2, we have

$$\ell(E_{B,\phi_B^{h^\star}}(x_t), y_t) \leq \ell(h^\star(x_t), y_t) + \frac{M}{a}\ell(E_{B,\phi_B^{h^\star}}(x_t), \mathrm{BinRel}(h^\star(x_t), p))$$

pointwise, where $a = \min_{\pi,y}\{\ell(\pi, y) \mid \ell(\pi, y) \neq 0\}$. By definition of M, we further get

$$\begin{aligned}\ell(E_{B,\phi_B^{h^\star}}(x_t), \mathrm{BinRel}(h^\star(x_t), p)) &\leq M\,\mathbb{1}\{\ell(E_{B,\phi_B^{h^\star}}(x_t), \mathrm{BinRel}(h^\star(x_t), p)) > 0\} \\ &= M\,\mathbb{1}\{\ell_{\mathrm{prec}}^{@p}(E_{B,\phi_B^{h^\star}}(x_t), \mathrm{BinRel}(h^\star(x_t), p)) > 0\},\end{aligned}$$

where the equality follows from the fact that $\ell \in \mathcal{L}(\ell_{\mathrm{prec}}^{@p})$.

In order to upperbound the indicator above, we need some more notations. Given the realizable online learner $\mathcal{A}_i^p$ for $i \in [K] \times [p]$, an instance $x \in \mathcal{X}$, and an ordered finite sequence of labeled examples $L \in (\mathcal{X} \times \{0,1\})^*$, let $\mathcal{A}_i^p(x|L)$ be the random variable denoting the prediction of $\mathcal{A}_i^p$ on the instance x after running and updating on L. For any $b \in \{0,1\}^T$, $h \in \mathcal{H}$, and $t \in [T]$, let $L_{b_{<t}}^h(i,p) = \{(x_s, h_i^p(x_s)) : s < t \text{ and } b_s = 1\}$ denote the *subsequence* of the sequence of labeled instances $\{(x_s, h_i^p(x_s))\}_{s=1}^{t-1}$ where $b_s = 1$. Then, we have

$$\mathbb{1}\{\ell_{\mathrm{prec}}^{@p}(E_{B,\phi_B^{h^\star}}(x_t), \mathrm{BinRel}(h^\star(x_t), p)) > 0\} \leq \sum_{i=1}^{K} \mathbb{1}\{\mathcal{A}_i^p(x_t \mid L_{B_{<t}}^{h^\star}(i,p)) \neq h_i^{\star,p}(x_t)\}.$$

To prove this claimed inequality, consider the case when $\sum_{i=1}^{K} \mathbb{1}\{\mathcal{A}_i^p(x_t \mid L_{B_{<t}}^{h^\star}(i,p)) \neq h_i^{\star,p}(x_t)\} = 0$ because the inequality is trivial otherwise. Then, we must have $\mathcal{A}_i^p(x_t \mid L_{B_{<t}}^{h^\star}(i,p)) = h_i^{\star,p}(x_t)$ for all $i \in [K]$. Let $v_t \in \{0,1\}^K$ such that $v_t[i] = \mathcal{A}_i^p(x_t \mid L_{B_{<t}}^{h^\star}(i,p))$

be a binary vote vector that the expert $E_{B,\phi_B^{h^\star}}$ constructs in round t. Since $h^\star(x_t)$ is a permutation, the vote vector v_t must contain exactly p labels with 1 vote and $K-p$ labels with 0 votes. Thus, every $\hat{\pi}_t \in \arg\min_{\pi \in \mathcal{S}_K} \langle \pi, v_t \rangle$ must rank labels with 1 vote in top p and labels with 0 votes outside top p. In other words, we must have $\hat{\pi}_t \overset{p}{=} h^\star(x_t)$, and thus $\ell_{\text{prec}}^{@p}(\hat{\pi}_t, \text{BinRel}(h^\star(x_t), p)) = 0$ by definition of $\ell_{\text{prec}}^{@p}$. Our claim follows because $E_{B,\phi_B^{h^\star}}(x_t) \in \arg\min_{\pi \in \mathcal{S}_K} \langle \pi, v_t \rangle$.

Combining everything, we obtain

$$\ell(E_{B,\phi_B^{h^\star}}(x_t), y_t) \leq \ell(h^\star(x_t), y_t) + \frac{M^2}{a} \sum_{i=1}^{K} \mathbb{1}\{\mathcal{A}_i^p(x_t \mid L_{B_{<t}}^{h^\star}(i,p)) \neq h_i^{\star,p}(x_t)\}.$$

Taking expectations on both sides and summing over all $t \in [T]$ yields

$$\mathbb{E}\left[\sum_{t=1}^{T} \ell(E_{B,\phi_B^{h^\star}}(x_t), y_t)\right] \leq \sum_{t=1}^{T} \ell(h^\star(x_t), y_t) + \frac{M^2}{a} \sum_{i=1}^{K} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}_i^p(x_t \mid L_{B_{<t}}^{h^\star}(i,p)) \neq h_i^{\star,p}(x_t)\}\right].$$

So, it now suffices to show that $\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}_i^p(x_t \mid L_{B_{<t}}^{h^\star}(i,p)) \neq h_i^{\star,p}(x_t)\}\right]$ is a sub-linear function of T. We can write

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}_i^p(x_t \mid L_{B_{<t}}^{h^\star}(i,p)) \neq h_i^{\star,p}(x_t)\}\right] = \sum_{t=1}^{T} \mathbb{E}\left[\mathbb{1}\{\mathcal{A}_i^p(x_t \mid L_{B_{<t}}^{h^\star}(i,p)) \neq h_i^{\star,p}(x_t)\}\right] \frac{\mathbb{P}\left[B_t = 1\right]}{\mathbb{P}\left[B_t = 1\right]}.$$

Since $\mathbb{P}\left[B_t = 1\right] = \mathbb{E}\left[\mathbb{1}\{B_t = 1\}\right] = \frac{T^\beta}{T}$, we have

$$\sum_{t=1}^{T} \mathbb{E}\left[\mathbb{1}\{\mathcal{A}_i^p(x_t \mid L_{B_{<t}}^{h^\star}(i,p)) \neq h_i^{\star,p}(x_t)\}\right] \frac{\mathbb{P}\left[B_t = 1\right]}{\mathbb{P}\left[B_t = 1\right]} = \frac{T}{T^\beta} \sum_{t=1}^{T} \mathbb{E}\left[\mathbb{1}\{\mathcal{A}_i^p(x_t \mid L_{B_{<t}}^{h^\star}(i,p)) \neq h_i^{\star,p}(x_t)\}\right] \mathbb{E}\left[\mathbb{1}\{B_t = 1\}\right]$$

Using the independence of B_t and the algorithm's prediction in round t,

$$\frac{T}{T^\beta}\sum_{t=1}^{T}\mathbb{E}\left[\mathbb{1}\{\mathcal{A}_i^p(x_t \mid L_{B_{<t}}^{h^\star}(i,p)) \neq h_i^{\star,p}(x_t)\}\right]\mathbb{E}\left[\mathbb{1}\{B_t=1\}\right] \\ = \frac{T}{T^\beta}\sum_{t=1}^{T}\mathbb{E}\left[\mathbb{1}\{\mathcal{A}_i^p(x_t \mid L_{B_{<t}}^{h^\star}(i,p)) \neq h_i^{\star,p}(x_t)\}\mathbb{1}\left\{B_t=1\right\}\right].$$

Hence, all together,

$$\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}\{\mathcal{A}_i^p(x_t \mid L_{B_{<t}}^{h^\star}(i,p)) \neq h_i^{\star,p}(x_t)\}\right] \\ = \frac{T}{T^\beta}\sum_{t=1}^{T}\mathbb{E}\left[\mathbb{1}\{\mathcal{A}_i^p(x_t \mid L_{B_{<t}}^{h^\star}(i,p)) \neq h_i^{\star,p}(x_t)\}\mathbb{1}\left\{B_t=1\right\}\right].$$

Next, we can use the regret guarantee of the algorithm $\mathcal{A}_i^p$ on the rounds it was updated. That is,

$$\sum_{t=1}^{T}\mathbb{E}\left[\mathbb{1}\{\mathcal{A}_i^p(x_t \mid L_{B_{<t}}^{h^\star}(i,p)) \neq h_i^{\star,p}(x_t)\}\mathbb{1}\left\{B_t=1\right\}\right] \\ = \mathbb{E}\left[\mathbb{E}\left[\sum_{t:B_t=1}\mathbb{1}\{\mathcal{A}_i^p(x_t \mid L_{B_{<t}}^{h^\star}(i,p)) \neq h_i^{\star,p}(x_t)\}\,\middle|\,B\right]\right] \leq \mathbb{E}_B\left[R_i^p(|B|)\right],$$

where $R_i^p(|B|)$ is the regret of $\mathcal{A}_i^p$, a sub-linear function of $|B|$. In the last step, we use the fact that $\mathcal{A}_i^p$ is a realizable algorithm for $\mathcal{H}_i^p$ and the feedback that the algorithm received was $(x_t, h_i^{\star,p}(x_t))$ in the rounds whenever $B_t = 1$. By Lemma 5.17 from Woess [2017], there exists a concave sub-linear function $\tilde{R}_i^p(|B|)$ that upperbounds $R_i^p(|B|)$. By Jensen's inequality, $\mathbb{E}_B\left[R_i^p(|B|)\right] \leq \tilde{R}_i^p(T^\beta)$, a sub-linear function of T^β.

Putting everything together, we obtain

$$\mathbb{E}\left[\sum_{t=1}^{T}\ell(\mathcal{Q}(x_t),y_t)\right] \leq \sum_{t=1}^{T}\ell(h^\star(x_t),y_t) + \frac{M^2}{a}\sum_{i=1}^{K}\frac{T}{T^\beta}\,\tilde{R}_i^p(T^\beta) + M\sqrt{2T^{1+\beta}K\ln K} \\ = \inf_{h\in\mathcal{H}}\sum_{t=1}^{T}\ell(h(x_t),y_t) + \frac{pM^2}{a}\sum_{i=1}^{K}\frac{T}{T^\beta}\,\tilde{R}_i^p(T^\beta) + M\sqrt{2T^{1+\beta}K\ln K}.$$

Since $\tilde{R}_i^p(T^\beta)$ is a sublinear function of T^β, we have that $\frac{T}{T^\beta}\tilde{R}_i^p(T^\beta)$ is a sublinear function of T. As the sum of sublinear functions is sublinear, the second term above must be a sublinear function of T. Thus, the regret is sub-linear for any choice of $\beta \in (0,1)$. This

completes our proof as we have shown that the algorithm $\mathcal{Q}$ achieves sub-linear regret in T. ■

We will now show that the online learnability of $\mathcal{H}$ w.r.t ℓ implies that $\mathcal{H}_i^p$ for each $i \in [K]$ is online learnable w.r.t 0-1 loss.

Proof. (of necessity in Theorem 7.5.2)

Fix $\ell \in \mathcal{L}(\ell_{\text{prec}}^{@p})$ and let $M = \max_{\pi,y} \ell(\pi, y)$. Given an online learner $\mathcal{A}$ for $\mathcal{H}$ w.r.t ℓ, our goal is to construct an agnostic online learner $\mathcal{A}_i^p$ for $\mathcal{H}_i^p$ for a fixed $i \in [K]$. One can construct agnostic online learners for $\mathcal{H}_i^p$ for all $i \in [K]$ by symmetry. Our construction uses the REWA and is similar to the sufficiency proof above.

Let us define function ϕ's, the collection of functions Φ_b for every b in the same way we did before. For every $b \in \{0,1\}^T$ and $\phi \in \Phi_b$, define an Expert $E_{b,\phi}$. Expert $E_{b,\phi}$ is the expert presented in Algorithm 24 after setting $j = p$ and uses $\mathcal{A}$ to make predictions in each round. For every $b \in \{0,1\}^T$, let $\mathcal{E}_b = \bigcup_{\phi \in \Phi_b} \{E_{b,\phi}\}$ denote the set of all Experts parameterized by functions $\phi \in \Phi_b$. As before, we will actually define $\mathcal{E}_b = \{E_0\} \cup \bigcup_{\phi \in \Phi_b} \{E_{b,\phi}\}$, where E_0 is the expert that never updates $\mathcal{A}$ and only uses it to make predictions in each round. Note that $1 \leq |\mathcal{E}_b| \leq (K!)^{|b|} \leq K^{K|b|}$.

The online learner for $\mathcal{H}_i^p$, henceforth denoted by $\mathcal{Q}$, is similar to Algorithm 11. First, it samples a $B \in \{0,1\}^T$ s.t. $B_t \sim \text{Bernoulli}(T^\beta/T)$, constructs a set of experts $\mathcal{E}_B$ using Algorithm 24 and runs REWA, denoted by $\mathcal{P}$, on the 0-1 loss over the stream $(x_1, y_1), ..., (x_T, y_T) \in (\mathcal{X} \times \{0,1\})^T$. Let $h_i^{\star,p} = \arg\min_{h_i^p \in \mathcal{H}_i^p} \sum_{t=1}^T \mathbb{1}\{h_i^p(x_t) \neq y_t\}$ be the optimal function in hindsight and $h^\star$ be any arbitrary completion of $h_i^{\star,p}$.

Using REWA guarantees and following exactly the same calculation as in the sufficiency proof, we arrive at

$$\mathbb{E}\left[\sum_{t=1}^T \mathbb{1}\{\mathcal{Q}(x_t) \neq y_t\}\right] \leq \mathbb{E}\left[\sum_{t=1}^T \mathbb{1}\{E_{B,\phi_B^{h^\star}}(x_t) \neq y_t\}\right] + \sqrt{2T^{1+\beta}K \ln K}.$$

The inequality above is the adaptation of Equation (7.1) for this proof. Since $\mathbb{1}\{E_{B,\phi_B^{h^\star}}(x_t) \neq y_t\} \leq \mathbb{1}\{h_i^{\star,p}(x_t) \neq y_t\} + \mathbb{1}\{E_{B,\phi_B^{h^\star}}(x_t) \neq h_i^{\star,p}(x_t)\}$, the inequality above reduces to

$$\mathbb{E}\left[\sum_{t=1}^T \mathbb{1}\{\mathcal{Q}(x_t) \neq y_t\}\right] \leq \sum_{t=1}^T \mathbb{1}\{h_i^{\star,p}(x_t) \neq y_t\} + \mathbb{E}\left[\sum_{t=1}^T \mathbb{1}\{E_{B,\phi_B^{h^\star}}(x_t) \neq h_i^{\star,p}(x_t)\}\right] + \sqrt{2T^{1+\beta}K \ln K}.$$

It now suffices to show that $\mathbb{E}\left[\sum_{t=1}^T \mathbb{1}\{E_{B,\phi_B^{h^\star}}(x_t) \neq h_i^{\star,p}(x_t)\}\right]$ is sub-linear in T.

Given an online learner $\mathcal{A}$ for $\mathcal{H}$, an instance $x \in \mathcal{X}$, and an ordered finite sequence of labeled examples $L \in (\mathcal{X} \times \mathcal{Y})^*$, let $\mathcal{A}(x|L)$ be the random variable denoting the prediction of $\mathcal{A}$ on the instance x after running and updating on L. For any $b \in \{0,1\}^T$, $h \in \mathcal{H}$, and $t \in [T]$, let $L^h_{b_{<t}} = \{(x_i, \mathrm{BinRel}(h(x_s), p)) : s < t \text{ and } b_s = 1\}$ denote the *subsequence* of the sequence of labeled instances $\{(x_s, \mathrm{BinRel}(h(x_s), p))\}_{s=1}^{t-1}$ where $b_s = 1$. Using Lemma E.5.4, we have

$$\begin{aligned}
\mathbb{1}\{E_{B,\phi^{h^\star}_B}(x_t) \neq h_i^{\star,p}(x_t)\} &\leq \mathbb{1}\{\ell^{@p}_{\mathrm{prec}}(\mathcal{A}(x_t \mid L^{h^\star}_{B_{<t}}), \mathrm{BinRel}(h^\star(x_t), p)) > 0\} \\
&= \mathbb{1}\{\ell(\mathcal{A}(x_t \mid L^{h^\star}_{B_{<t}}), \mathrm{BinRel}(h^\star(x_t), p)) > 0\} \\
&\leq \frac{1}{a}\,\ell(\mathcal{A}(x_t \mid L^{h^\star}_{B_{<t}}), \mathrm{BinRel}(h^\star(x_t), p)),
\end{aligned}$$

where the equality follows from the definition of the loss class. Here, a is the lower bound on ℓ whenever it is non-zero. Thus, we obtain

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{E_{B,\phi^{h^\star}_B}(x_t) \neq h_i^{\star,p}(x_t)\}\right] \leq \frac{1}{a}\mathbb{E}\left[\sum_{t=1}^{T} \ell(\mathcal{A}(x_t \mid L^{h^\star}_{B_{<t}}), \mathrm{BinRel}(h^\star(x_t), p))\right]$$

Now, we will again use the fact that the prediction $\mathcal{A}(x_t \mid L^{h^\star}_{B_{<t}})$ only depends on $(B_1, \ldots, B_{t-1})$, but is independent of B_t. Using this independence, we obtain

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T} \ell(\mathcal{A}(x_t \mid L^{h^\star}_{B_{<t}}), \mathrm{BinRel}(h^\star(x_t), p))\right] \\
&\qquad = \frac{T}{T^\beta}\,\mathbb{E}\left[\mathbb{E}\left[\sum_{t:B_t=1} \ell(\mathcal{A}(x_t \mid L^{h^\star}_{B_{<t}}), \mathrm{BinRel}(h^\star(x_t), p)) \,\middle|\, B\right]\right] \leq \frac{T}{T^\beta}\,\mathbb{E}\left[R(|B|, K)\right],
\end{aligned}$$

where $R(|B|, K)$ is the regret of the algorithm $\mathcal{A}$ and is a sub-linear function of $|B|$. In the last step, we use the fact that $\mathcal{A}$ is a (realizable) online learner for $\mathcal{H}$ w.r.t. ℓ and the feedback that the algorithm received was $(x_t, \mathrm{BinRel}(h^\star(x_t), p))$ in the rounds whenever $B_t = 1$. Again, using Lemma 5.17 from Woess [2017] and Jensen's inequality yields $\mathbb{E}_B\left[R(|B|, K)\right] \leq \tilde{R}(T^\beta, K)$, a concave, sub-linear function of T^β. Combining everything, we get

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{Q}(x_t) \neq h_i^{\star,p}(x_t)\}\right] &\leq \sum_{t=1}^{T} \mathbb{1}\{h_i^{\star,p}(x_t) \neq y_t\} + \frac{T}{a\,T^\beta}\tilde{R}(T^\beta, K) + \sqrt{2T^{1+\beta}K\ln K} \\
&\leq \inf_{h_i^p \in \mathcal{H}_i^p} \sum_{t=1}^{T} \mathbb{1}\{h_i^p(x_t) \neq y_t\} + \frac{T}{a\,T^\beta}\tilde{R}(T^\beta, K) + \sqrt{2T^{1+\beta}K\ln K}
\end{aligned}$$

For any choice of $\beta \in (0,1)$, the regret above is a sub-linear function of T. Therefore, we have shown that $\mathcal{Q}$ is an agnostic learner for $\mathcal{H}_i^p$ w.r.t. 0-1 loss. This completes our proof. ■

E.5 Technical Lemmas

Throughout this section, for any ranking (permutation) $\pi \in \mathcal{S}_K$, we let $\pi_i^j = \mathbb{1}\{\pi_i \leq j\}$ for all $(i,j) \in [K]$.

Lemma E.5.1. *For any $y \in \mathcal{Y}$, $(\pi, \hat{\pi}) \in \mathcal{S}_k$, and $\ell \in \mathcal{L}(\ell_{sum}^{@p})$*

$$\ell(\pi, y) \leq \ell(\hat{\pi}, y) + c\, p\, \mathbb{E}_{j \sim Unif([p])} \left[\ell(\pi, BinRel(\hat{\pi}, j))\right].$$

where $c = \frac{\max_{\tilde{\pi}, y} \ell(\tilde{\pi}, y)}{\min_{\tilde{\pi}, y}\{\ell(\tilde{\pi}, y) \,|\, \ell(\tilde{\pi}, y) \neq 0\}}$.

Proof. Assume that $\ell(\pi, y) > \ell(\hat{\pi}, y) \geq 0$ (as otherwise the inequality trivially holds). Then, since $\ell \in \mathcal{L}(\ell_{\text{sum}}^{@p})$, it must be the case that $\hat{\pi} \overset{[p]}{\neq} \pi$. That is, $\hat{\pi}$ and π assign different ranks to the labels in the top p. Therefore, there exists $i \in [p]$ s.t. $\ell_{\text{sum}}^{@p}(\pi, \text{BinRel}(\hat{\pi}, i)) > 0$. Since $\ell \in \mathcal{L}(\ell_{\text{sum}}^{@p})$, for this same $i \in [p]$, $\ell(\pi, \text{BinRel}(\hat{\pi}, i)) > 0$. Therefore, we have

$$\begin{aligned}
c\, p\, \mathbb{E}_{j \sim \text{Unif}([p])} \left[\ell(\pi, \text{BinRel}(\hat{\pi}, j))\right] &\geq c\ell(\pi, \text{BinRel}(\hat{\pi}, i)) \\
&= \frac{\max_{\tilde{\pi}, y} \ell(\tilde{\pi}, y)}{\min_{\tilde{\pi}, y}\{\ell(\tilde{\pi}, y) \;|\; \ell(\tilde{\pi}, y) \neq 0\}} \ell(\pi, \text{BinRel}(\hat{\pi}, i)) \\
&\geq \max_{\tilde{\pi}, y} \ell(\tilde{\pi}, y) \\
&\geq \ell(\pi, y).
\end{aligned}$$

Combining the upperbounds in both cases gives the desired inequality. ■

Lemma E.5.2. *For any $y \in \mathcal{Y}$, $(\pi, \hat{\pi}) \in \mathcal{S}_k$, and $\ell \in \mathcal{L}(\ell_{prec}^{@p})$*

$$\ell(\pi, y) \leq \ell(\hat{\pi}, y) + c\, \ell(\pi, BinRel(\hat{\pi}, p)).$$

where $c = \frac{\max_{\tilde{\pi}, y} \ell(\tilde{\pi}, y)}{\min_{\tilde{\pi}, y}\{\ell(\tilde{\pi}, y) \,|\, \ell(\tilde{\pi}, y) \neq 0\}}$.

Proof. Assume that $\ell(\pi, y) > \ell(\hat{\pi}, y) \geq 0$ (as otherwise the inequality trivially holds). Then, since $\ell \in \mathcal{L}(\ell_{\text{prec}}^{@p})$, it must be the case that $\hat{\pi} \overset{p}{\neq} \pi$. That is, $\hat{\pi}$ and π assign different labels in the top p. Therefore, $\ell_{\text{prec}}^{@p}(\pi, \text{BinRel}(\hat{\pi}, p)) > 0$. Since $\ell \in \mathcal{L}(\ell_{\text{prec}}^{@p})$, $\ell(\pi, \text{BinRel}(\hat{\pi}, p)) > 0$.

Therefore, we have

$$\begin{aligned} c\,\ell(\pi, \mathrm{BinRel}(\hat{\pi}, p)) &= \frac{\max_{\tilde{\pi},y} \ell(\tilde{\pi}, y)}{\min_{\tilde{\pi},y}\{\ell(\tilde{\pi}, y) \mid \ell(\tilde{\pi}, y) \neq 0\}} \ell(\pi, \mathrm{BinRel}(\hat{\pi}, p)) \\ &\geq \max_{\tilde{\pi},y} \ell(\tilde{\pi}, y) \\ &\geq \ell(\pi, y). \end{aligned}$$

Combining the upperbounds in both cases gives the desired inequality. ■

Lemma E.5.3. *Let $\pi, \hat{\pi} \in \mathcal{S}_k$. Then, for all $(i, j) \in [K] \times [p]$, $\ell^{@p}_{sum}(\pi, BinRel(\hat{\pi}, j)) \geq \mathbb{1}\{\pi_i^j \neq \hat{\pi}_i^j\}$.*

Proof. Fix label $i^\star \in [K]$ and threshold $j^\star \in [p]$. Our goal is to show that $\ell^{@p}_{\mathrm{sum}}(\pi, \mathrm{BinRel}(\hat{\pi}, j^\star)) \geq \mathbb{1}\{\pi_{i^\star}^{j^\star} \neq \hat{\pi}_{i^\star}^{j^\star}\}$. Recall that $\mathrm{BinRel}(\hat{\pi}, j^\star)[i^\star] = \mathbb{1}\{\hat{\pi}_{i^\star} \leq j^\star\}$ by definition. Since $\ell^{@p}_{\mathrm{sum}}(\hat{\pi}, \mathrm{BinRel}(\hat{\pi}, j^\star)) = 0$, we have that

$$\begin{aligned} \ell^{@p}_{\mathrm{sum}}(\pi, \mathrm{BinRel}(\hat{\pi}, j^\star)) &= \ell^{@p}_{\mathrm{sum}}(\pi, \mathrm{BinRel}(\hat{\pi}, j^\star)) - \ell^{@p}_{\mathrm{sum}}(\hat{\pi}, \mathrm{BinRel}(\hat{\pi}, j^\star)) \\ &= \sum_{i=1}^{K} \min(\pi_i, p+1)\mathrm{BinRel}(\hat{\pi}, j^\star)[i] - \min(\hat{\pi}_i, p+1)\mathrm{BinRel}(\hat{\pi}, j^\star)[i] \\ &= \sum_{i=1}^{K} \min(\pi_i, p+1)\mathbb{1}\{\hat{\pi}_i \leq j^\star\} - \sum_{i=1}^{K} \min(\hat{\pi}_i, p+1)\mathbb{1}\{\hat{\pi}_i \leq j^\star\} \\ &= \sum_{i=1}^{K} \min(\pi_i, p+1)\mathbb{1}\{\hat{\pi}_i \leq j^\star\} - \sum_{i=1}^{K} \hat{\pi}_i \mathbb{1}\{\hat{\pi}_i \leq j^\star\} \end{aligned}$$

Let $\mathcal{I} \subseteq [K]$ s.t. for all $i \in \mathcal{I}$, $\hat{\pi}_i^{j^\star} = \mathbb{1}\{\hat{\pi}_i \leq j^\star\} = 1$. Then, we have that

$$\begin{aligned} \ell^{@p}_{\mathrm{sum}}(\pi, \mathrm{BinRel}(\hat{\pi}, j^\star)) &= \sum_{i \in \mathcal{I}} \min(\pi_i, p+1) - \sum_{i \in \mathcal{I}} \hat{\pi}_i \\ &= \sum_{i \in \mathcal{I}} \min(\pi_i, p+1) - \sum_{i=1}^{j^\star} i \end{aligned}$$

Suppose that $\mathbb{1}\{\pi_{i^\star}^{j^\star} \neq \hat{\pi}_{i^\star}^{j^\star}\} = 1$. It suffices to show that $\ell^{@p}_{\mathrm{sum}}(\pi, \mathrm{BinRel}(\hat{\pi}, j^\star)) \geq 1$. There are two cases to consider. Suppose $i^\star \in \mathcal{I}$. Then, it must be the case that $\mathbb{1}\{\pi_{i^\star} \leq j^\star\} = \pi_{i^\star}^{j^\star} = 0$, implying that $\pi_{i^\star} \geq j^\star + 1$. It then follows that in the best case $\sum_{i \in \mathcal{I}} \min(\pi_i, p+1) \geq \sum_{i=1}^{j^\star - 1} i + (j^\star + 1) > \sum_{i=1}^{j^\star} i$ showcasing that indeed $\ell^{@p}_{\mathrm{sum}}(\pi, \mathrm{BinRel}(\hat{\pi}, j)) \geq 1$. Now, suppose $i^\star \notin \mathcal{I}$. Then, $\mathbb{1}\{\hat{\pi}_{i^\star} \leq j^\star\} = 0$, which means that $\mathbb{1}\{\pi_{i^\star} \leq j^\star\} = 1$. Accordingly, while $\hat{\pi}$

did not rank label $i^\star$ in the top $j^\star$, π *did* rank label $i^\star$ in the top $j^\star$. Since $|\mathcal{I}| = j^\star$, there must exist an label $\hat{i} \in \mathcal{I}$ which π does not rank in the top $j^\star$. That is, there exists $\hat{i} \in \mathcal{I}$ s.t. $\pi_{\hat{i}} \geq j^\star + 1$. Using the same logic, in the best case $\sum_{i \in \mathcal{I}} \min(\pi_i, p+1) \geq \sum_{i=1}^{j-1} i + (j^\star + 1)$ showcasing that again $\ell^{@p}_{\text{sum}}(\pi, \text{BinRel}(\hat{\pi}, j^\star)) \geq 1$. Thus, we have shown that when $\mathbb{1}\{\pi^{j^\star}_{i^\star} \neq \hat{\pi}^{j^\star}_{i^\star}\} = 1$, $\ell^{@p}_{\text{sum}}(\pi, \text{BinRel}(\hat{\pi}, j^\star)) \geq 1$. Since $i^\star$ and $j^\star$ were arbitrary, this must be true for any $(i, j) \in [K] \times [p]$, completing the proof. ■

Lemma E.5.4. *Let $\pi, \hat{\pi} \in \mathcal{S}_k$. Then, for all $i \in [K]$, $\ell^{@p}_{prec}(\pi, BinRel(\hat{\pi}, p)) \geq \mathbb{1}\{\pi^p_i \neq \hat{\pi}^p_i\}$.*

Proof. Fix label $i^\star \in [K]$. Our goal is to show that $\ell^{@p}_{\text{prec}}(\pi, \text{BinRel}(\hat{\pi}, p)) \geq \mathbb{1}\{\pi^p_{i^\star} \neq \hat{\pi}^p_{i^\star}\}$. Recall that $\text{BinRel}(\hat{\pi}, p)[i^\star] = \mathbb{1}\{\hat{\pi}_{i^\star} \leq p\}$ by definition. Since $\ell^{@p}_{\text{prec}}(\hat{\pi}, \text{BinRel}(\hat{\pi}, p)) = 0$, we have that

$$\begin{aligned}
\ell^{@p}_{\text{prec}}(\pi, \text{BinRel}(\hat{\pi}, p)) &= \ell^{@p}_{\text{prec}}(\pi, \text{BinRel}(\hat{\pi}, p)) - \ell^{@p}_{\text{prec}}(\hat{\pi}, \text{BinRel}(\hat{\pi}, p)) \\
&= \sum_{i=1}^{K} \mathbb{1}\{\hat{\pi}_i \leq p\}\text{BinRel}(\hat{\pi}, p)[i] - \sum_{i=1}^{K} \mathbb{1}\{\pi_i \leq p\}\text{BinRel}(\hat{\pi}, p)[i] \\
&= p - \sum_{i=1}^{K} \mathbb{1}\{\pi_i \leq p\}\mathbb{1}\{\hat{\pi}_i \leq p\}
\end{aligned}$$

Let $\mathcal{I} \subseteq [K]$ s.t. for all $i \in \mathcal{I}$, $\hat{\pi}^p_i = \mathbb{1}\{\hat{\pi}_i \leq p\} = 1$. Then, we have that

$$\ell^{@p}_{\text{prec}}(\pi, \text{BinRel}(\hat{\pi}, p)) = p - \sum_{i \in \mathcal{I}} \mathbb{1}\{\pi_i \leq p\}.$$

Suppose that $\mathbb{1}\{\pi^p_{i^\star} \neq \hat{\pi}^p_{i^\star}\} = 1$. It suffices to show that $\ell^{@p}_{\text{prec}}(\pi, \text{BinRel}(\hat{\pi}, p)) \geq 1$. There are two cases to consider. Suppose $i^\star \in \mathcal{I}$. Then, it must be the case that $\mathbb{1}\{\pi_{i^\star} \leq p\} = \pi^p_{i^\star} = 0$, implying that $\pi_{i^\star} \geq p+1$. It then follows that in the best case $\sum_{i \in \mathcal{I}} \mathbb{1}\{\pi_i \leq p\} \leq p-1 < p$ showcasing that indeed $\ell^{@p}_{\text{sum}}(\pi, \text{BinRel}(\hat{\pi}, p)) \geq 1$. Now, suppose $i^\star \notin \mathcal{I}$. Then, $\mathbb{1}\{\hat{\pi}_{i^\star} \leq p\} = 0$, which means that $\mathbb{1}\{\pi_{i^\star} \leq p\} = 1$. Accordingly, while $\hat{\pi}$ did not rank label $i^\star$ in the top p, π *did* rank label $i^\star$ in the top p. Since $|\mathcal{I}| = p$, there must exist an label $\hat{i} \in \mathcal{I}$ which π does not rank in the top p. That is, there exists $\hat{i} \in \mathcal{I}$ s.t. $\pi_{\hat{i}} \geq p + 1$. Using the same logic, in the best case $\sum_{i \in \mathcal{I}} \mathbb{1}\{\pi_i \leq p\} \leq p - 1 < p$ showcasing that again $\ell^{@p}_{\text{prec}}(\pi, \text{BinRel}(\hat{\pi}, p)) \geq 1$. Thus, we have shown that when $\mathbb{1}\{\pi^p_{i^\star} \neq \hat{\pi}^p_{i^\star}\} = 1$, $\ell^{@p}_{\text{prec}}(\pi, \text{BinRel}(\hat{\pi}, p)) \geq 1$. Since $i^\star$ was arbitrary, this must be true for any $i \in [K]$, completing the proof. ■

APPENDIX F

Estimating the (Un)seen: Sample-dependent Mass Estimation

F.1 Definition of VC dimension

Definition F.1.1 (VC dimension [Vapnik and Chervonenkis, 1971])**.** *A set $\{x_1, ..., x_n\} \in \mathcal{X}$ is shattered by $\mathcal{H}$, if $\forall y_1, ..., y_n \in \{0, 1\}$, $\exists h \in \mathcal{H}$, such that $\forall i \in [n]$, $h(x_i) = y_i$. The VC dimension of $\mathcal{H}$, denoted $\mathrm{VC}(\mathcal{H})$, is defined as the largest natural number $n \in \mathbb{N}$ such that there exists a set $\{x_1, ..., x_n\} \in \mathcal{X}$ that is shattered by $\mathcal{H}$.*

F.2 Missing Proofs

Here, we give a proof of the claim that we can upgrade Theorem 8.4.1 to hold for the MSE loss after replacing $\tau_g(n)$ with $\sup_{x_{1:n} \in \mathcal{X}^n} g(x_{1:n})$.

Theorem F.2.1. *Let $g : \mathcal{X}^\star \to 2^{\mathcal{X}}$ be any function such that for all $n \geq 1$, we have that $c_n := \sup_{x_{1:n} \in \mathcal{X}^n} |g(x_{1:n})| < \infty$. Then,*

$$\sup_{D \in \Delta\mathcal{X}} \mathop{\mathbb{E}}_{x_{1:n} \sim D^n} \left[\left(\hat{f}^{\mathrm{emp}}(x_{1:n}) - D(g(x_{1:n})) \right)^2 \right] \leq \frac{c_n}{n}.$$

Proof. Fix a distribution $D \in \Delta\mathcal{X}$ and a sample $x_{1:n} \sim D^n$. Define $\hat{D}_n(x) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}\{x_i = x\}$. Then, we can write

$$\hat{f}^{\mathrm{emp}}(x_{1:n}) - D(g(x_{1:n}))) = \sum_{x \in \mathcal{X}} (\hat{D}_n(x) - D(x)) \mathbf{1}\{x \in g(x_{1:n})\}.$$

By Cauchy-Schwarz, we have that

$$\left(\sum_{x\in\mathcal{X}}(\hat{D}_n(x)-D(x))\mathbf{1}\{x\in g(x_{1:n})\}\right)^2 \leq \left(\sum_{x\in\mathcal{X}}\mathbf{1}\{x\in g(x_{1:n})\}^2\right)\left(\sum_{x\in\mathcal{X}}(\hat{D}_n(x)-D(x))^2\right)$$
$$= c_n\left(\sum_{x\in\mathcal{X}}(\hat{D}_n(x)-D(x))^2\right)$$

After taking expectation on both sides, it suffices to upper bound

$$\mathbb{E}_{x_{1:n}\sim D^n}\left[\sum_{x\in\mathcal{X}}(\hat{D}_n(x)-D(x))^2\right] = \sum_{x\in\mathcal{X}}\mathbb{E}_{x_{1:n}\sim D^n}\left[(\hat{D}_n(x)-D(x))^2\right].$$

Since $\hat{D}_n(x)$ is an unbiased estimator of $D(x)$, we have that

$$\mathbb{E}_{x_{1:n}\sim D^n}\left[(\hat{D}_n(x)-D(x))^2\right] = \mathrm{Var}(\hat{D}_n(x)) = \frac{D(x)(1-D(x))}{n} \leq \frac{D(x)}{n}.$$

hence,

$$\mathbb{E}_{x_{1:n}\sim D^n}\left[\sum_{x\in\mathcal{X}}(\hat{D}_n(x)-D(x))^2\right] \leq \frac{1}{n},$$

which completes the proof. ■

F.3 Mass Estimation is not equivalent to Classical Mass Estimation

In this section, we give an example of a g function that is estimable accordingly to Definition 8.3.2 but not classically according to Definition 8.3.3. This results highlights that one needs to go beyond standard estimation lower bound techniques, like LeCam's two-point method [LeCam, 1973] and Fano's inequality [Fano, 1966], when considering estimation tasks where the parameter we are trying to estimate is not fixed, but rather is sample dependent.

The following theorem gives a lower bound for classical estimation.

Theorem F.3.1. *Let $\mathcal{X} = \mathbb{N}\cup\{0\}$ and consider the function $h : \mathcal{X} \to 2^{\mathcal{X}}$ defined as $h(0) = \mathbb{N}\cup\{0\}$ and $h(x) = \{x\}$ for $x \neq 0$. Let $g(x_{1:n}) = \bigcup_{i\in[n]} h(x_i)$. Then,*

$$\inf_{\hat{f}}\sup_{D\in\Delta\mathcal{X}}\mathbb{E}_{x_{1:n}\sim D^n}\left[\left|\hat{f}(x_{1:n}) - \mathbb{E}_{z_{1:n}\sim D^n}\left[D(g(z_{1:n}))\right]\right|\right] = \Omega(1).$$

Moreover, there exists an estimator $\hat{f}$ such that

$$\sup_{D\in\Delta\mathcal{X}} \mathop{\mathbb{E}}_{x_{1:n}\sim D^n}\left[\left(\hat{f}(x_{1:n}) - D(g(x_{1:n}))\right)^2\right] = O\left(\frac{1}{n}\right).$$

The upper bound follows from Theorem 8.5.2, hence we only focus on proving the lower bound.

Proof. (of lower bound in Theorem F.3.1) Fix a sample size $n \geq 2$ and let $\delta = \frac{1}{2n}$. Let $M = n^2$. Define the distribution D_0 such that $D_0(0) = 0$, $D_0(i) = \frac{1}{M}$ for $i \in [M]$ and $D_0(i) = 0$ for all $i \geq M+1$. Define the distribution D_1 such that $D_1(0) = \delta$, $D_1(i) = \frac{1-\delta}{M}$ for $i \in [M]$. and $D_1(i) = 0$ for $i \geq M+1$. First, note that

$$\mathrm{TV}(D_0, D_1) = \frac{1}{2}\sum_{i\in\mathbb{N}\cup\{0\}} |D_1(i) - D_0(i)| = \frac{\delta}{2} + \frac{1}{2}\sum_{i\in[M]}(\frac{1}{M} - \frac{1-\delta}{M}) = \delta = \frac{1}{2n}.$$

Hence,

$$\mathrm{TV}(D_0^n, D_1^n) \leq \frac{1}{2}.$$

Now, observe that

$$\mathbb{E}_{x_{1:n}\sim D_0}\left[D_0(g(x_{1:n}))\right] = D_0(\{x_{1:n}\}) = \frac{\mathbb{E}\left[\sum_{j=1}^M \mathbf{1}\{j \in \{x_{1:n}\}\}\right]}{M} = 1 - \left(1 - \frac{1}{M}\right)^n \leq \frac{n}{M}.$$

Likewise,

$$\mathbb{E}_{x_{1:n}\sim D_1}\left[D_1(g(x_{1:n}))\right] = \mathbb{E}_{x_{1:n}\sim D_1}\left[D_1(g(x_{1:n}))|0 \in \{x_{1:n}\}\right]\mathbb{P}\left[0 \in \{x_{1:n}\}\right] + \mathbb{E}_{x_{1:n}\sim D_1}\left[D_1(g(x_{1:n}))|0 \notin \{x_{1:n}\}\right]\mathbb{P}\left[0 \notin \{x_{1:n}\}\right].$$

Note that $\mathbb{P}\left[0 \in \{x_{1:n}\}\right] = 1 - \mathbb{P}\left[0 \notin \{x_{1:n}\}\right] = 1 - (1-\delta)^n$.Hence,

$$\mathbb{E}_{x_{1:n}\sim D_1}\left[D_1(g(x_{1:n}))\right] = \mathbb{E}_{x_{1:n}\sim D_1}\left[D_1(g(x_{1:n}))|0 \in \{x_{1:n}\}\right](1 - (1-\delta)^n) + \mathbb{E}_{x_{1:n}\sim D_1}\left[D_1(g(x_{1:n}))|0 \notin \{x_{1:n}\}\right]((1-\delta)^n).$$

If $0 \in \{x_{1:n}\}$, $g(x_{1:n}) = \mathbb{N}\cup\{0\}$, hence

$$\mathbb{E}_{x_{1:n}\sim D_1}\left[D_1(g(x_{1:n}))\right] = (1 - (1-\delta)^n) + \mathbb{E}_{x_{1:n}\sim D_1}\left[D_1(g(x_{1:n}))|0 \notin \{x_{1:n}\}\right]((1-\delta)^n).$$

The conditional distribution D_1 given $0 \notin \{x_{1:n}\}$ is uniform over $[M]$, hence

$$\mathbb{E}_{x_{1:n}\sim D_1}\left[D_1(g(x_{1:n}))|0 \notin \{x_{1:n}\}\right] = (1-\delta)\left(1-\left(1-\frac{1}{M}\right)^n\right).$$

Let $p_n = 1-(1-\delta)^n$ and $B = 1-\left(1-\frac{1}{M}\right)^n$. Then, we can write

$$\mathbb{E}_{x_{1:n}\sim D_1}\left[D_1(g(x_{1:n}))\right] - \mathbb{E}_{x_{1:n}\sim D_0}\left[D_0(g(x_{1:n}))\right] = p_n + (1-p_n)(1-\delta)B - B = p_n(1-B) - B\delta(1-p_n).$$

Note that

$$p_n = 1-(1-\delta)^n \geq 1-e^{n\delta} = 1-e^{-1/2}.$$

Moreover, for $n \geq 2$, we have that

$$1-B = (1-\frac{1}{M})^n = (1-\frac{1}{n^2})^n \geq 1-\frac{1}{n} \geq \frac{1}{2}$$

giving that $p_n(B-1) \geq \frac{1-e^{-1/2}}{2}$. On the other hand,

$$B\delta(1-p_n) \leq \frac{n}{M}\frac{1}{2n}e^{-1/2} = \frac{e^{-1/2}}{2n^2} \leq \frac{e^{-1/2}}{8}.$$

Thus,

$$|\mathbb{E}_{x_{1:n}\sim D_1}\left[D_1(g(x_{1:n}))\right] - \mathbb{E}_{x_{1:n}\sim D_0}\left[D_0(g(x_{1:n}))\right]| \geq \frac{1-e^{-1/2}}{2} - \frac{e^{-1/2}}{8} =: c > 0.$$

Now, we use LeCam's two point method [Yu, 1997]. Namely, we have that

$$\inf_{\hat{f}} \sup_{D\in\Delta\mathcal{X}} \mathop{\mathbb{E}}_{x_{1:n}\sim D^n}\left[|\hat{f}(x_{1:n}) - \mathop{\mathbb{E}}_{z_{1:n}\sim D^n}\left[D(g(z_{1:n})\right]|\right] \geq \frac{c}{2}(1-\mathrm{TV}(D_0^n, D_1^n)) \geq \frac{c}{4}.$$

Since the right-hand side does not decay with $n \geq 2$, we have shown that g is not classically strongly estimable. ■

BIBLIOGRAPHY

Yasin Abbasi-Yadkori and Csaba Szepesvári. Regret bounds for the adaptive control of linear quadratic systems. In Proceedings of the 24th Annual Conference on Learning Theory, pages 1–26. JMLR Workshop and Conference Proceedings, 2011.

Jayadev Acharya, Yelun Bao, Yuheng Kang, and Ziteng Sun. Improved bounds for minimax risk of estimating missing mass. In 2018 IEEE International Symposium on Information Theory (ISIT), pages 326–330. IEEE, 2018.

Roy L Adler, Alan G Konheim, and M Harry McAndrew. Topological entropy. Transactions of the American Mathematical Society, 114(2):309–319, 1965.

Naman Agarwal, Brian Bullins, Elad Hazan, Sham Kakade, and Karan Singh. Online control with adversarial disturbances. In International Conference on Machine Learning, pages 111–119. PMLR, 2019.

Matteo Almanza, Flavio Chierichetti, Silvio Lattanzi, Alessandro Panconesi, and Giuseppe Re. Online facility location with multiple advice. Advances in Neural Information Processing Systems, 34:4661–4673, 2021.

N. Alon, O. Ben-Eliezer, Y. Dagan, S. Moran, M. Naor, and E. Yogev. Adversarial laws of large numbers and optimal regret in online classification. arXiv:2101.09054, 2021a.

Noga Alon, Nicolo Cesa-Bianchi, Ofer Dekel, and Tomer Koren. Online learning with feedback graphs: Beyond bandits. In Conference on Learning Theory, pages 23–35. PMLR, 2015.

Noga Alon, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private pac learning implies finite littlestone dimension. In Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, pages 852–860, 2019.

Noga Alon, Amos Beimel, Shay Moran, and Uri Stemmer. Closure properties for private classification and online prediction. In Conference on Learning Theory, pages 119–152. PMLR, 2020.

Noga Alon, Omri Ben-Eliezer, Yuval Dagan, Shay Moran, Moni Naor, and Eylon Yogev. Adversarial laws of large numbers and optimal regret in online classification. In Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing, pages 447–455, 2021b.

Oren Anava, Elad Hazan, Shie Mannor, and Ohad Shamir. Online learning for time series prediction. In Conference on learning theory, pages 172–184. PMLR, 2013.

Oren Anava, Elad Hazan, and Assaf Zeevi. Online time series prediction with missing data. In International conference on machine learning, pages 2191–2199. PMLR, 2015.

Antonios Antoniadis, Christian Coester, Marek Eliáš, Adam Polak, and Bertrand Simon. Online metric algorithms with untrusted predictions. ACM Transactions on Algorithms, 19(2):1–34, 2023.

Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. Theory of computing, 8(1):121–164, 2012.

Karl Johan Åström and Peter Eykhoff. System identification—a survey. Automatica, 7(2): 123–162, 1971.

Peter Auer and Philip M Long. Structural results about on-line learning models with and without queries. Machine Learning, 36(3):147–181, 1999.

Sohail Bahmani and Justin Romberg. Convex programming for estimation in nonlinear recurrent models. The Journal of Machine Learning Research, 21(1):9563–9582, 2020.

Ivo M Baltruschat, Hannes Nickisch, Michael Grass, Tobias Knopp, and Axel Saalbach. Comparison of deep learning approaches for multi-label chest x-ray classification. Scientific reports, 9(1):1–10, 2019.

Etienne Bamas, Andreas Maggiori, and Ola Svensson. The primal-dual method for learning augmented algorithms. Advances in Neural Information Processing Systems, 33:20083–20094, 2020.

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. Journal of Machine Learning Research, 3(Nov):463–482, 2002.

Gábor Bartók. The role of information in online learning. 2012.

Gábor Bartók, Dean P Foster, Dávid Pál, Alexander Rakhlin, and Csaba Szepesvári. Partial monitoring—classification, regret bounds, and algorithms. Mathematics of Operations Research, 39(4):967–997, 2014.

David Belanger and Sham Kakade. A linear dynamical system model for text. In International Conference on Machine Learning, pages 833–842. PMLR, 2015.

S. Ben-David, N. Cesa-Bianchi, D. Haussler, and P. Long. Characterizations of learnability for classes of $\{0, \ldots, n\}$-valued functions. Journal of Computer and System Sciences, 50: 74–86, 1995.

Shai Ben-David, Eyal Kushilevitz, and Yishay Mansour. Online learning versus offline learning. Machine Learning, 29:45–63, 1997.

Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz. Agnostic online learning. In COLT, volume 3, page 1, 2009.

Daniel Berend and Aryeh Kontorovich. The missing mass problem. Statistics & Probability Letters, 82(6):1102–1110, 2012.

Daniel Berend and Aryeh Kontorovich. On the concentration of the missing mass. 2013.

Tyrus Berry and Suddhasattwa Das. Learning theory for dynamical systems. SIAM Journal on Applied Dynamical Systems, 22(3):2082–2122, 2023.

Tyrus Berry and Suddhasattwa Das. Limits of learning dynamical systems. SIAM Review, 67(1):107–137, 2025.

Adam Block, Yuval Dagan, Noah Golowich, and Alexander Rakhlin. Smoothed online learning is as easy as statistical learning. In Conference on Learning Theory, pages 1716–1786. PMLR, 2022.

Olivier Bousquet and André Elisseeff. Stability and generalization. Journal of machine learning research, 2(Mar):499–526, 2002.

Nataly Brukhim, Daniel Carmon, Irit Dinur, Shay Moran, and Amir Yehudayoff. A characterization of multiclass learnability. In 2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS), pages 943–955. IEEE, 2022.

Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. Foundations and Trends® in Machine Learning, 5(1):1–122, 2012.

Serhat S Bucak, Pavan Kumar Mallapragada, Rong Jin, and Anil K Jain. Efficient multi-label ranking for multi-class learning: application to object recognition. In 2009 IEEE 12th International Conference on Computer Vision, pages 2098–2105. IEEE, 2009.

David Buffoni, Clément Calauzenes, Patrick Gallinari, and Nicolas Usunier. Learning scoring functions with order-preserving losses and standardized supervision. In The 28th International Conference on Machine Learning (ICML 2011), pages 825–832, 2011.

John Bunge, Amy Willis, and Fiona Walsh. Estimating the number of species in microbial diversity studies. Annual Review of Statistics and Its Application, 1(1):427–445, 2014.

Clément Calauzenes, Nicolas Usunier, and Patrick Gallinari. On the (non-) existence of convex, calibrated surrogate losses for ranking. Advances in Neural Information Processing Systems, 25, 2012.

Marco C Campi and Erik Weyer. Finite sample properties of system identification methods. IEEE Transactions on Automatic Control, 47(8):1329–1334, 2002.

Nicolo Cesa-Bianchi and Gábor Lugosi. Prediction, learning, and games. Cambridge university press, 2006.

Prafulla Chandra and Andrew Thangaraj. Missing g-mass: Investigating the missing parts of distributions. IEEE Transactions on Information Theory, 2024.

Karthik Chandrasekhar, Claus Kadelka, Reinhard Laubenbacher, and David Murrugarra. Stability of linear boolean networks. Physica D: Nonlinear Phenomena, 451:133775, 2023.

Anne Chao and Shen-Ming Lee. Estimating the number of classes via sample coverage. Journal of the American statistical Association, 87(417):210–217, 1992.

Justin Chen, Sandeep Silwal, Ali Vakilian, and Fred Zhang. Faster fundamental graph algorithms via learned predictions. In International Conference on Machine Learning, pages 3583–3602. PMLR, 2022.

Jan Chorowski and Navdeep Jaitly. Towards better decoding and language model integration in sequence to sequence models. arXiv preprint arXiv:1612.02695, 2016.

Amanda Clare and Ross D King. Knowledge discovery in multi-label phenotype data. In Principles of Data Mining and Knowledge Discovery: 5th European Conference, PKDD 2001, Freiburg, Germany, September 3–5, 2001 Proceedings 5, pages 42–53. Springer, 2001.

Stéphan Clémenccon, Gábor Lugosi, and Nicolas Vayatis. Ranking and empirical minimization of u-statistics. The Annals of Statistics, 36(2):844–874, 2008. doi: 10.1214/009053607000000910.

Alon Cohen, Avinatan Hasidim, Tomer Koren, Nevena Lazic, Yishay Mansour, and Kunal Talwar. Online linear quadratic control. In International Conference on Machine Learning, pages 1029–1038. PMLR, 2018.

A. Daniely, S. Sabato, S. Ben-David, and S. Shalev-Shwartz. Multiclass learnability and the ERM principle. Journal of Machine Learning Research, 16(12):2377–2404, 2015.

Amit Daniely and Tom Helbertal. The price of bandit information in multiclass online classification. In Conference on Learning Theory, pages 93–104. PMLR, 2013.

Amit Daniely and Shai Shalev-Shwartz. Optimal learners for multiclass problems. In Conference on Learning Theory, pages 287–316. PMLR, 2014.

Amit Daniely, Sivan Sabato, Shai Ben-David, and Shai Shalev-Shwartz. Multiclass learnability and the erm principle. In Sham M. Kakade and Ulrike von Luxburg, editors, Proceedings of the 24th Annual Conference on Learning Theory, volume 19 of Proceedings of Machine Learning Research, pages 207–232, Budapest, Hungary, 09–11 Jun 2011. PMLR.

O. David, S. Moran, and A. Yehudayoff. On statistical learning via the lens of compression. In Advances in Neural Information Processing 29, 2016.

Krzysztof Dembczynski, Wojciech Kotlowski, and Eyke Hüllermeier. Consistent multilabel ranking through univariate losses. arXiv preprint arXiv:1206.6401, 2012.

Gianfranco Doretto, Alessandro Chiuso, Ying Nian Wu, and Stefano Soatto. Dynamic textures. International journal of computer vision, 51:91–109, 2003.

John C Duchi, Lester W Mackey, and Michael I Jordan. On the consistency of ranking algorithms. In ICML, pages 327–334, 2010.

Richard M Dudley. Central limit theorems for empirical measures. The Annals of Probability, pages 899–929, 1978.

Marek Elias, Haim Kaplan, Yishay Mansour, and Shay Moran. Learning-augmented algorithms with explicit predictors. arXiv preprint arXiv:2403.07413, 2024.

Jeffrey L Elman. Language as a dynamical system. Mind as motion: Explorations in the dynamics of cognition, pages 195–223, 1995.

Jon C Ergun, Zhili Feng, Sandeep Silwal, David P Woodruff, and Samson Zhou. Learning-augmented k-means clustering. arXiv preprint arXiv:2110.14094, 2021.

Robert Mario Fano. Transmission of Information: A statistical theory of communications. Mit Press, 1966.

Vitaly Feldman and Jan Vondrak. Generalization bounds for uniformly stable algorithms. Advances in Neural Information Processing Systems, 31, 2018.

Vitaly Feldman and Jan Vondrak. High probability generalization bounds for uniformly stable algorithms with nearly optimal rate. In Conference on learning theory, pages 1270–1279. PMLR, 2019.

Yuval Filmus, Steve Hanneke, Idan Mehalel, and Shay Moran. Optimal prediction using expert advice and randomized littlestone dimension, 2023.

Dylan Foster, Tuhin Sarkar, and Alexander Rakhlin. Learning nonlinear dynamical systems from a single trajectory. In Learning for Dynamics and Control, pages 851–861. PMLR, 2020.

Wei Gao and Zhi-Hua Zhou. On the consistency of multi-label learning. In Proceedings of the 24th annual conference on learning theory, pages 341–358. JMLR Workshop and Conference Proceedings, 2011.

Jesse Geneson. A note on the price of bandit feedback for mistake-bounded online learning. Theoretical Computer Science, 874:42–45, 2021.

Amin Ghadami and Bogdan I Epureanu. Data-driven prediction in dynamical systems: recent developments. Philosophical Transactions of the Royal Society A, 380(2229): 20210213, 2022.

Udaya Ghai, Holden Lee, Karan Singh, Cyril Zhang, and Yi Zhang. No-regret prediction in marginally stable systems. In Conference on Learning Theory, pages 1714–1757. PMLR, 2020.

Dimitrios Giannakis, Amelia Henriksen, Joel A Tropp, and Rachel Ward. Learning to forecast dynamical systems from streaming data. SIAM Journal on Applied Dynamical Systems, 22(2):527–558, 2023.

Sreenivas Gollapudi and Debmalya Panigrahi. Online algorithms for rent-or-buy with expert advice. In International Conference on Machine Learning, pages 2319–2327. PMLR, 2019.

Yunchao Gong, Yangqing Jia, Thomas Leung, Alexander Toshev, and Sergey Ioffe. Deep convolutional ranking for multilabel image annotation. arXiv preprint arXiv:1312.4894, 2013.

Irving J Good. The population frequencies of species and the estimation of population parameters. Biometrika, 40(3-4):237–264, 1953.

Daniele Grattarola, Lorenzo Livi, and Cesare Alippi. Learning graph cellular automata. Advances in Neural Information Processing Systems, 34:20983–20994, 2021.

Nika Haghtalab. Foundation of Machine Learning, by the People, for the People. PhD thesis, Carnegie Mellon University, 2018.

Nika Haghtalab, Tim Roughgarden, and Abhishek Shetty. Smoothed analysis of online and differentially private learning. Advances in Neural Information Processing Systems, 33: 9203–9215, 2020.

Steve Hanneke, Shay Moran, Vinod Raman, Unique Subedi, and Ambuj Tewari. Multiclass online learning and uniform convergence. Proceedings of the 36th Annual Conference on Learning Theory (COLT), 2023.

Steve Hanneke, Shay Moran, and Jonathan Shafer. A trichotomy for transductive online learning. Advances in Neural Information Processing Systems, 36, 2024.

Elad Hazan, Karan Singh, and Cyril Zhang. Learning linear dynamical systems via spectral filtering. Advances in Neural Information Processing Systems, 30, 2017.

Elad Hazan, Holden Lee, Karan Singh, Cyril Zhang, and Yi Zhang. Spectral filtering for general linear dynamical systems. Advances in Neural Information Processing Systems, 31, 2018.

David P Helmbold, Nicholas Littlestone, and Philip M Long. Apple tasting. Information and Computation, 161(2):85–139, 2000a.

David P Helmbold, Nicholas Littlestone, and Philip M Long. On-line learning with linear loss constraints. Information and Computation, 161(2):140–171, 2000b.

Alfons G Hoekstra, Jiri Kroc, and Peter MA Sloot. Simulating complex systems by cellular automata. Springer, 2010.

Max Hopkins, Daniel M. Kane, Shachar Lovett, and Gaurav Mahajan. Realizable learning is all you need. In Proceedings of Thirty Fifth Conference on Learning Theory, volume 178 of Proceedings of Machine Learning Research, pages 3015–3069. PMLR, 02–05 Jul 2022.

Chen-Yu Hsu, Piotr Indyk, Dina Katabi, and Ali Vakilian. Learning-based frequency estimation algorithms. In International Conference on Learning Representations, 2019.

Prateek Jain, Suhas S Kowshik, Dheeraj Nagaraj, and Praneeth Netrapalli. Streaming linear system identification with reverse experience replay. arXiv preprint arXiv:2103.05896, 2021.

Shaofeng H-C Jiang, Erzhi Liu, You Lyu, Zhihao Gavin Tang, and Yubo Zhang. Online facility location with predictions. arXiv preprint arXiv:2110.08840, 2021.

Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In Machine Learning: ECML-98: 10th European Conference on Machine Learning Chemnitz, Germany, April 21–23, 1998 Proceedings, pages 137–142. Springer, 2005.

Young Hun Jung and Ambuj Tewari. Online boosting algorithms for multi-label ranking. In International Conference on Artificial Intelligence and Statistics, pages 279–287. PMLR, 2018.

Sham Kakade, Akshay Krishnamurthy, Kendall Lowrey, Motoya Ohnishi, and Wen Sun. Information theoretic regret bounds for online nonlinear control. Advances in Neural Information Processing Systems, 33:15312–15325, 2020.

Sham M Kakade, Shai Shalev-Shwartz, and Ambuj Tewari. Efficient bandit algorithms for online multiclass prediction. In Proceedings of the 25th international conference on Machine learning, pages 440–447, 2008.

Adam Tauman Kalai and Santosh S Vempala. Calibrated language models must hallucinate. In Proceedings of the 56th Annual ACM Symposium on Theory of Computing, pages 160–171, 2024.

Adam Tauman Kalai, Ofir Nachum, Santosh S Vempala, and Edwin Zhang. Why language models hallucinate. arXiv preprint arXiv:2509.04664, 2025.

Satyen Kale, Ravi Kumar, and Sergei Vassilvitskii. Cross-validation and mean-square stability. In ICS, pages 487–495, 2011.

Stuart Kauffman. Homeostasis and differentiation in random genetic control networks. Nature, 224(5215):177–178, 1969.

David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 137–146, 2003.

Anna Korba, Alexandre Garcia, and Florence d'Alché Buc. A structured prediction approach for label ranking. Advances in Neural Information Processing Systems, 31, 2018.

Milan Korda and Igor Mezić. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. Automatica, 93:149–160, 2018.

Suhas Kowshik, Dheeraj Nagaraj, Prateek Jain, and Praneeth Netrapalli. Near-optimal offline and streaming algorithms for learning non-linear dynamical systems. Advances in Neural Information Processing Systems, 34:8518–8531, 2021a.

Suhas Kowshik, Dheeraj Nagaraj, Prateek Jain, and Praneeth Netrapalli. Streaming linear system identification with reverse experience replay. Advances in Neural Information Processing Systems, 34:30140–30152, 2021b.

Oluwasanmi O Koyejo, Nagarajan Natarajan, Pradeep K Ravikumar, and Inderjit S Dhillon. Consistent multilabel classification. Advances in Neural Information Processing Systems, 28, 2015.

Mark Kozdoba, Jakub Marecek, Tigran Tchrakian, and Shie Mannor. On-line learning of linear dynamical systems: Exponential forgetting in kalman filters. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 4098–4105, 2019.

Tim Kraska, Alex Beutel, Ed H Chi, Jeffrey Dean, and Neoklis Polyzotis. The case for learned index structures. In Proceedings of the 2018 international conference on management of data, pages 489–504, 2018.

Ravi Kumar, Daniel Lokshtanov, Sergei Vassilvitskii, and Andrea Vattani. Near-optimal bounds for cross-validation via loss stability. In International Conference on Machine Learning, pages 27–35. PMLR, 2013.

Sahin Lale, Kamyar Azizzadenesheli, Babak Hassibi, and Anima Anandkumar. Logarithmic regret bound in partially observable linear dynamical systems. Advances in Neural Information Processing Systems, 33:20876–20888, 2020.

Silvio Lattanzi, Thomas Lavastida, Benjamin Moseley, and Sergei Vassilvitskii. Online scheduling via learned weights. In Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pages 1859–1877. SIAM, 2020.

Lucien LeCam. Convergence of estimates under dimensionality restrictions. The Annals of Statistics, pages 38–53, 1973.

Michel Ledoux and Michel Talagrand. Probability in Banach Spaces: isoperimetry and processes, volume 23. Springer Science & Business Media, 1991.

Holden Lee. Improved rates for prediction and identification of partially observed linear dynamical systems. In International Conference on Algorithmic Learning Theory, pages 668–698. PMLR, 2022.

Xiang Li, Jiayi Xin, Qi Long, and Weijie J Su. Evaluating the unseen capabilities: How many theorems do llms know? arXiv preprint arXiv:2506.02058, 2025.

Yingying Li, Xin Chen, and Na Li. Online optimal control with linear dynamics and predictions: Algorithms and regret analysis. Advances in Neural Information Processing Systems, 32, 2019.

Honghao Lin, Tian Luo, and David Woodruff. Learning augmented binary search trees. In International Conference on Machine Learning, pages 13431–13440. PMLR, 2022.

N. Littlestone and M. K. Warmuth. The weighted majority algorithm. Information and Computation, 108(2):212–261, 1994.

Nicholas Littlestone. Mistake bounds and logarithmic linear-threshold learning algorithms. University of California, Santa Cruz, 1989.

Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. Machine Learning, 2:285–318, 1987.

Chenghao Liu, Steven CH Hoi, Peilin Zhao, and Jianling Sun. Online arima algorithms for time series prediction. In Proceedings of the AAAI conference on artificial intelligence, volume 30, 2016.

Tie-Yan Liu et al. Learning to rank for information retrieval. Foundations and Trends® in Information Retrieval, 3(3):225–331, 2009.

Lennart Ljung. System identification: theory for the user. PTR Prentice Hall, Upper Saddle River, NJ, 28:540, 1999.

Philip M Long. New bounds on the price of bandit feedback for mistake-bounded online multiclass learning. In International Conference on Algorithmic Learning Theory, pages 3–10. PMLR, 2017.

Thodoris Lykouris and Sergei Vassilvitskii. Competitive caching with machine learned advice. Journal of the ACM (JACM), 68(4):1–25, 2021.

Andreas Maurer. Concentration of the missing mass in metric spaces. arXiv preprint arXiv:2206.02012, 2022.

David McAllester and Luis Ortiz. Concentration inequalities for the missing mass and for histogram rule error. Journal of Machine Learning Research, 4(Oct):895–911, 2003.

David A McAllester and Robert E Schapire. On the convergence rate of good-turing estimators. In COLT, pages 1–6, 2000.

Andrew Kachites McCallum. Multi-label text classification with a mixture model trained by em. In AAAI'99 workshop on text learning, 1999.

Luis Mendoza and Elena R Alvarez-Buylla. Dynamics of the genetic regulatory network forarabidopsis thalianaflower morphogenesis. Journal of theoretical biology, 193(2):307–319, 1998.

Michael Mitzenmacher and Sergei Vassilvitskii. Algorithms with predictions. Communications of the ACM, 65(7):33–35, 2022.

Omar Montasser, Steve Hanneke, and Nathan Srebro. Vc classes are adversarially robustly learnable, but only improperly. In Conference on Learning Theory, pages 2512–2530. PMLR, 2019.

Shay Moran, Ohad Sharon, Iska Tsubari, and Sivan Yosebashvili. List online classification. In The Thirty Sixth Annual Conference on Learning Theory, pages 1885–1913. PMLR, 2023.

Deepan Muthirayan and Pramod P Khargonekar. Online learning robust control of nonlinear dynamical systems. arXiv preprint arXiv:2106.04092, 2021.

B. K. Natarajan. On learning sets and functions. Mach. Learn., 4(1): 67–97, oct 1989. ISSN 0885-6125. doi: 10.1023/A:1022605311895. URL https://doi.org/10.1023/A:1022605311895.

Balas K. Natarajan. Some results on learning. Unpublished manuscript, 1988.

Balas K. Natarajan and Prasad Tadepalli. Two new frameworks for learning. In ICML, pages 402–415, 1988.

Sima Noorani, Shayan Kiyani, George Pappas, and Hamed Hassani. Conformal prediction beyond the seen: A missing mass perspective for uncertainty quantification in generative models. arXiv preprint arXiv:2506.05497, 2025.

Ashwin Pananjady, Vidya Muthukumar, and Andrew Thangaraj. Just wing it: Near-optimal estimation of missing mass in a markovian sequence. Journal of Machine Learning Research, 25(312):1–43, 2024.

Liam Paninski. Estimation of entropy and mutual information. Neural computation, 15(6): 1191–1253, 2003.

Manish Purohit, Zoya Svitkina, and Ravi Kumar. Improving online algorithms via ml predictions. Advances in Neural Information Processing Systems, 31, 2018.

Zirou Qiu, Abhijin Adiga, Madhav V Marathe, SS Ravi, Daniel J Rosenkrantz, Richard E Stearns, and Anil Vullikanti. Learning the topology and behavior of discrete dynamical systems.

Nikhilesh Rajaraman, Andrew Thangaraj, and Ananda Theertha Suresh. Minimax risk for missing mass estimation. In 2017 IEEE International Symposium on Information Theory (ISIT), pages 3025–3029. IEEE, 2017.

A. Rakhlin, K. Sridharan, and A. Tewari. Online learning: Random averages, combinatorial parameters, and learnability. In Advances in Neural Information Processing Systems 23, 2010.

Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari. Online learning: Stochastic, constrained, and smoothed adversaries. Advances in neural information processing systems, 24, 2011.

Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari. Online learning via sequential complexities. J. Mach. Learn. Res., 16(1):155–186, 2015a.

Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari. Sequential complexities and uniform martingale laws of large numbers. Probability theory and related fields, 161: 111–153, 2015b.

Ananth Raman, Vinod Raman, Unique Subedi, Idan Mehalel, and Ambuj Tewari. Multiclass online learnability under bandit feedback. In Proceedings of 35th International Conference on Algorithmic Learning Theory, 2024a. accepted.

Vinod Raman, Unique Subedi, and Ambuj Tewari. A characterization of multioutput learnability. arXiv cs.LG, 2023a. preprint arXiv:2303.17716.

Vinod Raman, Unique Subedi, and Ambuj Tewari. A combinatorial characterization of online learning games with bounded losses. arXiv preprint arXiv:2307.03816, 2023b.

Vinod Raman, Unique Subedi, and Ambuj Tewari. Online learning with set-valued feedback. arXiv preprint arXiv:2306.06247, 2023c.

Vinod Raman, Unique Subedi, and Ambuj Tewari. The complexity of sequential prediction in dynamical systems. arXiv preprint arXiv:2402.06614, 2024b.

Paria Rashidinejad, Jiantao Jiao, and Stuart Russell. Slip: Learning to predict in unknown dynamical systems with long-term memory. Advances in Neural Information Processing Systems, 33:5716–5728, 2020.

Pradeep Ravikumar, Ambuj Tewari, and Eunho Yang. On ndcg consistency of listwise ranking methods. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, pages 618–626. JMLR Workshop and Conference Proceedings, 2011.

Daniel J Rosenkrantz, Abhijin Adiga, Madhav Marathe, Zirou Qiu, SS Ravi, Richard Stearns, and Anil Vullikanti. Efficiently learning the topology and behavior of a networked dynamical system via active queries. In International Conference on Machine Learning, pages 18796–18808. PMLR, 2022.

Tim Roughgarden. Beyond the worst-case analysis of algorithms. Cambridge University Press, 2021.

Yahya Sattar and Samet Oymak. Non-asymptotic and accurate learning of nonlinear dynamical systems. The Journal of Machine Learning Research, 23(1):6248–6296, 2022.

Robert E Schapire and Yoram Singer. Boostexter: A boosting-based system for text categorization. Machine learning, 39:135–168, 2000.

Ziv Scully, Isaac Grosof, and Michael Mitzenmacher. Uniform bounds for scheduling with job size estimates. arXiv preprint arXiv:2110.00633, 2021.

Shai Shalev-Shwartz and Shai Ben-David. Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press, USA, 2014a.

Shai Shalev-Shwartz and Shai Ben-David. Understanding machine learning: From theory to algorithms. Cambridge university press, 2014b.

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. The Journal of Machine Learning Research, 11:2635–2670, 2010.

Ilya Shmulevich, Edward R Dougherty, and Wei Zhang. From boolean to probabilistic boolean networks as models of genetic regulatory networks. Proceedings of the IEEE, 90 (11):1778–1792, 2002.

Daniel A Spielman and Shang-Hua Teng. Smoothed analysis: an attempt to explain the behavior of algorithms in practice. Communications of the ACM, 52(10):76–84, 2009.

Thomas Steinke and Lydia Zakynthinou. Reasoning about generalization via conditional mutual information. In Conference on Learning Theory, pages 3437–3452. PMLR, 2020.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. Advances in neural information processing systems, 27, 2014.

M. Talagrand. Sharper bounds for gaussian and empirical processes. The Annals of Probability, 22:28–76, 1994.

Anastasios Tsiamis and George J Pappas. Online learning of the kalman filter with logarithmic regret. IEEE Transactions on Automatic Control, 68(5):2774–2789, 2022.

Gregory Valiant and Paul Valiant. Estimating the unseen: an n/log (n)-sample estimator for entropy and support size, shown optimal via new clts. In Proceedings of the forty-third annual ACM symposium on Theory of computing, pages 685–694, 2011.

Gregory Valiant and Paul Valiant. Estimating the unseen: improved estimators for entropy and other properties. Journal of the ACM (JACM), 64(6):1–41, 2017.

Leslie G. Valiant. A theory of the learnable. In Symposium on the Theory of Computing, 1984.

A. W. van der Vaart and J. A. Wellner. Weak Convergence and Empirical Processes. Springer, 1996.

V. Vapnik and A. Chervonenkis. Theory of Pattern Recognition [in Russian]. 1974a.

Vladimir Vapnik and Alexey Chervonenkis. Theory of pattern recognition, 1974b.

Vladimir Naumovich Vapnik and Aleksei Yakovlevich Chervonenkis. On uniform convergence of the frequencies of events to their probabilities. Teoriya Veroyatnostei i ee Primeneniya, 16(2):264–279, 1971.

M. Vidyasagar. Learning and Generalization with Applications to Neural Networks. Springer-Verlag, 2nd edition, 2003.

Mathukumalli Vidyasagar and Rajeeva L Karandikar. A learning theory approach to system identification and stochastic adaptive control. Probabilistic and randomized methods for design under uncertainty, pages 265–302, 2006.

V. Vovk. Aggregating strategies. In Proceedings of the 3rd Annual Workshop on Computational Learning Theory, 1990.

V. Vovk. Universal forecasting algorithms. Information and Computation, 96(2):245–277, 1992.

Shufan Wang, Jian Li, and Shiqiang Wang. Online algorithms for multi-shop ski rental with machine learned advice. Advances in Neural Information Processing Systems, 33: 8150–8160, 2020.

Wen-Xu Wang, Ying-Cheng Lai, and Celso Grebogi. Data-based identification and prediction of nonlinear and complex dynamical systems. Physics Reports, 644:1–76, 2016.

Xi Wang and Gita Sukthankar. Multi-label relational neighbor classification using social context features. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 464–472, 2013.

Alexander Wei and Fred Zhang. Optimal robustness-consistency trade-offs for learning-augmented online algorithms. Advances in Neural Information Processing Systems, 33: 8042–8053, 2020.

Wolfgang Woess. Groups, graphs and random walks, volume 436. Cambridge University Press, 2017.

Stephen Wolfram. Theory and applications of cellular automata. World Scientific, 1986.

Changlong Wu, Ananth Grama, and Wojciech Szpankowski. Online learning in dynamically changing environments. In The Thirty Sixth Annual Conference on Learning Theory, pages 325–358. PMLR, 2023.

N Wulff and J A Hertz. Learning cellular automaton dynamics with neural networks. Advances in Neural Information Processing Systems, 5, 1992.

Haimin Yang, Zhisong Pan, Qing Tao, and Junyang Qiu. Online learning for vector autoregressive moving-average time series prediction. Neurocomputing, 315:9–17, 2018.

Hao Yang, Joey Tianyi Zhou, Yu Zhang, Bin-Bin Gao, Jianxin Wu, and Jianfei Cai. Exploit bounding box annotations for multi-label object recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 280–288, 2016.

Bin Yu. Assouad, fano, and le cam. In Festschrift for Lucien Le Cam: research papers in probability and statistics, pages 423–435. Springer, 1997.

Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. IEEE transactions on knowledge and data engineering, 26(8):1819–1837, 2013.