University of Alberta

PROBE-EFFICIENT LEARNING

by

Navid Zolghadr

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science

in

Statistical Machine Learning

Department of Computing Science

Spring 2013
Edmonton, Alberta

Abstract

This work introduces the "online probing" problem: In each round, the learner is able to purchase the values of a subset of features for the current instance. After the learner uses this information to produce a prediction for this instance, it then has the option of paying for seeing the full loss function for this instance that he is evaluated against. Either way, the learner pays for the errors of its predictions, and the cost of observing the features and loss function. We consider two variations of this problem, depending on whether the learner can observe the label for free. We provide algorithms and upper and lower bounds of the regret for both variants. We show that the paying a positive cost for the label significantly increases the regret of the problem. At the end we also convert the online algorithms to variants for batch settings.

Acknowledgements

This thesis would not have happened without the guidance and support of my advisors, Csaba Szepesvári and Russell Greiner. I would like to express my sincere appreciation to them for accompanying me in the past couple of years, dedicating their valuable time to this work, and assisting me an incredible amount of support and courage for research.

I greatly thank András György and Gábor Bartók for helping me with their valuable advises and ideas at all stages of this work. This thesis could not have been done without their helps. It is also a pleasure to thank my friends who were beside me throughout these years.

I would also like to thank Dale Schuurmans for participating in my examining committee, and providing me valuable comments and suggestions about this thesis.

Finally, I would like to express my deep gratitude to my family for all their encouragements and supports in my whole life.

Table of Contents

[1] Joint work with Gábor Bartók

Chapter 1

Introduction

To introduce the ideas, consider the challenge of classifying a sequence of parts, as they individually appear in a conveyor belt, as either defective or not, based on the values of its observable features. The challenge here is to find a predictor (here, a classifier) that maps the observed features into the label. However, first we must extract these features, which may involve non-trivial imaging operations, and so have some appreciable cost – here computational time. We therefore have only a prediction loss function that penalizes the predictor for any wrong prediction. But here we care not only about prediction loss of the classifier but also about its speed. So we might prefer classifier $C1$ over $C2$ even if $C1$ has slightly greater prediction error, if C_1 uses dramatically fewer features and so can be so much faster. Our goal is to find a classifier that minimizes the expected risk, where the risk of a classifier is a weighted sum of its prediction loss and the total cost of the features that the classifier pays to observe to make its prediction. Of course, the choice of the best classifier depends on how we weight the prediction loss compared to cost of features.

We consider the challenge of learning this "minimal risk predictor", using a learner that must decide which features it wants to observe for its learning process. In particular, this learner has access to once instance at a time: it can pay to observe any of its feature. It can also choose to observe the label of this instance; if so, it must pay an extra cost. Typically a learner is penalized only by the wrong prediction of its classifier with a given prediction loss function – *e.g.,* quadratic loss or zero-one loss. However in this context, the incurred loss will be the sum of the cost for

all observed features, the cost of the label if it is requested and the prediction loss of the selected predictor. We will later formulate our task more precisely in several frameworks. We will see that the cost of observing the label also plays an important role.

Below this, Section 1.1 provides more detail about this problem more and generalize from this example to the more general framework.

1.1 Costly Attribute Observation

In the classical settings, machine learning techniques try to learn an accurate predictor from a set of labeled examples and then use that predictor to produce label for unlabeled examples. We look at the general problem that includes the given example in the previous section. We focus on a variant where every feature, as well as the label, has an associated observation cost. The risk of a predictor is a combination of the prediction error and the costs that learner has paid for the features of the instances. Here we want a learner that find the best predictor in terms of its risk. This is what we call *Costly Attribute Observation* (CAO) framework. In the general "batch" problem, the goal is to find a predictor that achieves the minimum error (or in this context, the prediction loss) for the unforeseen examples. Note that the learner has access to all features and labels in the training phase in this general model. There are several variations to this general problem. A possible variation is to define the goal finding a predictor that achieves the best error in the prediction phase, while we have a limited budget in our learning phase for observing features and labels of training examples. Another variation can be optimizing a combination of the cost of features and labels in the training phase, with the risk of the final predictor in the prediction phase. Also the problem might be in *online* learning settings instead of *batch* which we will discuss later. We put all these variations under general CAO framework. However in this dissertation we look at some of those variations and not all of them. After we introduce the other frameworks, we will provide the formal definitions later in the following chapters for each cases and how previous works addressed these variations.

In this dissertation, we look at this problem in two different frameworks, the *online* framework and the *batch* framework. We will solve CAO in the *online* framework for two different settings – free label vs costly label – which leads to two different optimal regret bounds for general prediction loss functions. We will also provide more constrained bounds for the work by considering a more common loss function (*i.e.,* quadratic loss) and will be able to further improve the bounds. Chapter 2 will explain how the CAO framework fits into the online framework and provide several variations of this problem in the online framework and some algorithms and regret upper bounds for those algorithms as well as regret lower bounds for the problems. Chapter 3 shows how CAO fits into the batch learning framework and applies standard online-to-batch methods to transform the online algorithms to solve the problem in the batch framework. Chapters 2 and 3 begin by relating our work to previous works and earlier results. Finally, Chapter 4 summarizes all the results and provides future avenues for extending this work. Appendix A provides brief definitions of frequent used terms and Appendix B has proofs to the lemmas and theorems that are not included in the text.

Chapter 2

CAO in Online Framework

In this chapter, we explain the meaning of having cost for observing the features in an online problem. Going back to the example of classifying objects in a conveyor belt, we can look at this problem as an online problem as follows. Firstly, we do not have any training examples. We have a conveyor belt that is bringing a sequence of parts. These objects have several features that the learner can extract – *e.g.,* its weight, or the average intensity of the pixels. Each feature takes a specific amount of time to extract. For each object, the learner stops the belt to extract the features that it needs. The goal here for the learner is finding a predictor that uses a subset of the features that the learner decides to observe and predicts the label as accurate as possible for that object. Since we are in online settings, we care about minimizing the regret of learner, which is described with more details in Section 2.1. After the learner finds the predictor for that object, it has the option of asking an specialist to provide the true label of the object. Of course that might cost a non-negative amount for the learner to obtain true label, depending on the settings of the problem. In this chapter, we will look at the problem of having costs for features and labels in an online framework, which we call *online probing*. Section 2.4 shows more examples of this problem and summarizes the results of this chapter.

In this chapter we look at this problem more closely and provide more in-depth analysis of different variations and some upper bounds and lower bounds for the regret of the problem. Section 2.2 summarizes previous works related to this problem and Section 2.3 provides formal definitions and formulates the problem in our online framework. Sections 2.5 and 2.7 consider the problem when the labels are

free and non-free (respectively) and provide algorithms and upper bounds for the regret in each case. Also Section 2.6 and Section 2.8 show the lower bounds for the regret in each of the aforementioned case.

2.1 Online Framework and Notion of Regret

An instance of an *online learning* problem contains a sequence of rounds. At each round the learner is given an instance and has to predict something for that instance – *e.g.,* a label or a vector. At the end of each round, the learner suffers a loss. In this setting, we usually evaluate a learner based on its cumulative loss over all rounds. In online learning framework, the difference between the learner's cumulative loss and cumulative loss of any predictor from a given set of predictors is called *regret* of the learner competing against that set of predictors. A learning algorithm is good for an online problem if it has a bounded cumulative loss, or a bounded regret against the set of defined predictors. A typical result in this setting is upper bounding the regret after T rounds with function that is sub-linear in T – *e.g.,* $\mathcal{O}(\sqrt{T})$ or $\mathcal{O}(T^{2/3})$. Note that having a regret that scales linearly in T means that actually learner does not learn from the samples, as it suffers a constant amount at each round on average.

There is a wide range of problems in this framework. Below we summarize some of the relevant online problems and algorithms that we use later on. As one problem, at each round the learner has to choose a prediction from a set of experts' predictions[1] for the current instance (Cesa-Bianchi and Lugosi, 2006) and at the end, we compare its cumulative loss with cumulative loss of each of these experts. In this problem, the learner may have full information at each round, which means it has access to the loss of every single expert at each round. Cesa-Bianchi et al. (1997) proposed EWA[2] for this problem, which achieves regret bound $\mathcal{O}(\sqrt{T \ln N})$ where N is the number of experts. They also proved that this is a tight bound, apart from logarithmic factors. Note this assumes that the learner has access to complete information about each instance it observes. A variant considers the partial information case where the learner has access to the loss of only a subset of experts

[1] This problem is also known as "prediction with expert advice"
[2] Exponentially Weighted Algorithm

and in particular only the chosen expert at each round (Cesa-Bianchi and Lugosi, 2006). Auer et al. (2002a) considered this problem when choosing an expert reveals only its own loss[3] and proposed the EXP3 algorithm that achieves the regret bound of $\mathcal{O}(\sqrt{TN \ln N})$ where N is the number of experts[4]. They also proved that no algorithm can achieve better than $\Omega(\sqrt{TN})$ regret. Later, Beygelzimer et al. (2010) proposed an algorithm that achieves $\mathcal{O}(\sqrt{TN})$ regret, which matches the lower bound for this problem. Bubeck and Cesa-Bianchi (2012) recently surveyed all variations and results of Multi-armed Bandits in online framework. Mannor and Shamir (2011) looked at the problem of partial feedbacks where we have a graph of experts and choosing an expert reveals the loss of each expert that is connected by an edge in the graph. They solved this problem in the general case where we can have different graphs at each round; they proposed the ELP[5] algorithm and proved that its regret is bounded by $\mathcal{O}(\sqrt{\sum_{i=1}^{T} \chi(G_i) \ln N})$ where $\chi(G_i)$ is the clique-partition number of graph at round i and N is the number of experts. Note that the regret bounds for their algorithm matches with EWA in the full information case and EXP3 in bandit settings. We will use some of these algorithms later for our problem.

We view our CAO as an online problem as follows. It has multiple rounds, where at each round the learner is required to find a predictor, the predict the label using the predictor for the current example, exactly like a typical online problem. Here, however, the learner must pay to see the values of the features of each example, as well as the cost to obtain its true label after its prediction at each round. As we mentioned before, the total loss of learner (*i.e.,* risk) is the sum of the features costs it paid and its prediction loss, and it competes with other predictors that (1) never see the label and so do not pay for it and (2) can use any subset of features, and so they need to pay only for that subset. This cost model shows that there is an advantage to finding a predictor that involves few features, as long as it is sufficiently accurate. The challenge, of course, is finding these relevant features, which happens during this online learning process.

[3] This problem is also known as Multi-armed Bandits

[4] In this setting, experts are also called "arms"

[5] Exponentially-weighted algorithm with Linear Programming

2.2 Related Online Problems

In the previous section, we introduced some of the other problems in the online framework and compared those problems with ours, to show the similarities and the differences. We continue with some of the online problems that are similar to our problem, focusing on problems where the features and labels are not necessarily available for free.

Obviously, when there is a cost for observing the loss, the problem is related to active learning (Settles, 2009; Cesa-Bianchi et al., 2005). To our best knowledge, the case when observing the features is costly has not been studied before in the online literature. Cesa-Bianchi et al. (2005) considered the full information prediction with expert advice problem, constrained by the fact that learner can access the loss of the experts in at most m rounds[6]. They showed a lower bound of $\Omega(T\sqrt{(\ln N)/m})$ regret on this problem and also proposed an algorithm achieving this regret upper bound within a constant factor. Our problem is more general that we can have costs for features. Also the learner can choose in how many rounds it wants to see the label and there is no hard limit on that. Also in our problem, the learner decides which feature to see.

Our problem is different than having missing features in the data (Little and Rubin, 1986; Dempster et al., 1977). For example, Rostamizadeh et al. (2011) and Dekel et al. (2010) assume the features of different examples might be corrupted, missed, or partially observed due to the various problems, such as failure of sensors to gather values for these features. The missing features means the environment chooses a subset of features to give to the learner. So even though this assumption might be realistic in some of the applications, it is completely different than our problem where the learner (and not the environment) gets to decide which features to observe. They also assume that, at the end of each round, they have the the loss, without any more cost, which again our framework is extending to more general assumption of having a cost for revealing the loss as well. In general there might be delay in obtaining the information about the label and therefore the loss at each

[6]This problem is also known as label efficient prediction.

round. However, for simplicity, in this work we have decided not to study the effect of this delay. Preliminary works on related problems show that delays usually increase the regret in a moderate fashion (cf. Weinberger and Ordentlich (2006), Mesterharm (2005), Agarwal and Duchi (2011), the thesis of Joulani (2012) and the references therein).

Going back to the framework that the learner can pick features, Hazan and Koren (2012) and Cesa-Bianchi et al. (2010) assume that the learner can choose which feature to observe; this is similar to our problem. However they allow the learner to observer at most k features for any example[7]. Since Cesa-Bianchi et al. (2010) is more in the batch settings, we look at theirs more closely in Section 3.1. Hazan and Koren (2012) also focus on LAO in the case of different types of losses and provide theoretical bounds for their algorithm. Their algorithm can be categorized in the local budget constraint that we describe in the next chapter. They assume that data is coming from an unknown but fixed distribution and propose an algorithm that finds a predictor after T examples that achieves an optimal loss on the distribution of data. So they basically focus on learning with partial observation in training phase and then try to find a predictor that achieves the best loss at the end. In the LAO setting, the learner is not penalized for observing the values of features. So this learner will always choose the maximum allowed number of the features, k. This makes the problem different than ours. We will solve our problem in the online framework and bound the regret while we face an adversarial environment. Note that the adversarial setting is more general than stochastic setting because an adversary can choose a distribution and choose the data from that.

2.3 Formal Definition

In this section we study *online probing* motivated by practical problems, where there is a cost to observe features that may help one's predictions. Online probing is a special online learning problem. Like standard online learning problems, the learner's goal is to produce a good predictor. In each time step t, the learner pro-

[7]This settings is also known as Limited Attribute Observation (LAO).

duces its predictor based on a set of feature values $x_t = (x_{t,1}, \dots, x_{t,d})^\top \in \mathcal{X} \subset \mathbb{R}^d$.[8] However, unlike in the standard online learning settings, if the predictor wants to use the value of feature i to produce a prediction, the learner has to purchase the value at some fixed, *a priori* known cost, $c_i \geq 0$. Features whose value is not purchased in a given round remain unobserved by the learner. Once a prediction, $\hat{y}_t \in \mathcal{Y}$, is produced, it is evaluated using a prediction loss function, $\ell_t : \mathcal{Y} \to \mathbb{R}$. At the end of a round, the learner has the option of purchasing the full prediction loss function, again at a fixed pre-specified cost, $c_{d+1} \geq 0$ (otherwise, the prediction loss function is not revealed to the learner). The learner's performance is measured by its regret as it competes against some pre-specified set of predictors. Just like the learner, a competing predictor also needs to purchase the feature values needed in the prediction. Let $s_t \in \{0,1\}^{d+1}$ denote the indicator denoting what the learner purchased in round t. In particular, $s_{t,d+1}$ denotes whether the learner purchased the label. Also let $c \in [0,\infty)^{d+1}$ denote the respective observation costs. Then the regret with respect to a class of functions $\mathcal{F} \subset \{f \mid f : \mathcal{X} \to \mathcal{Y}\}$ is defined by

$$R_T = \sum_{t=1}^{T} \{\ell_t(\hat{y}_t) + \langle\, s_t, c \,\rangle\} \; - \inf_{f \in \mathcal{F}} \left\{ T \langle\, \mathrm{s}(f), c_{1:d} \,\rangle + \sum_{t=1}^{T} \ell_t(f(x_t)) \right\},$$

where $c_{1:d} \in \mathbb{R}^d$ is the vector obtained from c by dropping its last component and for a given function $f : \mathbb{R}^d \to \mathcal{Y}$, $\mathrm{s}(f) \in \{0,1\}^d$ is an indicator vector whose ith component indicates whether f is sensitive to its ith input (in particular, $\mathrm{s}_i(f) = 0$ by definition when $f(x_1, \dots, x_i, \dots, x_d) = f(x_1, \dots, x_i', \dots, x_d)$ holds for all $(x_1, \dots, x_i, \dots, x_d), (x_1, \dots, x_i', \dots, x_d) \in \mathcal{X}$, otherwise $\mathrm{s}_i(f) = 1$). When defining the best competitor in hindsight, we did not include the cost of observing the prediction loss function. This is because, if we do include that cost, we would essentially introduce a constant cost of size $c_{d+1}T$, basically making it regret-free for a learning algorithm to observe the loss function, in which case introducing the cost c_{d+1} would not make any difference, since they will cancel each other in the definition of the regret. Thus, the current regret definition is preferred as it promotes the study of regret when there is a price attached to learning the loss functions.

[8] We use $^\top$ to denote the transpose of vectors. In what follows, all vectors $x \in \mathbb{R}^d$ will denote column vectors.

2.4 Online Probing Problem

We provide several examples to further motivate online probing in our framework. Consider the problem of developing a computer-assisted diagnostic tool to determine what treatment to apply to a patient in a subpopulation of patients. When a patient arrives, the computer can order a number of tests, each of which costs money, to augment other information (*e.g.,* the medical record of the patient) that is available for free. Based on the available information, the system chooses a treatment. Following-up the patient may or may not incur additional cost. The goal here is to find a treatment that minimizes a combination of its treatment error and the cost of tests. As another example, consider the problem of product testing in a manufacturing process (*e.g.,* the production of electronic consumer devices). When the product arrives, it can be subjected to a large number of diagnostic tests that differ in terms of their costs and effectiveness. The goal is to find a predictor to decide whether the product is defect-free. Obtaining the ground truth can be quite expensive in the case of complex products. The situation is that the effectiveness of the various tests is often *a priori* unknown and that different tests may provide complementary information. Hence, it might be challenging to decide what the most cost-effective diagnostic procedure is. Yet another example is the problem of developing a cost-effective way of instrument calibration. In this problem, the goal is to predict one or more real-valued parameters of some product. Again, various tests with different costs and reliability can be used as the input to the predictor. In all these cases one can easily formulate them using the proposed model in the previous section.

This chapter analyzes two types of the online probing. In the first version, *free-label online probing*, there is no cost to see the prediction loss function, $c_{d+1} = 0$ (the prediction loss function is often comparing the predicted value with the true label in a known way, in which case learning the value of the label for the round means that the whole prediction loss function becomes known; hence the choice of the name). Thus, the learner naturally will choose to see the loss function after it provides its prediction; this provides feedback that the learner can use, to improve

the predictor it produces. In the second version, *non-free-label online probing*, the cost of seeing the prediction loss function is positive: $c_{d+1} > 0$.

In the case of free-label online probing, we give an algorithm that enjoys a regret of

$$\mathcal{O}(\sqrt{2^d LT \ln \mathcal{N}_T(1/(TL))})$$

when the losses are L-equi-Lipschitz (Theorem 2.5.2). Here $\mathcal{N}_T(\varepsilon)$ is the ε-covering number of $\mathcal{F}$ on sequences of length T. This leads to an $\tilde{\mathcal{O}}(\sqrt{2^d LT})$ regret bound for typical function classes, such as the class of linear predictors with bounded weights and bounded inputs (Corollary 2.5.1). Next, for the case of linear prediction with quadratic loss, in Section 2.5.2 we give an algorithm whose regret scales only polynomially with the dimension d (Theorem 2.5.3). Also Section 2.6 provides a matching lower bound for this problem (Theorem 2.6.1).

In the case of non-free-label online probing, in contrast to the free-label case, we prove that the minimax growth rate of the regret is of the order $\tilde{\Theta}(T^{2/3})$ (Theorems 2.7.1, 2.8.1). The increase of regret-rate stems from the fact that the "best competitor in hindsight" does not have to pay for the label. In contrast to the previous case, since the label is costly, if the algorithm decides to see the label it does not even have to reason about which features to observe: the main source of the excess cost over that of the best predictor in hindsight is due to the cost of seeing the labels. However, in practice (for shorter horizons) it still makes sense to select the ones that provide the best balance between the feature-cost and the prediction loss. Although we do not study this, we note in passing that, by combining the algorithmic ideas developed for the free-label case with the ideas developed for the non-free-label case, it is possible to derive an algorithm that reasons actively about the cost of observing the features, too.

In the part dealing with the free-label problem, we build heavily on the results of Mannor and Shamir (2011), while our results for the non-free-label problem are based on the ideas of (Cesa-Bianchi et al., 2006).

2.5 Free-Label Probing

In this section, we consider the case when the cost of observing the prediction loss function is zero. Thus, we can assume without loss of generality that the learner receives the loss function at the end of each round (*i.e.,* $s_{t,d+1} = 1$) – as the learner can ask for it for free. We will first consider the general setting where the only restriction are that the losses are equi-Lipschitz and the function set $\mathcal{F}$ has finite empirical worst-case covering numbers, $\mathcal{N}_T(\mathcal{F}, \alpha)$ (Section 2.5.1). For this case, we derive an upper bound $\mathcal{O}(\sqrt{2^d T \ln \mathcal{N}_T(\mathcal{F}, 1/(TL))})$ on the regret (Theorem 2.5.2). For linear predictors with bounded inputs and weights, this bound results in the bound $\mathcal{O}(\sqrt{d 2^d T \ln T})$ on the regret (Corollary 2.5.1). Besides covering numbers, our main tool in proving the upper bound is the work of Mannor and Shamir (2011), who consider prediction with expert advice in a setting when choosing one expert will reveal the losses of some other pre-specified experts. Next, we consider a special case when the set of competitors are the linear predictors and the prediction losses are quadratic (Section 2.5.2). For this setting, exploiting the algebraic properties of prediction loss functions and predictors, we design an algorithm and prove that its regret is bounded by $\mathcal{O}(\sqrt{dT})$, vastly improving the dependence of the regret on the dimension d. The algorithm proposed, although it tames the exponential dependence of the regret, is computationally expensive: Both its memory and computational requirements scale exponentially with the dimension d. It remains an important open problem to design an algorithm whose computational complexity, as well as regret, scale polynomially with the dimension, while keeping the root-T dependence of the regret on time.

2.5.1 The case of Lipschitz losses

In this section we assume that the prediction loss functions, ℓ_t, are Lipschitz with a known, common Lipschitz constant $L \in \mathbb{R}^+$ over $\mathcal{Y}$ w.r.t. to some semi-metric $d_{\mathcal{Y}}$ of $\mathcal{Y}$:

$$\max_{t \geq 1} \sup_{y, y' \in \mathcal{Y}} |\ell_t(y) - \ell_t(y')| \leq L \, d_{\mathcal{Y}}(y, y'). \tag{2.1}$$

The idea is to study first the case when $\mathcal{F}$ is finite and then reduce the general

infinite case to the finite case by considering appropriate finite coverings of the space $\mathcal{F}$. The regret will then depend on how the covering numbers of the space $\mathcal{F}$ behave.

Let us thus first consider the case when $\mathcal{F}$ is finite. In this case, the problem is an instance of prediction with expert advice under partial information feedback (Auer et al., 2002a), each expert being identified by an element of $\mathcal{F}$. The important observation is that, if the learner chooses to observe the values of some features then it will also be able to evaluate the losses of all the predictors $f \in \mathcal{F}$ that use only the selected features. This can be formalized as follows: By a slight abuse of notation let $s_t \in \{0,1\}^d$ be the indicator showing the features selected by the learner at time t (we drop the last element of s_t in our earlier notation since, in the current setting, the prediction loss will always be observed as it costs nothing). Then, the learner can compute the loss of any predictor $f \in \mathcal{F}$ such that $\mathrm{s}(f) \leq s_t$, where $\leq$ denotes the component-wise comparison. Note, however, that depending on the function, it may be possible to estimate the prediction losses of other predictors, too. This is what we will exploit when we study some interesting special cases of the general problem. However, in general, this might be not possible.

Mannor and Shamir (2011) studied problems similar to this in a general framework, where in addition to the loss of the selected predictor (expert), the losses of some other predictors are also communicated to the learner in every round. In their problem, there is a graph at each round whose vertices are the elements of $\mathcal{F}$ (*i.e.,* the experts). If the learner chooses expert $f \in \mathcal{F}$, the environment will reveal the loss for all other experts $g \in \mathcal{F}$ that has an edge to f in the graph. It is assumed that the graph of any round t, $G_t = (\mathcal{F}, E_t)$ becomes known to the learner at the beginning of the round. Further, it is assumed that $(f,f) \in E_t$ for any $t \geq 1$ and $f \in \mathcal{F}$. Mannor and Shamir (2011) provide Algorithm 1 and prove the following:

Theorem 2.5.1 (Mannor and Shamir (2011))**.** *Consider a prediction with expert advice problem over $\mathcal{F}$ where in round t, $G_t = (\mathcal{F}, E_t)$ is the directed graph that encodes which losses become available to the learner. Assume that for any $t \geq 1$, at most $\chi(G_t)$ cliques of G_t can cover all vertices of G_t. Let B be a bound on the non-negative losses ℓ_t. Then, there exists a constant $C_{Elp} > 0$ such that for any*

Algorithm 1 The ELP Algorithm. In the pseudocode, Δ_N denotes the N-dimensional simplex: $\Delta_N = \{s \in [0,1]^N \mid \sum_{i=1}^N s_i = 1\}$.

Parameters: Neighborhood graphs $G_t = (\mathcal{F}, E_t)$, $1 \le t \le T$, a bound B on the losses.
Initialization: $N = |\mathcal{F}|$, $\beta = \sqrt{(\ln N)/(3B^2 \sum_t \chi(G_t))}$, $w_{0,j} = 1/N$, $1 \le j \le N$.
for $t = 1$ **to** T **do**
  Let $s_t = \arg\max_{q \in \Delta_N} \min_{1 \le i \le N} \sum_{(i,k) \in E_t} q_k$.
  Let $s_t^* = \min_{1 \le i \le N} \sum_{(i,k) \in E_t} s_{t,i}$.
  Let $\gamma_t = \beta B / s_t^*$.
  Choose action i_t randomly from probability mass function

$$p_{t,i} = (1 - \gamma_t) \frac{w_{t,i}}{\sum_{j=1}^N w_{t,j}} + \gamma_t s_{t,i} \qquad (1 \le i \le N).$$

  Receive loss $(\ell_{t,k})_{(i_t,k) \in E_t}$.
  Compute $\tilde{g}_{t,j} = \frac{B - \ell_{t,j}}{\sum_{(l,j) \in E_t} p_{t,l}}$ if $(j, i_t) \in E_t$, and $\tilde{g}_j(t) = 0$ otherwise.
  $w_{t+1,j} = w_{t,j} \exp(\beta \tilde{g}_{t,j})$, $1 \le j \le N$.
end for

$T > 0$, the regret of Algorithm 1 when competing against the best predictor using the algorithm satisfies

$$\mathbb{E}[R_T] \le C_{Elp} B \sqrt{\ln |\mathcal{F}| \sum_{t=1}^T \chi(G_t)}\,. \tag{2.2}$$

In particular, the algorithm's computational cost at any given round is $\mathrm{poly}(|\mathcal{F}|)$.

The algorithm of Mannor and Shamir (2011) builds on the exponential weights algorithm, which they call ELP for exponential weights with linear programming, but modifies it to explore less, and so exploit the information structure of the problem. The exploration distribution is found by solving a linear program, explaining the name of the algorithm.

In our case $E_t \equiv E \doteq \{(f,g) \mid \mathrm{s}(g) \le \mathrm{s}(f)\}$. Thus, $\chi = 2^d$. Further, $B = \|c_{1:d}\|_1 + \max_{t \ge 1, y \in \mathcal{Y}} \ell_t(y) \doteq C_1 + \ell_{\max}$ (where $C_1 = \|c_{1:d}\|_1$). It is important to note that the loss of each expert in Algorithm 1 is sum of the prediction loss and the cost of the features that the expert needs to see. Plugging these into (2.2) gives

$$\mathbb{E}[R_T] \le C_{Elp}(C_1 + \ell_{\max}) \sqrt{2^d T \ln |\mathcal{F}|}\,. \tag{2.3}$$

Now, let us consider the case when $\mathcal{F}$ is not finite. Fix $\mathcal{F}' \subset \{f \mid f : X \to \mathcal{Y}\}$, $T > 0$. Introduce the worst-case average approximation error of $\mathcal{F}$ using $\mathcal{F}'$ over sequences of length T as follows:

$$A_T(\mathcal{F}', \mathcal{F}) = \max_{x \in \mathcal{X}^T} \sup_{f \in \mathcal{F}} \inf_{f' \in \mathcal{F}'} \frac{1}{T} \sum_{t=1}^{T} d_{\mathcal{Y}}(f(x_t), f'(x_t)) \,.$$

The average error can also be viewed as a (normalized) $d_{\mathcal{Y}}$-"distance" between the vectors $(f(x_t))_{1 \le t \le T}$ and $(f'(x_t))_{1 \le t \le T}$. For a given positive number α, define the *worst-case empirical covering number* of $\mathcal{F}$ at level α and horizon $T > 0$ by

$$\mathcal{N}_T(\mathcal{F}, \alpha) = \min\{\ |\mathcal{F}'| \ \mid\ \mathcal{F}' \subset \{f \mid f : X \to \mathcal{Y}\},\ A_T(\mathcal{F}', \mathcal{F}) \le \alpha\ \}.$$

With these definitions, we have the following result:

Theorem 2.5.2. *Assume that the losses $(\ell_t)_{t \ge 1}$ are L-Lipschitz (cf. (2.1)). Then, for every $\alpha > 0$, there exists an algorithm such that for any $T > 0$, knowing T, the regret satisfies*

$$\mathbb{E}[R_T] \le C_{Elp}(C_1 + \ell_{\max}) \sqrt{2^d T \ln \mathcal{N}_T(\mathcal{F}, \alpha)} + TL\alpha \,.$$

In particular, by choosing $\alpha = 1/(TL)$, we have

$$\mathbb{E}[R_T] \le C_{Elp}(C_1 + \ell_{\max}) \sqrt{2^d T \ln \mathcal{N}_T(\mathcal{F}, 1/(TL))} + 1 \,.$$

Proof. Consider the algorithm that starts by constructing a worst-case covering $\mathcal{F}_\alpha$ of $\mathcal{F}$ at level α. The regret relative to the best choice of $f^* \in \mathcal{F}$ in hindsight can then be written as the regret relative to the best approximation of f^* within $\mathcal{F}_\alpha$, plus the error of approximating f^* by f. The latter, by the definition of $\mathcal{F}_\alpha$ and thanks the Lipschitzness of the losses, is bounded by $TL\alpha$, while the former can be bounded using (2.3). This gives rise to the first stated regret bound. The second bound is obtained by simply plugging in the definition of $\alpha = 1/(TL)$ in the first bound. □

We note in passing that using the well-known "guess and double trick" (Auer et al., 2002b), the requirement that the algorithm has to know the horizon T at the

beginning can be removed at the expense of increasing the constant multiplier in the regret bound.

In order to turn the above bound into a concrete bound, one must investigate the behavior of the *metric entropy*, $\ln \mathcal{N}_T(\mathcal{F}, \alpha)$. In many cases, the metric entropy can be bounded independently of T. In fact, often, $\ln \mathcal{N}_T(\mathcal{F}, \alpha) = D \ln(1 + c/\alpha)$ for some $c, D > 0$. When this holds, D is often called the "dimension" of $\mathcal{F}$ and we get that

$$\mathbb{E}[R_T] \leq C_{Elp}(C_1 + \ell_{\max})\sqrt{2^d T D \ln(1 + cTL)} + 1\,.$$

As a specific example, we will consider the case of real-valued linear functions over a ball in a Euclidean space with weights belonging to some other ball. For a normed vector V with norm $\|\cdot\|$, $x \in V$, $r \geq 0$, let $B_{\|\cdot\|}(x, r) = \{v \in V \mid \|v\| \leq r\}$ denote the ball in V centered at x that has radius r. For $\mathcal{X} \subset \mathbb{R}^d$, $\mathcal{W} \subset \mathbb{R}^d$, let

$$\mathcal{F} \subset \mathrm{Lin}(\mathcal{X}, \mathcal{W}) \doteq \{g : \mathcal{X} \to \mathbb{R} \mid g(\cdot) = \langle\, w, \cdot \,\rangle\,, w \in \mathcal{W}\} \tag{2.4}$$

be the space of linear mappings from $\mathcal{X}$ to reals with weights belonging to $\mathcal{W}$. We have the following lemma:

Lemma 2.5.1. *Let $X, W > 0$, $d_Y(y, y') = |y - y'|$, $\mathcal{X} \subset B_{\|\cdot\|}(0, X)$ and $\mathcal{W} \subset B_{\|\cdot\|_*}(0, W)$. Consider a set of real-valued linear predictors $\mathcal{F} \subset \mathrm{Lin}(\mathcal{X}, \mathcal{W})$. Then, for any $\alpha > 0$,*

$$\ln^+ \mathcal{N}_T(\mathcal{F}, \alpha) \leq d \ln(1 + 2WX/\alpha).$$

Proof. An appropriate covering of $\mathcal{F}$ can be constructed as follows: Consider an ε-covering $\mathcal{W}'$ of the ball $\mathcal{W}$ with respect to $\|\cdot\|_*$ for some $\varepsilon > 0$ (*i.e.,* for any $w \in \mathcal{W}$ there exists $w' \in \mathcal{W}'$ such that $\|w - w'\|_* \leq \varepsilon$). Then,

$$\mathcal{F}' = \{g : \mathcal{X} \to \mathbb{R} \mid g(x) = \langle\, x, w \,\rangle, w \in \mathcal{W}'\} \tag{2.5}$$

is an εX-covering of $\mathcal{F}$. To see this pick any $f \in \mathcal{F}$. Thus, $f(x) = \langle\, w, x \,\rangle$ for some $w \in \mathcal{W}$. Let w' be the vector in $\mathcal{W}'$ that is closest to w. Thus, $\|w - w'\|_* \leq \varepsilon$. Let $g \in \mathcal{F}'$ be given by $g(x) = \langle\, x, w' \,\rangle$. Then,

$$\frac{1}{T}\sum_{t=1}^{T} |f(x_t) - g(x_t)| = \frac{1}{T}\sum_{t=1}^{T} |\langle\, w - w', x_t \,\rangle| \leq \varepsilon X, \tag{2.6}$$

where the last step required Hölder's inequality and that $x_t \in \mathcal{X}$ means $\|x_t\| \le X$. This argument thus shows that, to get an α-covering of $\mathcal{F}$, we need an ε-covering of $\mathcal{W}$ with $\varepsilon = \alpha/X$ and therefore $\mathcal{N}_T(\mathcal{F}, \alpha) \le \mathcal{N}(\mathcal{W}, \alpha/X)$. As it is well known,[9] $\mathcal{N}(\mathcal{W}, \varepsilon) \le (2W/\varepsilon + 1)^d$ and thus $\ln^+ \mathcal{N}_T(\mathcal{F}, \alpha) \le d \ln(1 + 2WX/\alpha)$. □

The previous lemma, together with Theorem 2.5.2, immediately gives the following result:

Corollary 2.5.1. *Assume that $\mathcal{F} \subset \mathrm{Lin}(\mathcal{X}, \mathcal{W})$, $\mathcal{X} \subset B_{\|\cdot\|}(0, X)$, $\mathcal{W} \subset B_{\|\cdot\|_*}(0, W)$ for some $X, W > 0$. Further, assume that the losses $(\ell_t)_{t \ge 1}$ are L-Lipschitz. Then, there exists an algorithm such that, for any $T > 0$, the regret of the algorithm will satisfy,*

$$\mathbb{E}\left[R_T\right] \le C_{Elp}(C_1 + \ell_{\max})\sqrt{d 2^d T \ln(1 + 2TLWX)} + 1\,.$$

If one is given an *a prior bound* p on the maximum number of features that can be used in a single round (allowing the algorithm to use fewer than p features, but not more) then 2^d in the above bound could be replaced by $\sum_{1 \le i \le p} \binom{d}{i} \approx d^p$, where the approximation assumes that $p < d/2$. Such a bound on the number of features available per round may arise from strict budgetary considerations. When d^p is small, this makes the bound non-vacuous even for small horizons. In addition, in such cases the algorithm also becomes computationally feasible. It remains an interesting open question to study the computational complexity when there is no restriction on the number of features used.

2.5.2 Linear prediction with quadratic losses

In this section we study the problem under the assumption that the predictors have a linear form, *i.e.*, $\mathcal{F} = \mathrm{Lin}(\mathcal{W}, \mathcal{X}) \doteq \{g : \mathcal{X} \to \mathbb{R} \,|\, g(x) = \langle\, x, w \,\rangle, w \in \mathcal{W}\}$, where in our case $\mathcal{W} = \{w \in \mathbb{R}^d \,|\, \|w\|_* \le w_{\lim}\} \subset \mathbb{R}^d$ and $\mathcal{X} = \{x \in \mathbb{R}^d \,|\, \|x\| \le x_{\lim}\} \subset \mathbb{R}^d$ and the prediction loss is quadratic,

$$\ell_t(y) = (y - y_t)^2\,,$$

[9] The proof of this is given in Section B.1.1 in the appendix for the convenience of the reader.

where $|y_t| \leq x_{\lim} w_{\lim}$. Thus, choosing a predictor is akin to selecting a weight vector $w_t \in \mathcal{W}$, as well as a binary vector $s_t \in \mathcal{G} \subset \{0,1\}^d$ that encodes the features to use in round t. Let $s(w)$ denote the binary vector whose ith component is one if ith component of w is non-zero and otherwise zero. We may also look at this binary vector as a subset of features in which w is non-zero. The prediction for round t is then $\hat{y}_t = \langle\, w_t, s_t \odot x_t \,\rangle$, and the prediction loss suffered is $(\hat{y}_t - y_t)^2$. The set $\mathcal{G}$ is an arbitrary non-empty, a priori specified subset of $\{0,1\}^d$ that allows the user of the algorithm to encode extra constraints on what subsets of features can be selected. Note that $\mathcal{G}$ might be all 2^d subsets of $\{0,1\}^d$. Further, it is assumed that $x_t \in \mathcal{X} \doteq \{x \in \mathbb{R}^d \,|\, \|x\| \leq x_{\lim}\}$.

In this section we show, that in this case, a regret bound $\tilde{\mathcal{O}}(\sqrt{\mathrm{poly}(d)T})$ is possible. The key idea, which permits the improvement of the regret bound, is that a randomized choice of a weight vector W_t (and thus, of a subset) helps one to construct unbiased estimates of the losses $\ell_t(\langle\, w, s \odot x_t \,\rangle)$ for all weight vectors w and all subsets $s \in \mathcal{G}$ under some mild conditions on the distribution of W_t. The construction of such unbiased estimates is possible, even though some feature values are unobserved, because of the special algebraic structure of the prediction and loss functions. A similar construction has appeared in a different context (Cesa-Bianchi et al., 2010).

The construction works as follows. Define the $d \times d$ matrix, X_t by $(X_t)_{i,j} = x_{t,i} x_{t,j}$ $(1 \leq i, j \leq d)$. Expanding the loss of the prediction $\hat{y}_t = \langle\, w, x_t \,\rangle$, we get that the prediction loss of using $w \in \mathcal{W}$ is

$$\ell_t(w) \doteq \ell_t(\langle\, w, x_t \,\rangle) = w^\top X_t\, w - 2\, w^\top x_t y_t + y_t^2,$$

where, with a slight abuse of notation, we have introduced the loss function $\ell_t : \mathcal{W} \to \mathbb{R}$ (we'll keep abusing the use of ℓ_t by overloading it based on the type of its argument). Clearly, it suffices if we construct unbiased estimates of $\ell_t(w)$ for any $w \in \mathcal{W}$.

We will use a discretization approach. Therefore, assume that we are given a finite subset $\mathcal{W}'$ of $\mathcal{W}$ (that will be constructed later). In each step t, our algorithm will choose a random weight vector W_t from a probability distribution supported

on $\mathcal{W}'$. Let $p_t(w)$ be the probability of selecting the weight vector, $w \in \mathcal{W}'$.

For $1 \le i \le d$, let

$$q_t(i) = \sum_{w \in \mathcal{W}' : i \in s(w)} p_t(w)\,,$$

be the probability that $s(W_t)$ will contain i,[10] while for $1 \le i, j \le d$, let

$$q_t(i,j) = \sum_{w \in \mathcal{W}' : i,j \in s(w)} p_t(w)\,,$$

be the probability that both $i, j \in s(W_t)$.[11] Assume that $p_t(\cdot)$ is constructed such that $q_t(i,j) > 0$ holds for any time t and indices $1 \le i, j \le d$. This implies that $q_t(i) > 0$ for all $1 \le i \le d$.

Define the vector $\tilde{x}_t \in \mathbb{R}^d$ and matrix $\tilde{X}_t \in \mathbb{R}^{d \times d}$ by the following equations:

$$\tilde{x}_{t,i} = \frac{\mathbb{1}_{\{i \in s(W_t)\}} x_{t,i}}{q_t(i)}\,, \qquad (\tilde{X}_t)_{i,j} = \frac{\mathbb{1}_{\{i,j \in s(W_t)\}} x_{t,i} x_{t,j}}{q_t(i,j)}\,. \tag{2.7}$$

and observe that $\mathbb{E}\left[\tilde{x}_t \,|\, p_t\right] = x_t$ and $\mathbb{E}\left[\tilde{X}_t \,|\, p_t\right] = X_t$. Further, notice that both $\tilde{x}_t$ and $\tilde{X}_t$ can be computed based on the information available at the end of round t, *i.e.,* based on the feature values $(x_{t,i})_{i \in s(W_t)}$. Now, define the estimate of prediction loss

$$\tilde{\ell}_t(w) = w^\top \tilde{X}_t\, w - 2\, w^\top \tilde{x}_t y_t + y_t^2\,. \tag{2.8}$$

Note that y_t can be readily computed from $\ell_t(\cdot)$, which is available to the algorithm (equivalently, we may assume that the algorithm implicitly observed y_t). Due to the linearity of expectation, we have $\mathbb{E}\left[\tilde{\ell}_t(w) | p_t\right] = \ell_t(w)$. That is, $\tilde{\ell}_t(w)$ provides an unbiased estimate of the prediction loss $\ell_t(w)$ for any $w \in \mathcal{W}$. Hence, by adding feature cost term, we get $\tilde{\ell}_t(w) + \langle\, s(w), c\,\rangle$ as an estimate of the risk that the learner would have suffered at round t had it chosen the weight vector w.

The algorithm that we propose is based on the standard EXP3 algorithm. Let $\eta > 0$ be the learning parameter (to be chosen later). For each $w \in \mathcal{W}'$, the learner updates a weight $u_t(w)$ with an exponential update-rule using the estimated losses:

$$u_{t+1}(w) = u_t(w) e^{-\eta(\tilde{\ell}_t(w) + \langle\, c, s(w)\,\rangle)}, \quad w \in \mathcal{W}'\,.$$

[10] That is, the ith feature will be used

[11] Note that, following our earlier suggestion, we view each d-dimensional binary vectors as a subset of $\{1, \ldots, d\}$.

Algorithm 2 The LQDEXP3 Algorithm

Parameters: Real numbers $0 \le \eta$, $0 < \gamma \le 1$, $\mathcal{W}' \subset \mathcal{W}$ finite set, a distribution μ over $\mathcal{W}'$, Real number $T > 0$.
Initialization: $u_1(w) = 1$ ($w \in \mathcal{W}'$).
for $t = 1$ **to** T **do**
Draw $W_t \in \mathcal{W}'$ from the probability mass function

$$p_t(w) = (1-\gamma)\frac{u_t(w)}{U_t} + \gamma\mu(w), \qquad w \in \mathcal{W}'\,.$$

Obtain the features values, $(x_{t,i})_{i \in s(W_t)}$.
Predict $\hat{y}_t = \sum_{i \in s(W_t)} w_{t,i} x_{t,i}$.
for $w \in \mathcal{W}'$ **do**
Update the distribution (cf. Equations (2.8) for the definitions of $\tilde{\ell}_t(w)$):

$$u_{t+1}(w) = u_t(w) e^{-\eta(\tilde{\ell}_t(w) + \langle c, s(w)\rangle)}, \quad w \in \mathcal{W}'\,.$$

end for
end for

The probability distribution p_t is obtained from the weights $(u_t(\cdot))_{w \in \mathcal{W}'}$ after normalization and mixing the resulting distribution with some "exploration distribution", $(\mu(\cdot))_{w \in \mathcal{W}'}$. Let $0 < \gamma < 1$ be the "mixing" or "exploration" parameter (to be chosen later), and let $U_t = \sum_{w \in \mathcal{W}'} u_t(w)$. Then,

$$p_t(w) = (1-\gamma)\frac{u_t(w)}{U_t} + \gamma\mu(w), \qquad w \in \mathcal{W}'\,.$$

Note that if μ is such that, for any $1 \le i, j \le d$, $\sum_{w \in W' : i,j \in s(w)} \mu(w) > 0$, then $q_t(i,j) > 0$ will be guaranteed for all time steps.

The pseudocode of the resulting algorithm, which we call LQDEXP3, is given as Algorithm 2. In the name of the algorithm, LQ stands for Linear prediction, Quadratic losses and D stands for discretization. Define

$$E_{\mathcal{G}} = \max_{s \in \mathcal{G}} \sup_{w \in \mathcal{W} : \|w\|_* = 1} \|w \odot s\|_*, \tag{2.9}$$

$$y_{\lim} = w_{\lim} x_{\lim}\,. \tag{2.10}$$

Now we are ready to state the main theorem of this section.

Theorem 2.5.3. *Let $w_{\lim}, x_{\lim} > 0$, $c \in [0,\infty)^d$ be given, $\mathcal{W} \subset B_{\|\cdot\|_*}(0, w_{\lim})$ convex, $\mathcal{X} \subset B_{\|\cdot\|}(0, x_{\lim})$ and fix $T \ge 1$. Then, there exist a parameter setting*

for LQDEXP3 *such that the following holds: Let* R_T *denote the regret of* LQDEXP3 *against the best linear predictor from* $\mathrm{Lin}(\mathcal{W}, \mathcal{X})$ *when* LQDEXP3 *is used in an online free-label probing problem defined with the sequence* $((x_t, y_t))_{1 \le t \le T}$ *for* $\|x_t\| \le x_{\lim}$, $|y_t| \le w_{\lim} x_{\lim}$, $1 \le t \le T$, *quadratic losses* $\ell_t(y) = (y - y_t)^2$, *and feature-costs given by the vector* c. *Then,*

$$\mathbb{E}\left[R_T\right] \le C \sqrt{T d \left(4 y_{\lim}^2 + \|c\|_1\right)\left(W_\infty^2 X_1^2 + 2 y_{\lim} W_\infty X_1 + 4 y_{\lim}^2 + \|c\|_1\right) \ln(E_{\mathcal{G}} T)}\,,$$

where $C > 0$ *is a universal constant (i.e., the value of* C *does not depend on the problem parameters).*

The actual parameter setting to be used with the algorithm is constructed in the proof.

Before stating the proof, let us state a lemma that we will need in the proof of this theorem. This lemma, which is essentially extracted from the paper by Auer et al. (2002a), gives a bound on the regret of an exponential weights algorithm as a function of some "statistics" of the losses fed to the algorithm:

Lemma 2.5.2. *Fix the integers* $N, T > 0$, *the real numbers* $0 < \gamma < 1$, $\eta > 0$ *and let* μ *be a probability mass function over the set* $\underline{N} = \{1, \dots, N\}$. *Let* $\ell_t : \underline{N} \to \mathbb{R}$ *be a sequence of loss functions such that*

$$\eta \ell_t(i) \ge -1 \tag{2.11}$$

for all $1 \le t \le T$ *and* $i \in \underline{N}$. *Define the sequence of functions* $(u_t)_{1 \le t \le T}, (p_t)_{1 \le t \le T}$ $(u_t : \underline{N} \to \mathbb{R}^+,\ p_t : \underline{N} \to [0,1])$ *by* $u_t \equiv 1$,

$$u_t(i) = \exp\left(\eta \sum_{s=1}^{t-1} \ell_s(i)\right), \qquad i \in \underline{N},\ 1 \le t \le T,$$

and

$$p_t(i) = (1-\gamma)\frac{u_t(i)}{\sum_{j \in \underline{N}} u_t(j)} + \gamma \mu(i), \qquad i \in \underline{N},\ 1 \le t \le T.$$

Let $\hat{L}_T = \sum_{t=1}^T \sum_{j \in \underline{N}} p_t(j) \ell_t(j)$ *and* $L_T(i) = \sum_{t=1}^T \ell_t(i)$. *Then, for any* $i \in \underline{N}$,

$$\hat{L}_T - L_T(i) \le \frac{\ln N}{\eta} + \eta \sum_{t=1}^T \sum_{j \in \underline{N}} p_t(j) \ell_t^2(j) + \gamma \sum_{t=1}^T \sum_{j \in \underline{N}} \mu(j) \left\{\ell_t(j) - \ell_t(i)\right\}.$$

The proof is provided in Section B.1.2.

Proof of Theorem 2.5.3. Fix the sequence of $((x_t, y_t))_{1\le t\le T}$ as in the statement of the theorem and let $\ell_t(y) = (y - y_t)^2$. Remember that (with a slight abuse of notation), the loss of using weight $w \in W$ in time step t is

$$\ell_t(w) = \ell_t(\langle\, w, x_t \,\rangle), \quad 1 \le t \le T\,.$$

Then total cumulative loss of the algorithm is

$$\hat{L}_T = \sum_{t=1}^{T} \left[\langle\, s(W_t), c \,\rangle + \ell_t(W_t)\right],$$

where $s(W_t) \in \mathcal{G} \subset \{0,1\}^d$ is the subset of features selected at time step t and W_t are the random prediction weights at the same time step. Let

$$L_T(w) = T\langle\, s(w), c \,\rangle + \sum_{t=1}^{T} \ell_t(w), \qquad w \in \mathbb{R}^d,$$

be the total loss of using the weight vector w. Then the regret of LQDEXP3 up to time T on the sequence $((x_t, y_t))_{1\le t\le T}$ can be written as

$$R_T = \max_{w\in\mathcal{W}} R_T(w)\,,$$

where

$$R_T(w) \doteq \hat{L}_T - L_T(w), \qquad w \in \mathbb{R}^d.$$

Using the discretized weight vector set, $\mathcal{W}'$, the regret can be written as

$$\begin{aligned} R_T &= \max_{w\in\mathcal{W}} R_T(w) \\ &= \left\{\hat{L}_T - \min_{w'\in\mathcal{W}'} L_T(w')\right\} + \left\{\min_{w'\in\mathcal{W}'} L_T(w') - \min_{w\in\mathcal{W}} L_T(w)\right\} \\ &= \left\{\hat{L}_T - \min_{w'\in\mathcal{W}'} L_T(w')\right\} + \max_{w\in\mathcal{W}} \min_{w'\in\mathcal{W}'} \left\{L_T(w') - L_T(w)\right\}. \end{aligned} \tag{2.12}$$

Now, fix $w \in \mathcal{W}$. By construction, $\mathcal{W}'$ is such that, for any $s \in \{0,1\}^d$, there exists some vector $w' \in \mathcal{W}'$ such that $s(w') = s$. Then,

$$\begin{aligned} \min_{w'\in\mathcal{W}'} \left\{L_T(w') - L_T(w)\right\} &\le \min_{w'\in\mathcal{W}':s(w')=s(w)} \left\{L_T(w') - L_T(w)\right\} \\ &= \min_{w'\in\mathcal{W}':s(w')=s(w)} \sum_{t=1}^{T} \ell_t(w') - \ell_t(w). \end{aligned}$$

Let us first deal with the second term. A simple calculation shows that $\ell_t : [-y_{\lim}, y_{\lim}] \to \mathbb{R}$ where $\ell_t(y) = (y - y_t)^2$ is $4y_{\lim}$-Lipschitz. Hence, as long as $w' \in W'$ is such that $s(w') = s(w)$,

$$L_T(w') - L_T(w) = \sum_{t=1}^{T} \ell(\langle\, w', x_t \,\rangle, y_t) - \ell_t(\langle\, w, x_t \,\rangle, y_t) \leq 4Ty_{\lim} \left(\frac{1}{T} \sum_{t=1}^{T} |\langle\, w - w', x_t \,\rangle| \right).$$

For $s \in \mathcal{G}$, define $\mathcal{W}'(s) = \{w \in \mathcal{W}' \,|\, s(w) = s\}$ and $W(s) = \{w \in \mathcal{W} \,|\, s(w) = s\}$. For $\alpha > 0$, let $W_\alpha(s) \subset \mathcal{W}$ be the minimal cardinality subset of $\mathcal{W}(s)$ such that $\mathrm{Lin}(\mathcal{X}, \mathcal{W}_\alpha(s))$ is an α-cover of $\mathrm{Lin}(\mathcal{X}, \mathcal{W}(s))$ w.r.t. $d_{\mathcal{Y}}(y, y') = |y - y'|$. Choose

$$\mathcal{W}' = \cup_{s \in \mathcal{G}} \mathcal{W}_\alpha(s)\,.$$

Then, by construction,

$$\min_{w' \in \mathcal{W}'} L_T(w') - L_T(w) \leq 4Ty_{\lim}\alpha \tag{2.13}$$

and since this holds for any $w \in W$, we get that the same bound applies to $\max_{w \in \mathcal{W}} \min_{w' \in \mathcal{W}'} L_T(w') - L_T(w)$. Before we turn to bounding the first term of (2.12), let us bound the cardinality of $\mathcal{W}'$, which we will need later.

Notice that

$$|\mathcal{W}'| \leq \sum_{s \in \mathcal{G}} |\mathcal{W}_\alpha(s)| \leq |\mathcal{G}| \max_{s \in \mathcal{G}} |\mathcal{W}_\alpha(s)|.$$

Now, note also that, thanks to the definition of $E_{\mathcal{G}}$ (cf. (2.9)), for any $s \in \mathcal{G}$, $w \in \mathcal{W}$, $\|w\|_* \leq E_{\mathcal{G}} \cdot \|w \odot s\|_*$. Let $\mathcal{W}_\alpha$ denote a minimum cardinality α-cover of $\mathcal{W}$. Then, for any $s \in \mathcal{G}$, $\mathrm{Lin}(\mathcal{X}, \mathcal{W}_{\alpha/E_{\mathcal{G}}})$ is an α-cover of $\mathrm{Lin}(\mathcal{X}, \mathcal{W}(s))$ w.r.t. $d_{\mathcal{Y}}(y, y') = |y - y'|$. Hence, by the minimum cardinality property of $\mathcal{W}_\alpha(s)$, we have $|\mathcal{W}_\alpha(s)| \leq |\mathcal{W}_{\alpha/E_{\mathcal{G}}}|$ and, by Lemma 2.5.1, we get that $\ln^+ |\mathcal{W}_\alpha(s)| \leq d \ln(1 + 2E_{\mathcal{G}} y_{\lim}/\alpha)$. Hence,

$$\ln |\mathcal{W}'| \leq \ln(|\mathcal{G}|) + d \ln(1 + 2E_{\mathcal{G}} y_{\lim}/\alpha). \tag{2.14}$$

Let us now turn to bounding the expectation of the first term of (2.12). We have

$$\mathbb{E}\left[\hat{L}_T - \min_{w \in \mathcal{W}'} L_T(w)\right] = \max_{w \in \mathcal{W}'} \mathbb{E}\left[\hat{L}_T - L_T(w)\right],$$

where we have exploited that $L_T(w)$ is deterministic. Therefore, it suffices to bound $\mathbb{E}\left[\hat{L}_T - L_T(w)\right]$ for any fixed $w \in \mathcal{W}'$. Thus, fix $w \in \mathcal{W}'$.

By the construction of $\tilde{\ell}_t$, $\mathbb{E}\left[\tilde{\ell}_t(w)\right] = \ell_t(w)$ holds. Further, it also holds that

$$\mathbb{E}\left[\tilde{\ell}_t(W_t)\right] = \mathbb{E}\left[\ell_t(W_t)\right] \tag{2.15}$$

Indeed, by the tower rule,

$$\mathbb{E}\left[\tilde{\ell}_t(W_t)\right] = \mathbb{E}\left[\mathbb{E}\left[\tilde{\ell}_t(W_t)|p_t\right]\right]$$

and

$$\mathbb{E}\left[\tilde{\ell}_t(W_t)|p_t\right] = \sum_{w' \in \mathcal{W}'} p_t(w')\mathbb{E}\left[\tilde{\ell}_t(w')|p_t\right] = \sum_{w' \in \mathcal{W}'} p_t(w')\ell_t(w').$$

The expectation of the right-hand side is $\mathbb{E}\left[\ell_t(W_t)\right]$, while the expectation on the left-hand side (by our earlier remark) is equal to $\mathbb{E}\left[\tilde{\ell}_t(W_t)\right]$. Therefore, (2.15) holds. Introduce $\hat{\tilde{\ell}}_t(w) = \tilde{\ell}_t(w) + \langle\, s(w), c\,\rangle$. Then, we see that it suffices to bound

$$\mathbb{E}\left[\hat{L}_T - L_T(w)\right] = \mathbb{E}\left[\sum_{t=1}^{T}\sum_{w' \in W'} p_t(w')\hat{\tilde{\ell}}_t(w') - \sum_{t=1}^{T}\hat{\tilde{\ell}}_t(w)\right].$$

Now, by Lemma 2.5.2, under the assumption that $0 < \gamma \leq 1$, $0 < \eta$ are such that for any $w' \in \mathcal{W}'$, $1 \leq t \leq T$

$$\eta\hat{\tilde{\ell}}_t(w') \geq -1 \tag{2.16}$$

holds, we have

$$\begin{aligned}
&\sum_{t=1}^{T}\sum_{w' \in W'} p_t(w')\hat{\tilde{\ell}}_t(w') - \sum_{t=1}^{T}\hat{\tilde{\ell}}_t(w) \\
&\leq \frac{\ln|\mathcal{W}'|}{\eta} + \eta\sum_{t=1}^{T}\sum_{w' \in \mathcal{W}'} p_t(w')\hat{\tilde{\ell}}_t^2(w') + \gamma\sum_{t=1}^{T}\sum_{w' \in \mathcal{W}'} \mu(w')(\hat{\tilde{\ell}}_t(w') - \hat{\tilde{\ell}}_t(w)).
\end{aligned}$$

Let us assume for a moment that η, γ can be chosen to satisfy Lemma 2.5.2 conditions – we shall return to the choice of these parameters soon. Taking expectations of both sides of the last inequality, we get

$$\begin{aligned}
\mathbb{E}\left[\hat{L}_T - L_T(w)\right] \leq \frac{\ln|\mathcal{W}'|}{\eta} &+ \eta\sum_{t=1}^{T}\sum_{w' \in \mathcal{W}'} \mathbb{E}\left[p_t(w')\hat{\tilde{\ell}}_t^2(w')\right] \\
&+ \gamma\sum_{t=1}^{T}\sum_{w' \in \mathcal{W}'} \mu(w')(\ell_t(w') + \langle\, s(w'), c\,\rangle),
\end{aligned}$$

where we have used that $\mathbb{E}\left[\hat{\tilde{\ell}}_t(w)\right] = \ell_t(w) + \langle\, s(w), c\,\rangle \geq 0$. Thus, we see that it remains to bound $\mathbb{E}\left[p_t(w')\hat{\tilde{\ell}}_t^2(w')\right]$. For this, we use the following lemma whose proof is provided in Section B.1.3:

Lemma 2.5.3. *Let $\mathcal{W}'$, $\tilde{\ell}_t$, p_t be as in* LQDExp3. *Also let $W_\infty = \sup_{w\in\mathcal{W}} \|w\|_\infty$ and $X_1 = \sup_{x\in\mathcal{X}} \|x\|_1$. Then, the following equation holds:*

$$\sum_{w\in\mathcal{W}'} p(w)\mathbb{E}\left[\hat{\tilde{\ell}}^2(w)\,|\,p\right] \leq (4y_{\lim}^2 + \|c\|_1)(W_\infty^2 X_1^2 + 2y_{\lim}W_\infty X_1 + y_{\lim}^2 + \|c\|_1).$$

It remains to bound $\sum_{w'\in\mathcal{W}'} \mu(w')\left(\ell_t(w') + \langle\, s(w'), c\,\rangle\right)$. Because of the bounds on weight vectors in $\mathcal{W}'$ and $((x_t, y_t))_{(1\leq t\leq T)}$, we know that $\ell_t(w') + \langle\, s(w'), c\,\rangle \leq 4y_{\lim}^2 + \|c\|_1$. Combining the inequalities obtained so far, we get

$$\begin{aligned}\mathbb{E}\left[\hat{L}_T - L_T(w)\right] \leq\ & \frac{\ln|\mathcal{W}'|}{\eta} \\ & + \eta(4y_{\lim}^2 + \|c\|_1)(W_\infty^2 X_1^2 + 2y_{\lim}W_\infty X_1 + y_{\lim}^2 + \|c\|_1) \\ & + \gamma T(4y_{\lim}^2 + \|c\|_1).\end{aligned} \tag{2.17}$$

Thus, it remains to select η, γ such that the earlier imposed conditions, amongst them (2.16), hold and the above bound on the expected regret is minimized. To ensure $\eta\hat{\tilde{\ell}}_t(w) \geq -1$, we start with a lower bound on $\tilde{\ell}_t(w)$:

$$\begin{aligned}\tilde{\ell}_t(w) &= w^\top \tilde{X}_t\, w - 2\, w^\top \tilde{x}_t\, y_t + y_t^2 \\ &\geq w^\top \tilde{X}_t\, w - 2\, w^\top \tilde{x}_t\, y_t \\ &= \sum_{i,j=1}^{d} w_i w_j (\tilde{X}_t)_{i,j} - 2y_t \sum_{j=1}^{d} w_j \tilde{x}_{t,j} \\ &\geq -\frac{\sum_{i,j} \mathbb{1}_{\{i\in s(w)\}}\mathbb{1}_{\{j\in s(w)\}}|x_{t,i}x_{t,j}w_i w_j|}{\gamma} - 2y_{\lim}\frac{\sum_i \mathbb{1}_{\{i\in s(w)\}}|x_{t,i}w_i|}{\gamma} \\ &\geq -\frac{\|w\|_*^2\|x_t\|^2}{\gamma} - 2y_{\lim}\frac{\|w\|_*\|x_t\|}{\gamma} \\ &\geq -\frac{3y_{\lim}^2}{\gamma}\,.\end{aligned}$$

The above derivation used the fact that μ is chosen such that $q_t(i) \geq \gamma$ and $q_t(i,j) \geq \gamma$ for all $1 \leq i, j \leq d$. To ensure that, we can set the probability distribution μ to be zero for all predictors except the one that observe all the features. Thus, as long as

$3\eta y_{\lim}^2 \leq \gamma$, it follows that (2.16) holds. To minimize (2.17), we choose

$$\gamma = 3\eta y_{\lim}^2 \tag{2.18}$$

to get

$$\begin{aligned}\mathbb{E}\left[\hat{L}_T - L_T(w)\right] \leq \frac{\ln|\mathcal{W}'|}{\eta} \\ + \eta(4y_{\lim}^2 + \|c\|_1)(W_\infty^2 X_1^2 + 2y_{\lim} W_\infty X_1 + 4y_{\lim}^2 + \|c\|_1)\,.\end{aligned}$$

Using $\eta = \sqrt{\frac{\ln|\mathcal{W}'|}{(4y_{\lim}^2+\|c\|_1)(W_\infty^2 X_1^2+2y_{\lim}W_\infty X_1+4y_{\lim}^2+\|c\|_1)}}$, we get

$$\begin{aligned}&\mathbb{E}\left[\hat{L}_T\right] - L_T(w) \\ &\leq 2\sqrt{T\,(4y_{\lim}^2 + \|c\|_1)(W_\infty^2 X_1^2 + 2y_{\lim}W_\infty X_1 + 4y_{\lim}^2 + \|c\|_1)\ln|\mathcal{W}'|}\,.\end{aligned}$$

Noting that here $w \in \mathcal{W}'$ was arbitrary, together with the regret decomposition (2.12), the bound (2.13) on the regret arising from discretization, the bound (2.14) on $\ln|\mathcal{W}'|$ and that $\ln|\mathcal{G}| \leq d\ln 2$, give

$$\begin{aligned}&\mathbb{E}\left[R_T\right] \\ &\leq 2\sqrt{T\,(4y_{\lim}^2 + \|c\|_1)(W_\infty^2 X_1^2 + 2y_{\lim}W_\infty X_1 + 4y_{\lim}^2 + \|c\|_1)\ln|\mathcal{W}'|} + 4\,T\,y_{\lim}\alpha \\ &\leq 2\sqrt{Td\,(4y_{\lim}^2 + \|c\|_1)(W_\infty^2 X_1^2 + 2y_{\lim}W_\infty X_1 + 4y_{\lim}^2 + \|c\|_1)\ln(2 + 4E_{\mathcal{G}}y_{\lim}/\alpha)} \\ &+ 4\,T\,y_{\lim}\alpha\,.\end{aligned}$$

Choosing $\alpha = y_{\lim}T^{-1/2}$, we get the bound

$$\mathbb{E}\left[R_T\right] \leq C\sqrt{Td\,(4y_{\lim}^2 + \|c\|_1)(W_\infty^2 X_1^2 + 2y_{\lim}W_\infty X_1 + 4y_{\lim}^2 + \|c\|_1)\ln(E_{\mathcal{G}}T)}\,. \tag{2.19}$$

for some constant $C > 0$. □

2.6 Lower bound for Free-Label Probing with Linear Predictors

In this section, we show that there exists an online free label probing game with linear predictors and quadratic losses such that the expected regret of any algorithm is $\Omega(\sqrt{Td})$. So the regret bound provided in the previous section is tight within logarithmic factors in terms of the number of rounds and the dimension of data.

Theorem 2.6.1. *Given $d > 0, T \geq \frac{4d}{8\ln(4/3)}$, $\varepsilon > 0$, for the following set of parameters*

$$\|w_t\|_1 \leq 1\,, \qquad \|x_t\|_\infty \leq 1\,, \qquad |y_t| \leq 1\,, \qquad c = \varepsilon \times \mathbf{1} \in \mathbb{R}^d\,,$$

and loss function $\ell_t(w_t) = (w_t^\top x_t - y_t)^2 + \langle\, s(w_t), c\,\rangle$ at round t, there exist a sequence of $((x_t, y_t))_{1\leq t\leq T}$ for the online free label probing with linear predictors such that the regret of any algorithm can be lower bounded by

$$\mathbb{E}[R_T] \geq \frac{\sqrt{2}-1}{\sqrt{32\ln(4/3)}}\sqrt{Td}\,.$$

Proof. The idea of the proof is similar to Mannor and Shamir (2011, Theorem 4). We will solve the problem of Multi-armed Bandits with d arms using an algorithm that can solve free-label probing with examples having d features. We will use the lower bound proved in Cesa-Bianchi and Lugosi (2006, Theorem 6.11) for Multi-armed Bandit game. They showed a method of choosing the losses and proved that there exist a universal constant C_{MAB} such that no algorithm can achieve an expected regret better than $C_{\text{MAB}}\sqrt{Td}$ in T rounds using d arms. In their method, an adversary chooses one of the arms beforehand and assign a random Bernoulli loss with parameter $1/2 + \varepsilon$ to that arm and a random Bernoulli loss with parameter $1/2$ to all other arms at each round. Then they proved that by choosing $\varepsilon = \sqrt{(1/(8\ln(4/3))d/T}$, no algorithm can achieve an expected regret bound better than $C_{\text{MAB}}\sqrt{Td}$ in T rounds. Note that they use the fact that losses are in range $[0, 1]$. Without loss of generality we can add ε to all the losses and assume that the losses are now in range $[\varepsilon, 1 + \varepsilon]$ and their result still hold.

Now we explain how we can solve that problem using an algorithm that solves free label probing game. More formally we will use the following lemma.

Lemma 2.6.1. *Give any learner $\mathcal{A}$ for an online free-label probing game, there exist a learner $\mathcal{A}'$ for Multi-armed Bandit problem with the adversaries proposed in Cesa-Bianchi and Lugosi (2006, Theorem 6.11) and an adversary for online free-label probing game such that*

$$\mathbb{E}\left[R_{\mathcal{A}'}(T, \textit{MAB})\right] - 2d\sqrt{(1/(8\ln(4/3))} \leq \mathbb{E}\left[R_{\mathcal{A}}(T, \textit{OFLP})\right]\,,$$

holds where $R_{\mathcal{A}'}(T, \textit{MAB})$ is the regret of the learner $\mathcal{A}'$ in the Multi-armed Bandit problem with the defined adversary and $R_{\mathcal{A}}(T, \textit{OFLP})$ is the regret of the learner $\mathcal{A}$ in the online free-label probing game.

Proof. We define the adversary in the online free label probing game. The adversary chooses $y_t = 1$ for all the rounds. Note that the challenge is finding a weight vector to predict the label and not only predicting the label. Consider the weight vector e_i that, for each $i \in \{1, 2, \ldots, d\}$ is a zero weight vector with a single one in its ith element. The adversary then chooses one of the components v in advance and sets $x_{t,i}$ to be a Bernoulli random variable with parameter one for every $i \neq v$ and sets $x_{t,v}$ to be a Bernoulli random variable with parameter $1/2 + \varepsilon$. Note that this component v is the same arm as the adversary in multi-arm bandit chooses. Now we know that for each e_i, the loss will be the cost of observing ith feature, which is $1/2$, and a prediction error, which is a Bernoulli random variable based on the assignments to the features. So you can easily see a correspondence between e_i and ith arm in Multi-armed Bandit problem with the adversary defined in Cesa-Bianchi and Lugosi (2006, Theorem 6.11).

Let $R_{\mathcal{A}}(T, \text{OFLP})$ denote the regret of the learner $\mathcal{A}$ in this online free-label probing. We know that if we make the set of competitors smaller, the regret cannot be increased. Note that we do not change the set of actions that algorithm $\mathcal{A}$ can take. Let $R^*_{\mathcal{A}}(T, \text{OFLP})$ denote the regret of the learner $\mathcal{A}$ in this online free-label probing when it competes only against e_i weight vectors for all $1 \leq i \leq d$. Since we make the set of competitors smaller, we have

$$R^*_{\mathcal{A}}(T, \text{OFLP}) \leq R_{\mathcal{A}}(T, \text{OFLP})\,. \tag{2.20}$$

Now consider the learner $\mathcal{A}$ that solves this online free-label probing game. We will construct another algorithm $\mathcal{A}'$ such that solves the Multi-armed Bandits problem. Let I_t denote the chosen arm by $\mathcal{A}'$ and $\ell_{t,i}$ denote the loss of arm i at round $t \geq 1$. Here are the different situations.

When $\mathcal{A}$ chooses $w_t = \mathbf{0} \in \mathbb{R}^d$ at round t, $\mathcal{A}'$ chooses one of the arms randomly in the Multi-armed Bandit problem. By this choice, $\mathcal{A}$ does not observe any feature and predicts zero for the label. The expected regret at these types of rounds for $\mathcal{A}$,

is:

$$\mathbb{E}\left[\ell_t(\mathbf{0}) - \ell_t(e_v)\right] = 1 - (1/2 + \mathbb{E}\left[(e_v^\top x_t - y_t)^2\right] = 1 - (\varepsilon + 1/2 - \varepsilon) = 1/2\,.$$

On the other hand, the expected regret of $\mathcal{A}'$ in the game of Multi-armed Bandits at each round is bounded by ε. By this we know that in the rounds that $\mathcal{A}$ chooses $w_t = \mathbf{0} \in \mathbb{R}^d$ we get

$$\mathbb{E}\left[\ell_{t,I_t} - \ell_{t,v}\right] = \mathbb{E}\left[\ell_t(e_{I_t}) - \ell_t(e_v)\right] \leq \varepsilon = \mathbb{E}\left[\ell_t(w_t) - \ell_t(e_t)\right]\,, \tag{2.21}$$

which means the regret of $\mathcal{A}'$ is not going to be increased more that regret of $\mathcal{A}$ in such rounds.

When $\mathcal{A}$ chooses a weight vector $w_t \neq \mathbf{0}$ in the free-label probing game, $\mathcal{A}'$ chooses all arms i in the bandit game whose corresponding ith component of w_t is not zero in the consecutive rounds. After finding all required component values of x, it gives it to $\mathcal{A}$ as the feedback for calculating the loss. Note that the weight vector chosen by $\mathcal{A}$ requires either one feature or more than one feature. As a result, $\mathcal{A}'$ plays the bandit games for T' rounds while $\mathcal{A}$ plays the online free-label probing game for T rounds. If w_t needs only one feature, due to the way the choice of $x_{t,i}$, the minimizer of expected loss is exactly e_i. Because if the ith component of w_t was α instead of one we get

$$\mathbb{E}\left[(w_t^\top x_t - y_t)^2\right] = \mathbb{E}\left[(\alpha x_{t,i} - 1)^2\right] = \mathbb{P}\left[x_{t,i} = 0\right] \times 1 + \mathbb{P}\left[x_{t,i} = 1\right] \times (1-\alpha)^2\,.$$

which achieves its minimum for $\alpha = 1$. So we get Eq.(2.21) for these types of rounds as well. Now if w_t has more than one non-zero components, as we said $\mathcal{A}'$ plays more rounds. At these extra rounds the expected regret of $\mathcal{A}'$ will be increased by at most ε. However $\mathcal{A}$ is also paying for those extra features that it needs. Since the cost of each feature is ε, we can conclude that the regret of $\mathcal{A}$ for all these extra rounds is still less than or equal the regret of $\mathcal{A}$ on the rounds that it chooses w_t. Let T' denote the random number of rounds that $\mathcal{A}'$ is playing the bandits game. We know that this number is bounded by dT since at each round $\mathcal{A}$ can choose at most all the features. Putting the above results together with Eq.(2.21), we get

$$\mathbb{E}\left[R_{\mathcal{A}'}(T', \text{MAB})\right] \leq \mathbb{E}\left[R^*_{\mathcal{A}}(T, \text{OFLP})\right]\,.$$

Because the expected regret is increasing in the number of rounds (Mannor and Shamir, 2011), we can use $\mathbb{E}\left[R_{\mathcal{A}'}(T, \text{MAB})\right] \leq \mathbb{E}\left[R_{\mathcal{A}'}(T', \text{MAB})\right]$ and also Eq.(2.20) to get

$$\mathbb{E}\left[R_{\mathcal{A}'}(T, \text{MAB})\right] \leq \mathbb{E}\left[R_{\mathcal{A}}(T, \text{OFLP})\right] .$$

Using the value of ε that Cesa-Bianchi and Lugosi (2006, Theorem 6.11) uses, we get the lemma statement. Also $T \geq \frac{4d}{8\ln(4/3)}$ in the lemma statement guarantees that $\varepsilon \leq 1/2$ which was needed in the middle of the proof. □

Using this lemma and also knowing that

$$\mathbb{E}\left[R_{\mathcal{A}'}(T, \text{MAB})\right] \geq \sqrt{dT}\frac{\sqrt{2}-1}{\sqrt{32\ln(4/3)}}$$

based on the result of Cesa-Bianchi and Lugosi (2006, Theorem 6.11), we can derive

$$\frac{\sqrt{2}-1}{\sqrt{32\ln(4/3)}}\sqrt{dT} \leq \mathbb{E}\left[R_{\mathcal{A}}(T, \text{OFLP})\right] .$$

□

2.7 Non-Free-Label Probing

Now we turn our attention to the problem with $c_{d+1} > 0$. Recall that the learner in this problem does not necessarily see the loss function at the end of each round; but if it does (*i.e.,* $s_{t,d+1} = 1$) it suffers an extra loss of $c_{d+1} > 0$ in that round. We will see that these problems are inherently harder than the ones with free labels. For this setting, we use an ε-*greedy* style algorithm.

We first consider the finitely many experts case since we can easily reduce the other case to this case in *non-free-label probing* with the same method we used in the previous sections.

The idea of the algorithm is very similar to the algorithm "Random Forecaster with a Revealing Action" (Cesa-Bianchi et al., 2006, Figure 2) that plays exponential weights on the experts and it observes the losses with probability γ. At time t, it selects $f_t \in \mathcal{F}$ based on the distribution over experts. It then draws a Bernoulli random variable Z_t with preset parameter γ. If this $Z_t = 0$, the learner then requests

Algorithm 3 Revealing action algorithm for non-free-label online probing

Parameters: Real numbers $0 \leq \eta, \gamma \leq 1$, Set of experts $\mathcal{F}$.
Initialization: $u_1(f) = 1$ ($f \in \mathcal{F}$).
for $t = 1$ **to** T **do**
  Draw $F_t \in \mathcal{F}$ from the probability mass function

$$p_t(f) = \frac{u_t(f)}{\sum_{f \in \mathcal{F}} u_t(f)}, \qquad f \in \mathcal{F}.$$

  Draw a Bernoulli random variable Z_t such that $\mathbb{P}[Z_t = 1] = \gamma$.
  if $Z_t = 0$ **then**
    $S_t = (s(F_t), 0)$ (*i.e.,* $s_{t,d+1} = 0$).
    Obtain the features values, $(x_{t,i})_{i \in s(F_t)}$.
    Predict $\hat{y}_t = F_t(x_t)$.
  else
    $S_t = \mathbf{1} \in \mathbb{R}^{d+1}$ (*i.e.,* all $d+1$ components are one).
    Observe all the features of x_t.
    Predict $\hat{y}_t = F_t(x_t)$.
    Receive the true label y_t.
  end if
  for each $f \in \mathcal{F}$ **do**
    $\tilde{\ell}_t(f) = \mathbb{1}_{\{Z_t = 1\}} \frac{\langle s(f), c_{1:d} \rangle + \ell_t(\hat{y}_t)}{\gamma}$.
    $u_{t+1}(f) = u_t(f) \exp(-\eta \tilde{\ell}_t(f))$.
  end for
end for

only the features that are "needed". Otherwise, if $Z_t = 1$, the learner requests the label and also asks for all the features (that is, $s_t = \mathbf{1} \in \mathbb{R}^{d+1}$). We will call these rounds *exploration rounds*. The extra loss suffered in these rounds is the cost of the label (*i.e.,* c_{d+1}) and the cost of features whose s_t coordinate is one.

In exploration rounds, the losses of all actions can be calculated, and thus the weights of all actions will be updated via importance weighting; see Algorithm 3.

The following lemma is an upper bound on the expected regret achieved by Algorithm 3.

Lemma 2.7.1. *Given any non-free-label online probing with finitely many experts, Algorithm 3 with appropriately set parameters achieves*

$$\mathbb{E}[R_T] \leq C T^{2/3} (\ell_{\max}^2 \|c\|_1 \ln |\mathcal{F}|)^{1/3}$$

for some constant $C > 0$.

Proof. The regret has two additive terms: the extra cost we pay in exploration rounds and the regret of the exponential weighted algorithm. We bound each one separately to get the total regret bound; see Section B.1.4 in the appendix. □

Algorithm 3 works for any prediction loss function and any finite set of experts. Also the regret is not exponential with respect to dimension. However it has to keep track of all experts and update each of them at each round which means the running time at each round is $\mathcal{O}(|\mathcal{F}|)$.

Now we solve non-free-label probing in the case of linear predictor experts. When $\mathcal{F}$ is the set of linear predictors (cf. (2.4)), since $|\mathcal{F}|$ appears under logarithm in the above bound, we can easily use the set $\mathcal{F}'$ (cf. (2.5)) introduced in previous section as the set of function for Algorithm 3. The following theorem states the regret bound for the case of linear predictors using this technique.

Theorem 2.7.1. *Given any non-free-label online probing with linear predictor experts and Lipschitz prediction loss function with constant L, Algorithm 3 with appropriately set parameters running on $\mathcal{D}_\alpha$ achieves*

$$\mathbb{E}[R_T] \leq C\, T^{2/3} \left[\ell_{\max}^2 \|c\|_1\, d\, \ln(TLWX)\right]^{1/3}$$

for some constant $C > 0$.

Proof. We have to add approximation error from Equation (2.6) for all rounds to the regret in Lemma 2.7.1 and substitute size $|\mathcal{F}|$ with $\mathcal{N}_T(\mathcal{F}, \alpha)$ using Lemma B.1.1 and optimize over α. Using $\alpha = \frac{1}{LT}$, we get the desired bound. □

We obtain the expected regret bound for the algorithm but like before, this algorithm must track all of the experts, which makes this algorithm quite impractical. Since the number of the experts are $\mathcal{O}(T^d)$. This will open to find a practical algorithm that achieves the same regret bound.

Finally, the regret bound in *non-free-label probing* is still in $\tilde{\mathcal{O}}(T^{2/3})$ as opposed to *free-label probing* that has $\tilde{\mathcal{O}}(\sqrt{T})$ regret bound. In the next section we provide a

lower bound for the expected regret *non-free-label probing* that proves this problem cannot be solved with better than $\mathcal{O}(T^{2/3})$ regret bound.

2.8 Lower Bound for the Non-Free-Label Probing[12]

In this section we present a lower bound on the expected regret of a non-trivial class of non-free-label probing problems. As we see, this lower bound is within a logarithmic factor of the upper bound from Section 2.7.

Theorem 2.8.1. *Let the prediction loss function be $\hat{\ell}_t(y) = (y_t - y)^2$ (the quadratic loss). There exists a constant C such that for any non-free-label probing with linear predictor expert and $c_j > (1/d)\sum_{i=1}^{d} c_i - \frac{1}{2d}$ for every $j = 1, \ldots, d$, the expected regret of any algorithm can be lower bounded by*

$$\mathbb{E}[R_T] \geq C(c_{d+1}d)^{1/3}T^{2/3}\,.$$

Proof. Here, we propose a set of strategies whose action losses are close to each other. However one of them is slightly better and we will show that no algorithm can find the optimal action without suffering $\mathcal{O}(T^{2/3})$ regret; see Section B.1.5 in the appendix. □

[12] Joint work with Gábor Bartók

Chapter 3

CAO in Batch Framework

This chapter focuses on applying CAO to the batch learning task. We can look at classifying the objects on conveyor belt as a batch problem. We have several training instances, with known labels. However extracting features from the training samples takes time. Now the learner not only should find an accurate predictor but also it should minimize the time it needs to extract the features in the training phase. So it has to decide wisely which feature to observe in the training phase. There are several different versions of this framework: For example we can have a hard limit for the *total time* of observing features throughout the training phase or we can have a time limit for extracting features from *each instance* in the training phase. Section 3.1 shows more variations of this problem and previous results. Section 3.2 shows how a low-regret *online* learning algorithm can be used to find a predictor whose risk (including the cost of purchasing the features) is almost as good as that of the predictor with the smallest expected total risk over a pre-specified set of predictors. Note that the treatment in Section 3.2 is based on both folklore results, though we were specifically inspired by the clean explanation of these results in the thesis of Shalev-Shwartz (2007).

3.1 Batch Framework

In this section, we explain the batch framework and fit our problem in this framework. The standard "batch learning" framework has a pure explore phase, of giving the learner a set of labeled, completely specified examples, followed by a pure

exploit phase, where the learned predictor is asked to predict the label for novel instances. Notice those standard learners are not required (nor even allowed) to decide which information to gather. By contrast, "active (batch) learning" requires the learner to identify which information to collect (Settles, 2009). Most such active learners begin with completely specified, but unlabeled instances; they then purchase labels for a subset of the instances. But our problem is more similar to the "active feature-purchasing learning" or "budgeted learning" framework (Greiner et al., 2002; Lizotte et al., 2003), which requires the learner to decide which feature values, of which instances, to pay to observe. This is extended in Kapoor and Greiner (2005) to a version that requires the eventual predictor (as well as the learner) to pay to see feature values as well. In this sense, in our problem we are also looking for the predictor that needs to pay for the required features at the test time as well and the goal is finding such a predictor with minimal risk while keeping the training cost small.

Following this, Cesa-Bianchi et al. (2010) divide LAO, which we described in Section 2.2 in the batch settings, into different categories – global budget (*i.e.,* the total number of the features in the training phase is limited), local budget (*i.e.,* the observed features from each individual example is limited) and prediction budget (*i.e.,* the learner has access to the entire dataset during learning but can access a limited number of features during prediction). They also provide theoretical upper bounds and lower bound of achievable loss for these problems. However they just focused on the hard budget constraints and they do not combine the loss with the cost that learner has to pay for the features. Also they assume that every feature has the same cost, which means the constraint reduces to restricting the total number of observed features. In our problem though, we have a cost vector that assigns different costs to different features, and also define the total loss or risk as the weighted sum of the prediction loss and the cost of the required features for the predictor. So by minimizing the risk, we are actually balancing between prediction loss and cost of the features, and not imposing hard limit for either.

Another work, Deng et al. (2007), employs Multi-armed Bandit algorithms (see Section 2.2) and proposes an algorithm for the global limit budget that suggest

the order of features from different examples to observe, based on the amount of information each gives to the learner. Although they do not provide any theoretical proof for their algorithm, they try different reward functions for the features and show some successful experimental results. Dulac-Arnold et al. (2012) combine the cost for the observing features with the prediction loss and show some equivalence between the resulting objective function with the reward function in a Markov Decision Process and address the problem of prediction budget case. Also they extend the problem of prediction budget when we have a hard budget for learning phase as well. However they do not show any convergence proof to the optimal solutions for their algorithm. We do this by extending our results in the online framework, using batch to online conversion methods to bound the quality of a predictor that works well on stochastically generated data at the end in our framework and show some expected and high probability theoretical bound. We do not address the problem of having a hard budget either in training phase nor in the prediction phase in this work though. However we theoretically bound the total loss of our predictor, which is sum of the costs and prediction losses in the training phase compared to the best possible predictor, and using this we can find a bound on the risk for the final predictor; see Section 3.2.

3.2 From regret bounds to generalization bounds

In this section, we use the results in Chapter 2 to investigate the statistical analogue of the online learning problem in the case when label cost is zero to be able to derive the bounds for the batch settings. For this section we fix a set of admissible loss functions $\mathcal{L} \subset \{\ell \,|\, \ell : \mathcal{Y} \to \mathbb{R}\}$. The results in this section use well-known ideas (cf. Section 5 of Shalev-Shwartz (2011), or Appendix B of Shalev-Shwartz (2007) and the references therein). They are included here mainly to clarify the "proper" statistical analogue of the online learning problem studied beforehand.

This analogue is defined as follows: Assume that we are given a sequence of pairs of random inputs and prediction loss functions, $D_T \doteq ((X_t, \mathfrak{L}_t), t = 1, \ldots, T)$ that are sampled from an unknown distribution P over $\mathcal{X} \times \mathcal{L}$ in an i.i.d. (inde-

pendent, identically distributed) manner. Denote the expected risk of a predictor $f : \mathcal{X} \to \mathcal{Y}$ under the common unknown distribution of the loss functions by

$$\mathfrak{R}(f) = \mathbb{E}\left[\langle\, s(f), c_{1:d}\,\rangle + \mathfrak{L}(f(X))\right],$$

where $(X, \mathfrak{L}_t) \sim P$ is a pair that is independent of D_T. Note that the expected risk, as defined here, also takes into account the cost of using the features.

In this statistical problem, the goal is to design a method that uses the data D_t to return a predictor f_T from a fixed set of predictors $\mathcal{F}$ whose risk $\mathfrak{R}(f_T)$ is close to the best possible within-class risk $\mathfrak{R}^*_{\mathcal{F}} = \inf_{f\in\mathcal{F}} \mathfrak{R}(f)$. That is, one is interested in finding a predictor whose expected risk on future data is close to optimum. Given any method that finds a predictor f_T, we are interested in bounding the *excess risk* $\mathfrak{R}(f_T) - \mathfrak{R}^*_{\mathcal{F}}$.

Assume that we are given an online learning method $\mathcal{A}$. Let $f_t^{\mathcal{A}}(\cdot; \{(x_s, \ell_s)\}_{1\le s\le t-1})$ denote the predictor returned by $\mathcal{A}$ at time step t given the past data $\{(x_s, \ell_s)\}_{1\le s\le t-1}$. Let $\tilde{\ell}_t(f, x) = \langle\, s(f), c_{1:d}\,\rangle + \ell_t(f(x))$ and

$$R_T^{\mathcal{A}} = \sup\Big\{\mathbb{E}\left[\sum_{t=1}^{T} \tilde{\ell}_t(f_t, x_t) - \tilde{\ell}_t(f, x_t)\right] \;\Big|\; (x_t, \ell_t) \in \mathcal{X}\times\mathcal{L},\ f_t(\cdot) = f_t^{\mathcal{A}}(\cdot; \{(x_s, \ell_s)\}_{1\le s\le t-1}), 1 \le t \le T, f \in \mathcal{F}\Big\}$$

denote the worst-case regret of $\mathcal{A}$. We have the following result:

Theorem 3.2.1 (Online to batch conversion)**.** *Let $((X_t, \mathfrak{L}_t), t = 1, \dots, T)$ be an i.i.d. sequence of examples from some distribution P. Let $f_t = f_t^{\mathcal{A}}(\cdot; \{(X_s, \mathfrak{L}_s)\}_{1\le s\le t-1})$ be the predictor returned in time step t when $\mathcal{A}$ is run on D_T in a sequential fashion and let F_T be the predictor chosen from the sequence $(f_t)_{1\le t\le T}$ uniformly at random. Then, the expected excess risk of using the predictor F_T can be bounded by*

$$\mathbb{E}\left[\mathfrak{R}(F_T)\right] - \mathfrak{R}^*_{\mathcal{F}} \le \mathbb{E}\left[R_T^{\mathcal{A}}\right]/T\,.$$

For the sake of brevity, let us introduce the combined loss

$$\tilde{\mathfrak{L}}_t(f, x) = \langle\, s(f), c_{1:d}\,\rangle + \mathfrak{L}_t(f(x)).$$

Proof. A simple conditioning argument shows that we have

$$\mathbb{E}\left[\mathfrak{R}(F_T)\right] = \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\tilde{\mathfrak{L}}_t(f_t, X_t)\right].$$

Now,

$$\sup_{f\in\mathcal{F}}\mathbb{E}\left[\sum_{t=1}^{T}\tilde{\mathfrak{L}}_t(f_t, X_t) - \tilde{\mathfrak{L}}_t(f, X_t)\right] \le R_T^{\mathcal{A}}$$

holds by the definition of $R_T^{\mathcal{A}}$. Using the previous equality and the definition of $\mathfrak{R}_{\mathcal{F}}^*$ gives the desired result. □

Remark 3.2.1. *Examining the proof, it is clear that the independence assumption about the data is not used. That is, the theorem continues to hold as long as the distribution of* $(X_t, \mathfrak{L}_t)$ *is* P.

It might be tempting to use the averaged predictor $\overline{F}_T = \frac{1}{T}\sum_{t=1}^{T} f_t$ instead of the "randomized ensemble predictor" F_T. When the loss functions are convex, it is known that this averaged predictor has also small expected excess risk. However, in our case, the loss functions are amended with the cost of using the features, which involves the term $\langle\, s(\overline{F}_T), c_{1:d}\,\rangle$, which is not convex in $\overline{F}_T$ and thus prevents this argument from going through.

The potential problem with the randomized ensemble predictor is that the variance of its loss may be high. One way of dealing with this is to apply the standard probability boosting method: Split the data into s even parts, each of length $m = T/s$ (assuming for simplicity that T is evenly dividable by s) and let F_i be the randomized ensemble obtained by running $\mathcal{A}$ on the ith part $(1 \le i \le s)$. Assume now that we also have access to a validation set $\{(X_t, \mathfrak{L}_t)\}_{T+1\le t\le T+T'}$ sampled from the same unknown distribution P, independently of the first set and in an i.i.d. fashion. Then select the predictor whose average empirical risk on the validation set is the smallest:

$$i^* = \mathrm{argmin}_{1\le i\le s}\sum_{t=T+1}^{T+T'}\tilde{\mathfrak{L}}_t(F_i, X_t)\,,$$

where with a slight abuse of notation we define

$$\tilde{\mathfrak{L}}_t(F_i, X_t) = \sum_{s=(i-1)m+1}^{im}\tilde{\mathfrak{L}}_t(f_s, X_t)\,.$$

Theorem 3.2.2. *Assume that for any loss $\ell \in \mathcal{L}$, the range of risks is included in the $[0,1]$ interval. Then using $s_\delta = \min(1, \lceil \ln(1/\delta) \rceil)$, with probability $1-3\delta$, it holds that*

$$\mathfrak{R}(F_{i^*}) \leq \mathfrak{R}^*_{\mathcal{F}} + \frac{e s_\delta R^{\mathcal{A}}_{T/s_\delta}}{T} + 2\sqrt{\frac{\ln(s_\delta) + \ln(1/\delta)}{2T'}}\,,$$

where e is the base of natural logarithm.

Remark 3.2.2. *In practice, we have an uniform dataset and we need to decide how to split it into the training set and the validation set. How to split the data to minimize this bound if we have $T'' = T + T'$ data points and what is the resulting bound? This will depend on $R^{\mathcal{A}}_T$. If $R^{\mathcal{A}}_T = \Theta(\sqrt{T})$, then an even split will do and we get an $\mathcal{O}(T^{-1/2})$-rate. When $R^{\mathcal{A}}_T$ is growing faster than $\Omega(T^{1/2})$, then less data should go into the validation set, and the rate will be $R^{\mathcal{A}}_T/T$. When $R^{\mathcal{A}}_T$ is growing slower than $\mathcal{O}(T^{1/2})$, then more data should go into the validation set and the rate will be $\mathcal{O}(T^{-1/2})$ (the second term is the slower term).*

Proof. Fix $0 \leq \delta \leq 1$. Define $\mathfrak{R}_i = \sum_{t=T+1}^{T+T'} \tilde{\mathfrak{L}}_t(F_i, X_t)$. A simple application of Hoeffding's inequality shows that w.p. $1-\delta$, simultaneously for all $1 \leq i \leq s$,

$$\mathfrak{R}_i \leq \mathfrak{R}(F_i) + \sqrt{\frac{\ln(s/\delta)}{2T'}}\,.$$

Taking the minimum of both sides w.r.t. i, we get that

$$\mathfrak{R}_{i^*} \leq \min_{1\leq i\leq s} \mathfrak{R}(F_i) + \sqrt{\frac{\ln(s/\delta)}{2T'}}\,.$$

A similar argument also gives that, w.p. $1-\delta$.

$$\mathfrak{R}(F_{i^*}) \leq \mathfrak{R}_{i^*} + \sqrt{\frac{\ln(s/\delta)}{2T'}}\,.$$

Therefore, w.p. $1-2\delta$,

$$\mathfrak{R}(F_{i^*}) \leq \min_{1\leq i\leq s} \mathfrak{R}(F_i) + 2\sqrt{\frac{\ln(s/\delta)}{2T'}}\,.$$

Now, we want to show that $\min_{1\leq i\leq s} \mathfrak{R}(F_i)$ is also close to $\mathfrak{R}^*_{\mathcal{F}}$ with high probability. For this let us first fix $1 \leq i \leq s$. From Markov's inequality we know that, for any $a > 0$,

$$\mathbb{P}\left[\mathfrak{R}(F_i) - \mathfrak{R}^*_{\mathcal{F}} \geq a\right] \leq \frac{\mathbb{E}\left[\mathfrak{R}(F_i) - \mathfrak{R}^*_{\mathcal{F}}\right]}{a}$$

and thus from Theorem 3.2.1 it follows that

$$\mathbb{P}\left[\mathfrak{R}(F_i) - \mathfrak{R}^*_{\mathcal{F}} \geq a\right] \leq \frac{s\mathbb{E}\left[R^{\mathcal{A}}_{T/s}\right]}{Ta}.$$

Solving $\mathbb{E}\left[sR^{\mathcal{A}}_{T/s}\right]/(Ta) = 1/e$ for a (e is the base of the natural logarithm function), we get that with probability $1 - 1/e$,

$$\mathfrak{R}(F_i) - \mathfrak{R}^*_{\mathcal{F}} \leq \frac{e\, s\, \mathbb{E}\left[R^{\mathcal{A}}_{T/s}\right]}{T}.$$

Due to the independence of the s blocks, the probability that this not hold for *all* $1 \leq i \leq s$ is at most e^{-s}. Thus, in the opposite case, *i.e.,* with probability at least $1 - e^{-s}$, there exists some index $1 \leq i \leq s$ such that

$$\mathfrak{R}(F_i) \leq \mathfrak{R}^*_F + \frac{e\, s\, \mathbb{E}\left[R^{\mathcal{A}}_{T/s}\right]}{T}.$$

Therefore,

$$\min_{1\leq i\leq s} \mathfrak{R}(F_i) \leq \mathfrak{R}^*_{\mathcal{F}} + \frac{e\, s\, \mathbb{E}\left[R^{\mathcal{A}}_{T/s}\right]}{T}.$$

Combining the inequalities obtained, we get that with probability at least $1 - (2\delta + e^{-s})$, it holds that

$$\mathfrak{R}(F_{i^*}) \leq \mathfrak{R}^*_{\mathcal{F}} + \frac{e\, s\, \mathbb{E}\left[R^{\mathcal{A}}_{T/s}\right]}{T} + 2\sqrt{\frac{\ln(s/\delta)}{2T'}}.$$

Choosing $e^{-s} = \delta$, *i.e.,* $s = \min(1, \lceil \ln(1/\delta) \rceil)$, we get that w.p. $1 - 3\delta$,

$$\mathfrak{R}(F_{i^*}) \leq \mathfrak{R}^*_{\mathcal{F}} + \frac{e\, s\, \mathbb{E}\left[R^{\mathcal{A}}_{T/s}\right]}{T} + 2\sqrt{\frac{\ln(1 + \ln(1/\delta)) + \ln(1/\delta)}{2T'}},$$

finishing the proof. □

It would be interesting to consider a fixed budget B and see how well we can do when we have this hard budget in the learning phase using the proposed online approach. Then we can compare this with Dulac-Arnold et al. (2012) and Cesa-Bianchi et al. (2010). However this remains as future work.

Chapter 4

Conclusions

4.1 Future Works

Here we mention several future avenues and extensions for this thesis.

Free label case, Lipschitz losses, covering number argument: We have focused on Lipschitz loss functions; it would be interesting to consider other loss functions, such as zero-one loss. Here, if $\mathcal{F}$ has a finite Littlestone dimension (see, *e.g.,* Ben-David et al. (2009)), our result will continue to hold, by just replacing the metric entropy, $\ln \mathcal{N}_T(\mathcal{F}, \alpha)$, with $\mathcal{F}$'s Littlestone dimension.

Free label case, linear prediction, with quadratic losses: We have exploited the *algebraic* properties of the quadratic loss functions and the predictors to design an algorithm that enjoys a $\mathcal{O}(\sqrt{dT})$ regret bound. However we have shown that, with fewer assumptions, the regret will scale exponentially with the dimension in the proposed algorithm. This raises the question of whether we can do better given finite experts with any loss function; and what properties are sufficient and necessary to ensure that the worst-case regret is $\Theta(\mathrm{poly}(d)\sqrt{T})$.

Also, the algorithm proposed for the case of quadratic losses and linear prediction, although it tames the exponential dependence of the regret, is computationally expensive: Both its memory and computational requirements scale exponentially with the dimension d. It remains an important open problem to design an algorithm whose computational complexity, as well as regret, scale polynomially with the dimension, while keeping the $\sqrt{T}$ dependence of the regret on time.

Of course, when one has a small *a priori* bound S on the number of features that can be used in a single time step, the exponential dependence on d becomes polynomial (*i.e.,* $\mathcal{O}(d^S)$): to achieve this only consider all possible subset of $\{1, \ldots, d\}$ of cardinality S or less. In practice, the bound S may arise from budget constraints.

Convex upper bound on the feature-cost: A major source of difficulty in our formulation is that the cost of features, $\langle\, s(f), c_{1:d} \,\rangle$, is non-convex in f. The situation is similar to the case of zero-one loss, where the standard solution is to use a convex upper bound on the zero-one loss, making it possible to derive tractable algorithms (for the new loss) that come with reasonable performance guarantees. We can use this for our case, as well. Let us consider, for simplicity, the case of linear predictors. Then for every function $f(x) = \langle\, w, x \,\rangle$, a relaxed convex function for $\langle\, s(f), c_{1:d} \,\rangle = \sum_{i=1}^{d} c_i \mathbb{1}_{\{|w_i|>0\}}$ (similar to the hinge loss for zero-one loss) is $\sum_{i=1}^{d} c_i |w_i|$, the $c_{1:d}$-weighted ℓ_1-norm of w. In particular, we see that with this approach we get the familiar Lasso-type penalty. Since the Lasso-type penalty is known to promote sparsity, the algorithm is expected to indeed take into account the varying cost of features. However, it remains for future work to study the behavior of this natural algorithm.

4.2 Contribution

In this work, we introduced a new framework called CAO and a new problem called *online probing* in online settings. This extends previous online learning models by giving the learner the option of choosing the subset of features it wants to observe for each instance, as well as option of observing the true label for that instance. However it has to pay for everything that it observes. In other words, it suffers from a risk function that combines the prediction loss and costs of observing. This assumption produced new challenges in solving the online problem. We solved this problem for two different settings – free label vs costly label – which leads to two different optimal regret bounds. We proved that no learner can solve the *non-free-label online probing* with better than $\mathcal{O}(T^{2/3})$ regret and that the novel FPE algorithm achieves $\tilde{\mathcal{O}}(\sqrt{T})$ for *free-label online probing* and an ε*-greedy* al-

gorithm achieves $\tilde{\mathcal{O}}(T^{2/3})$ for *non-free-label online probing*. These results hold for general loss functions; we also showed that the problem can be solved much more efficiently, and with better regret with respect to dimension of examples, by restricting the prediction loss function to a quadratic loss. Then we used online-to-batch methods to be able to use our online methods in the batch framework and proved the bound on the risk of final predictor.

Bibliography

Agarwal, A. and Duchi, J. C. (2011). Distributed delayed stochastic optimization. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F. C. N., and Weinberger, K. Q., editors, *NIPS*, pages 873–881.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. (2002a). The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77.

Auer, P., Cesa-Bianchi, N., and Gentile, C. (2002b). Adaptive and self-confident on-line learning algorithms. *J. Comput. Syst. Sci.*, 64(1):48–75.

Ben-David, S., Pál, D., and Shalev-Shwartz, S. (2009). Agnostic online learning. In *COLT*.

Beygelzimer, A., Langford, J., Li, L., Reyzin, L., and Schapire, R. E. (2010). An optimal high probability algorithm for the contextual bandit problem. *CoRR*, abs/1002.4058.

Bubeck, S. and Cesa-Bianchi, N. (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *CoRR*, abs/1204.5721.

Cesa-Bianchi, N., Freund, Y., Haussler, D., Helmbold, D. P., Schapire, R. E., and Warmuth, M. K. (1997). How to use expert advice. *J. ACM*, 44(3):427–485.

Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge Univ Pr.

Cesa-Bianchi, N., Lugosi, G., and Stoltz, G. (2005). Minimizing regret with label efficient prediction. *IEEE Transactions on Information Theory*, 51(6):2152–2162.

Cesa-Bianchi, N., Lugosi, G., and Stoltz, G. (2006). Regret minimization under partial monitoring. *Math. Oper. Res.*, 31(3):562–580.

Cesa-Bianchi, N., Shalev-Shwartz, S., and Shamir, O. (2010). Efficient learning with partially observed attributes. *CoRR*, abs/1004.4421.

Dekel, O., Shamir, O., and Xiao, L. (2010). Learning to classify with missing and corrupted features. *Machine Learning*, 81(2):149–178.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38.

Deng, K., Bourke, C., Scott, S. D., Sunderman, J., and Zheng, Y. (2007). Bandit-based algorithms for budgeted learning. In *ICDM*, pages 463–468. IEEE Computer Society.

Dulac-Arnold, G., Denoyer, L., Preux, P., and Gallinari, P. (2012). Sequential approaches for learning datum-wise sparse representations. *Machine Learning*, 89(1-2):87–122.

Greiner, R., Grove, A. J., and Roth, D. (2002). Learning cost-sensitive active classifiers. *Artif. Intell.*, 139(2):137–174.

Hazan, E. and Koren, T. (2012). Linear regression with limited observation. *CoRR*, abs/1206.4678.

Joulani, P. (2012). Multi-armed bandit problems under delayed feedback. Master's thesis, University of Alberta.

Kapoor, A. and Greiner, R. (2005). Learning and classifying under hard budgets. In *European Conference on Machine Learning (ECML)*, pages 166–173.

Little, R. J. A. and Rubin, D. B. (1986). *Statistical analysis with missing data*. John Wiley & Sons, Inc., New York, NY, USA.

Lizotte, D., Madani, O., and Greiner, R. (2003). Budgeted learning of naive-bayes classifiers. In *Conference on Uncertainty in Artificial Intelligence (UAI)*.

Mannor, S. and Shamir, O. (2011). From bandits to experts: On the value of side-observations. *CoRR*, abs/1106.2436.

Mesterharm, C. (2005). On-line learning with delayed label feedback. In *Proceedings of the 16th international conference on Algorithmic Learning Theory*, ALT'05, pages 399–413, Berlin, Heidelberg. Springer-Verlag.

Rostamizadeh, A., Agarwal, A., and Bartlett, P. L. (2011). Learning with missing features. In *UAI*, pages 635–642.

Settles, B. (2009). Active learning literature survey. Technical report, University of WisconsinMadison.

Shalev-Shwartz, S. (2007). *Online learning: Theory, algorithms, and applications*. PhD thesis, The Hebrew University.

Shalev-Shwartz, S. (2011). Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194.

Weinberger, M. J. and Ordentlich, E. (2006). On delayed prediction of individual sequences. *IEEE Trans. Inf. Theor.*, 48(7):1959–1976.

Appendix A

Glossary

approximation error The difference between the loss of a predictor and the loss of its approximated predictor. 15, 32

CAO Costly Attribute Observation. 3, 6, 34

classifier A predictor with discrete output space. Sometimes, the output space has only two elements: either positive or negative. 1

competitor A strategy that makes its prediction based on some inputs. This strategy in general might be a simple predictor function or a learner itself. This word means the same as expert in our context. 9, 11, 12

expert A strategy that makes its prediction based on some inputs. This strategy in general might be a simple predictor function or a learner itself. This word means the same as competitor in our context. 5, 7, 12, 30

LAO Limited Attribute Observation. 8, 35

learner An algorithm that learns from examples to produce a predictor that generates a label for unlabeled examples. 2, 4, 5, 7, 8, 10, 12, 30, 34

prediction loss A loss that a predictor suffers because of its prediction. Usually there is a prediction loss function that penalizes the predictor for its prediction and it maps the output space of the predictor to non-negative real numbers – *e.g.,* quadratic loss. 1, 2, 6, 9, 10, 12, 17, 35, 36

predictor A function that maps partially observed feature space into the output space. 1, 2, 4, 5, 8, 12, 26, 32, 34, 35, 37

regret The difference between the learner's cumulative loss and cumulative loss of any predictor from a given set of predictors in online learning framework. 4, 9, 26

risk This is the total loss of a predictor and goal of the problem is to minimize this risk. In this document it has two components: prediction loss and the costs of the required features. 1, 2, 6, 19, 34, 35, 37

Appendix B

Proofs

B.1 Proofs

B.1.1 Covering numbers for balls in Euclidean spaces

Lemma B.1.1. *Let $\|\cdot\|$ be a norm on the d-dimensional Euclidean space. Then for any $R, \alpha > 0$, $N^* = \left\lceil \left(1 + \frac{2R}{\alpha}\right)^d \right\rceil$ balls, each of radius α, suffice to cover the ball $B(0, R) = \{x \mid \|x\| \leq R\}$.*

Proof. Fix $R, \alpha > 0$. Let $X = \{x_1, \ldots, x_N\}$ be the largest set such that any two distinct points in the set are at least α-apart. Then $B(0, R) \subset \cup_{i=1}^N B(x_i, \alpha)$ because otherwise we could fit one more point into the set X. Pick any $c < 1/2$. The balls $B(x_i, c\alpha)$ for $i = 1, \ldots, N$ are disjoint and $\cup_{i=1}^N B(x_i, c\alpha) \subset B(0, R + c\alpha)$. The volume of a ball with radius r is $C_d r^d$ where C_d is a constant that depends on the norm $\|\cdot\|$ and the dimension d. Hence,

$$NC_d(c\alpha)^d = \text{Vol}\left(\cup_{i=1}^N B(x_i, c\alpha)\right) \leq \text{Vol}\left(B(0, R + c\alpha)\right) = C_d(R + c\alpha)^d .$$

From this we get that $N \leq \left(\frac{R+c\alpha}{c\alpha}\right)^d$. Since this is true for any $c < 1/2$, it is also true for $c = 1/2$. By substituting c with $1/2$ we get the desired bound. □

B.1.2 Proof of Lemma 2.5.2

Lemma 2.5.2. *Fix the integers $N, T > 0$, the real numbers $0 < \gamma < 1$, $\eta > 0$ and let μ be a probability mass function over the set $\underline{N} = \{1, \ldots, N\}$. Let $\ell_t : \underline{N} \to \mathbb{R}$ be a sequence of loss functions such that*

$$\eta \ell_t(i) \geq -1 \tag{2.11}$$

for all $1 \le t \le T$ *and* $i \in \underline{N}$. *Define the sequence of functions* $(u_t)_{1\le t\le T}, (p_t)_{1\le t\le T}$ $(u_t : \underline{N} \to \mathbb{R}^+,\ p_t : \underline{N} \to [0,1])$ *by* $u_t \equiv 1$,

$$u_t(i) = \exp\left(\eta \sum_{s=1}^{t-1} \ell_s(i)\right), \qquad i \in \underline{N},\ 1 \le t \le T,$$

and

$$p_t(i) = (1-\gamma)\frac{u_t(i)}{\sum_{j\in\underline{N}} u_t(j)} + \gamma\mu(i), \qquad i \in \underline{N},\ 1 \le t \le T.$$

Let $\hat{L}_T = \sum_{t=1}^T \sum_{j\in\underline{N}} p_t(j)\ell_t(j)$ *and* $L_T(i) = \sum_{t=1}^T \ell_t(i)$. *Then, for any* $i \in \underline{N}$,

$$\hat{L}_T - L_T(i) \le \frac{\ln N}{\eta} + \eta \sum_{t=1}^T \sum_{j\in\underline{N}} p_t(j)\ell_t^2(j) + \gamma \sum_{t=1}^T \sum_{j\in\underline{N}} \mu(j)\left\{\ell_t(j) - \ell_t(i)\right\}.$$

Proof. Let $U_t = \sum_{1\le i\le N} u_t(i)$. The bound will follow from upper and lower bounds on $\ln U_T$. As for the lower bound, we have

$$\ln U_T \ge \ln u_T(i) = -\eta L_T(i).$$

On the other hand,

$$\ln(U_T) = \ln(U_1) + \ln(\tfrac{U_2}{U_1}) + \ldots + \ln(\tfrac{U_T}{U_{T-1}}). \tag{B.1}$$

and thus it suffices to upper bound the terms $\ln(\frac{U_t}{U_{t-1}})$. Thanks to the weight update rule,

$$\frac{U_t}{U_{t-1}} = \sum_{i\in\underline{N}} \frac{u_t(i)}{U_{t-1}} = \sum_{i\in\underline{N}} \frac{u_{t-1}(i)}{U_{t-1}} e^{-\eta\ell_t(i)}.$$

By assumption, $-\eta\ell_t(i) \le 1$. Hence, applying $e^x \le 1 + x + x^2$ which holds for $x \le 1$ to bound $e^{-\eta\ell_t(i)} \le 1$ we get

$$\begin{aligned}
\frac{U_t}{U_{t-1}} &\le \sum_{i\in\underline{N}} \frac{u_{t-1}(i)}{U_{t-1}}\{1 - \eta\ell_t(i) + \eta^2\ell_t^2(i)\} \\
&= 1 - \eta \sum_{i\in\underline{N}} \frac{p_t(i) - \gamma\mu(i)}{1-\gamma}\ell_t(i) + \eta^2 \sum_{i\in\underline{N}} \frac{p_t(i) - \gamma\mu(i)}{1-\gamma}\ell_t^2(i) \\
&\le 1 + \frac{-\eta\sum_{i\in\underline{N}} p_t(i)\ell_t(i) + \eta\gamma\sum_{i\in\underline{N}} \mu(i)\ell_t(i) + \eta^2\sum_{i\in\underline{N}} p_t(i)\ell_t^2(i)}{1-\gamma}.
\end{aligned}$$

Note that the right-hand side is positive since the left-hand side is positive Using $\ln(x) \le x - 1$ which holds for any $x > 0$, we get

$$\ln(\tfrac{U_t}{U_{t-1}}) \le \frac{-\eta\sum_{i\in\underline{N}} p_t(i)\tilde{\ell}_t(i) + \eta\gamma\sum_{i\in\underline{N}} \mu(i)\ell_t(i) + \eta^2\sum_{i\in\underline{N}} p_t(i)\ell_t^2(i)}{1-\gamma}.$$

Plugging these upper bounds into (B.1) and since $U_1 = N$, we get

$$\ln(U_T) \leq \ln(N) + \frac{-\eta \sum_{t=1}^{T} \sum_{i \in \underline{N}} p_t(i)\tilde{\ell}_t(i) + \eta\gamma \sum_{t=1}^{T} \sum_{i \in \underline{N}} \mu(i)\ell_t(i) + \eta^2 \sum_{t=1}^{T} \sum_{i \in \underline{N}} p_t(i)\ell_t^2(i)}{1-\gamma}.$$

Putting the lower and upper bounds of $\ln(U_T)$ together and introducing

$$\overline{L}_T = \sum_{t=1}^{T} \sum_{i \in \underline{N}} \mu(i)\ell_t(i), \qquad Q_T = \sum_{t=1}^{T} \sum_{i \in \underline{N}} p_t(i)\ell_t^2(i),$$

gives

$$-\eta\tilde{L}_T(i) \leq \ln N - \frac{\eta\hat{L}_T}{1-\gamma} + \frac{\eta\gamma\overline{L}_T}{1-\gamma} + \frac{\eta^2 Q_T}{1-\gamma}.$$

Multiplying both sides by $1-\gamma$ and reordering the terms yields

$$\begin{aligned}\eta\hat{L}_T - \eta L_T(i) &\leq (1-\gamma)\ln N + \eta\gamma(\overline{L}_T - L_T) + \eta^2 Q_T \\ &\leq \ln N + \eta\gamma(\overline{L}_T - L_T(i)) + \eta^2 Q_T\,.\end{aligned}$$

Dividing both sides by η gives the final result. □

B.1.3 Proof of Lemma 2.5.3

Lemma 2.5.3. *Let $\mathcal{W}'$, $\tilde{\ell}_t$, p_t be as in* LQDExp3. *Also let $W_\infty = \sup_{w\in\mathcal{W}} \|w\|_\infty$ and $X_1 = \sup_{x\in\mathcal{X}} \|x\|_1$. Then, the following equation holds:*

$$\sum_{w\in\mathcal{W}'} p(w)\mathbb{E}\left[\hat{\tilde{\ell}}^2(w) \,|\, p\right] \leq (4y_{\lim}^2 + \|c\|_1)(W_\infty^2 X_1^2 + 2y_{\lim}W_\infty X_1 + y_{\lim}^2 + \|c\|_1).$$

Proof. By the tower rule, we have

$$\mathbb{E}\left[\sum_{w\in\mathcal{W}'} p_t(w)\hat{\tilde{\ell}}_t^2(w)\right] = \mathbb{E}\left[\sum_{w\in\mathcal{W}'} p_t(w)\mathbb{E}\left[\hat{\tilde{\ell}}_t^2(w) \,|\, p_t\right]\right]$$

Therefore, it suffices to bound

$$\sum_{w\in\mathcal{W}'} p_t(w)\mathbb{E}\left[\hat{\tilde{\ell}}_t^2(w) \,|\, p_t\right].$$

For simplifying the presentation, since t is fixed, from now on we will remove the subindex t from the quantities involved and write $\hat{\tilde{\ell}}$ instead of $\hat{\tilde{\ell}}_t$, p instead of p_t, etc.

The plan of the proof is as follows: We construct a deterministic upper bound $h(w)$ on $|\hat{\tilde{\ell}}(w)|$ and an upper bound B on $\sum_{w\in\mathcal{W}'} p(w)h(w)$. Then, we provide an upper bound B' on $\mathbb{E}\left[\hat{\tilde{\ell}}(w)|p\right]$ so that

$$\sum_{w\in\mathcal{W}'} p(w)\mathbb{E}\left[\hat{\tilde{\ell}}^2(w)\,\middle|\,p\right] \le \sum_{w\in\mathcal{W}'} p(w)h(w)\mathbb{E}\left[\hat{\tilde{\ell}}(w)\,\middle|\,p\right] \le B'\sum_{w\in\mathcal{W}'} p(w)h(w) \le BB'.$$

Before providing these bounds, let's review some basic relations. Remember that $W_\infty = \sup_{w\in\mathcal{W}}\|w\|_\infty$ and $X_1 = \sup_{x\in\mathcal{X}}\|x\|_1$. Further, note that for any $1 \le j, j' \le d$, we have

$$\mathbb{E}\left[\mathbb{1}_{\{j\in s(W)\}}\,\middle|\,p\right] = \sum_{w\in\mathcal{W}':j\in s(w)} p(w) = \sum_{w\in\mathcal{W}'} \mathbb{1}_{\{j\in s(w)\}}p(w) = q(j), \tag{B.2}$$

$$\mathbb{E}\left[\mathbb{1}_{\{j,j'\in s(W)\}}\,\middle|\,p\right] = \sum_{w\in\mathcal{W}':j,j'\in s(w)} p(w) = \sum_{w\in\mathcal{W}'} \mathbb{1}_{\{j,j'\in s(w)\}}p(w) = q(j,j'). \tag{B.3}$$

As to the upper bound $h(w)$ on $|\hat{\tilde{\ell}}(w)|$, we start with

$$|\hat{\tilde{\ell}}(w)| \le |w^\top \tilde{X} w| + 2|y|\,|w^\top \tilde{x}| + |y|^2 + \|c\|_1. \tag{B.4}$$

Now, $|y| \le y_{\lim}$ and

$$|w^\top \tilde{x}| \le W_\infty \sum_{j=1}^{d} \mathbb{1}_{\{j\in s(w)\}} \frac{|x_j|}{q(j)} \doteq g(w,x),$$

$$|w^\top \tilde{X} w| \le W_\infty^2 \sum_{j,j'} \mathbb{1}_{\{j,j'\in s(w)\}} \frac{|x_j x_{j'}|}{q(j,j')} \doteq G(w,x).$$

Hence,

$$|\hat{\tilde{\ell}}(w)| \le G(w,x) + 2y_{\lim}g(w,x) + y_{\lim}^2 + \|c\|_1 \doteq h(w)$$

which is indeed a deterministic upper bound on $|\hat{\tilde{\ell}}(w)|$. To bound $\sum_{w\in\mathcal{W}'} p(w)h(w)$, it remains to upper bound $\sum_{w\in\mathcal{W}'} p(w)g(w,x)$ and $\sum_{w\in\mathcal{W}'} p(w)G(w,x)$. To upper bound these, we move the sum over the weights w inside the other sums in the

definitions of g and G to get:

$$\begin{aligned}
\sum_{w\in\mathcal{W}'} p(w)g(w,x) &= W_\infty \sum_{j=1}^{d} \frac{|x_j|}{q(j)} \sum_{w\in\mathcal{W}'} p(w)\mathbb{1}_{\{j\in s(w)\}} \\
&= W_\infty X_1, \qquad \text{(by (B.2) and } \|x\|_1 \le X_1\text{)} \\
\sum_{w\in\mathcal{W}'} p(w)G(w,x) &= W_\infty^2 \sum_{j,j'} \frac{|x_j x_{j'}|}{q(j,j')} \sum_{w\in\mathcal{W}'} p(w)\mathbb{1}_{\{j,j'\in s(w)\}} \\
&= W_\infty^2 \sum_{j,j'} |x_j x_{j'}| = W_\infty^2 \| x \|_1^2 \\
&\le W_\infty^2 X_1^2. \qquad \text{(by (B.3) and } \|x\|_1 \le X_1\text{)}
\end{aligned}$$

Hence,

$$\sum_{w\in\mathcal{W}'} p(w)h(w) \le W_\infty^2 X_1^2 + 2y_{\lim} W_\infty X_1 + y_{\lim}^2 + \|c\|_1.$$

Let us now turn to bounding $\mathbb{E}\left[|\hat{\tilde{\ell}}(w)| \,\middle|\, p\right]$. From (B.4), it is clear that it suffices to upper bound $\mathbb{E}\left[|w^\top \tilde{X} w| \,|\, p\right]$ and $\mathbb{E}\left[|w^\top \tilde{x}| \,|\, p\right]$. From (B.2) and (B.3), the definitions of $\tilde{x}$ and $\tilde{X}$ and because by assumption $\||w|\|_* \||x|\| \le w_{\lim} x_{\lim} = y_{\lim}$, we obtain

$$\begin{aligned}
\mathbb{E}\left[|w^\top \tilde{x}| \,\middle|\, p\right] &= \sum_j |w_j x_j| \le y_{\lim} \quad \text{and} \\
\mathbb{E}\left[|w^\top \tilde{X} w| \,\middle|\, p\right] &= \sum_{j,j'} |w_j w_{j'} x_j x_{j'}| = \Big(\sum_j |w_j x_j|\Big)^2 \le y_{\lim}^2.
\end{aligned}$$

Thus,

$$\mathbb{E}\left[|\hat{\tilde{\ell}}(w)| \,\middle|\, p\right] \le \mathbb{E}\left[|w^\top \tilde{X} w| + 2y_{\lim}|w^\top \tilde{x}| + y_{\lim}^2 + \|c\|_1 \,\middle|\, p\right] \le 4y_{\lim}^2 + \|c\|_1.$$

Putting together all the bounds, we get

$$\sum_{w\in\mathcal{W}'} p(w)\mathbb{E}\left[\hat{\tilde{\ell}}^2(w) \,|\, p\right] \le (4y_{\lim}^2 + \|c\|_1)(W_\infty^2 X_1^2 + 2y_{\lim} W_\infty X_1 + y_{\lim}^2 + \|c\|_1).$$

□

B.1.4 Proof of Lemma 2.7.1

The regret of the algorithm is decomposed into two additive terms:

1. The extra loss suffered in exploration rounds. The cumulative expectation of this extra loss can be upper bounded by $T\gamma\|c\|_1$.

2. The regret of the algorithm compared to each expert, excluding rounds that request the label and extra features. To upper bound this term, we follow the classical "exponential weights" proof (see *e.g.,* Cesa-Bianchi et al. (2006)).

   First we make the trivial observation that for every time step t and $f \in \mathcal{F}, \mathbb{E}[\tilde{\ell}_t(f)] = \langle s(f), c_{1:d} \rangle + \ell_t(f(s \odot x_t))$. That is, $\tilde{\ell}_t(f)$ is an unbiased estimate of the true loss of function f. Let $U_t = \sum_{f\in\mathcal{F}} u_t(f)$. Now we continue with lower and upper bounding the term U_T:

$$U_T \geq \sum_{f\in\mathcal{F}} u_T(f) \geq u_T(f^*) = \exp\left(-\eta \sum_{t=1}^{T} \tilde{\ell}_t(f^*)\right) ,$$

   where f^* is and arbitrary expert in $\mathcal{F}$. For the upper bound we write

$$\begin{aligned}
\frac{U_t}{U_{t-1}} &= \sum_{f\in\mathcal{F}} \frac{u_{t-1}(f)\exp(-\eta\tilde{\ell}_t(f))}{U_{t-1}} \\
&= \sum_{f\in\mathcal{F}} p_t(f)(1-\eta\tilde{\ell}_t(f)+\eta^2\tilde{\ell}_t^2(f)) \qquad \text{(B.5)} \\
&= 1-\eta\sum_{f\in\mathcal{F}} p_t(f)\tilde{\ell}_t(f)+\eta^2\sum_{f\in\mathcal{F}} p_t(f)\tilde{\ell}_t^2(f) \\
&\leq \exp\left(-\eta\sum_{f\in\mathcal{F}} p_t(f)\tilde{\ell}_t(f)+\eta^2\sum_{f\in\mathcal{F}} p_t(f)\tilde{\ell}_t^2(f)\right) , \qquad \text{(B.6)}
\end{aligned}$$

   where in (B.5) we used that $u_{t-1}(f)/U_{t-1} = p_t(f)$ and the inequality $e^x \leq 1+x+x^2$ if $x \leq 1$, and in (B.6) we used that $e^x \geq 1+x$. Multiplying the above inequality for $t = 1, \ldots, T$ and also U_1 we get

$$U_T \leq |\mathcal{F}| \exp\left(-\eta\sum_{t=1}^{T}\sum_{f\in\mathcal{F}} p_t(f)\tilde{\ell}_t(f)+\eta^2\sum_{t=1}^{T}\sum_{(s,f(.))\in\mathcal{F}} p_t(f)\tilde{\ell}_t^2(f)\right) .$$

   We now merge the lower and upper bounds and take logarithm of both sides:

$$-\eta\sum_{t=1}^{T}\tilde{\ell}_t(f^*) - \ln|\mathcal{F}| \leq -\eta\sum_{t=1}^{T}\sum_{f\in\mathcal{F}} p_t(f)\tilde{\ell}_t(f)+\eta^2\sum_{t=1}^{T}\sum_{f\in\mathcal{F}} p_t(f)\tilde{\ell}_t^2(f) .$$

Rearranging gives

$$\sum_{t=1}^{T}\sum_{f\in\mathcal{F}} p_t(f)\tilde{\ell}_t(f) - \sum_{t=1}^{T}\tilde{\ell}_t(f^*) \le \eta\sum_{t=1}^{T}\sum_{f\in\mathcal{F}} p_t(f)\tilde{\ell}_t^2(f) + \frac{\ln|\mathcal{F}|}{\eta}\,.$$

After taking expectation of both sides, the first term on the left hand side is the expected cumulative loss of the algorithm excluding the extra loss suffered in exploration rounds, while the second term is the expected cumulative loss of the any arbitrary expert f. The first term on the right hand side can be upper bounded as

$$\begin{aligned}\eta\sum_{t=1}^{T}\sum_{f\in\mathcal{F}} \mathbb{E}[p_t(f)\tilde{\ell}_t^2(f)] &\le \eta\sum_{t=1}^{T}\sum_{f\in\mathcal{F}} \mathbb{E}[p_t(f)\tilde{\ell}_t(f)]\frac{\ell_{\max}}{\gamma}\\ &\le \frac{\eta\ell_{\max}^2 T}{\gamma}\,,\end{aligned}$$

where $\ell_{\max}$ is the maximum loss an action can suffer, ignoring the label cost c_{d+1}.

Adding up the two terms of the expected regret, we get

$$\mathbb{E}[R_T] \le T\gamma\|c\|_1 + \frac{\eta\ell_{\max}^2 T}{\gamma} + \frac{\ln|\mathcal{F}|}{\eta}\,.$$

Setting the parameters to

$$\eta = (\ln|\mathcal{F}|)^{2/3}T^{-2/3}(4\ell_{\max}^2\|c\|_1)^{-1/3} \qquad \gamma = \sqrt{\frac{\eta\ell_{\max}^2}{\|c\|_1}}$$

we get

$$\mathbb{E}[R_T] \le CT^{2/3}(\ell_{\max}^2\|c\|_1\ln|\mathcal{F}|)^{1/3}$$

for some constant $C > 0$. □

B.1.5 Proof of Theorem 2.8.1

To prove that we propose a set of strategies which is basically a subset of weight vectors. We construct a set of opponent strategies and show that the expected regret of any algorithm is high against at least one of them. The features $x_{t,i}$ for

$t = 1, \dots, T$ and $i = 1, \dots, d$ are generated by the iid random variables $X_{t,i}$ whose distribution is Bernoulli with parameter 0.5. Let $Z_t \in \{1, \dots, d\}$ be random variables whose distribution will be specified later. The labels y_t are generated by the random variable defined as $Y_t = X_{t,Z}$.

To construct the distribution of Z_t we introduce the following notation. For every $i = 1, \dots, d$, let

$$a_i = \frac{1}{d} + 2c_i - \frac{2}{d} \sum_{j=1}^{d} c_j \,.$$

The assumptions on c ensures that $a_i > 0$ for every $i = 1, \dots, d$. For opponent strategy k, let the distribution of Z_t defined as

$$\mathbb{P}_k \left[Z_t = i \right] = \begin{cases} a_i - \varepsilon, & i \neq k; \\ a_i + (d-1)\varepsilon, & \text{i=k} \,, \end{cases}$$

with some $\varepsilon > 0$ to be defined later.

Lemma B.1.2. *Let e_k denote the k^{th} basis vector of dimension d. Against opponent strategy k, the instantaneous expected regret for any action such that $(s, s_\ell) \neq (e_k, 0)$ is at least $\frac{d\varepsilon}{2}$.*

For $i = 1, \dots, d$, let N_i denote the number of times the player's action is (e_i, w, s_{d+1}). Similarly, let N_L denote the number of times the player requests the label. Now it is easy to see that the expected regret under opponent strategy k can be lower bounded by

$$\mathbb{E}_k[R_T] \geq (T - \mathbb{E}_k[N_k]) \, \frac{d\varepsilon}{2} + c_{d+1} \mathbb{E}_k[N_L] \,.$$

The rest of the proof is devoted to show that for any algorithm, the average of the above value, $1/d \sum_{i=1}^{d} \mathbb{E}_i[R_T]$ can be lower bounded. We only show this for deterministic algorithms. The statement follows for randomizing algorithms with the help of a simple argument, see *e.g.,* Cesa-Bianchi and Lugosi (2006, Theorem 6.11).

A deterministic algorithm is defined as a sequence of functions $A_t(\cdot)$, where the argument of A_t is a sequence of observations up to time step $t-1$ and the value is the action taken at time step t. We denote the observation at time step t

by $h_t \in \{0,1,*\}^{d+1}$, where $h_{t,i} = x_{t,i}$ if $s_{t,i} = 1$ and $h_{t,i} = *$ if $s_{t,i} = 0$ for all $1 \le i \le d$. Similarly, $h_{t,d+1} = y_t$ if $s_{t,d+1} = 1$ and $h_{t,d+1} = *$ if $s_{t,d+1} = 0$. That is, $*$ is the symbol for not observing a feature or the label. The next lemma, which is the key lemma of the proof, shows that the expected value of N_i does not change too much if we change the opponent strategy.

Lemma B.1.3. *There exists a constant C_1 such that for any $i, j \in \{1, \ldots, d\}$,*

$$\mathbb{E}_i[N_i] - \mathbb{E}_j[N_i] \le C_1 T\varepsilon\sqrt{d\mathbb{E}_j[N_L]}\,.$$

Now we are equipped to lower bound the expected regret. Let

$$j = \operatorname{argmin}_{k\in\{1,\ldots,d\}} \mathbb{E}_k[N_L].$$

By Lemma B.1.3,

$$\begin{aligned}\mathbb{E}_i[R_T] &\ge (T - \mathbb{E}_i[N_i])\,\frac{d\varepsilon}{2} + c_{d+1}\mathbb{E}_i[N_L] \\ &\ge \left(T - \mathbb{E}_j[N_i] - C_1 T\varepsilon\sqrt{d\mathbb{E}_j[N_L]}\right)\frac{d\varepsilon}{2} + c_{d+1}\mathbb{E}_j[N_L]\end{aligned}$$

Denoting $\sqrt{\mathbb{E}_j[N_L]}$ by ν we have

$$\begin{aligned}\frac{1}{d}\sum_{i=1}^{d}\mathbb{E}_i[R_T] &\ge \left(T - \frac{1}{d}\sum_{i=1}^{d}\mathbb{E}_j[N_i] - C_1 T\varepsilon\sqrt{d}\nu\right)\frac{d\varepsilon}{2} + c_{d+1}\nu^2 \\ &\ge \left(T - \frac{T}{d} - C_1 T\varepsilon\sqrt{d}\nu\right)\frac{d\varepsilon}{2} + c_{d+1}\nu^2\end{aligned}$$

What is left is to optimize this bound in terms of ν and ε. Since ν is the property of the algorithm, we have to minimize the expression in ν, with ε as a parameter. After simple algebra we get

$$\nu_{opt} = \frac{C_1 T\varepsilon^2 d^{3/2}}{4c_{d+1}}\,.$$

Substituting it back results in

$$\frac{1}{d}\sum_{i=1}^{d}\mathbb{E}_i[R_T] \ge (d-1)\frac{T\varepsilon}{2} - \frac{C_1^2 T^2\varepsilon^4 d^3}{16c_{d+1}}$$

Now we set

$$\varepsilon = \left(\frac{2}{C_1^2}\right)^{1/3} (c_{d+1})^{1/3} \, d^{-2/3} T^{-1/3}$$

to get

$$\mathbb{E}[R_T] \geq C_3 \, (c_{d+1})^{1/3} \, d^{1/3} T^{2/3}$$

whenever $d \geq 2$. $\square$