Journal of Machine Learning Research 15 (2014) 2451-2487 Submitted 7/13; Revised 1/14; Published 7/14

On Multilabel Classification and Ranking with Bandit Feedback

Claudio Gentile claudio.gentile@uninsubria.it
DiSTA, Università dell'Insubria
via Mazzini 5
21100 Varese, Italy

Francesco Orabona francesco@orabona.com
Toyota Technological Institute at Chicago
6045 South Kenwood Avenue
60637 Chicago, IL, USA

Editor: Peter Auer

Abstract

We present a novel multilabel/ranking algorithm working in partial information settings. The algorithm is based on 2nd-order descent methods, and relies on upper-confidence bounds to trade-off exploration and exploitation. We analyze this algorithm in a partial adversarial setting, where covariates can be adversarial, but multilabel probabilities are ruled by (generalized) linear models. We show $O(T^{1/2} \log T)$ regret bounds, which improve in several ways on the existing results. We test the effectiveness of our upper-confidence scheme by contrasting against full-information baselines on diverse real-world multilabel data sets, often obtaining comparable performance.

Keywords: contextual bandits, structured prediction, ranking, online learning, regret bounds, generalized linear

1. Introduction

Consider a book recommendation system. Given a customer's profile, the system recommends a few possible books to the user by means of, e.g., a limited number of banners placed at different positions on a webpage. The system's goal is to select books that the user likes and possibly purchases. Typical feedback in such systems is the actual action of the user or, in particular, what books he has bought/preferred, if any. The system cannot observe what would have been the user's actions had other books got recommended, or had the same book ads been placed in a different order within the webpage.

Such problems are collectively referred to as learning with partial feedback. As opposed to the full information case, where the system (the learning algorithm) knows the outcome of each possible response (e.g., the user's action for each and every possible book recommendation placed in the largest banner ad), in the partial feedback setting the system only observes the response to very limited options and, specifically, the option that was actually recommended.

In this and many other examples of this sort, it is reasonable to assume that recommended options are not given the same treatment by the system, e.g., large banners which are displayed on top of the page should somehow be more committing as a recommendation than smaller ones placed elsewhere. Moreover, it is often plausible to interpret the user feedback as a preference (if any) *restricted to* the displayed alternatives.

In this paper, we consider instantiations of this problem in the multilabel and learning-to-rank settings. Learning proceeds in rounds: in round t, the algorithm receives an instance $\boldsymbol{x}_t$ and outputs an ordered subset $\hat{Y}_t$ of labels from a finite set of possible labels $[K] = \{1, 2, \ldots, K\}$. Restrictions might apply to the size of $\hat{Y}_t$ (due, e.g., to the number of available slots in the webpage, or to the specifics of the targeted user). The set $\hat{Y}_t$ corresponds to the aforementioned recommendations, and is intended to approximate the true set of preferences associated with $\boldsymbol{x}_t$. However, the latter set is never observed. In its stead, the algorithm receives $Y_t \cap \hat{Y}_t$, where $Y_t \subseteq [K]$ is a *noisy version* of the true set of user preferences on $\boldsymbol{x}_t$. When we are restricted to $|\hat{Y}_t| = 1$ for all t, this becomes a multiclass classification problem with bandit feedback—see below.

1.1 Related Work

This paper lies at the intersection between online learning with partial feedback and multilabel classification/ranking. Both fields include a substantial amount of work, so we can hardly do it justice here. In the sequel, we outline some of the main contributions in the two fields, with an emphasis on those we believe are the most related to this paper.

A well-known tool for facing the problem of partial feedback in online learning is to trade off exploration and exploitation through upper confidence bounds. This technique has been introduced by Lai and Robbins (1985), and can by now be considered a standard tool. In the so-called *bandit* setting with contextual information (sometimes called bandits with side information or bandits with covariates, e.g., Auer 2002; Dani et al. 2008; Filippi et al. 2010; Crammer and Gentile 2011; Krause and Ong 2011, and references therein) an online algorithm receives at each time step a *context* (typically, in the form of a feature vector $\boldsymbol{x}$) and is compelled to select an action (e.g., a label), whose goodness is quantified by a predefined loss function. Full information about the loss function (one that would perhaps allow to minimize the total loss over the contexts seen so far) is not available. The specifics of the interaction model determines which pieces of loss will be observed by the algorithm, e.g., the actual value of the loss on the chosen action, some information on more profitable directions on the action space, noisy versions thereof, etc. The overall goal is to compete against classes of functions that map contexts to (expected) losses in a regret sense, that is, to obtain *sublinear* cumulative regret bounds.

All these algorithms share the common need to somehow trade off an exploratory attitude for gathering loss information on unchosen directions of the context-action space, and an exploitatory attitude for choosing actions that are deemed best according to the available data. For instance, Auer (2002); Dani et al. (2008); Filippi et al. (2010); Abbasi-Yadkori et al. (2011) work in a finite action space where the mappings context-to-loss for each action are linear (or generalized linear, as Filippi et al., 2010's) functions of the features. They all obtain $T^{1/2}$-like regret bounds, where T is the time horizon. This is extended by Krause and Ong (2011), where the loss function is modeled as a sample from a Gaussian process over the joint context-action space. We are using a similar (generalized) linear modeling here. An earlier (but somehow more general) setting that models such mappings by VC-classes is considered by Langford and Zhang (2008), where a $T^{2/3}$ regret bound has been proven under i.i.d. assumptions. Linear multiclass classification problems with bandit feedback are considered by, e.g., Kakade et al. (2008); Crammer and Gentile (2011); Hazan and Kale (2011), where either $T^{2/3}$ or $T^{1/2}$ or even logarithmic regret bounds are proven, depending on the noise model and the underlying loss functions.

All the above papers do not consider *structured* action spaces, where the learner is allowed to select *sets* of actions, which is more suitable to multilabel and ranking problems. Along these lines are the papers by Hazan and Kale (2009); Streeter et al. (2009); Kale et al. (2010); Slivkins et al. (2010); Shivaswamy and Joachims (2012); Amin et al. (2011). The general problem of online minimization of a submodular loss function under both full and bandit information without covariates is considered by Hazan and Kale (2009), achieving a regret $T^{2/3}$ in the bandit case. Streeter et al. (2009) consider the problem of online learning of assignments, where at each round an algorithm is requested to assign positions (e.g., rankings) to sets of items (e.g., ads) with given constraints on the set of items that can be placed in each position. Their problem shares similar motivations as ours but, again, the bandit version of their algorithm does not explicitly take side information into account, and leads to a $T^{2/3}$ regret bound. Another paper with similar goals but a different mathematical model is by Kale et al. (2010), where the aim is to learn a suitable ordering (an "ordered slate") of the available actions. Among other things, the authors prove a $T^{1/2}$ regret bound in the bandit setting with a multiplicative weight updating scheme. Yet, no contextual information is incorporated. Slivkins et al. (2010) motivate the ability of selecting sets of actions by a problem of diverse retrieval in large document collections which are meant to live in a general metric space. In contrast to our paper, that approach does not lead to strong regret guarantees for specific (e.g., smooth) loss functions. Shivaswamy and Joachims (2012) use a simple linear model for the hidden utility function of users interacting with a web system and providing partial feedback in any form that allows the system to make significant progress in learning this function (this is called an α-informative feedback by the authors). Under these assumptions, a regret bound of $T^{1/2}$ is again provided that depends on the degree of informativeness of the feedback, as measured by the progress made during the learning process. It is experimentally argued that this feedback is typically made available by a user that clicks on relevant URLs out of a list presented by a search engine. Despite the neatness of the argument, no formal effort is put into relating this information to the context information at hand or, more generally, to the way data are generated. The recent paper by Amin et al. (2011) investigates classes of graphical models for contextual bandit settings that afford richer interaction between contexts and actions leading again to a $T^{2/3}$ regret bound.

Finally, further interesting recent works that came to our attention at the time of writing this extended version of our conference paper (Gentile and Orabona, 2012) are the papers by Bartók and Szepesvári (2012), by Bartók (2013), and by Agarwal (2013). In Bartók and Szepesvári (2012), the authors provide sufficient conditions ("local observability") that insure rates of the form $T^{1/2}$ in partial monitoring games with side information. Partial monitoring is an attempt to formalize through a unifying language the partial information settings where the algorithm is observing only partial information about the loss of its action, in the form of some kind of feedback or "signal". The results presented by Bartók and Szepesvári (2012) do not seem to conveniently extend to the structured action space setting we are interested in (or, if they do, we do not see it in the current version of their paper). Moreover, being very general in scope, that paper is missing a tight dependence of the regret bound on the number of available actions, which can be very large in structured action spaces. Progress in this directions has very recently been made by Bartók (2013), where the dependence on the number of actions is replaced by a quantity depending on the structure of the action space in the locally observable game. Yet, no side information is considered in that paper. The paper by Agarwal (2013) investigates multiclass selective sampling settings (similar to Cavallanti et al., 2011; Cesa-Bianchi et al., 2009; Dekel et al., 2012; Orabona and Cesa-Bianchi, 2011) with essentially the same generalized linear models as the ones we consider here. As such, that paper is close to ours only from a technical viewpoint.

The literature on multilabel learning and learning to rank is overwhelming. The wide attention this literature attracts is often motivated by its web-search-engine or recommender-system applications, and many of the papers are experimental in nature. Relevant references include the work by Tsoumakas et al. (2011); Furnkranz et al. (2008); Dembczynski et al. (2012), along with references therein. Moreover, when dealing with multilabel, the typical assumption is full supervision, an important concern being modeling correlations among classes. In contrast to that, the specific setting we are considering here need not face such a modeling issue (Dembczynski et al., 2012). The more recent work by Wang et al. (2012) reduces any online algorithm working on pairwise loss functions (like a ranking loss) to a batch algorithm with generalization bound guarantees. But, again, only fully supervised settings are considered. Other related references are the papers by Herbrich et al. (2000); Freund et al. (2003), where learning is by pairs of examples. Yet, these approaches need i.i.d. assumptions on the data, and typically deliver batch learning procedures. Finally, more recent efforts related to proving consistency of pairwise ranking methods are Clémençon et al. (2005); Cossock and Zhang (2006); Duchi et al. (2010); Buffoni et al. (2011); Lan et al. (2012) where, unlike this paper, multi-level user ratings are assumed to be available.

To summarize, whereas we are technically closer to the linear modeling approaches by Auer (2002); Dani et al. (2008); Dekel et al. (2012); Crammer and Gentile (2011); Filippi et al. (2010); Abbasi-Yadkori et al. (2011); Krause and Ong (2011); Bartók and Szepesvári (2012); Agarwal (2013), from a motivational standpoint we are perhaps closest to Streeter et al. (2009); Kale et al. (2010); Shivaswamy and Joachims (2012).

1.2 Our Results

We investigate the multilabel and learning-to-rank problems in a partial feedback scenario with contextual information, where we assume a probabilistic linear model over the labels, although the contexts can be chosen by an adaptive adversary. We consider two families of loss functions, one is a cost-sensitive multilabel loss that generalizes the standard Hamming loss in several respects, the other is a kind of (unnormalized) ranking loss. In both cases, the learning algorithm is maintaining a (generalized) linear predictor for the probability that a given label occurs, the ranking being produced by upper confidence-corrected estimated probabilities. In such settings, we prove $T^{1/2}\log T$ cumulative regret bounds, which are essentially optimal (up to log factors) in some cases. A distinguishing feature of our user feedback model is that, unlike previous papers (e.g., Hazan and Kale 2009; Streeter et al.

2009; Abbasi-Yadkori et al. 2011; Krause and Ong 2011), we are not assuming the algorithm is observing a noisy version of the risk function on the currently selected action. In fact, when a generalized linear model is adopted, the mapping context-to-risk turns out to be nonconvex in the parameter space. Furthermore, when operating on structured action spaces this more traditional form of bandit model does not seem appropriate to capture the typical user preference feedback. Our approach is based on having the loss decoupled from the label generating model, the user feedback being a noisy version of the gradient of a *surrogate* convex loss associated with the model itself. As a consequence, the algorithm is not directly dealing with the original loss when making exploration. In this sense, we are more similar to the multiclass bandit algorithm by Crammer and Gentile (2011). Yet, our work is a substantial departure from Crammer and Gentile's (2011) in that we lift their machinery to nontrivial structured action spaces, and we do so by means of generalized linear models. On one hand, these extensions pose several extra technical challenges; on the other, they provide additional modeling power and practical advantage.

Though the emphasis is on theoretical results, we also validate our algorithms on real-world multilabel data sets under several experimental conditions: data set size, label set size, loss functions, training mode and performance (online vs. batch), label generation model (linear vs. logistic). Under all such conditions, our algorithms are contrasted against the corresponding multilabel/ranking baselines that operate with full information, often showing (surprisingly enough) comparable prediction performance.

1.3 Structure of the Paper

The paper is organized as follows. In Section 2 we introduce our learning model, our first loss function, the label generation model, and some preliminary results and notation used throughout the rest of the paper. In Section 3 we describe our partial feedback algorithm working under the loss function introduced in Section 2, along with the associated regret analysis. In Section 4 we show that a very similar machinery applies to ranking with partial feedback, where the loss function is a kind of pairwise ranking loss (with partial feedback). Similar regret bounds are then presented that work under additional modeling restrictions. In Section 5 we provide our experimental comparison. Section 6 gives proof ideas and technical details. The paper is concluded with Section 7, where possible directions for future research are mentioned.

2. Model and Preliminaries

We consider a setting where the algorithm receives at time t the side information vector $\boldsymbol{x}_t \in \mathbb{R}^d$, is allowed to output a (possibly ordered) subset[1] $\hat{Y}_t \subseteq [K]$ of the set of possible labels, then the subset of labels $Y_t \subseteq [K]$ associated with $\boldsymbol{x}_t$ is generated, and the algorithm gets as feedback $\hat{Y}_t \cap Y_t$. The loss suffered by the algorithm may take into account several things: the *distance* between Y_t and $\hat{Y}_t$ (both viewed as sets), as well as the *cost* for playing $\hat{Y}_t$. The cost $c(\hat{Y}_t)$ associated with $\hat{Y}_t$ might be given by the sum of costs suffered on each class $i \in \hat{Y}_t$, where we possibly take into account the *order* in which i occurs within $\hat{Y}_t$ (viewed as an ordered list of labels). Specifically, given constant $a \in [0, 1]$ and costs

[1] An ordered subset is like a list with *no repeated* items.

$c = \{c(i,s), i = 1,\ldots,s, s \in [K]\}$, such that $1 \geq c(1,s) \geq c(2,s) \geq \ldots c(s,s) \geq 0$, for all $s \in [K]$, we consider the loss function

$$\ell_{a,c}(Y_t,\hat{Y}_t) = a\,|Y_t \setminus \hat{Y}_t| + (1-a) \sum_{i \in \hat{Y}_t \setminus Y_t} c(j_i, |\hat{Y}_t|),$$

where j_i is the position of class i in $\hat{Y}_t$, and $c(j_i,\cdot)$ depends on $\hat{Y}_t$ only through its size $|\hat{Y}_t|$. In the above, the first term accounts for the false negative mistakes, hence there is no specific ordering of labels therein. The second term collects the loss contribution provided by all false positive classes, taking into account through the costs $c(j_i, |\hat{Y}_t|)$ the order in which labels occur in $\hat{Y}_t$. The constant a serves as weighting the relative importance of false positive vs. false negative mistakes.[2] As a specific example, suppose that $K = 10$, the costs $c(i,s)$ are given by $c(i,s) = (s-i+1)/s, i = 1,\ldots,s$, the algorithm plays the ordered list $\hat{Y}_t = (4,3,6)$, but Y_t is the (unordered) set $\{1,3,8\}$. In this case, $|Y_t \setminus \hat{Y}_t| = 2$, and $\sum_{i \in \hat{Y}_t \setminus Y_t} c(j_i, |\hat{Y}_t|) = 3/3 + 1/3$, i.e., the cost for mistakenly playing class 4 in the top slot of $\hat{Y}_t$ is more damaging than mistakenly playing class 6 in the third slot. In the special case when all costs are unitary, there is no longer need to view $\hat{Y}_t$ as an ordered collection, and the above loss reduces to a standard Hamming-like loss between sets Y_t and $\hat{Y}_t$, i.e., $a\,|Y_t \setminus \hat{Y}_t| + (1-a)\,|\hat{Y}_t \setminus Y_t|$. Notice that the partial feedback $\hat{Y}_t \cap Y_t$ allows the algorithm to know which of the chosen classes in $\hat{Y}_t$ are good or bad (and to what extent, because of the selected ordering within $\hat{Y}_t$).

The reader should also observe the asymmetry between the label set $\hat{Y}_t$ produced by the algorithm and the true label set Y_t: the algorithm predicts an ordered set of labels, but the true set of labels is unordered. In fact, it is often the case in, e.g., recommender system practice, that the user feedback does not contain preference information in the form of an ordered set of items. Still, in such systems we would like to get back to the user with an appropriate ranking over the items.

Working with the above loss function makes the algorithm's output $\hat{Y}_t$ become a ranked list of classes, where ranking is *restricted* to the deemed relevant classes only. In this sense, the above problem can be seen as a partial information version of the multilabel ranking problem (see the work by Furnkranz et al., 2008, and references therein). In a standard multilabel ranking problem a classifier has to provide for any given instance $\boldsymbol{x}_t$, both a separation between relevant and irrelevant classes and a ranking of the classes within the two sets (or, perhaps, over the whole set of classes, as long as ranking is consistent with the relevance separation). In our setting, instead, ranking applies to the selected classes only, but the information gathered by the algorithm while training is partial. That is, only a relevance feedback among the selected classes is observed (the set $Y_t \cap \hat{Y}_t$), but no supervised ranking information (e.g., in the form of pairwise preferences) is provided to the algorithm within this set. Alternatively, we can think of a ranking framework where restrictions on the size of $\hat{Y}_t$ are set by an exogenous (and possibly time-varying) parameter of the problem, and the algorithm is required to provide a ranking complying with these restrictions. In this sense, an alternative interpretation of the ranking-sensitive term $\sum_{i \in \hat{Y}_t \setminus Y_t} c(j_i, |\hat{Y}_t|)$ in $\ell_{a,c}(Y_t,\hat{Y}_t)$ is a Discounted Cumulative Gain (DCG) difference between the optimal ranking

[2] Parameter a is not redundant here, since the costs $c(i,s)$ have been normalized to [0,1].

(i.e., the one sorting the $|Y_t|$ classes in Y_t according to decreasing value of $c(i, |\hat{Y}_t|)$) and the actual ranking contained in $\hat{Y}_t$, the discounting function being just the coefficients $c(i, |\hat{Y}_t|), i = 1, \dots |\hat{Y}_t|$. DCG is a standard metric for measuring the effectiveness of Web search engine algorithms (e.g., Jarvelin and Kekalainen, 2002).

Another important concern we would like to address with our loss function $\ell_{a,c}$ is to avoid combinatorial explosions due to the exponential number of possible choices for $\hat{Y}_t$. As we shall see below, this is guaranteed by the chosen structure for costs $c(i, s)$. Another loss function providing similar guarantees (though with additional modeling restrictions) is the (pairwise) ranking loss considered in Section 4, where more on the connection to the ranking setting with partial feedback is given.

The problem arises as to which noise model we should adopt so as to encompass significant real-world settings while at the same time affording *efficient implementation* of the resulting algorithms. For any subset $Y_t \subseteq [K]$, we let $(y_{1,t}, \dots, y_{K,t}) \in \{0,1\}^K$ be the corresponding indicator vector. Then it is easy to see that

$$\begin{aligned}\ell_{a,c}(Y_t, \hat{Y}_t) &= a \sum_{i \notin \hat{Y}_t} y_{i,t} + (1-a) \sum_{i \in \hat{Y}_t} c(j_i, |\hat{Y}_t|)\,(1 - y_{i,t}) \\ &= a \sum_{i=1}^{K} y_{i,t} + (1-a) \sum_{i \in \hat{Y}_t} \Big(c(j_i, |\hat{Y}_t|) - \Big(\tfrac{a}{1-a} + c(j_i, |\hat{Y}_t|) \Big)\, y_{i,t} \Big)\,.\end{aligned}$$

Moreover, because the first sum does not depend on $\hat{Y}_t$, for the sake of optimizing over $\hat{Y}_t$ (but also for the sake of defining the regret R_T—see below) we can equivalently define

$$\ell_{a,c}(Y_t, \hat{Y}_t) = (1-a) \sum_{i \in \hat{Y}_t} \Big(c(j_i, |\hat{Y}_t|) - \Big(\tfrac{a}{1-a} + c(j_i, |\hat{Y}_t|) \Big)\, y_{i,t} \Big)\,. \qquad (1)$$

Note that the algorithm can evaluate the value of this loss, using the feedback received. Let $\mathbb{P}_t(\cdot)$ be a shorthand for the conditional probability $\mathbb{P}(\cdot \,|\, \boldsymbol{x}_t)$, where the side information vector $\boldsymbol{x}_t$ can in principle be generated by an adaptive adversary as a function of the past. Then

$$\mathbb{P}_t(y_{1,t}, \dots, y_{K,t}) = \mathbb{P}(y_{1,t}, \dots, y_{K,t} \,|\, \boldsymbol{x}_t).$$

We will assume that the marginals $\mathbb{P}_t(y_{i,t} = 1)$ satisfy[3]

$$\mathbb{P}_t(y_{i,t} = 1) = \frac{g(-\boldsymbol{u}_i^\top \boldsymbol{x}_t)}{g(\boldsymbol{u}_i^\top \boldsymbol{x}_t) + g(-\boldsymbol{u}_i^\top \boldsymbol{x}_t)}, \qquad i = 1, \dots, K, \qquad (2)$$

for some K vectors $\boldsymbol{u}_1, \dots, \boldsymbol{u}_K \in \mathcal{R}^d$, and a (known) function $g : D \subseteq \mathcal{R} \to \mathcal{R}^+$, that is the negative derivative of a suitable convex and nonincreasing function. The model is well defined if $\boldsymbol{u}_i^\top \boldsymbol{x} \in D$ for all i and all $\boldsymbol{x} \in \mathcal{R}^d$ chosen by the adversary. We assume for the sake of simplicity that $||\boldsymbol{x}_t|| = 1$ for all t. Notice that here the variables $y_{i,t}$ *need not* be conditionally independent. We are only defining a family of allowed joint distributions

[3] The reader familiar with generalized linear models will recognize the derivative of the function $p(\Delta) = \frac{g(-\Delta)}{g(\Delta)+g(-\Delta)}$ as the (inverse) link function of the associated canonical exponential family of distributions (McCullagh and Nelder, 1989).

$\mathbb{P}_t(y_{1,t}, \ldots, y_{K,t})$ through the properties of their marginals $\mathbb{P}_t(y_{i,t})$. A classical result in the theory of copulas (Sklar, 1959) makes one derive all allowed joint distributions starting from the corresponding one-dimensional marginals. It is also important to point out the arbitrary dependence of $\boldsymbol{x}_t$ on the past, since the typical scenarios we are modeling here (human interaction) are producing data sequences which are nonstationary in nature, implying that traditional statistical inference methods (e.g., empirical risk minimization) should be used cautiously.

Our algorithm will be based on the loss function L, which is such that the function g above is equal to the negative derivative of L. For instance, if L is the square loss $L(\Delta) = (1-\Delta)^2/2$, then $g(\Delta) = 1 - \Delta$, resulting in $\mathbb{P}_t(y_{i,t} = 1) = (1 + \boldsymbol{u}_i^\top \boldsymbol{x}_t)/2$, under the assumption $D = [-1, 1]$. If L is the logistic loss $L(\Delta) = \ln(1 + e^{-\Delta})$, then $g(\Delta) = \frac{1}{e^{\Delta}+1}$, and $\mathbb{P}_t(y_{i,t} = 1) = e^{\boldsymbol{u}_i^\top \boldsymbol{x}_t}/(e^{\boldsymbol{u}_i^\top \boldsymbol{x}_t} + 1)$, with domain $D = \mathcal{R}$. Observe that in both cases $\mathbb{P}_t(y_{i,t} = 1)$ is an increasing function of $\boldsymbol{u}_i^\top \boldsymbol{x}_t$. This will be true in general.

Set for brevity $\Delta_{i,t} = \boldsymbol{u}_i^\top \boldsymbol{x}_t$. Taking into account (1), this model allows us to write the (conditional) expected loss of the algorithm playing $\hat{Y}_t$ as

$$\mathbb{E}_t[\ell_{a,c}(Y_t, \hat{Y}_t)] = (1-a) \sum_{i \in \hat{Y}_t} \left(c(j_i, |\hat{Y}_t|) - \left(\tfrac{a}{1-a} + c(j_i, |\hat{Y}_t|) \right) p_{i,t} \right), \tag{3}$$

where we introduced the shorthands

$$p_{i,t} = p(\Delta_{i,t}), \qquad\qquad p(\Delta) = \frac{g(-\Delta)}{g(\Delta) + g(-\Delta)},$$

and the expectation $\mathbb{E}_t$ in (3) is w.r.t. the generation of labels Y_t, conditioned on both $\boldsymbol{x}_t$, and all previous $\boldsymbol{x}$ and Y.

A key aspect of this formalization is that the Bayes optimal ordered subset

$$Y_t^* = \mathrm{argmin}_{Y=(j_1,j_2,\ldots,j_{|Y|}) \subseteq [K]} \mathbb{E}_t[\ell_{a,c}(Y_t, Y)]$$

can be computed efficiently when knowing $\Delta_{1,t}, \ldots, \Delta_{K,t}$. This is handled by the next lemma. In words, this lemma says that, in order to minimize (3), it suffices to try out all possible sizes $s = 0, 1, \ldots, K$ for Y_t^* and, for each such value, determine the sequence $Y_{s,t}^*$ that minimizes (3) over all sequences of size s. In turn, $Y_{s,t}^*$ can be computed just by sorting classes $i \in [K]$ in decreasing order of $p_{i,t}$, sequence $Y_{s,t}^*$ being given by the first s classes in this sorted list.

Lemma 1 *With the notation introduced so far, let $p_{i_1,t} \geq p_{i_2,t} \geq \ldots p_{i_K,t}$ be the sequence of $p_{i,t}$ sorted in nonincreasing order. Then we have that*

$$Y_t^* = \mathrm{argmin}_{s=0,1,\ldots K} \mathbb{E}_t[\ell_{a,c}(Y_t, Y_{s,t}^*)],$$

where $Y_{s,t}^* = (i_1, i_2, \ldots, i_s)$, and $Y_{0,t}^* = \emptyset$.

Proof First observe that, for any given size s, the sequence $Y_{s,t}^*$ must contain the s top-ranked classes in the sorted order of $p_{i,t}$. This is because, for any candidate sequence $Y_s = \{j_1, j_2, \ldots, j_s\}$, we have $\mathbb{E}_t[\ell_{a,c}(Y_t^*, Y_s)] = (1-a) \sum_{i \in Y_s} \left(c(j_i, s) - \left(\frac{a}{1-a} + c(j_i, s) \right) p_{i,t} \right)$. If there exists $i \in Y_s$ which is not among the s-top ranked ones, then we could replace class i in position j_i within Y_s with class $k \notin Y_s$ such that $p_{k,t} > p_{i,t}$ obtaining a smaller loss.

Next, we show that the optimal ordering within $Y^*_{s,t}$ is precisely ruled by the nonincreasing order of $p_{i,t}$. By the sake of contradiction, assume there are i and k in $Y^*_{s,t}$ such that i precedes k in $Y^*_{s,t}$ but $p_{k,t} > p_{i,t}$. Specifically, let i be in position j_1 and k be in position j_2 with $j_1 < j_2$ and such that $c(j_1, s) > c(j_2, s)$. Then, disregarding the common $(1-a)$-factor, switching the two classes within $Y^*_{s,t}$ yields an expected loss difference of

$$\begin{aligned}
&c(j_1,s) - \left(\tfrac{a}{1-a} + c(j_1,s)\right) p_{i,t} + c(j_2,s) - \left(\tfrac{a}{1-a} + c(j_2,s)\right) p_{k,t}\\
&- \left(c(j_1,s) - \left(\tfrac{a}{1-a} + c(j_1,s)\right) p_{k,t}\right) - \left(c(j_2,s) - \left(\tfrac{a}{1-a} + c(j_2,s)\right) p_{i,t}\right)\\
&= (p_{k,t} - p_{i,t})\,(c(j_1,s) - c(j_2,s)) > 0,
\end{aligned}$$

since $p_{k,t} > p_{i,t}$ and $c(j_1, s) > c(j_2, s)$. Hence switching would get a smaller loss which leads as a consequence to $Y^*_{s,t} = (i_1, i_2, \ldots, i_s)$. ■

Notice the way costs $c(i, s)$ influence the Bayes optimal computation. We see from (3) that placing class i within $\hat{Y}_t$ in position j_i is beneficial (i.e., it leads to a reduction of loss) if and only if $p_{i,t} > c(j_i, |\hat{Y}_t|)/(\frac{a}{1-a} + c(j_i, |\hat{Y}_t|))$. Hence, the higher is the slot i_j in $\hat{Y}_t$ the larger should be $p_{i,t}$ in order for this inclusion to be convenient.[4]

It is Y^*_t above that we interpret as the true set of user preferences on $\boldsymbol{x}_t$. We would like to compete against Y^*_t in a cumulative regret sense, i.e., we would like to bound

$$R_T = \sum_{t=1}^{T} \mathbb{E}_t[\ell_{a,c}(Y_t, \hat{Y}_t)] - \mathbb{E}_t[\ell_{a,c}(Y_t, Y^*_t)]$$

with high probability.

We use a similar but largely more general analysis than Crammer and Gentile (2011)'s to devise an online second-order descent algorithm whose updating rule makes the comparison vector $U = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K) \in \mathcal{R}^{dK}$ defined through (2) be Bayes optimal w.r.t. a surrogate convex loss $L(\cdot)$ such that $g(\Delta) = -L'(\Delta)$. Observe that the expected loss function defined in (3) is, generally speaking, nonconvex in the margins $\Delta_{i,t}$ (consider, for instance the logistic case $g(\Delta) = \frac{1}{e^{\Delta}+1}$). Thus, we cannot directly minimize this expected loss.

3. Algorithm and Regret Bounds

In Figure 2 is our bandit algorithm for (ordered) multiple labels. In order to acquaint the reader with this algorithm, a simplified version of it is first presented (Figure 1) which applies to the linear model $p(\Delta) = \frac{1+\Delta}{2}$, $g(\Delta) = 1 - \Delta$, under the simplifying assumption $||\boldsymbol{u}_i|| \le 1$, for $i \in [K]$.

[4] Notice that this depends on the actual size of $\hat{Y}_t$, so we cannot decompose this problem into K independent problems. The decomposition does occur if the costs $c(i, s)$ are constants, independent of i and s, the criterion for inclusion becoming $p_{i,t} \ge \theta$, for some constant threshold θ.

```
Parameters:
    • Loss parameters $a \in [0,1]$, and cost values $c(i,s)$;
    • Confidence level $\delta \in [0,1]$.
Initialization: $A_{i,0} = I \in \mathcal{R}^{d \times d}$, $i = 1, \dots, K$,  $\boldsymbol{w}_{i,1} = 0 \in \mathcal{R}^d$,  $i = 1, \dots, K$;
For $t = 1, 2 \dots, T$ :
    1. Get instance $\boldsymbol{x}_t \in \mathcal{R}^d \,:\, ||\boldsymbol{x}_t|| = 1$;
    2. For $i \in [K]$, set $\widehat{\Delta}'_{i,t} = \boldsymbol{x}_t^\top \boldsymbol{w}'_{i,t}$, where
        $\boldsymbol{w}'_{i,t} = \begin{cases} \boldsymbol{w}_{i,t} & \text{if } \boldsymbol{w}_{i,t}^\top \boldsymbol{x}_t \in [-1,1], \\ \boldsymbol{w}_{i,t} - \left( \frac{\boldsymbol{w}_{i,t}^\top \boldsymbol{x}_t - 1}{\boldsymbol{x}_t^\top A_{i,t-1}^{-1} \boldsymbol{x}_t} \right) A_{i,t-1}^{-1} \boldsymbol{x}_t & \text{if } \boldsymbol{w}_{i,t}^\top \boldsymbol{x}_t > 1, \\ \boldsymbol{w}_{i,t} - \left( \frac{\boldsymbol{w}_{i,t}^\top \boldsymbol{x}_t + 1}{\boldsymbol{x}_t^\top A_{i,t-1}^{-1} \boldsymbol{x}_t} \right) A_{i,t-1}^{-1} \boldsymbol{x}_t & \text{if } \boldsymbol{w}_{i,t}^\top \boldsymbol{x}_t < -1; \end{cases}$
    3. Output
        $\hat{Y}_t = \operatorname*{argmin}_{Y = (j_1, j_2, \dots j_{|Y|}) \subseteq [K]} \left( \sum_{i \in Y} \left( c(j_i, |Y|) - \left( \frac{a}{1-a} + c(j_i, |Y|) \right) \widehat{p}_{i,t} \right) \right),$
       where
        $\widehat{p}_{i,t} = \frac{1 + [\widehat{\Delta}'_{i,t} + \epsilon_{i,t}]_{[-1,1]}}{2},$
        $\epsilon^2_{i,t} = \boldsymbol{x}_t^\top A_{i,t-1}^{-1} \boldsymbol{x}_t \left( 1 + 4\, d \,\ln \left( 1 + \frac{t-1}{d} \right) + 48 \,\ln \frac{K(t+4)}{\delta} \right)$ ;
    4. Get feedback $Y_t \cap \hat{Y}_t$;
    5. For $i \in [K]$, update:
        $A_{i,t} = A_{i,t-1} + |s_{i,t}| \boldsymbol{x}_t \boldsymbol{x}_t^\top, \qquad \boldsymbol{w}_{i,t+1} = \boldsymbol{w}'_{i,t} - A_{i,t}^{-1} \nabla_{i,t},$
       where
        $s_{i,t} = \begin{cases} 1 & \text{if } i \in Y_t \cap \hat{Y}_t, \\ -1 & \text{if } i \in \hat{Y}_t \setminus Y_t = \hat{Y}_t \setminus (Y_t \cap \hat{Y}_t), \\ 0 & \text{otherwise;} \end{cases}$
       and
        $\nabla_{i,t} = (s_{i,t}\, \widehat{\Delta}'_{i,t} - 1)\, s_{i,t}\, \boldsymbol{x}_t.$
```

Figure 1: The partial feedback algorithm in the (ordered) multiple label setting—the linear model case.

Both algorithms are based on replacing the unknown model vectors $\boldsymbol{u}_1, \dots, \boldsymbol{u}_K$ with prototype vectors $\boldsymbol{w}'_{1,t}, \dots, \boldsymbol{w}'_{K,t}$, being $\boldsymbol{w}'_{i,t}$ the time-t approximation to $\boldsymbol{u}_i$, satisfying sim-

```
Parameters:
  • Loss parameters $a \in [0,1]$, and cost values $c(i,s)$;
  • Interval $D = [-R,R]$, function $g \,:\, D \to \mathcal{R}$;
  • Confidence level $\delta \in [0,1]$, and norm upper bound $U > 0$.
Initialization: $A_{i,0} = I \in \mathcal{R}^{d \times d}$, $i = 1,\ldots,K$, $\boldsymbol{w}_{i,1} = 0 \in \mathcal{R}^d$, $i = 1,\ldots,K$;
For $t = 1, 2 \ldots, T$ :
  1. Get instance $\boldsymbol{x}_t \in \mathcal{R}^d \,:\, ||\boldsymbol{x}_t|| = 1$;
  2. For $i \in [K]$, set $\widehat{\Delta}'_{i,t} = \boldsymbol{x}_t^\top \boldsymbol{w}'_{i,t}$, where
       $$\boldsymbol{w}'_{i,t} = \begin{cases} \boldsymbol{w}_{i,t} & \text{if } \boldsymbol{w}_{i,t}^\top \boldsymbol{x}_t \in [-R,R], \\ \boldsymbol{w}_{i,t} - \left(\frac{\boldsymbol{w}_{i,t}^\top \boldsymbol{x}_t - R}{\boldsymbol{x}_t^\top A_{i,t-1}^{-1} \boldsymbol{x}_t}\right) A_{i,t-1}^{-1} \boldsymbol{x}_t & \text{if } \boldsymbol{w}_{i,t}^\top \boldsymbol{x}_t > R, \\ \boldsymbol{w}_{i,t} - \left(\frac{\boldsymbol{w}_{i,t}^\top \boldsymbol{x}_t + R}{\boldsymbol{x}_t^\top A_{i,t-1}^{-1} \boldsymbol{x}_t}\right) A_{i,t-1}^{-1} \boldsymbol{x}_t & \text{if } \boldsymbol{w}_{i,t}^\top \boldsymbol{x}_t < -R; \end{cases}$$
  3. Output
       $$\hat{Y}_t = \operatorname*{argmin}_{Y = (j_1, j_2, \ldots j_{|Y|}) \subseteq [K]} \left( \sum_{i \in Y} \left( c(j_i, |Y|) - \left( \tfrac{a}{1-a} + c(j_i, |Y|) \right) \widehat{p}_{i,t} \right) \right),$$
     where
       $$\widehat{p}_{i,t} = p\left([\widehat{\Delta}'_{i,t} + \epsilon_{i,t}]_D\right) = \frac{g\left(-[\widehat{\Delta}'_{i,t} + \epsilon_{i,t}]_D\right)}{g\left([\widehat{\Delta}'_{i,t} + \epsilon_{i,t}]_D\right) + g\left(-[\widehat{\Delta}'_{i,t} + \epsilon_{i,t}]_D\right)},$$
       $$\epsilon^2_{i,t} = \boldsymbol{x}_t^\top A_{i,t-1}^{-1} \boldsymbol{x}_t \left( U^2 + \frac{d\, c'_L}{(c''_L)^2} \ln\left(1 + \frac{t-1}{d}\right) + \frac{12}{c''_L} \left( \frac{c'_L}{c''_L} + 3L(-R) \right) \ln \frac{K(t+4)}{\delta} \right) \; ;$$
  4. Get feedback $Y_t \cap \hat{Y}_t$;
  5. For $i \in [K]$, update:
       $$A_{i,t} = A_{i,t-1} + |s_{i,t}| \boldsymbol{x}_t \boldsymbol{x}_t^\top, \qquad \boldsymbol{w}_{i,t+1} = \boldsymbol{w}'_{i,t} - \frac{1}{c''_L} A_{i,t}^{-1} \nabla_{i,t},$$
     where
       $$s_{i,t} = \begin{cases} 1 & \text{if } i \in Y_t \cap \hat{Y}_t, \\ -1 & \text{if } i \in \hat{Y}_t \setminus Y_t = \hat{Y}_t \setminus (Y_t \cap \hat{Y}_t), \\ 0 & \text{otherwise;} \end{cases}$$
     and
       $$\nabla_{i,t} = \nabla_{\boldsymbol{w}} L(s_{i,t}\, \boldsymbol{w}^\top \boldsymbol{x}_t)|_{\boldsymbol{w} = \boldsymbol{w}'_{i,t}} = -g(s_{i,t}\, \widehat{\Delta}'_{i,t})\, s_{i,t}\, \boldsymbol{x}_t.$$
```

Figure 2: The partial feedback algorithm in the (ordered) multiple label setting—the *generalized* linear model case.

ilar constraints we set for the $\boldsymbol{u}_i$ vectors. For the sake of brevity, we let $\widehat{\Delta}'_{i,t} = \boldsymbol{x}_t^\top \boldsymbol{w}'_{i,t}$, and $\Delta_{i,t} = \boldsymbol{u}_i^\top \boldsymbol{x}_t$, $i \in [K]$.

The algorithms use $\widehat{\Delta}'_{i,t}$ as proxies for the underlying $\Delta_{i,t}$ according to the (upper confidence) approximation scheme $\Delta_{i,t} \approx [\widehat{\Delta}'_{i,t} + \epsilon_{i,t}]_D$, where $\epsilon_{i,t} \geq 0$ is a suitable upper-confidence level for class i at time t, and $[\cdot]_D$ denotes the clipping-to-D operation: if $D = [-R, R]$, then

$$[x]_D = \begin{cases} R & \text{if } x > R \\ x & \text{if } -R \leq x \leq R \\ -R & \text{if } x < -R. \end{cases}$$

The algorithms' prediction at time t has the same form as the computation of the Bayes optimal sequence Y_t^*, where we replace the true (and unknown) $p_{i,t} = p(\Delta_{i,t})$ with the corresponding upper confidence proxy

$$\widehat{p}_{i,t} = p([\widehat{\Delta}'_{i,t} + \epsilon_{i,t}]_D),$$

being

$$\hat{Y}_t = \underset{Y=(j_1,j_2,\ldots j_{|Y|})\subseteq[K]}{\operatorname{argmin}} \left(\sum_{i \in Y} \left(c(j_i, |Y|) - \left(\tfrac{a}{1-a} + c(j_i, |Y|) \right) \widehat{p}_{i,t} \right) \right).$$

Computing $\hat{Y}_t$ above can be done by mimicking the computation of the Bayes optimal ordered subset Y_t^* (just replace $p_{i,t}$ by $\widehat{p}_{i,t}$). From a computational viewpoint, this essentially amounts to sorting classes $i \in [K]$ in decreasing value of $\widehat{p}_{i,t}$, i.e., order of $K \log K$ running time per prediction. Thus the algorithms are producing a ranked list of relevant classes based on upper-confidence-corrected scores $\widehat{p}_{i,t}$. Class i is deemed relevant and ranked high among the relevant ones when either $\widehat{\Delta}'_{i,t}$ is a good approximation to $\Delta_{i,t}$ and $p_{i,t}$ is large, or when the algorithms are not very confident on their own approximation about i (that is, the upper confidence level $\epsilon_{i,t}$ is large).

Specifically, the algorithm in Figure 1 receives in input the loss parameters a and $c(i, s)$, and the desired confidence level δ, and maintains both K positive definite matrices $A_{i,t}$ of dimension d (initially set to the $d \times d$ identity matrix), and K weight vectors $\boldsymbol{w}_{i,t} \in \mathcal{R}^d$ (initially set to the zero vector). At each time step t, upon receiving the d-dimensional instance vector $\boldsymbol{x}_t$ the algorithm uses the weight vectors $\boldsymbol{w}_{i,t}$ to compute the prediction vectors $\boldsymbol{w}'_{i,t}$. These vectors can easily be seen as the result of projecting $\boldsymbol{w}_{i,t}$ onto interval $[-1, 1]$ w.r.t. the distance function $d_{i,t-1}$, i.e.,

$$\boldsymbol{w}'_{i,t} = \underset{\boldsymbol{w}\in\mathcal{R}^d \,:\, \boldsymbol{w}^\top \boldsymbol{x}_t \in [-1,1]}{\operatorname{argmin}} d_{i,t-1}(\boldsymbol{w}, \boldsymbol{w}_{i,t}), \qquad i \in [K],$$

where

$$d_{i,t-1}(\boldsymbol{u}, \boldsymbol{w}) = (\boldsymbol{u} - \boldsymbol{w})^\top A_{i,t-1} (\boldsymbol{u} - \boldsymbol{w}).$$

Vectors $\boldsymbol{w}'_{i,t}$ are then used to produce prediction values $\widehat{\Delta}'_{i,t}$ involved in the upper-confidence calculation of the predicted ordered subset $\hat{Y}_t \subseteq [K]$. Next, the feedback $Y_t \cap \hat{Y}_t$ is observed, and the algorithm in Figure 1 promotes all classes $i \in Y_t \cap \hat{Y}_t$ (sign $s_{i,t} = 1$), demotes all classes $i \in \hat{Y}_t \setminus Y_t$ (sign $s_{i,t} = -1$), and leaves all remaining classes $i \notin \hat{Y}_t$ unchanged (sign $s_{i,t} = 0$). Promotion of class i on $\boldsymbol{x}_t$ implies that if the new vector $\boldsymbol{x}_{t+1}$ is close to $\boldsymbol{x}_t$ then i will be ranked higher on $\boldsymbol{x}_{t+1}$. The update $\boldsymbol{w}'_{i,t} \rightarrow \boldsymbol{w}_{i,t+1}$ is based on the gradients $\nabla_{i,t}$ of the square loss function $L(\Delta) = (1-\Delta)^2/2$. On the other hand, the update $A_{i,t-1} \rightarrow A_{i,t}$ uses the rank-one matrix[5] $\boldsymbol{x}_t\boldsymbol{x}_t^\top$. The matrix $A_{i,t-1}$ is used to calculate the upper confidence level on each prediction. Matrix $A_{i,t-1}$ is the empirical covariance matrix of the samples on which we received some feedback, either positive ($s_{i,t} = 1$) or negative ($s_{i,t} = -1$), and is used in the expression for the confidence $\epsilon^2_{i,t}$ involving the quadratic form $\boldsymbol{x}_t^\top A^{-1}_{i,t-1}\boldsymbol{x}_t$. Notice that $\epsilon^2_{i,t}$ will be small when the current sample $\boldsymbol{x}_t$ is in the span of the previous samples on which we received feedback, and will be large otherwise. In both the update of $\boldsymbol{w}'_{i,t}$ and the one involving $A_{i,t-1}$, the reader should observe the role played by the signs $s_{i,t}$.

The algorithm contained in Figure 2 is just a more general version of the one in Figure 1, where we also receive in input the specifics of the generalized linear model through the model function $g(\cdot)$ and the associated margin domain $D = [-R, R]$, and the norm upper bound U, such that $\|\boldsymbol{u}_i\| \le U$ for all $i \in [K]$. The update $\boldsymbol{w}'_{i,t} \rightarrow \boldsymbol{w}_{i,t+1}$ in Figure 2 is based on the gradients $\nabla_{i,t}$ of a loss function $L(\cdot)$ satisfying $L'(\Delta) = -g(\Delta)$. On the other hand, the update $A_{i,t-1} \rightarrow A_{i,t}$ uses again the rank-one matrix $\boldsymbol{x}_t\boldsymbol{x}_t^\top$. The constants c'_L and c''_L occurring in the expression for $\epsilon^2_{i,t}$ in Figure 2 are related to smoothness properties of $L(\cdot)$. In particular, $\epsilon^2_{i,t}$ in Figure 1 is obtained from $\epsilon^2_{i,t}$ in Figure 2 by setting $R = 1$, $L(-R) = L(-1) = 0$, along with $c'_L = 4$ and $c''_L = 1$, as explained in the next theorem.[6]

Theorem 2 *Let $L : D = [-R, R] \subseteq \mathcal{R} \rightarrow \mathcal{R}^+$ be a $C^2(D)$ convex and nonincreasing function of its argument, $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K) \in \mathcal{R}^{dK}$ be defined in (2) with $g(\Delta) = -L'(\Delta)$ for all $\Delta \in D$, and such that $\|\boldsymbol{u}_i\| \le U$ for all $i \in [K]$. Assume there are positive constants c_L, c'_L and c''_L such that*

i. $\frac{L'(\Delta)\, L''(-\Delta) + L''(\Delta)\, L'(-\Delta)}{(L'(\Delta)+L'(-\Delta))^2} \ge -c_L$,

ii. $(L'(\Delta))^2 \le c'_L$,

iii. $L''(\Delta) \ge c''_L$

simultaneously hold for all $\Delta \in D$. Then the cumulative regret R_T of the algorithm in Figure 2 satisfies, with probability at least $1-\delta$,

$$R_T = O\left((1-a)\, c_L\, K \sqrt{T\, C\, d \ln\left(1 + \frac{T}{d}\right)}\right),$$

where

$$C = O\left(U^2 + \frac{d\, c'_L}{(c''_L)^2} \ln\left(1 + \frac{T}{d}\right) + \left(\frac{c'_L}{(c''_L)^2} + \frac{L(-R)}{c''_L}\right) \ln \frac{KT}{\delta}\right).$$

[5] The rank-one update is based on $\boldsymbol{x}_t\boldsymbol{x}_t^\top$ rather than $\nabla_{i,t}\nabla_{i,t}^\top$, as in , e.g., the paper by Hazan et al. (2007). This is due to technical reasons that will be made clear in Section 6. This feature tells this algorithm slightly apart from the Online Newton step algorithm (Hazan et al., 2007), which is the starting point of our analysis. The very same comment applies to the algorithm in Figure 2.

[6] The proof is given in Section 6.

It is easy to see that when $L(\cdot)$ is the square loss $L(\Delta) = (1-\Delta)^2/2$ and $D = [-1, 1]$, we have $c_L = 1/2$, $c'_L = 4$ and $c''_L = 1$; when $L(\cdot)$ is the logistic loss $L(\Delta) = \ln(1 + e^{-\Delta})$ and $D = [-R, R]$, we have $c_L = 1/4$, $c'_L \leq 1$ and $c''_L = \frac{1}{2(1+\cosh(R))}$, where $\cosh(x) = \frac{e^x + e^{-x}}{2}$.

The following remarks are in order at this point.

Remark 3 *A drawback of Theorem 2 is that, in order to properly set the upper confidence levels $\epsilon_{i,t}$, we assume prior knowledge of the norm upper bound U. Because this information is often unavailable, we present here a simple modification to the algorithm that copes with this limitation, similar to the one proposed in Orabona and Cesa-Bianchi (2011). We change the definition of $\epsilon^2_{i,t}$ in Figure 2 to*

$$\epsilon^2_{i,t} = \max\left\{\boldsymbol{x}^\top A^{-1}_{i,t-1}\boldsymbol{x}\,\left(\frac{2\,d\,c'_L}{(c''_L)^2}\ln\left(1+\frac{t-1}{d}\right)+\frac{12}{c''_L}\left(\frac{c'_L}{c''_L}+3L(-R)\right)\ln\frac{K(t+4)}{\delta}\right), 4\,R^2\right\},$$

that is, we substitute U^2 by $\frac{d\,c'_L}{(c''_L)^2}\ln\left(1+\frac{t-1}{d}\right)$, and cap the maximal value of $\epsilon^2_{i,t}$ to $4\,R^2$. This immediately leads to the following result.[7]

Theorem 4 *With the same assumptions and notation as in Theorem 2, if we replace $\epsilon^2_{i,t}$ as explained above we have that, with probability at least $1-\delta$, R_T satisfies*

$$R_T = O\left((1-a)\,c_L\,K\sqrt{T\,C\,d\ln\left(1+\frac{T}{d}\right)}+(1-a)\,c_L\,K\,R\,d\left(\exp\left(\frac{(c''_L)^2\,U^2}{c'_L\,d}\right)-1\right)\right).$$

Remark 5 *From a computational standpoint, the most demanding operation in Figure 2 is computing the upper confidence levels $\epsilon_{i,t}$ involving the inverse matrices $A^{-1}_{i,t-1}$, $i \in [K]$. Note that the matrices can be safely inverted because they are full rank, being initialized with identity matrices. The matrix inversion can be done incrementally in $\mathcal{O}(K\,d^2)$ time per round. This can be hardly practical if both d and K are large. In practice (as explained, e.g., by Crammer and Gentile, 2011), one can use an approximated version of the algorithm which maintains* diagonal *matrices $A_{i,t}$ instead of full ones. All the steps remain the same except Step 5 of Algorithm 2 where one defines the rth diagonal element of matrix $A_{i,t}$ as $(A_{i,t})_{r,r} = (A_{i,t-1})_{r,r} + x^2_{r,t}$, being $\boldsymbol{x}_t = (x_{1,t}, x_{2,t}, \ldots, x_{r,t}, \ldots, x_{K,t})^\top$. The resulting running time per round (including prediction and update) becomes $\mathcal{O}(dK + K\log K)$. In fact, when a limitation on the size of $\hat{Y}_t$ is given, the running time may be further reduced, see Remark 8.*

4. On Ranking with Partial Feedback

As Lemma 1 points out, when the cost values $c(i,s)$ in the loss function $\ell_{a,c}$ are *strictly* decreasing i.e., $c(1,s) > c(2,s) > \ldots > c(s,s)$, for all $s \in [K]$, then the Bayes optimal ordered sequence Y^*_t on $\boldsymbol{x}_t$ is unique can be obtained by sorting classes in decreasing values of $p_{i,t}$, and then decide on a cutoff point[8] induced by the loss parameters, so as to tell relevant classes apart from irrelevant ones. In turn, because $p(\Delta) = \frac{g(-\Delta)}{g(\Delta)+g(-\Delta)}$ is increasing in Δ,

[7] The proof is deferred to Section 6.

[8] This is called the *zero point* by Furnkranz et al. (2008).

this ordering corresponds to sorting classes in decreasing values of $\Delta_{i,t}$. Now, if parameter a in $\ell_{a,c}$ is very close[9] to 1, then $|Y_t^*| = K$, and the algorithm itself will produce ordered subsets $\hat{Y}_t$ such that $|\hat{Y}_t| = K$. Moreover, it does so by receiving *full* feedback on the relevant classes at time t (since $Y_t \cap \hat{Y}_t = Y_t$). As is customary (e.g., Dembczynski et al. 2012), one can view any multilabel assignment $Y = (y_1, \ldots, y_K) \in \{0,1\}^K$ as a ranking among the K classes in the most natural way: i precedes j if and only if $y_i > y_j$. The (unnormalized) ranking loss function $\ell_{rank}(Y, f)$ between the multilabel Y and a ranking function $f : \mathcal{R}^d \to \mathcal{R}^K$, representing degrees of class relevance sorted in a decreasing order $f_{j_1}(\boldsymbol{x}_t) \geq f_{j_2}(\boldsymbol{x}_t) \geq \ldots \geq f_{j_K}(\boldsymbol{x}_t) \geq 0$, counts the number of class pairs that disagree in the two rankings:

$$\ell_{rank}(Y, f) = \sum_{i,j \in [K]\,:\, y_i > y_j} \left(\{f_i(\boldsymbol{x}_t) < f_j(\boldsymbol{x}_t)\} + \tfrac{1}{2}\{f_i(\boldsymbol{x}_t) = f_j(\boldsymbol{x}_t)\} \right),$$

where $\{\ldots\}$ is the indicator function of the predicate at argument. As pointed out by Dembczynski et al. (2012), the ranking function $f(\boldsymbol{x}_t) = (p_{1,t}, \ldots, p_{K,t})$ is also Bayes optimal w.r.t. $\ell_{rank}(Y, f)$, *no matter if* the class labels y_i are conditionally independent or not. Hence we can use the algorithm in Figure 2 with a close to 1 for tackling ranking problems derived from multilabel ones, when the measure of choice is ℓ_{rank} and the feedback is full.

We now consider a partial information version of the above ranking problem. Suppose that at each time t, the environment discloses both $\boldsymbol{x}_t$ and a maximal *size* S_t for the ordered subset $\hat{Y}_t = (j_1, j_2, \ldots, j_{|\hat{Y}_t|})$ (both $\boldsymbol{x}_t$ and S_t can be chosen adaptively by an adversary). Here S_t might be the number of available slots in a webpage or the maximal number of URLs returned by a search engine in response to query $\boldsymbol{x}_t$. Then it is plausible to compete in a regret sense against the best time-t offline ranking of the form

$$f^*(\boldsymbol{x}_t) = f^*(\boldsymbol{x}_t; S_t) = (f_1^*(\boldsymbol{x}_t), f_2^*(\boldsymbol{x}_t), \ldots, f_K^*(\boldsymbol{x}_t)),$$

where the number of strictly positive $f_i^*(\boldsymbol{x}_t)$ values is at most S_t. Further, the ranking loss could be reasonably restricted to count the number of class pairs disagreeing within $\hat{Y}_t$ plus a quantity related to the number of false negative mistakes. If $\hat{Y}_t$ is the sequence of length S_t associated with a ranking function f, we consider the loss function $\ell_{p-rank,t}$ ("partial information ℓ_{rank} at time t")

$$\ell_{p-rank,t}(Y, f) = \sum_{i,j \in \hat{Y}_t\,:\, y_i > y_j} \left(\{f_i(\boldsymbol{x}_t) < f_j(\boldsymbol{x}_t)\} + \tfrac{1}{2}\{f_i(\boldsymbol{x}_t) = f_j(\boldsymbol{x}_t)\} \right) + S_t\,|Y_t \setminus \hat{Y}_t|.$$

In this loss function, the factor S_t multiplying $|Y_t \setminus \hat{Y}_t|$ serves as balancing the contribution of the double sum $\sum_{i,j \in \hat{Y}_t\,:\, y_i > y_j}$ (potentially involving a quadratic number of terms) with the contribution of false negative mistakes $|Y_t \setminus \hat{Y}_t|$. As for loss $\ell_{a,c}$, we can rewrite $\ell_{p-rank,t}(Y, f)$ as

$$\ell_{p-rank,t}(Y, f) = \sum_{i,j \in \hat{Y}_t\,:\, y_i > y_j} \left(\{f_i(\boldsymbol{x}_t) < f_j(\boldsymbol{x}_t)\} + \tfrac{1}{2}\{f_i(\boldsymbol{x}_t) = f_j(\boldsymbol{x}_t)\} \right) - S_t\,|Y_t \cap \hat{Y}_t| + S_t\,|Y_t|,$$

[9] If $a = 1$, the algorithm only cares about false negative mistakes, the best strategy being always predicting $\hat{Y}_t = [K]$. Unsurprisingly, this yields zero regret in both Theorems 2 and 4.

where the first two terms can be calculated by the algorithm, and the last one does not depend on $\hat{Y}_t$. For convenience, we will interchangeably use the notations $\ell_{p-rank,t}(Y, f)$ and $\ell_{p-rank,t}(Y, \hat{Y}_t)$, whenever it is clear from the surrounding context that $\hat{Y}_t$ is the sequence corresponding to f.

The next lemma[10] is the ranking counterpart to Lemma 1. It shows that the Bayes optimal ranking for $\ell_{p-rank,t}$ is given by

$$f^*(\boldsymbol{x}_t; S_t) = (p'_{1,t}, p'_{2,t}, \ldots, p'_{K,t}),$$

where $p'_{j,t} = p_{j,t}$ if $p_{j,t}$ is among the S_t largest values in the sequence $(p_{1,t}, \ldots, p_{K,t})$, and 0 otherwise. That is, $f^*(\boldsymbol{x}_t; S_t)$ is the function that ranks classes according to decreasing values of $p_{i,t}$ and cuts off exactly at position S_t. This is in contrast to what happens for loss $\ell_{a,c}$, where, depending on the cost parameters $c(i, s)$, the cut off point can even be smaller than the total number of available slots—see Lemma 1 and surrounding comments. In order for this result to go through, we need to restrict model (2) to the case of conditionally independent classes, i.e., to the case when

$$\mathbb{P}_t(y_{1,t}, \ldots, y_{K,t}) = \prod_{i \in [K]} p_{i,t}\,. \tag{4}$$

This is a significant departure from the full information setting, where the Bayes optimal ranking only depends on the marginal distribution values $p_{i,t}$ (Dembczynski et al., 2012). Due to the interaction between the two terms in the definition of $\ell_{p-rank,t}$, the Bayes optimal ranking for $\ell_{p-rank,t}$ turns out to depend on both marginal and pairwise correlation values of the joint class distribution. Assumption (4) may be avoided by maintaining $O(K^2)$ upper confidence values $\epsilon_{i,j}$, one for each pair $(i, j), i < j$, leading to an extra computational burden which can become prohibitive even in the presence of a moderate number of classes K.

Lemma 6 *With the notation introduced so far, let the joint distribution $\mathbb{P}_t(y_{1,t}, \ldots, y_{K,t})$ factorize as in (4). Then $f^*(\boldsymbol{x}_t; S_t)$ introduced above satisfies*

$$f^*(\boldsymbol{x}_t; S_t) = \operatorname*{argmin}_{Y=(i_1, i_2, \ldots i_h)\,, h \le S_t} \mathbb{E}_t[\ell_{p-rank,t}(Y_t, Y)].$$

If we add to the argmin of our algorithm (Step 3 in Figure 2) the further constraint $|Y| \le S_t$ (notice that the resulting computation is still about sorting classes according to decreasing values of $\widehat{p}_{i,t}$), we are defining a partial information ranking algorithm that ranks classes according to decreasing values of $\widehat{p}_{i,t}$ up to position S_t (i.e., $|\hat{Y}_t| = S_t$). Let $\widehat{f}(\boldsymbol{x}_t, S_t)$ be the resulting ranking. We can then define the cumulative regret R_T w.r.t. $\ell_{p-rank,t}$ as

$$R_T = \sum_{t=1}^{T} \mathbb{E}_t[\ell_{p-rank,t}(Y_t, \widehat{f}(\boldsymbol{x}_t, S_t))] - \mathbb{E}_t[\ell_{p-rank,t}(Y_t, f^*(\boldsymbol{x}_t, S_t)], \tag{5}$$

that is, the extent to which the conditional $\ell_{p-rank,t}$-risk of $\widehat{f}(\boldsymbol{x}_t, S_t)$ exceeds the one of the Bayes optimal ranking $f^*(\boldsymbol{x}_t; S_t)$, accumulated over time.

We have the following ranking counterpart to Theorem 2.

[10] We postpone its lengthy proof to Section 6.

Theorem 7 *With the same assumptions and notation as in Theorem 2, combined with the independence assumption (4), let the cumulative regret R_T w.r.t. $\ell_{p-rank,t}$ be defined as in (5). Then, with probability at least $1-\delta$, we have that the algorithm in Figure 2 working with $a \to 1$ and strictly decreasing cost values $c(i,s)$ (i.e., the algorithm computing in round t the ranking function $\widehat{f}(\boldsymbol{x}_t, S_t)$) achieves*

$$R_T = O\left(c_L \sqrt{S\,K\,T\,C\,d\,\ln\left(1+\frac{T}{d}\right)}\right),$$

where $S = \max_{t=1,\dots,T} S_t$.

The proof (see Section 6) is very similar to the one of Theorem 2. This suggests that, to some extent, we are decoupling the label generating model from the loss function ℓ under consideration.

Remark 8 *As is typical in many multilabel classification settings, the number of classes K can be very large and/or have an inner structure (e.g., a hierarchical or DAG-like structure). It is often the case that in such a large label space, many classes are relatively rare. This has lead researchers to consider methods that are specifically tailored to leverage the label sparsity of the chosen classifier (e.g., Hsu et al. 2009 and references therein) and/or the specific structure of the set of labels (e.g., Cesa-Bianchi et al. 2006a; Bi and Kwok 2011, and references therein). Though our algorithm is not designed to exploit the label structure, we would like to stress that the restriction $|\hat{Y}_t| \le S_t \le S$ in Theorem 7 allows us to replace the linear dependence on the total number of classes K (which is often much larger than S) by $\sqrt{SK}$. It is very easy to see that this restriction would bring similar benefits to Theorem 2.*

In fact, the above restriction is not only beneficial from a "statistical" point of view, but also from a computational one. As is by now standard, algorithms like the one in Figure 2 can easily be cast in dual variables (i.e., in a RKHS). This comes with at least two consequences:

1. *We can depart from the (generalized) linear modeling assumption (2), and allow for more general nonlinear dependencies of $p_{i,t}$ on the input vectors $\boldsymbol{x}_t$, possibly resorting to the universal approximation properties of Gaussian RKHS (e.g., Steinwart, 2002).*

2. *We can maintain a dual variable representation for margins $\widehat{\Delta}'_{i,t}$ and quadratic forms $\boldsymbol{x}_t^\top A_{i,t-1}^{-1}\boldsymbol{x}_t$, so that computing each one of them takes $O(N_{i,t-1}^2)$ inner products, where $N_{i,t}$ is the number of times class i has been updated up to time t, each inner product being $O(d)$. Now, each of the (at most $S_t \le S$) updates is $O(N_{i,t-1}^2)$. Hence, the overall running time in round t is coarsely overapproximated by $O(d\sum_{i\in[K]} N_{i,T}^2 + K\log K)$. From $\sum_{i\in[K]} N_{i,T} \le ST$, we see that when S is small compared to K, then $N_{i,t-1}$ tends to be small as well. For instance, if $S \le \sqrt{K}$ this leads to a running time per round of the form SdT^2, which can be smaller than the bound Kd^2 mentioned in Remark 5.*

Finally, observe that one can also combine Theorem 7 with the argument contained in Remark 1.

Task	Train+Test	d	K	Avg	Avg + std	95%	99%
Mediamill	30,993+12,914	120	101	5	7	8	10
Sony	16,452+16,519	98	632	38	44	48	52
Yeast	1,500+917	103	14	5	6	7	8

Table 1: Main statistics related to the three data sets used in our experiments. The last four columns give information on the distribution of the number of labels per instance. "Avg" denotes the (rounded) average number of labels over the training examples, and "Avg+std" gives the average augmented by one unit of standard deviation. So, for instance, in the Mediamill data set, the average number of labels per instance in the training set is 5, with a standard deviation of 2. The columns tagged "95%" and "99%" give an idea of the quantiles of this distribution. E.g., on Mediamill, 95% of the training examples have at most 8 classes (out of 101), on the Sony data set, 99% of the training examples have at most 52 classes (out of 632).

5. Experiments

The experiments we report here are meant to validate the exploration-exploitation tradeoff implemented by our algorithm along different axes: data set size, label set size, loss function, label generation model, training mode of operation, and restrictions on the total number of classes predicted. Moreover, we explicitly tested the effectiveness of ranking classes based on upper confidence-corrected probability estimates.

5.1 Data Sets

We used three diverse multilabel data sets, intended to represent different real-world conditions. The first one, called Mediamill, was introduced in a video annotation challenge (Snoek et al., 2006). It comprises $30,993$ training samples and $12,914$ test ones. The number of features d is 120, and the number of classes K is 101. The second data set is the music annotated Sony CSL Paris data set (Pachet and Roy, 2009), made up of $16,452$ training samples and 16,519 test samples, each sample being described by $d = 98$ features. The number of classes K is 632, which is significantly larger than Mediamill's. The third one is the smaller Yeast data set (Elisseeff and Weston, 2002), made up of $1,500$ training samples, 917 test samples, with $d = 103$ and $K = 14$. In all cases, the feature vectors have been normalized to unit Euclidean norm. Table 1 summarizes relevant statistics about these data sets. This table also gives an idea of the distribution of the number of classes per instance.

5.2 Parameter Setting and Loss Measures

For the practical implementation of the algorithm in Figure 2, we simplified the formula for $\epsilon^2_{i,t}$. This is justified by the fact that the actual constants in the definition of $\epsilon^2_{i,t}$ are artifacts of our high-probability upper bounds. Hence, we used

$$\epsilon^2_{i,t} = \alpha \, \boldsymbol{x}_t^\top A^{-1}_{i,t-1} \boldsymbol{x}_t \, \log(t+1),$$

where α is a parameter that we found by cross-validation on each data set across the range $\alpha = 2^{-8}, 2^{-7}, \ldots, 2^{7}, 2^{8}$, for each choice of the label-generation model, loss setting, and value of S—see below. We have considered two different loss functions L, the square loss and the logistic loss (denoted by “Log Loss” in our plots). Correspondingly, the two label-generation models we tested are the linear model $\mathbb{P}_t(y_{i,t} = 1) = (1 + \boldsymbol{u}_i^\top \boldsymbol{x}_t)/2$ with domain $D = [-1, 1]$, and the logistic model $\mathbb{P}_t(y_{i,t} = 1) = e^{\boldsymbol{u}_i^\top \boldsymbol{x}_t}/(e^{\boldsymbol{u}_i^\top \boldsymbol{x}_t} + 1)$. In the logistic case, it makes sense in practice not to place any restrictions on the margin domain D, so that we set $R = \infty$. Again, because our upper bounding analysis would yield as a consequence $c''_L = 0$, we instead set c''_L to a small positive constant, specifically $c''_L = 0.1$, with no special attention to its fine-tuning. The setting of the cost function $c(i, s)$ depends on the task at hand, and we decided to evaluate two possible settings. The first one, denoted by “decreasing” is $c(i, s) = \frac{s-i+1}{s}, i = 1, \ldots, s$, the second one, denoted by “constant”, is $c(i, s) = 1$, for all i and s. In all experiments with $\ell_{a,c}$, the a parameter was set to 0.5 (so that $\ell_{a,c}$ with constant c reduces to half the Hamming loss). In the decreasing c scenario, we evaluated the performance of the algorithm on the loss $\ell_{a,c}$ that the algorithm is minimizing, but also its ability to produce meaningful (partial) rankings through $\ell_{p-rank,t}$. In the constant c scenario, we only evaluated the Hamming loss, its natural loss function.

As is typical of multilabel problems, the label *density* of our data sets, i.e., the average fraction of labels associated with the examples, is quite small. Hence, it is clearly beneficial to our learning algorithm to bias its inference process so as to produce short ranked lists $\hat{Y}_t$. We did so by imposing, for all t, an upper bound $S_t = S$ on $|\hat{Y}_t|$. For each of the three data sets, we tried out the four different values of S reported in the last four columns of Table 1: the average number of labels; the average plus one standard deviation, the number of labels that covers 95% of the examples, and the number of labels that covers 99% of the examples, all figures only referring to the corresponding training sets.

5.3 Baselines

As a baseline, we considered a full information version of Algorithm 2, denoted by “Full Info”, that receives after each prediction the full array of true labels Y_t for each sample. Comparing to full information algorithms stresses the effectiveness of the exploration/exploitation rule above and beyond the details of underlying generalized linear predictor. We also compared against the random predictor (denoted by “Random”) that simply outputs at time t a ranked list $\hat{Y}_t$ made up of S labels chosen (and ranked) at random. Finally, an interesting ranking baseline which targets the ranking ability of our algorithm is one that lets our partial feedback algorithm select which classes to include in $\hat{Y}_t$, and then shuffles them at random within $\hat{Y}_t$ to produce the ranked list. This baseline we only used with the ranking loss $\ell_{p-rank,t}$, and is denoted by “Shuffled” in our plots.

5.4 Results

Our results are summarized in Figures 3, 4, and 5. The top row of each figure shows the results in the online setting, while the bottom row is for the batch setting. Each column corresponds to a different data set. In both the online and batch cases, the algorithms were fed with the training set in a sequential fashion, sweeping over it only *once*.

The plots report online or batch loss measures as a function of S,[11] averaged over 5 random permutation of the training sequence. Specifically, whereas the online measure of performance ("Final Average ... Loss") is the cumulative loss accumulated during training, divided by the number of samples in the training set, the batch measure ("Test ...") is simply the average loss over the test set achieved by the last solution produced by training. For the partial-feedback algorithms ("Square Loss", "Log Loss" and, in the ranking case, also "Square Loss Shuffled" and "Log Loss Shuffled"), only the best α-cross-validated performances are shown. Moreover, in the ranking experiments, because of the explicit dependence of $\ell_{p-rank,t}$ on S, we instead considered the scaled version of the loss $\ell_{p-rank,t}/S$. Notice that the theoretical results contained in Section 4 still apply to this scaled loss function.

The first thing to observe from the evidence we collected is that performance in the batch setting closely follows the one in the online setting, across all the data sets, conditions and losses. In a sense, this is to be expected, since the order of samples in the training set is randomly shuffled.

The optimal value of S that allows us to best balance exploration and the exploitation of the algorithm seems to be depending on the particular data set and task at hand. So, for instance, on Mediamill with Hamming loss, this value is $S = 8$, corresponding to the 95% coverage of the training set, while on Yeast it is the average value $S = 5$, covering around 50% of the training examples. When the loss is $\ell_{a,c}$, the best value of S clearly depends on the costs c. In the ranking case, performance increases as S gets larger, but this is very likely to be due to the scaling factor $1/S$ in the loss we plotted. Notice that, from our theoretical analysis in Section 3, the algorithm (e.g., in the special case on Hamming loss) should in principle be able to determine the best size of $\hat{Y}_t$ at each round, so that setting $S_t = K$ for all t is still a fair choice. Yet, this conservative setting makes the algorithm face an unnecessarily large action space (of size $K!$), and correspondingly a harder inference problem, rather than the substantially smaller space (of size $K(K-1)(K-2)\dots(K-S+1)$) obtained by setting $S_t = S$. This is evinced by the fact that all plots (regarding both partial and full information algorithms) in Figure 4 tend to be increasing with S. For the very sake of this inference, the fact that all algorithms see the examples only once seems to be a severe limitation.[12]

The performance of our partial information algorithms are always pretty close to those of the corresponding full information algorithms. This empirically validates the exploration/exploitation scheme we used. Also, in all cases, all algorithms clearly outperform the random predictor. In most of the experiments, the linear model ("Square Loss") seems to deliver slightly better results in the bandit setting than the logistic model ("Log Loss"), while the performance of the two models is very similar in the full information case. Exceptions are the constant and the decreasing cost settings in the batch case on the Yeast data set (Figure 3, bottom right, and Figure 4, bottom right), where the bandit algorithm has an even better performance than the full information one. This is perhaps due to the

[11] The plots are actually piecewise linear interpolations with knots corresponding to the 4 values of S mentioned in the main text.

[12] Training for a single epoch is a restriction needed to carry out a fair comparison between full and partial information algorithms: Cycling more than once on a training set may turn a partial information algorithm into a full information one.

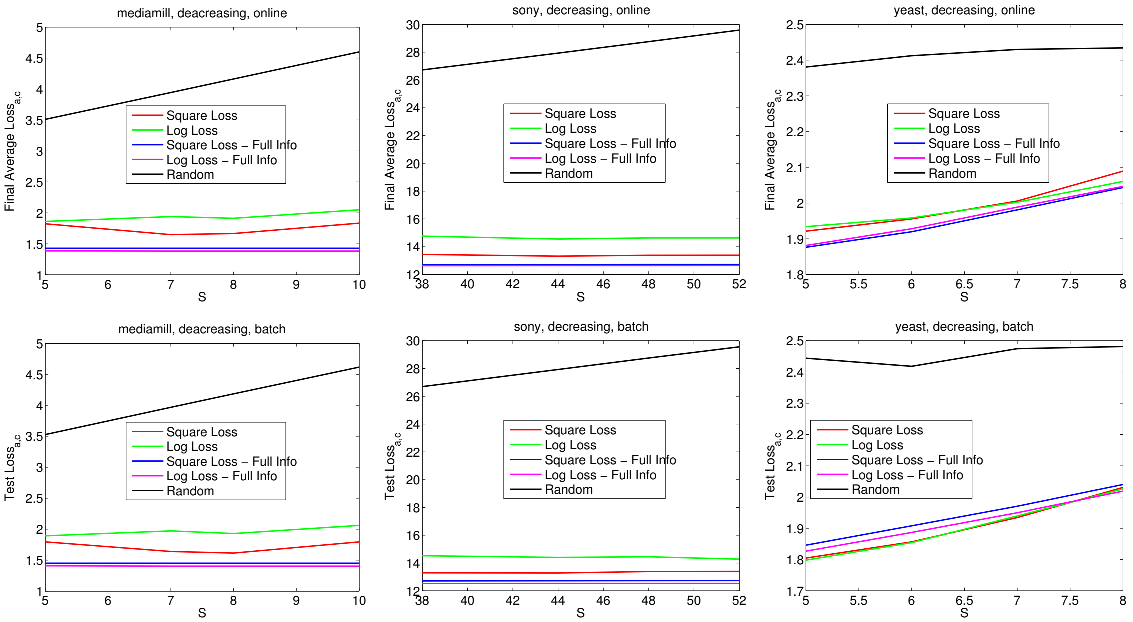

Figure 3: Experiments with $\ell_{a,c}$ and decreasing costs.

noise introduced during exploration, that acts as a kind of regularization, improving generalization performance in such a small data set. In general, however, the comparison linear vs. logistic is somewhat mixed.

In the ranking setting (Figure 5) we also show the performance of our algorithm when the order of predicted labels is randomly permuted ("Shuffled"). It is shown that, uniformly over all settings, shuffling causes performance degradation, thereby proving that our algorithm is indeed learning a meaningful ranking over the labels in the set $\hat{Y}_t$, even without receiving any ranking feedback within this set from the user.

6. Technical Details

This section contains all proofs missing from the main text, along with ancillary results and comments.

The algorithm in Figure 2 works by updating through the gradients $\nabla_{i,t}$ of a modular margin-based loss function $\sum_{i=1}^{K} L(\boldsymbol{w}_i^\top \boldsymbol{x})$ associated with the label generation model (2), i.e., associated with function g, so as to make the parameters $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K) \in \mathcal{R}^{dK}$ therein achieve the Bayes optimality condition

$$(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K) = \arg \min_{\boldsymbol{w}_1,\ldots,\boldsymbol{w}_K \,:\, \boldsymbol{w}_i^\top \boldsymbol{x}_t \in D} \mathbb{E}_t \left[\sum_{i=1}^{K} L(s_{i,t}\, \boldsymbol{w}_i^\top \boldsymbol{x}_t) \right], \qquad (6)$$

where $\mathbb{E}_t[\cdot]$ above is over the generation of Y_t in producing the sign value $s_{i,t} \in \{-1, 0, +1\}$, conditioned on the past (in particular, conditioned on $\hat{Y}_t$). The requirement in (6) is akin to the classical construction of *proper scoring rules* in the statistical literature (e.g., Savage, 1973).

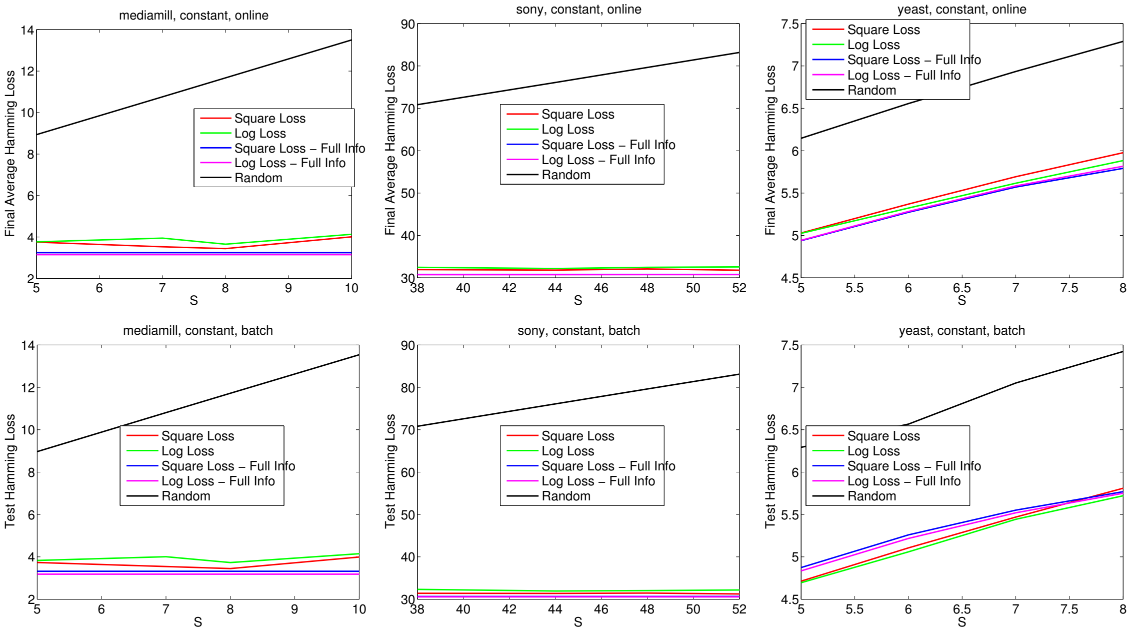

Figure 4: Experiments with $\ell_{a,c}$ and constant costs (Hamming loss).

The above is combined with the ability of the algorithm to guarantee the high probability convergence of the prototype vectors $\boldsymbol{w}'_{i,t}$ to the corresponding $\boldsymbol{u}_i$ (Lemma 13). The rate of convergence is ruled by the fact that the associated upper confidence values $\epsilon_{i,t}$ shrink to zero as $\frac{1}{\sqrt{t}}$ when t grows large. In order for this convergence to take place, it is important to insure that the algorithm is observing informative feedback (either "correct", i.e., $s_{i,t} = 1$, or "mistaken", i.e., $s_{i,t} = -1$) for each class i contained in the selected $\hat{Y}_t$. This in turn implies regret bounds for both $\ell_{a,c}$ (Lemma 11) and $\ell_{p-rank,t}$ (Lemma 12).

The following lemma faces the problem of hand-crafting a convenient loss function $L(\cdot)$ such that (6) holds.

Lemma 9 *Let $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K \in \mathcal{R}^{dK}$ be arbitrary weight vectors such that $\boldsymbol{w}_i^\top \boldsymbol{x}_t \in D$, $i \in [K]$, $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K) \in \mathcal{R}^{dK}$ be defined in (2), $s_{i,t}$ be the updating signs computed by the algorithm at the end (Step 5) of time t, $L : D = [-R, R] \subseteq \mathcal{R} \to \mathcal{R}^+$ be a convex and differentiable function of its argument, with $g(\Delta) = -L'(\Delta)$. Then for any t we have*

$$\mathbb{E}_t\left[\sum_{i=1}^{K} L(s_{i,t}\,\boldsymbol{w}_i^\top \boldsymbol{x}_t)\right] \geq \mathbb{E}_t\left[\sum_{i=1}^{K} L(s_{i,t}\,\boldsymbol{u}_i^\top \boldsymbol{x}_t)\right],$$

i.e., (6) holds.

Proof Let us introduce the shorthands $\Delta_i = \boldsymbol{u}_i^\top \boldsymbol{x}_t$, $\widehat{\Delta}_i = \boldsymbol{w}_{i,t}^\top \boldsymbol{x}_t$, $s_i = s_{i,t}$, and $p_i = \mathbb{P}(y_{i,t} = 1 \,|\, \boldsymbol{x}_t) = \frac{L'(-\Delta_i)}{L'(\Delta_i)+L'(-\Delta_i)} = \frac{g(-\Delta_i)}{g(\Delta_i)+g(-\Delta_i)}$. Moreover, let $\mathbb{P}_t(\cdot)$ be an abbreviation for the conditional probability $\mathbb{P}(\cdot \,|\, (y_1, \boldsymbol{x}_1), \ldots, (y_{t-1}, \boldsymbol{x}_{t-1}), \boldsymbol{x}_t)$. Recalling the way $s_{i,t}$ is

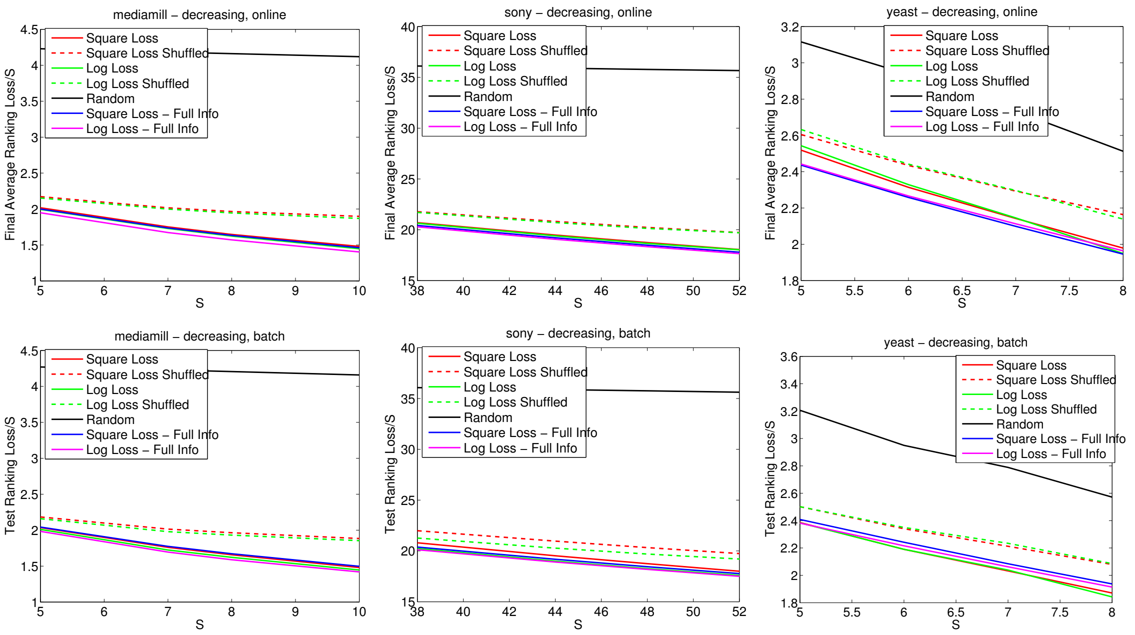

Figure 5: Experiments with the ranking loss $\ell_{p-rank,t}$. In order to obtain "scale-independent" results, in this figure we actually used $\ell_{p-rank,t}/S$ rather than $\ell_{p-rank,t}$ itself.

constructed (Figure 2), we can write

$$\begin{aligned}\mathbb{E}_t\left[\sum_{i=1}^{K} L(s_{i,t}\,\widehat{\Delta}_i)\right] &= \sum_{i\in\hat{Y}_t}\Big(\mathbb{P}_t(s_{i,t}=1)\,L(\widehat{\Delta}_i)+\mathbb{P}_t(s_{i,t}=-1)\,L(-\widehat{\Delta}_i)\Big)+(K-|\hat{Y}_t|)\,L(0)\\ &= \sum_{i\in\hat{Y}_t}\Big(p_i\,L(\widehat{\Delta}_i)+(1-p_i)\,L(-\widehat{\Delta}_i)\Big)+(K-|\hat{Y}_t|)\,L(0),\end{aligned}$$

For similar reasons,

$$\mathbb{E}_t\left[\sum_{i=1}^{K} L(s_{i,t}\,\Delta_i)\right] = \sum_{i\in\hat{Y}_t}\big(p_i\,L(\Delta_i)+(1-p_i)\,L(-\Delta_i)\big)+(K-|\hat{Y}_t|)\,L(0).$$

Since $L(\cdot)$ is convex, so is $\mathbb{E}_t\left[\sum_{i=1}^{K} L(s_{i,t}\,\widehat{\Delta}_i)\right]$ when viewed as a function of the $\widehat{\Delta}_i$. We have that $\frac{\partial\,\mathbb{E}_t\left[\sum_{i=1}^{K} L(s_{i,t}\,\widehat{\Delta}_i)\right]}{\partial\widehat{\Delta}_i}=0$ if and only if for all $i\in\hat{Y}_t$ we have that $\widehat{\Delta}_i$ satisfies

$$p_i = \frac{L'(-\widehat{\Delta}_i)}{L'(\widehat{\Delta}_i)+L'(-\widehat{\Delta}_i)}.$$

Since $p_i = \frac{L'(-\Delta_i)}{L'(\Delta_i)+L'(-\Delta_i)}$, we have that $\mathbb{E}_t\left[\sum_{i=1}^{K} L(s_{i,t}\,\widehat{\Delta}_i)\right]$ is minimized when $\widehat{\Delta}_i=\Delta_i$ for all $i\in[K]$. The claimed result immediately follows. ■

Let now $Var_t(\cdot)$ be a shorthand for $Var(\cdot \,|\, (y_1, \boldsymbol{x}_1), \ldots, (y_{t-1}, \boldsymbol{x}_{t-1}), \boldsymbol{x}_t)$. The following lemma shows that under additional assumptions on the loss $L(\cdot)$, we can bound the variance of a difference of losses $L(\cdot)$ by the expectation of this difference. This will be key to proving the fast rates of convergence contained in the subsequent Lemma 13.

Lemma 10 *Let $(\boldsymbol{w}'_{1,t}, \ldots, \boldsymbol{w}'_{K,t}) \in \mathcal{R}^{dK}$ be the weight vectors computed by the algorithm in Figure 2 at the beginning (Step 2) of time t, $s_{i,t}$ be the updating signs computed at the end (Step 5) of time t, and $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K) \in \mathcal{R}^{dK}$ be the comparison vectors defined through (2). Let $L : D = [-R, R] \subseteq \mathcal{R} \to \mathcal{R}^+$ be a $C^2(D)$ convex function of its argument, with $g(\Delta) = -L'(\Delta)$ and such that there are positive constants c'_L and c''_L with $(L'(\Delta))^2 \le c'_L$ and $L''(\Delta) \ge c''_L$ for all $\Delta \in D$. Then for any $i \in \hat{Y}_t$*

$$0 \le Var_t \left(L(s_{i,t}\, \boldsymbol{x}_t^\top \boldsymbol{w}'_{i,t}) - L(s_{i,t}\, \boldsymbol{u}_i^\top \boldsymbol{x}_t) \right) \le \frac{2c'_L}{c''_L}\, \mathbb{E}_t \left[L(s_{i,t}\, \boldsymbol{x}_t^\top \boldsymbol{w}'_{i,t}) - L(s_{i,t}\, \boldsymbol{u}_i^\top \boldsymbol{x}_t) \right].$$

Proof Let us introduce the shorthands $\Delta_i = \boldsymbol{x}_t^\top \boldsymbol{u}_i$, $\widehat{\Delta}_i = \boldsymbol{x}_t^\top \boldsymbol{w}'_{i,t}$, $s_i = s_{i,t}$, and $p_i = \mathbb{P}(y_{i,t} = 1 \,|\, \boldsymbol{x}_t) = \frac{L'(-\Delta_i)}{L'(\Delta_i)+L'(-\Delta_i)} = \frac{g(-\Delta_i)}{g(\Delta_i)+g(-\Delta_i)}$. Then, for any $i \in [K]$,

$$Var_t \left(L(s_{i,t}\, \boldsymbol{x}_t^\top \boldsymbol{w}'_{i,t}) - L(s_{i,t}\, \boldsymbol{u}_i^\top \boldsymbol{x}_t) \right) \le \mathbb{E}_t \left(\left(L(s_i\, \widehat{\Delta}_i) - L(s_i\, \Delta_i) \right)^2 \right) \le c'_L\, (\widehat{\Delta}_i - \Delta_i)^2. \quad (7)$$

Moreover, for any $i \in \hat{Y}_t$ we can write

$$\begin{aligned}
\mathbb{E}_t \left[L(s_i\, \widehat{\Delta}_i) - L(s_i\, \Delta_i) \right] &= p_i\, (L(\widehat{\Delta}_i) - L(\Delta_i)) + (1 - p_i)\, (L(-\widehat{\Delta}_i) - L(-\Delta_i)) \\
&\ge p_i \left(L'(\Delta_i)(\widehat{\Delta}_i - \Delta_i) + \frac{c''_L}{2}(\widehat{\Delta}_i - \Delta_i)^2 \right) \\
&\quad + (1 - p_i) \left(L'(-\Delta_i)(\Delta_i - \widehat{\Delta}) + \frac{c''_L}{2}(\widehat{\Delta}_i - \Delta_i)^2 \right) \\
&= p_i\, \frac{c''_L}{2}(\widehat{\Delta}_i - \Delta_i)^2 + (1 - p_i)\, \frac{c''_L}{2}(\widehat{\Delta}_i - \Delta_i)^2 \\
&= \frac{c''_L}{2}(\widehat{\Delta}_i - \Delta_i)^2, \qquad (8)
\end{aligned}$$

where the second equality uses the definition of p_i. Combining (7) with (8) gives the desired bound. ■

We continue by showing a one-step regret bound *for our original* loss $\ell_{a,c}$. The precise connection to loss $L(\cdot)$ will be established with the help of a later lemma (Lemma 13).

Lemma 11 *Let $L : D = [-R, R] \subseteq \mathcal{R} \to \mathcal{R}^+$ be a convex, twice differentiable, and nonincreasing function of its argument. Let $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K) \in \mathcal{R}^{dK}$ be defined in (2) with $g(\Delta) = -L'(\Delta)$ for all $\Delta \in D$. Let also c_L be a positive constant such that*

$$\frac{L'(\Delta)\, L''(-\Delta) + L''(\Delta)\, L'(-\Delta)}{(L'(\Delta) + L'(-\Delta))^2} \ge -c_L$$

holds for all $\Delta \in D$. Finally, let $\Delta_{i,t}$ denote $\boldsymbol{u}_i^\top \boldsymbol{x}_t$, and $\widehat{\Delta}'_{i,t}$ denote $\boldsymbol{x}_t^\top \boldsymbol{w}'_{i,t}$, where $\boldsymbol{w}'_{i,t}$ is the i-the weight vector computed by the algorithm at the beginning (Step 2) of time t. If time t is such that $|\Delta_{i,t} - \widehat{\Delta}'_{i,t}| \le \epsilon_{i,t}$ for all $i \in [K]$, then

$$\mathbb{E}_t[\ell_{a,c}(Y_t, \hat{Y}_t)] - \mathbb{E}_t[\ell_{a,c}(Y_t, Y_t^*)] \le 2\,(1-a)\,c_L \sum_{i \in \hat{Y}_t} \epsilon_{i,t}.$$

Proof Recall the shorthand notation $p(\Delta) = \frac{g(-\Delta)}{g(\Delta)+g(-\Delta)}$. We can write

$$\begin{aligned}
&\mathbb{E}_t[\ell_{a,c}(Y_t, \hat{Y}_t)] - \mathbb{E}_t[\ell_{a,c}(Y_t, Y_t^*)] \\
&\qquad = (1-a) \sum_{i \in \hat{Y}_t} \Bigl(c(\hat{j}_i, |\hat{Y}_t|) - \Bigl(\tfrac{a}{1-a} + c(\hat{j}_i, |\hat{Y}_t|) \Bigr)\, p(\Delta_{i,t}) \Bigr) \\
&\qquad\quad - (1-a) \sum_{i \in Y_t^*} \Bigl(c(j_i^*, |Y_t^*|) - \Bigl(\tfrac{a}{1-a} + c(j_i^*, |Y_t^*|) \Bigr)\, p(\Delta_{i,t}) \Bigr),
\end{aligned}$$

where $\hat{j}_i$ denotes the position of class i in $\hat{Y}_t$ and j_i^* is the position of class i in Y_t^*. Now,

$$p'(\Delta) = \frac{-g'(-\Delta)\,g(\Delta) - g'(\Delta)\,g(-\Delta)}{(g(\Delta)+g(-\Delta))^2} = \frac{-L'(\Delta)\,L''(-\Delta) - L'(-\Delta)\,L''(\Delta)}{(L'(\Delta)+L'(-\Delta))^2} \ge 0$$

since $g(\Delta) = -L'(\Delta)$, and $L(\cdot)$ is convex and nonincreasing. Hence $p(\Delta)$ is itself a nondecreasing function of Δ. Moreover, the extra condition on L involving L' and L'' is a Lipschitz condition on $p(\Delta)$ via a uniform bound on $p'(\Delta)$. Hence, from $|\Delta_{i,t} - \widehat{\Delta}'_{i,t}| \le \epsilon_{i,t}$ and the definition of $\hat{Y}_t$ we can write

$$\begin{aligned}
&\mathbb{E}_t[\ell_{a,c}(Y_t, \hat{Y}_t)] - \mathbb{E}_t[\ell_{a,c}(Y_t, Y_t^*)] \\
&\qquad \le (1-a) \sum_{i \in \hat{Y}_t} \Bigl(c(\hat{j}_i, |\hat{Y}_t|) - \Bigl(\tfrac{a}{1-a} + c(\hat{j}_i, |\hat{Y}_t|) \Bigr)\, p([\widehat{\Delta}'_{i,t} - \epsilon_{i,t}]_D) \Bigr) \\
&\qquad\quad - (1-a) \sum_{i \in Y_t^*} \Bigl(c(j_i^*, |Y_t^*|) - \Bigl(\tfrac{a}{1-a} + c(j_i^*, |Y_t^*|) \Bigr)\, p([\widehat{\Delta}'_{i,t} + \epsilon_{i,t}]_D) \Bigr) \\
&\qquad \le (1-a) \sum_{i \in \hat{Y}_t} \Bigl(c(\hat{j}_i, |\hat{Y}_t|) - \Bigl(\tfrac{a}{1-a} + c(\hat{j}_i, |\hat{Y}_t|) \Bigr)\, p([\widehat{\Delta}'_{i,t} - \epsilon_{i,t}]_D) \Bigr) \\
&\qquad\quad - (1-a) \sum_{i \in \hat{Y}_t} \Bigl(c(\hat{j}_i, |\hat{Y}_t|) - \Bigl(\tfrac{a}{1-a} + c(\hat{j}_i, |\hat{Y}_t|) \Bigr)\, p([\widehat{\Delta}'_{i,t} + \epsilon_{i,t}]_D) \Bigr) \\
&\qquad = (1-a) \sum_{i \in \hat{Y}_t} \Bigl(c(\hat{j}_i, |\hat{Y}_t|) \Bigl(p([\widehat{\Delta}'_{i,t} + \epsilon_{i,t}]_D) - p([\widehat{\Delta}'_{i,t} - \epsilon_{i,t}]_D) \Bigr) \Bigr) \\
&\qquad \le 2\,(1-a)\,c_L \sum_{i \in \hat{Y}_t} \epsilon_{i,t},
\end{aligned}$$

the last inequality deriving from $c(i,s) \le 1$ for all $i \le s \le K$, and

$$p([\widehat{\Delta}'_{i,t} + \epsilon_{i,t}]_D) - p([\widehat{\Delta}'_{i,t} - \epsilon_{i,t}]_D) \le c_L \bigl([\widehat{\Delta}'_{i,t} + \epsilon_{i,t}]_D - [\widehat{\Delta}'_{i,t} - \epsilon_{i,t}]_D \bigr) \le 2\,c_L\,\epsilon_{i,t}.$$

■

Now, we first give a proof of Lemma 6, and then provide a one step regret for the partial information ranking loss.

Proof [Lemma 6] Recall the notation $\mathbb{P}_t(\cdot) = \mathbb{P}(\cdot\,|\,\boldsymbol{x}_t)$, and $p_{i,t} = p(\Delta_{i,t}) = \frac{g(-\Delta_{i,t})}{g(\Delta_{i,t})+g(-\Delta_{i,t})}$. For notational convenience, in this proof we drop subscript t from $p_{i,t}$, S_t, $y_{i,t}$, $\hat{Y}_t$, and $\ell_{p-rank,t}$. A simple adaptation of Dembczynski et al. (2012) (proof of Theorem 1 therein) shows that for a generic sequence $\widehat{a} = (\widehat{a}_1, \ldots, \widehat{a}_K)$ with at most S nonzero values $\widehat{a}_i$ and associated set of indices $\hat{Y}$, one has

$$\mathbb{E}_t[\ell_{p-rank}(Y_t, \widehat{a})] = \sum_{i,j\in\hat{Y},\, i<j} (\widehat{r}_{i,j} + \widehat{r}_{j,i}) + S\left(\sum_{i\in[K]} p_i - \sum_{i\in\hat{Y}} p_i\right)$$

where

$$\widehat{r}_{i,j} = \widehat{r}_{i,j}(\widehat{a}) = \mathbb{P}_t(y_i > y_j)\,\left(\{\widehat{a}_i < \widehat{a}_j\} + \tfrac{1}{2}\,\{\widehat{a}_i = \widehat{a}_j\}\right)\,.$$

Moreover, if p^* denotes the sequence made up of at most S nonzero values taken from $\{p_i\,, i\in[K]\}$, *where i ranges again in $\hat{Y}$*, we have

$$\mathbb{E}_t[\ell_{p-rank}(Y_t, p^*)] = \sum_{i,j\in\hat{Y},\, i<j} (r_{i,j} + r_{j,i}) + S\left(\sum_{i\in[K]} p_i - \sum_{i\in\hat{Y}} p_i\right)$$

with

$$r_{i,j} = r_{i,j}(p^*) = \mathbb{P}_t(y_i > y_j)\,\left(\{p_i < p_j\} + \tfrac{1}{2}\,\{p_i = p_j\}\right)\,.$$

Hence

$$\mathbb{E}_t[\ell_{p-rank}(Y_t, \widehat{a})] - \mathbb{E}_t[\ell_{p-rank}(Y_t, p^*)] = \sum_{i,j\in\hat{Y},\, i<j} (\widehat{r}_{i,j} - r_{i,j} + \widehat{r}_{j,i} - r_{j,i})\,.$$

Since

$$\mathbb{P}_t(y_i > y_j) - \mathbb{P}_t(y_j > y_i) = \mathbb{P}_t(y_i = 1) - \mathbb{P}_t(y_j = 1) = p_i - p_j,$$

a simple (but lengthy) case analysis reveals that

$$\widehat{r}_{i,j} - r_{i,j} + \widehat{r}_{j,i} - r_{j,i} = \begin{cases} \frac{1}{2}\,(p_i - p_j) & \text{If } \widehat{a}_i < \widehat{a}_j,\ p_i = p_j \text{ or } \widehat{a}_i = \widehat{a}_j,\ p_i > p_j \\ \frac{1}{2}\,(p_j - p_i) & \text{If } \widehat{a}_i = \widehat{a}_j,\ p_i < p_j \text{ or } \widehat{a}_i > \widehat{a}_j,\ p_i = p_j \\ p_i - p_j & \text{If } \widehat{a}_i < \widehat{a}_j,\ p_i > p_j \\ p_j - p_i & \text{If } \widehat{a}_i > \widehat{a}_j,\ p_i < p_j\,. \end{cases}$$

Notice that the above quantity is always nonnegative, and is strictly positive if the p_i are all different. The nonnegativity implies that *whatever set of indices $\hat{Y}$ we select*, the best way to sort them within $\hat{Y}$ in order to minimize $\mathbb{E}_t[\ell_{p-rank}(Y_t, \cdot)]$ is by following the ordering of the corresponding p_i.

We are left to show that the best choice for $\hat{Y}$ is to collect the S largest[13] values in $\{p_i\,, i \in [K]\}$. To this effect, consider again $\mathbb{E}_t[\ell_{p-rank}(Y_t, p^*)] = \mathbb{E}_t[\ell_{p-rank}(Y_t, \hat{Y})]$, and introduce the shorthand $p_{i,j} = p_i\, p_j = p_i - \mathbb{P}_t(y_i > y_j)$. Disregarding the term $S \sum_{i \in [K]} p_i$, which is independent of $\hat{Y}$, we can write

$$\begin{aligned}
\mathbb{E}_t[\ell_{p-rank}(Y_t, \hat{Y})] &= \sum_{i,j \in \hat{Y},\, i<j} \mathbb{P}_t(y_i > y_j)\, \big(\{p_i < p_j\} + \tfrac{1}{2}\,\{p_i = p_j\}\big) \\
&\quad + \sum_{i,j \in \hat{Y},\, i<j} \mathbb{P}_t(y_j > y_i)\, \big(\{p_j < p_i\} + \tfrac{1}{2}\,\{p_j = p_i\}\big) - S \sum_{i \in \hat{Y}} p_i \\
&= \sum_{i,j \in \hat{Y},\, i<j} (p_i - p_{i,j})\{p_i < p_j\} + (p_i - p_{i,j})\tfrac{1}{2}\,\{p_i = p_j\} \\
&\quad + \sum_{i,j \in \hat{Y},\, i<j} (p_j - p_{i,j})\{p_j < p_i\} + (p_j - p_{i,j})\tfrac{1}{2}\,\{p_j = p_i\} - S \sum_{i \in \hat{Y}} p_i \\
&= \sum_{i,j \in \hat{Y},\, i<j} (p_i - p_j)\{p_i < p_j\} + \tfrac{1}{2}\,(p_i - p_j)\,\{p_i = p_j\} + p_j - p_{i,j} - S \sum_{i \in \hat{Y}} p_i \\
&= \sum_{i,j \in \hat{Y},\, i<j} (\min\{p_i, p_j\} - p_i p_j) - S \sum_{i \in \hat{Y}} p_i
\end{aligned}$$

which can be finally seen to be equal to

$$-\sum_{i \in \hat{Y}} (S + 1 - \hat{j}_i)\, p_i - \sum_{i,j \in \hat{Y},\, i<j} p_i\, p_j\,, \qquad (9)$$

where $\hat{j}_i$ is the position of class i within $\hat{Y}_t$ in decreasing order of p_i.

Now, rename the indices in $\hat{Y}$ as $1, 2, \ldots, S$, in such a way that $p_1 > p_2 > \ldots > p_S$ (so that $\hat{j}_i = i$), and consider the way to increase (9) by adding to $\hat{Y}$ item $k \notin \hat{Y}$ such that $p_S > p_k$ and removing from $\hat{Y}$ the item in position ℓ. Denote the resulting sequence by $\hat{Y}'$. From (9), it is not hard to see that

$$\begin{aligned}
&\mathbb{E}_t[\ell_{p-rank}(Y_t, \hat{Y})] - \mathbb{E}_t[\ell_{p-rank}(Y_t, \hat{Y}')] \\
&\quad = (\ell - 1)\, p_\ell + \sum_{i=\ell+1}^{S} p_i - \sum_{i=1}^{\ell-1} p_i\, p_\ell - \sum_{i=\ell+1}^{S} p_\ell\, p_i - (S-1)\, p_k + \sum_{i=1, i \neq \ell}^{S} p_i\, p_k - S(p_\ell - p_k) \\
&\quad = (\ell - 1)\, p_\ell + \sum_{i=\ell+1}^{S} p_i - (p_\ell - p_k) \sum_{i=1, i \neq \ell}^{S} p_i - (S-1)\, p_k - S(p_\ell - p_k) \\
&\quad \le (S - 1)\, p_\ell - (p_\ell - p_k) \sum_{i=1, i \neq \ell}^{S} p_i - (S-1)\, p_k - S(p_\ell - p_k) \\
&\quad = (p_k - p_\ell) \left(1 + \sum_{i=1, i \neq \ell}^{S} p_i\right) \qquad (10)
\end{aligned}$$

[13] It is at this point that we need the conditional independence assumption over the classes.

which is smaller than zero since, by assumption, $p_\ell > p_k$. Reversing the direction, if we maintain a sequence $\hat{Y}$ of size S, we can always reduce (9) by removing its smallest element and replacing it with a larger element outside the sequence. We continue until no element outside the current sequence exists which is larger than the smallest one in the sequence. Clearly, we end up collecting the S largest elements in $\{p_i\,, i \in [K]\}$.

Finally, from (9) it is very clear that removing an element from a sequence $\hat{Y}$ of length $h \le S$ can only increase the value of (9). Since this holds for an arbitrary $\hat{Y}$ and an arbitrary $h \le S$, this shows that, no matter which set $\hat{Y}$ we start off from, we always converge to the same set containing exactly the S largest elements in $\{p_i\,, i \in [K]\}$. This concludes the proof. ■

Lemma 12 *Under the same assumptions and notation as in Lemma 11, combined with the independence assumption (4), let the Algorithm in Figure 2 be working with $a \to 1$ and strictly decreasing cost values $c(i,s)$, i.e., the algorithm is computing in round t the ranking function $\widehat{f}(\boldsymbol{x}_t; S_t)$ defined in Section 4. Let $\boldsymbol{w}'_{i,t}$ be the i-th weight vector computed by this algorithm at the beginning (Step 2) of time t. If time t is such that $|\Delta_{i,t} - \widehat{\Delta}'_{i,t}| \le \epsilon_{i,t}$ for all $i \in [K]$, then*

$$\mathbb{E}_t[\ell_{rank,t}(Y_t, \widehat{f}(\boldsymbol{x}_t; S_t)] - \mathbb{E}_t[\ell_{rank,t}(Y_t, f^*(\boldsymbol{x}_t; S_t)] \le 4\, S_t\, c_L \sum_{i \in \hat{Y}_t} \epsilon_{i,t}.$$

Proof We use the same notation as in the proof of Lemma 6, where $\widehat{a}$ is now $\hat{Y}_t$, the sequence produced by ranking $\widehat{f}(\boldsymbol{x}_t; S_t)$ operating on $\widehat{p}_{i,t}$. Denote by Y^*_t the sequences determined by $f^*(\boldsymbol{x}_t; S_t)$, and let $\hat{j}_i$ and j^*_i be the position of class i in decreasing order of $p_{i,t}$ within $\hat{Y}_t$ and Y^*_t, respectively.

Proceeding as in Lemma 11 and recalling (9) we can write

$$\begin{aligned}
&\mathbb{E}_t[\ell_{p-rank,t}(Y_t, \widehat{f}(\boldsymbol{x}_t; S_t))] - \mathbb{E}_t[\ell_{p-rank,t}(Y_t, f^*(\boldsymbol{x}_t; S_t)] \\
&\quad= \sum_{i \in Y^*_t} (S_t + 1 - j^*_i)\, p_i + \sum_{i,j \in Y^*_t,\, i<j} p_i\, p_j - \sum_{i \in \hat{Y}_t} (S_t + 1 - \hat{j}_i)\, p_i - \sum_{i,j \in \hat{Y}_t,\, i<j} p_i\, p_j \\
&\quad\le \sum_{i \in Y^*_t} (S_t + 1 - j^*_i)\, p([\widehat{\Delta}'_{i,t} + \epsilon_{i,t}]_D) + \sum_{i,j \in Y^*_t,\, i<j} p([\widehat{\Delta}'_{i,t} + \epsilon_{i,t}]_D)\, p([\widehat{\Delta}'_{j,t} + \epsilon_{j,t}]_D) \\
&\qquad\quad - \sum_{i \in \hat{Y}_t} (S_t + 1 - \hat{j}_i)\, p([\widehat{\Delta}'_{i,t} - \epsilon_{i,t}]_D) - \sum_{i,j \in \hat{Y}_t,\, i<j} p([\widehat{\Delta}'_{i,t} - \epsilon_{i,t}]_D)\, p([\widehat{\Delta}'_{j,t} - \epsilon_{j,t}]_D) \\
&\quad\le \sum_{i \in \hat{Y}_t} (S_t + 1 - \hat{j}_i)\, \Big(p([\widehat{\Delta}'_{i,t} + \epsilon_{i,t}]_D) - p([\widehat{\Delta}'_{i,t} - \epsilon_{i,t}]_D) \Big) \\
&\qquad\quad + \sum_{i,j \in \hat{Y}_t,\, i<j} \Big(p([\widehat{\Delta}'_{i,t} + \epsilon_{i,t}]_D)\, p([\widehat{\Delta}'_{j,t} + \epsilon_{j,t}]_D) - p([\widehat{\Delta}'_{i,t} - \epsilon_{i,t}]_D)\, p([\widehat{\Delta}'_{j,t} - \epsilon_{j,t}]_D) \Big)\,.
\end{aligned}$$

This, in turn, can be upper bounded by

$$2S_t c_L \sum_{i \in \hat{Y}_t} \epsilon_{i,t} + \sum_{i,j \in \hat{Y}_t,\, i<j} 2c_L\,(\epsilon_{i,t} + \epsilon_{j,t}) = 2\,S_t\,c_L \sum_{i \in \hat{Y}_t} \epsilon_{i,t} + 2\,(S_t - 1)\,c_L \sum_{i \in \hat{Y}_t} \epsilon_{i,t}$$
$$< 4\,S_t\,c_L \sum_{i \in \hat{Y}_t} \epsilon_{i,t}\,,$$

as claimed. ■

Lemma 13 *Let $L : D = [-R, R] \subseteq \mathcal{R} \rightarrow \mathcal{R}^+$ be a $C^2(D)$ convex and nonincreasing function of its argument, $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K) \in \mathcal{R}^{dK}$ be defined in (2) with $g(\Delta) = -L'(\Delta)$ for all $\Delta \in D$, and such that $\|\boldsymbol{u}_i\| \le U$ for all $i \in [K]$. Assume there are positive constants c'_L and c''_L with $(L'(\Delta))^2 \le c'_L$ and $L''(\Delta) \ge c''_L$ for all $\Delta \in D$. With the notation introduced in Figure 2, we have that*

$$(\boldsymbol{x}^\top \boldsymbol{w}'_{i,t} - \boldsymbol{u}_i^\top \boldsymbol{x})^2 \le \boldsymbol{x}^\top A_{i,t-1}^{-1} \boldsymbol{x} \left(U^2 + \frac{d\,c'_L}{(c''_L)^2} \ln\left(1 + \frac{t-1}{d}\right) + \frac{12}{c''_L} \left(\frac{c'_L}{c''_L} + 3L(-R)\right) \ln \frac{K(t+4)}{\delta} \right)$$

holds with probability at least $1 - \delta$ for any $\delta < 1/e$, uniformly *over $i \in [K]$, $t = 1, 2, \ldots,$ and $\boldsymbol{x} \in \mathcal{R}^d$.*

Proof For any given class i, the time-t update rule $\boldsymbol{w}'_{i,t} \rightarrow \boldsymbol{w}_{i,t+1} \rightarrow \boldsymbol{w}'_{i,t+1}$ in Figure 2 allows us to start off from the paper by Hazan et al. (2007) (proof of Theorem 2 therein), from which one can extract the following inequality

$$d_{i,t-1}(\boldsymbol{u}_i, \boldsymbol{w}'_{i,t}) \le U^2 + \frac{1}{(c''_L)^2} \sum_{k=1}^{t-1} r_{i,k} - \frac{2}{c''_L} \sum_{k=1}^{t-1} \left(\nabla_{i,k}^\top (\boldsymbol{w}'_{i,k} - \boldsymbol{u}_i) - \frac{c''_L}{2} \left(s_{i,k}\, \boldsymbol{x}_k^\top (\boldsymbol{w}'_{i,k} - \boldsymbol{u}_i) \right)^2 \right), \quad (11)$$

where we set $r_{i,k} = \nabla_{i,k}^\top\, A_{i,k}^{-1}\, \nabla_{i,k}$.

We now observe that we can construct a quadratic lower bound to L, using the lower bound on the second derivative of L. More explicitly, using the Taylor expansion of L, we have

$$L(x) \ge L(y) + L'(y)(x-y) + \frac{c''_L}{2}(x-y)^2,$$

for any x, y in D. Hence, setting $y = s_{i,k}\, \boldsymbol{x}_k^\top \boldsymbol{w}'_{i,k}$ and $x = s_{i,k}\, \boldsymbol{u}_i^\top \boldsymbol{x}_k$, we have

$$L(s_{i,k}\, \boldsymbol{x}_k^\top \boldsymbol{w}'_{i,k}) - L(s_{i,k}\, \boldsymbol{u}_i^\top \boldsymbol{x}_k)$$
$$\le L'(s_{i,k}\, \boldsymbol{x}_k^\top \boldsymbol{w}'_{i,k})(s_{i,k} \boldsymbol{x}_k^\top \boldsymbol{w}'_{i,k} - s_{i,k}\, \boldsymbol{u}_i^\top \boldsymbol{x}_k) - \frac{c''_L}{2}(s_{i,k}\, \boldsymbol{x}_k^\top \boldsymbol{w}'_{i,k} - s_{i,k}\, \boldsymbol{u}_i^\top \boldsymbol{x}_k)^2$$
$$= \nabla_{i,k}^\top (\boldsymbol{w}'_{i,k} - \boldsymbol{u}_i) - \frac{c''_L}{2} \left(s_{i,k}\, \boldsymbol{x}_k^\top (\boldsymbol{w}'_{i,k} - \boldsymbol{u}_i) \right)^2.$$

Plugging back into (11) yields

$$d_{i,t-1}(\boldsymbol{u}_i, \boldsymbol{w}'_{i,t}) \le U^2 + \frac{1}{(c''_L)^2}\sum_{k=1}^{t-1} r_{i,k} - \frac{2}{c''_L}\sum_{k=1}^{t-1}\Big(L(s_{i,k}\,\boldsymbol{x}_k^\top \boldsymbol{w}'_{i,k}) - L(s_{i,k}\,\boldsymbol{u}_i^\top \boldsymbol{x}_k)\Big)\,. \qquad (12)$$

We now borrow a proof technique from Dekel et al. (2012) (see also the papers by Crammer and Gentile 2011; Abbasi-Yadkori et al. 2011 and references therein). Define

$$L_{i,k} = L(s_{i,k}\,\boldsymbol{x}_k^\top \boldsymbol{w}'_{i,k}) - L(s_{i,k}\,\boldsymbol{u}_i^\top \boldsymbol{x}_k),$$

and $L'_{i,k} = \mathbb{E}_k[L_{i,k}] - L_{i,k}$. Notice that the sequence of random variables $L'_{i,1}, L'_{i,2}, \ldots$, forms a martingale difference sequence such that, for any $i \in \hat{Y}_k$:

i. $\mathbb{E}_k[L_{i,k}] \ge 0$, by Lemma 10 (or Lemma 9);

ii. $|L'_{i,k}| \le 2L(-R)$, since $L(\cdot)$ is nonincreasing over D, and $s_{i,k}\,\boldsymbol{x}_k^\top \boldsymbol{w}'_{i,k}$, $s_{i,k}\,\boldsymbol{u}_i^\top \boldsymbol{x}_k \in D$;

iii. $Var_k(L'_{i,k}) = Var_k(L_{i,k}) \le \frac{2c'_L}{c''_L}\,\mathbb{E}_k[L_{i,k}]$ (again, because of Lemma 10).

On the other hand, when $i \notin \hat{Y}_k$ then $s_{i,k} = 0$, and the above three properties are trivially satisfied. Under the above conditions, we are in a position to apply any fast concentration result for bounded martingale difference sequences. For instance, setting for brevity $B = B(t,\delta) = 3\,\ln\frac{K(t+4)}{\delta}$, a result contained in the paper by Kakade and Tewari (2009) allows us derive the inequality

$$\sum_{k=1}^{t-1}\mathbb{E}_k[L_{i,k}] - \sum_{k=1}^{t-1} L_{i,k} \ge \max\left\{\sqrt{\frac{8c'_L}{c''_L}\,B\sum_{k=1}^{t-1}\mathbb{E}_k[L_{i,k}]},\, 6L(-R)\,B\right\},$$

that holds with probability at most $\frac{\delta}{Kt(t+1)}$ for any $t \ge 1$. We use the inequality $\sqrt{cb} \le \frac{1}{2}(c+b)$ with $c = \frac{4c'_L}{c''_L}\,B$, and $b = 2\sum_{k=1}^{t-1}\mathbb{E}_k[L_{i,k}]$, and simplify. This gives

$$-\sum_{k=1}^{t-1} L_{i,k} \le \left(\frac{2c'_L}{c''_L} + 6L(-R)\right)\,B$$

with probability at least $1 - \frac{\delta}{Kt(t+1)}$. Using the Cauchy-Schwarz inequality

$$(\boldsymbol{x}^\top \boldsymbol{w}'_{i,t} - \boldsymbol{u}_i^\top \boldsymbol{x})^2 \le \boldsymbol{x}^\top A_{i,t-1}^{-1}\,\boldsymbol{x}\, d_{i,t-1}(\boldsymbol{u}_i, \boldsymbol{w}'_{i,t})$$

holding for any $\boldsymbol{x} \in \mathcal{R}^d$, and replacing back into (12) allows us to conclude that

$$(\boldsymbol{x}^\top \boldsymbol{w}'_{i,t} - \boldsymbol{u}_i^\top \boldsymbol{x})^2 \le \boldsymbol{x}^\top A_{i,t-1}^{-1}\boldsymbol{x}\,\left(U^2 + \frac{1}{(c''_L)^2}\sum_{k=1}^{t-1} r_{i,k} + \frac{12}{c''_L}\left(\frac{c'_L}{c''_L} + 3L(-R)\right)\ln\frac{K(t+4)}{\delta}\right) \qquad (13)$$

holds with probability at least $1 - \frac{\delta}{Kt(t+1)}$, *uniformly* over $\boldsymbol{x} \in \mathcal{R}^d$.

The bounds on $\sum_{k=1}^{t-1} r_{i,k}$ can be obtained in a standard way. Applying known inequalities (Azoury and Warmuth, 2001; Cesa-Bianchi et al., 2002, 2009; Cavallanti et al., 2011; Hazan et al., 2007; Dekel et al., 2012), and using the fact that $\nabla_{i,k} = L'(s_{i,k}\,\boldsymbol{x}_k^\top \boldsymbol{w}'_{i,k})\,s_{i,k}\boldsymbol{x}_k$ we have

$$\begin{aligned}\sum_{k=1}^{t-1} r_{i,k} &= \sum_{k=1}^{t-1} |s_{i,j}|\,(L'(s_{i,k}\,\boldsymbol{x}_k^\top \boldsymbol{w}'_{i,k}))^2\,\boldsymbol{x}_k^\top A_{i,k}^{-1}\boldsymbol{x}_k \le c'_L \sum_{k=1}^{t-1} |s_{i,k}|\boldsymbol{x}_k^\top A_{i,k}^{-1}\boldsymbol{x}_k\\ &\le c'_L \sum_{k=1}^{t-1} \ln\frac{|A_{i,k}|}{|A_{i,k-1}|} = c'_L\,\ln\frac{|A_{i,t-1}|}{|A_{i,0}|} \le d\,c'_L\,\ln\left(1+\frac{t-1}{d}\right).\end{aligned}$$

Combining as in (13) and stratifying over $t = 1, 2, \ldots,$ and $i \in [K]$ concludes the proof. ■

We are now ready to put all pieces together.

Proof [Theorem 2] From Lemma 11 and Lemma 13, we see that with probability at least $1-\delta$,

$$R_T \le 2\,(1-a)\,c_L \sum_{t=1}^{T}\sum_{i\in\hat{Y}_t} \epsilon_{i,t}\,, \tag{14}$$

when $\epsilon_{i,t}^2$ is the one given in Figure 2. We continue by proving a pointwise upper bound on the sum in the RHS. More in detail, we will find an upper bound on $\sum_{t=1}^{T}\sum_{i\in\hat{Y}_t}\epsilon_{i,t}^2$, and then derive a resulting upper bound on the RHS of (14).

From Lemma 13 and the update rule (Step 5) of the algorithm we can write[14]

$$\begin{aligned}\epsilon_{i,t}^2 &\le C\,\boldsymbol{x}_t^\top A_{i,t-1}^{-1}\boldsymbol{x}_t = C\,\frac{\boldsymbol{x}_t^\top (A_{i,t-1} + |s_{i,t}|\,\boldsymbol{x}_t\boldsymbol{x}_t^\top)^{-1}\boldsymbol{x}_t}{1 - |s_{i,t}|\boldsymbol{x}_t^\top (A_{i,t-1} + |s_{i,t}|\,\boldsymbol{x}_t\boldsymbol{x}_t^\top)^{-1}\boldsymbol{x}_t}\\ &= C\,\frac{\boldsymbol{x}_t^\top A_{i,t}^{-1}\boldsymbol{x}_t}{1 - |s_{i,t}|\boldsymbol{x}_t^\top (A_{i,t-1} + |s_{i,t}|\,\boldsymbol{x}_t\boldsymbol{x}_t^\top)^{-1}\boldsymbol{x}_t}\\ &\le C\,\frac{\boldsymbol{x}_t^\top A_{i,t}^{-1}\boldsymbol{x}_t}{1 - |s_{i,t}|\boldsymbol{x}_t^\top (A_0 + |s_{i,t}|\,\boldsymbol{x}_t\boldsymbol{x}_t^\top)^{-1}\boldsymbol{x}_t} = C\,\frac{\boldsymbol{x}_t^\top A_{i,t}^{-1}\boldsymbol{x}_t}{1-\frac{1}{2}} = 2\,C\,\boldsymbol{x}_t^\top A_{i,t}^{-1}\boldsymbol{x}_t.\end{aligned}$$

Hence, if we set $r_{i,t} = \boldsymbol{x}_t^\top A_{i,t}^{-1}\boldsymbol{x}_t$ and proceed as in the proof of Lemma 13, we end up with the upper bound $\sum_{t=1}^{T}\epsilon_{i,t}^2 \le 2\,C\,d\,\ln\left(1+\frac{T}{d}\right)$, holding for all $i \in [K]$. Denoting by M the quantity $2\,C\,d\,\ln\left(1+\frac{T}{d}\right)$, we conclude from (14) that

$$R_T \le 2\,(1-a)\,c_L\,\max\left\{\sum_{i\in[K]}\sum_{t=1}^{T}\epsilon_{i,t}\,\middle|\,\sum_{t=1}^{T}\epsilon_{i,t}^2 \le M,\;\; i\in[K]\right\} = 2\,(1-a)\,c_L\,K\,\sqrt{T\,M},$$

as claimed. ■

[14] It is in this chain of inequalities that we exploit the rank-one update of $A_{i,t-1}$ based on $\boldsymbol{x}_t\boldsymbol{x}_t^\top$ rather than $\nabla_{i,t}\nabla_{i,t}^\top$. Here we need to lower bound the eigenvalue of the rank-one matrix used in the update. Using the $\nabla_{i,t}\nabla_{i,t}^\top$ (as in the worst-case analysis by Hazan et al. 2007), the lower bound would be zero. This is due to the presence of the multiplicative factor $g(s_{i,t}\widehat{\Delta}'_{i,t})$ (Step 5 in Figure 2) which can be arbitrarily small.

Proof [Theorem 4] As we said, we change the definition of $\epsilon^2_{i,t}$ in the Algorithm in Figure 2 to

$$\epsilon^2_{i,t} = \max\left\{ \boldsymbol{x}^\top A^{-1}_{i,t-1}\boldsymbol{x} \left(\frac{2\,d\,c'_L}{(c''_L)^2} \ln\left(1+\frac{t-1}{d}\right) + \frac{12}{c''_L}\left(\frac{c'_L}{c''_L} + 3L(-R)\right) \ln\frac{K(t+4)}{\delta}\right), 4\,R^2 \right\}.$$

First, notice that the $4R^2$ cap seamlessly applies, since $(\boldsymbol{x}^\top \boldsymbol{w}'_{i,t} - \boldsymbol{u}_i^\top \boldsymbol{x})^2$ in Lemma 13 is bounded by $4\,R^2$ anyway. With this modification, we have that Theorem 2 only holds for t such that $\frac{d\,c'_L}{(c''_L)^2} \ln\left(1+\frac{t-1}{d}\right) \geq U^2$, i.e., for $t \geq d\left(\exp\left(\frac{(c''_L)^2\,U^2}{c'_L\,d}\right)-1\right)+1$, while for $t < d\left(\exp\left(\frac{(c''_L)^2\,U^2}{c'_L\,d}\right)-1\right)+1$ we have in the worst-case scenario the maximum amount of regret at each step. From Lemma 11 we see that this maximum amount (the cap on $\epsilon^2_{i,t}$ is needed here) can be bounded by $4\,(1-a)\,c_L\,|\hat{Y}_t|\,R \leq 4\,(1-a)\,c_L\,K\,R$. ■

Proof [Theorem 7] We start from the one step-regret delivered by Lemma 12, and proceed as in the proof of Theorem 2. This yields

$$R_T \leq 4\,c_L \sum_{t=1}^T S_t \sum_{i\in\hat{Y}_t} \epsilon_{i,t} \leq 4\,S\,c_L \sum_{t=1}^T \sum_{i\in\hat{Y}_t} \epsilon_{i,t} \leq 4\,S\,c_L \sum_{t=1}^T \sum_{i\in[K]} \epsilon_{i,t} = 4\,S\,c_L \sum_{i\in[K]} \sum_{t=1}^T \epsilon_{i,t}\,,$$

with probability at least $1-\delta$, where $\epsilon^2_{i,t}$ is the one given in Figure 2. Let M be as in the proof of Theorem 2. We have that $\sum_{t=1}^T \epsilon^2_{i,t} \leq M$. If $N_{i,T}$ denotes the total number of times class i occurs in $\hat{Y}_t$, this implies $\sum_{t=1}^T \epsilon_{i,t} \leq \sqrt{N_{i,T}\,M}$ for all $i \in [K]$. Moreover, from $\sum_{i\in[K]} N_{i,T} \leq ST$ we can write

$$R_T \leq 4\,S\,c_L \sum_{i\in K]} \sqrt{N_{i,T}\,M} \leq 4\,c_L\,\sqrt{M\,S\,K\,T}\,,$$

as claimed. ■

7. Conclusions and Open Questions

In this paper, we have used generalized linear models to formalize the exploration-exploitation tradeoff in a multilabel/ranking setting with partial feedback, providing $T^{1/2}$-like regret bounds under semi-adversarial settings. Our analysis decouples the multilabel/ranking loss at hand from the label-generation model, improving in various ways on the existing literature. Thanks to the usage of calibrated score values $\widehat{p}_{i,t}$, our algorithm is capable of automatically inferring where to split the ranking between relevant and nonrelevant classes (Furnkranz et al., 2008), the split depending on the loss function under consideration. We considered two partial-feedback loss functions: $\ell_{a,c}$ and $\ell_{p-rank,t}$. The former can be seen as a Discounted Cumulative Gain difference, the latter a version of the standard (unnormalized) ranking loss, both being restricted to the chosen ranked list $\hat{Y}_t$. These two losses are inherently different: whereas $\ell_{p-rank,t}$ has a pairwise component, $\ell_{a,c}$ does not; whereas the Bayes optimal Y_t^* w.r.t. $\ell_{p-rank,t}$ has the maximal allowed length, the Bayes optimal Y_t^* w.r.t. $\ell_{a,c}$ need not be full length; whereas Bayes optimality solely based on $p_{i,t}$ does not require conditional independence assumptions when the loss is $\ell_{a,c}$, such condition is needed when the loss is $\ell_{p-rank,t}$. Yet, both losses depend in a similar fashion on the classes contained in $\hat{Y}_t$, as well as on the way such classes are ranked within $\hat{Y}_t$.

We have investigated the practically important case when $\hat{Y}_t$ has to satisfy length constraints $|\hat{Y}_t| \le S_t$, which is a typical prior knowledge in the presence of large multilabel action spaces. When $S_t \le S$ for all t, our regret bounds turn the linear dependence on K into a linear dependence on $\sqrt{SK}$.

Finally, we have presented experiments aimed at validating our upper-confidence-based ranking scheme against several real-world conditions and modeling assumptions.

There are many directions along which this work could be extended. In what follows, we briefly mention three of them.

- Multilabel and ranking algorithms are usually evaluated using an array of loss measures, including 0/1, Average Precision, F-measure, AUC, normalized ranking losses, etc. It would be nice to extend the theory contained in this paper to such measures. However, many of these losses are likely to require modeling pairwise correlations among classes.

- In the case when $S_t \le S$, we showed regret bounds of the form $\sqrt{SK}\,\sqrt{T}$. Is it possible to modify our theoretical arguments (possibly combining with the compressed sensing machinery used by Hsu et al. 2009) so as to obtain the information-theoretic bound $(S \log K)\,\sqrt{T}$, instead? Clearly enough, it would be most interesting to do so via computationally efficient algorithms.

- As a broader goal, it would be interesting to extend this theory to other practically relevant structured action spaces. For instance, an interesting extension is to the case when class labels $y_{i,t}$ are not binary, but real valued. Such values can in fact be the results of click aggregations over time. In this case, we may want to interpret Y_t as a ranked list as well, and come up with appropriate (partial-information) losses between pairs of such lists. Another interesting extension is to (multilabel) hierarchical classification. To this effect, the Bayes optimality arguments developed by Cesa-Bianchi et al. (2006a,b) may be of some relevance.

Acknowledgments

We would like to thank the anonymous reviewers for their constructive comments that helped us to improve the presentation of this paper. We would also like to thank Koby Crammer, Krzysztof Dembczyński, and Willem Waegeman for early discussions regarding this paper. The first author has been partially supported by the PASCAL2 Network of Excellence under EC grant 216886. This publication only reflects the authors views.

References

Y. Abbasi-Yadkori, D. Pal, and C. Szepesvári. Improved algorithms for linear stochastic bandits. In J. Shawe-Taylor, R.S. Zemel, P.L. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2312–2320. Curran Associates, Inc., 2011.

A. Agarwal. Selective sampling algorithms for cost-sensitive multiclass prediction. In *ICML (3)*, volume 28 of *JMLR Proceedings*, pages 1220–1228. JMLR.org, 2013.

K. Amin, M. Kearns, and U. Syed. Graphical models for bandit problems. In F.G. Cozman and A. Pfeffer, editors, *UAI*, pages 1–10. AUAI Press, 2011.

P. Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3:397–422, 2002.

K. S. Azoury and M. K. Warmuth. Relative loss bounds for online density estimation with the exponential family of distributions. *Machine Learning*, 43(3):211–246, 2001.

G. Bartók. A near-optimal algorithm for finite partial-monitoring games against adversarial opponents. In S. Shalev-Shwartz and I. Steinwart, editors, *COLT*, volume 30 of *JMLR Proceedings*, pages 696–710. JMLR.org, 2013.

G. Bartók and C. Szepesvári. Partial monitoring with side information. In N.H. Bshouty, G. Stoltz, N. Vayatis, and T. Zeugmann, editors, *ALT*, volume 7568 of *Lecture Notes in Computer Science*, pages 305–319. Springer, 2012.

W. Bi and J. Kwok. Multilabel classification on tree- and DAG-structured hierarchies. In L. Getoor and T. Scheffer, editors, *ICML*, pages 17–24. Omnipress, 2011.

D. Buffoni, C. Calauzènes, P. Gallinari, and N. Usunier. Learning scoring functions with order-preserving losses and standardized supervision. In L. Getoor and T. Scheffer, editors, *ICML*, pages 825–832. Omnipress, 2011.

G. Cavallanti, N. Cesa-Bianchi, and C. Gentile. Learning noisy linear classifiers via adaptive and selective sampling. *Machine Learning*, 83:71–102, 2011.

N. Cesa-Bianchi, A. Conconi, and C. Gentile. A second-order Perceptron algorithm. In J. Kivinen and R. H. Sloan, editors, *COLT*, volume 2375 of *Lecture Notes in Computer Science*, pages 121–137. Springer, 2002.

N. Cesa-Bianchi, C. Gentile, and L. Zaniboni. Incremental algorithms for hierarchical classification. *Journal of Machine Learning Research*, 7:31–54, 2006a.

N. Cesa-Bianchi, C. Gentile, and L. Zaniboni. Hierarchical classification: combining Bayes with SVM. In W.W. Cohen and A. Moore, editors, *ICML*, volume 148 of *ACM International Conference Proceeding Series*, pages 177–184. ACM, 2006b.

N. Cesa-Bianchi, C. Gentile, and F. Orabona. Robust bounds for classification via selective sampling. In A.P. Danyluk, L. Bottou, and M.L. Littman, editors, *ICML*, volume 382 of *ACM International Conference Proceeding Series*. ACM, 2009.

S. Clémençon, G. Lugosi, and N. Vayatis. Ranking and scoring using empirical risk minimization. In P. Auer and R. Meir, editors, *COLT*, volume 3559 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2005.

D. Cossock and T. Zhang. Subset ranking using regression. In G. Lugosi and H.-U. Simon, editors, *COLT*, volume 4005 of *Lecture Notes in Computer Science*, pages 605–619. Springer, 2006.

K. Crammer and C. Gentile. Multiclass classification with bandit feedback using adaptive regularization. In L. Getoor and T. Scheffer, editors, *ICML*, pages 273–280. Omnipress, 2011.

V. Dani, T. Hayes, and S. Kakade. Stochastic linear optimization under bandit feedback. In R.A. Servedio and T. Zhang, editors, *COLT*, pages 355–366. Omnipress, 2008.

O. Dekel, C. Gentile, and K. Sridharan. Selective sampling and active learning from single and multiple teachers. *Journal of Machine Learning Research*, 13:2655–2697, 2012.

K. Dembczynski, W. Waegeman, W. Cheng, and E. Hullermeier. On label dependence and loss minimization in multi-label classification. *Machine Learning*, 88:5–45, 2012.

J. C. Duchi, L. W. Mackey, and M. I. Jordan. On the consistency of ranking algorithms. In J. Frnkranz and T. Joachims, editors, *ICML*, pages 327–334. Omnipress, 2010.

A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In T.G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, volume 14, pages 681–687. MIT Press, 2002.

S. Filippi, O. Cappé, A. Garivier, and C. Szepesvári. Parametric bandits: The generalized linear case. In J.D. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 586–594. Curran Associates, Inc., 2010.

Y. Freund, R. D. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003.

J. Furnkranz, E. Hullermeier, E. Loza Menca, and K. Brinker. Multilabel classification via calibrated label ranking. *Machine Learning*, 73:133–153, 2008.

C. Gentile and F. Orabona. On multilabel classification and ranking with partial feedback. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1151–1159. Curran Associates, Inc., 2012.

E. Hazan and S. Kale. Beyond convexity: Online submodular minimization. In Y. Bengio, D. Schuurmans, J.D. Lafferty, C.K.I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 700–708. Curran Associates, Inc., 2009.

E. Hazan and S. Kale. Newtron: an efficient bandit algorithm for online multiclass prediction. In J. Shawe-Taylor, R.S. Zemel, P.L. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 891–899. Curran Associates, Inc., 2011.

E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69:169–192, 2007.

R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. In *Advances in Large Margin Classifiers, MIT Press*, 2000.

D. Hsu, S. Kakade, J. Langford, and T. Zhang. Multi-label prediction via compressed sensing. In Y. Bengio, D. Schuurmans, J.D. Lafferty, C.K.I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 772–780. Curran Associates, Inc., 2009.

K. Jarvelin and J. Kekalainen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20:422–446, 2002.

S. Kakade and A. Tewari. On the generalization ability of online strongly convex programming algorithms. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 801–808. Curran Associates, Inc., 2009.

S. Kakade, S. Shalev-Shwartz, and A. Tewari. Efficient bandit algorithms for online multiclass prediction. In W.W. Cohen, A. McCallum, and S.T. Roweis, editors, *ICML*, volume 307 of *ACM International Conference Proceeding Series*, pages 440–447. ACM, 2008.

S. Kale, L. Reyzin, and R. Schapire. Non-stochastic bandit slate problems. In J.D. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1054–1062. Curran Associates, Inc., 2010.

A. Krause and C. S. Ong. Contextual Gaussian process bandit optimization. In J. Shawe-Taylor, R.S. Zemel, P.L. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2447–2455. Curran Associates, Inc., 2011.

T. H. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Adv. Appl. Math.*, 6, 1985.

Y. Lan, J. Guo, X. Cheng, and T. Liu. Statistical consistency of ranking methods in a rank-differentiable probability space. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1232–1240. Curran Associates, Inc., 2012.

J. Langford and T. Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In J.C. Platt, D. Koller, Y. Singer, and S.T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 817–824. Curran Associates, Inc., 2008.

P. McCullagh and J.A. Nelder. *Generalized Linear Models*. Chapman and Hall, 1989.

F. Orabona and N. Cesa-Bianchi. Better algorithms for selective sampling. In L. Getoor and T. Scheffer, editors, *ICML*, pages 433–440. Omnipress, 2011.

F. Pachet and P. Roy. Improving multilabel analysis of music titles: A large-scale validation of the correction approach. *IEEE Trans. on Audio, Speech, and Lang. Proc.*, 17(2):335–343, 2009.

L.J. Savage. Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association*, 336:783–801, 1973.

P. Shivaswamy and T. Joachims. Online structured prediction via coactive learning. In *ICML*. icml.cc / Omnipress, 2012.

A. Sklar. Fonctions de répartition à n dimensions et leurs marges. *Publ. Inst. Statist. Univ. Paris*, 8:229–231, 1959.

A. Slivkins, F. Radlinski, and S. Gollapudi. Learning optimally diverse rankings over large document collections. pages 983–990. Omnipress, 2010.

C.G.M. Snoek, M. Worring, J.C. van Gemert, J.-M. Geusebroek, and A.W.M. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In K. Nahrstedt, M. Turk, Y. Rui, W. Klas, and K. Mayer-Patel, editors, *ACM Multimedia*, pages 421–430. ACM, 2006.

I. Steinwart. Support vector machines are universally consistent. *J. Complexity*, 18(3): 768–791, 2002.

M. Streeter, D. Golovin, and A. Krause. Online learning of assignments. In Y. Bengio, D. Schuurmans, J.D. Lafferty, C.K.I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1794–1802. Curran Associates, Inc., 2009.

G. Tsoumakas, I. Katakis, and I. Vlahavas. Random k-labelsets for multilabel classification. *IEEE Transactions on Knowledge and Data Engineering*, 23:1079–1089, 2011.

Y. Wang, R. Khardon, D. Pechyony, and R. Jones. Generalization bounds for online learning algorithms with pairwise loss functions. In S. Mannor, N. Srebro, and R.C. Williamson, editors, *COLT*, volume 23 of *JMLR Proceedings*, pages 13.1–13.22. JMLR.org, 2012.