A Unified Algorithmic Approach for Efficient Online Label Ranking

Shai Shalev-Shwartz[1] **and Yoram Singer**[1,2]
[1] School of Computer Sci. & Eng., The Hebrew University, Jerusalem 91904, Israel
[2] Google Inc., 1600 Amphitheater Parkway, Mountain View, CA 94043, USA

Abstract

Label ranking is the task of ordering labels with respect to their relevance to an input instance. We describe a unified approach for the online label ranking task. We do so by casting the online learning problem as a game against a competitor who receives all the examples in advance and sets its label ranker to be the optimal solution of a constrained optimization problem. This optimization problem consists of two terms: the empirical label-ranking loss of the competitor and a complexity measure of the competitor's ranking function. We then describe and analyze a framework for online label ranking that *incrementally* ascends the dual problem corresponding to the competitor's optimization problem. The generality of our framework enables us to derive new online update schemes. In particular, we use the relative entropy as a complexity measure to derive efficient multiplicative algorithms for the label ranking task. Depending on the specific form of the instances, the multiplicative updates either have a closed form or can be calculated very efficiently by tailoring an interior point procedure to the label ranking task. We demonstrate the potential of our approach in a few experiments with email categorization tasks.

1 Introduction and Problem Setting

Label ranking is concerned with the task of ordering labels which are associated with a given instance in accordance to their relevance to the input instance. In this paper we describe an algorithmic framework for *online* label ranking. As an illustrative example consider an email categorization task in which the instances are email messages. Most email applications allow users to organize email massages into user-defined folders. For example, Google's Gmail users can tag each of their email messages with one or more labels. The set of labels is also user defined yet it is finite and typically constitutes of a few tens if not hundreds of different labels. The benefit of this approach is the flexibility it gives users categorizing emails, since an email may be associated with multiple labels. The approach stands in contrast to traditional systems in which an email is associated with a *single* physical folder. Users may browse all emails associated with a particular label and can also use the labels for searching through their emails. However, manually attaching the relevant labels to each incoming email message can be an all-out effort. An online label ranking algorithm automatically learns how to rank-order labels in accordance with their relevance to each of the incoming email messages. Quite a few learning algorithms have been devised for the category ranking problem such as a multiclass version of AdaBoost called AdaBoost.MH [10], a generalization of Vapnik's Support Vector Machines to the multilabel setting by Elisseeff and Weston [5], and generalizations of the Perceptron algorithm to category ranking [3, 2].

The category ranking hypotheses this work employs are closely related to the ones presented and used in [5, 3, 4]. However, we depart from the standard paradigm which is confined to a specific form of regularization and give a unified account for online learning for label ranking problems. Our starting point is the primal-dual perspective we initially presented in [12] and further developed in [13]. Following [12, 13], we cast the problem as an optimization problem which is solved incrementally as the online learning progresses. We then switch to the dual representation of the problem and show that by modifying only the set of variables corresponding to the example received on a given trial of the online algorithm we are able to obtain two goals. First, we devise a general procedure and derive specific online update schemes. Second, we use the primal-dual view in conjunction with the weak-duality theorem to obtain a general mistake bound for the label ranking problem. We demonstrate the power of our approach by deriving new updates for the label ranking task. In particular, we devise new multiplicative updates which outperform additive updates in the experiments reported in this paper.

Before proceeding with a formal description of the problem setting we would like to underscore the contribution of this work in the light of our previous theoretical work. The work presented here provides new updates for the label ranking problem. While our mistake bound analysis is based on previous work [12, 13], the former was mostly confined either to binary classification or to regret analysis for general functions. More importantly, we describe new specific algorithms for entropic regularization. Interestingly, our formal analysis is on par with the experimental results which give further validation to the formal results presented in [12, 13] and in this paper.

We start by formally describing the online label ranking problem. Let $\mathcal{X} \subset \mathbb{R}^n$ be an instance domain and let $\mathcal{Y} = \{1, \ldots, k\}$ be a predefined set of labels. Online learning is performed in a sequence of trials. On trial t the algorithm first receives an instance $\mathbf{x}^t \in \mathcal{X}$ and is required to rank the labels in $\mathcal{Y}$ according to their relevance to the instance $\mathbf{x}^t$. For simplicity, we assume that the predicted ranking is given in the form of a vector $\boldsymbol{\rho}^t \in \mathbb{R}^k$, where $\rho^t_r > \rho^t_s$ means that label r is ranked ahead of label s. After the online learning algorithm has predicted the ranking $\boldsymbol{\rho}^t$ it receives as feedback a subset of labels $Y^t \subseteq \mathcal{Y}$, which are mostly relevant to $\mathbf{x}^t$. We say that the ranking predicted by the algorithm is correct if all the labels in Y^t are at the top of the list. That is, if for all $r \in Y^t$ and $s \notin Y^t$ we have that $\rho^t_r > \rho^t_s$. Otherwise, if there exist $r \in Y^t$ and $s \notin Y^t$ for which $\rho^t_r \leq \rho^t_s$, we say that the algorithm made a prediction mistake on trial t. The ultimate goal of the algorithm is to minimize the total number of prediction mistakes it makes along its run. Throughout this paper, we use M to denote the number of prediction mistakes made by an online algorithm on a sequence of examples $(\mathbf{x}^1, Y^1), \ldots, (\mathbf{x}^m, Y^m)$.

We assume that the prediction of the algorithm at each trial is determined by a linear function which is parameterized by k weight vectors, $\{\boldsymbol{\omega}^t_1, \ldots, \boldsymbol{\omega}^t_k\}$. Namely, for all $r \in \mathcal{Y}$, the value of ρ^t_r is the inner product between $\boldsymbol{\omega}^t_r$ and $\mathbf{x}^t$, that is, $\rho^t_r = \langle \boldsymbol{\omega}^t_r, \mathbf{x}^t \rangle$. We use the notation $\bar{\boldsymbol{\omega}}^t$ as an abbreviation for the set $\{\boldsymbol{\omega}^t_1, \ldots, \boldsymbol{\omega}^t_k\}$. To evaluate the performance of $\bar{\boldsymbol{\omega}}^t$ on the example $(\mathbf{x}^t, Y^t)$ we check whether $\bar{\boldsymbol{\omega}}^t$ made a prediction mistake, by determining whether for all $r \in Y^t$ and $s \notin Y^t$ we have $\langle \boldsymbol{\omega}^t_r, \mathbf{x}^t \rangle > \langle \boldsymbol{\omega}^t_s, \mathbf{x}^t \rangle$. To obtain bounds on prediction mistakes we use a second evaluation scheme of the performance of $\bar{\boldsymbol{\omega}}^t$. This scheme is based on a generalization of the *hinge-loss* function, denoted $\ell^\gamma(\bar{\boldsymbol{\omega}}^t; (\mathbf{x}^t, Y^t))$, for ranking problems, defined as,

$$\max_{r \in Y^t, s \notin Y^t} \left[\gamma - \left(\langle \boldsymbol{\omega}^t_r, \mathbf{x}^t \rangle - \langle \boldsymbol{\omega}^t_s, \mathbf{x}^t \rangle\right)\right]_+ \ , \qquad (1)$$

where $[a]_+ = \max\{a, 0\}$ and where $\gamma > 0$ is a predefined parameter. The above definition of loss extends the hinge-loss used in binary classification problems [14] to the problem of label-ranking. The term $\langle \boldsymbol{\omega}^t_r, \mathbf{x}^t \rangle - \langle \boldsymbol{\omega}^t_s, \mathbf{x}^t \rangle$ in the definition of the hinge-loss is a generalization of the notion of *margin* from binary classification. The hinge-loss penalizes $\bar{\boldsymbol{\omega}}^t$ for any margin less than γ. Additionally, if $\bar{\boldsymbol{\omega}}^t$ errs on $(\mathbf{x}^t, Y^t)$ then there exist $r \in Y^t$ and $s \notin Y^t$ such that $\langle \boldsymbol{\omega}^t_r, \mathbf{x}^t \rangle - \langle \boldsymbol{\omega}^t_s, \mathbf{x}^t \rangle \leq 0$ and thus $\ell^\gamma(\bar{\boldsymbol{\omega}}^t; (\mathbf{x}^t, Y^t)) \geq \gamma$. Thus, the *cumulative hinge-loss* suffered over a sequence of examples upper bounds γM.

To obtain a concrete online learning algorithm we must determine the initial value of each weight vector and an update rule used to modify the weight vectors at the end of each trial. Recall that our goal is to derive online learning algorithms which make small number of prediction mistakes. Naturally, without further assumptions on the sequence of examples, any online learning algorithm can be forced to make a large number of mistakes. To state a more realistic goal, we follow the relative mistake bound model, and measure the performance of an online learning algorithm relatively to the performance of *any* fixed set of weight vectors $\bar{\boldsymbol{\omega}}^\star = \{\boldsymbol{\omega}^\star_1, \ldots, \boldsymbol{\omega}^\star_k\} \in \Omega^k$, where $\Omega \subset \mathbb{R}^n$ is a set of admissible vectors. The competitor $\bar{\boldsymbol{\omega}}^\star$ can be chosen in hindsight after observing the entire sequence of examples. In particular, if for a given sequence of examples, $(\mathbf{x}^1, Y^1), \ldots, (\mathbf{x}^m, Y^m)$, there exists a competitor $\bar{\boldsymbol{\omega}}^\star$ for which $\sum_{t=1}^m \ell^\gamma(\bar{\boldsymbol{\omega}}^\star; (\mathbf{x}^t, Y^t)) = 0$, then we would like M, the number of prediction mistakes of the online algorithm, to be independent of m. The requirement we cast is that M is upper bounded by $\mathbf{F}(\bar{\boldsymbol{\omega}}^\star)$. We make the assumption that $\mathbf{F}$, which operates on set of vectors $\bar{\boldsymbol{\omega}}$, is obtained by applying the same convex function to each of the constituents of $\bar{\boldsymbol{\omega}}$. Formally, given $F : \Omega \to \mathbb{R}$, which assesses the "complexity" of a single vector $\boldsymbol{\omega}$, and $\bar{\boldsymbol{\omega}} = \{\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_k\}$, we define $\mathbf{F}(\bar{\boldsymbol{\omega}})$ to be $\sum_{r=1}^k F(\boldsymbol{\omega}_r)$.

In the general case, we provide analysis in which M is upper-bounded by a sum of two terms: the first is the complexity of $\bar{\boldsymbol{\omega}}^\star$ as defined by $\mathbf{F}(\bar{\boldsymbol{\omega}}^\star)$ and the second is the cumulative hinge-loss suffered by $\bar{\boldsymbol{\omega}}^\star$. Formally, let λ and C be two positive scalars. We say that an online algorithm is (λ, C)-competitive with the set of vectors in Ω, with respect to a complexity function $\mathbf{F}$ and the hinge-loss ℓ^γ, if the following bound holds for any $\bar{\boldsymbol{\omega}}^\star \in \Omega^k$,

$$\lambda\, M \ \leq \ \mathbf{F}(\bar{\boldsymbol{\omega}}^\star) + C \sum_{t=1}^m \ell^\gamma(\bar{\boldsymbol{\omega}}^\star; (\mathbf{x}^t, Y^t)) \ . \qquad (2)$$

The parameter C controls the trade-off between the complexity of $\bar{\boldsymbol{\omega}}^\star$ (through F) and the cumulative hinge-loss of $\bar{\boldsymbol{\omega}}^\star$. The main goal of this paper is to develop efficient online learning algorithms which achieve mistake bounds of the form given in Eq. (2).

2 Online Learning by Dual Ascent

We now describe our approach for designing and analyzing online learning algorithms for label ranking. To motivate our construction, we would like to note first that the bound in Eq. (2) can be rewritten as,

$$\lambda\, M \ \leq \ \inf_{\bar{\boldsymbol{\omega}} \in \Omega^k} \ \mathbf{F}(\bar{\boldsymbol{\omega}}) + C \sum_{t=1}^m \ell^\gamma(\bar{\boldsymbol{\omega}}; (\mathbf{x}^t, Y^t)) \ . \qquad (3)$$

Denote by $\mathcal{P}(\bar{\boldsymbol{\omega}})$ the objective function of the minimization problem given on the right-hand side of Eq. (3). We would like to emphasize that $\mathcal{P}(\bar{\boldsymbol{\omega}})$ depends on the entire sequence of examples $\{(\mathbf{x}^1, Y^1), \ldots, (\mathbf{x}^m, Y^m)\}$ and therefore the minimization problem can only be solved in hindsight, that is, after observing the entire sequence of examples. Eq. (3) requires that λM lower bounds the optimum of the minimization problem $\min_{\bar{\boldsymbol{\omega}}} \mathcal{P}(\bar{\boldsymbol{\omega}})$.

Duality, which is a central notion in optimization theory, plays an important role in obtaining lower bounds for the minimal value of a minimization problem (see for example [1]). Formally, if the dual problem of $\min_{\bar{\boldsymbol{\omega}}} \mathcal{P}(\bar{\boldsymbol{\omega}})$ is to maximize a dual objective function $\mathcal{D}(\boldsymbol{\tau})$ over a dual domain $\boldsymbol{\tau} \in S$ then the weak duality theorem states that for any $\boldsymbol{\tau} \in S$ we have, $\mathcal{D}(\boldsymbol{\tau}) \le \inf_{\bar{\boldsymbol{\omega}} \in \Omega^k} \mathcal{P}(\bar{\boldsymbol{\omega}})$. Before we derive the dual function $\mathcal{D}(\boldsymbol{\tau})$ and its domain S, let us first underscore the implications of the weak duality theorem for online learning. Let $\mathcal{M}$ be the set of trials on which the online algorithm makes a prediction mistake. Assume that we can associate a feasible dual solution $\boldsymbol{\tau}^t$ with each trial $1 \le t \le m+1$ that satisfies the following two requirements: (i) The initial dual objective function is zero, $\mathcal{D}(\boldsymbol{\tau}^1) = 0$. (ii) For all $t \in \mathcal{M}$, the increase in the dual objective value is bounded from below: $\mathcal{D}(\boldsymbol{\tau}^{t+1}) - \mathcal{D}(\boldsymbol{\tau}^t) \ge \lambda$, and for $t \notin \mathcal{M}$ the increase in the dual is non-negative. Assuming that these two conditions are met we obtain that,

$$\begin{aligned} \lambda M \;&\le\; \mathcal{D}(\boldsymbol{\tau}^1) + \sum_{t=1}^{m} \left(\mathcal{D}(\boldsymbol{\tau}^{t+1}) - \mathcal{D}(\boldsymbol{\tau}^t)\right) \\ &=\; \mathcal{D}(\boldsymbol{\tau}^{m+1}) \;\le\; \inf_{\bar{\boldsymbol{\omega}} \in \Omega^k} \mathcal{P}(\bar{\boldsymbol{\omega}}) \;\;, \end{aligned}$$

where the last inequality follows from the weak duality theorem. Therefore, Eq. (3) holds and our online algorithm is (λ, C)–competitive.

We have thus shown that any sequence of feasible dual solutions $\boldsymbol{\tau}^1, \ldots, \boldsymbol{\tau}^{m+1}$ can be used for analyzing an online learning algorithm so long as the two conditions described above hold. As we show in the sequel, the first requirement is easily satisfied. For the second requirement to hold, we must relate the sequence of dual solutions to the predictions of the algorithm. Recall that an online learning algorithm constructs a sequence of primal solutions $\bar{\boldsymbol{\omega}}^1, \ldots, \bar{\boldsymbol{\omega}}^{m+1}$ where on trial t it employs $\bar{\boldsymbol{\omega}}^t$ for predicting $\boldsymbol{\rho}^t$. One way to connect between the sequence of dual solutions to the predictions of the algorithm can be achieved by constructing each primal solution $\bar{\boldsymbol{\omega}}^t$ from its dual variable $\boldsymbol{\tau}^t$. To explain this construction we first need to derive the dual optimization problem of the minimization problem given on the right-hand side of Eq. (3).

To simplify our notation, we denote by $[m]$ the set of integers $\{1, \ldots, m\}$. We also use the notation E^i as a shorthand for the set $Y^i \times (\mathcal{Y} \setminus Y^i)$. The dual problem associates a variable $\tau_{i,r,s}$ with each example $i \in [m]$ and pair $(r,s) \in E^i$. A fairly tedious yet routine usage of Lagrange multipliers yields that the dual domain is, $S = \{\boldsymbol{\tau} \ge \mathbf{0} : \forall i \in [m], \sum_{(r,s)\in E^i} \tau_{i,r,s} \le C\}$. Our expression for the dual objective function is based on a function $G(\boldsymbol{\theta})$ which is the Fenchel conjugate of the complexity function F,

$$G(\boldsymbol{\theta}) \;=\; \sup_{\boldsymbol{\omega}\in\Omega} \;\langle \boldsymbol{\omega}, \boldsymbol{\theta} \rangle - F(\boldsymbol{\omega}) \;\; . \tag{4}$$

The dual objective function is

$$\mathcal{D}(\boldsymbol{\tau}) \;=\; \gamma \sum_{i=1}^{m} \sum_{(r,s)\in E^i} \tau_{i,r,s} \;-\; \sum_{y=1}^{k} G\left(\boldsymbol{\theta}_y\right) \;\; , \tag{5}$$

where for all $y \in [k]$,

$$\boldsymbol{\theta}_y \;=\; \sum_{i: y \in Y^i} \mathbf{x}^i \sum_{s \notin Y^i} \tau_{i,y,s} - \sum_{i: y \notin Y^i} \mathbf{x}^i \sum_{r \in Y^i} \tau_{i,r,y} \;\; . \tag{6}$$

To conclude this section, we would like to underscore a few important properties of the dual problem given by Eq. (5). Recall that we would like to associate a dual solution $\boldsymbol{\tau}^t$ with each online trial and we required that $\mathcal{D}(\boldsymbol{\tau}^1) = 0$ and that $\mathcal{D}(\boldsymbol{\tau}^{t+1}) - \mathcal{D}(\boldsymbol{\tau}^t) \ge \lambda$ whenever $t \in \mathcal{M}$. To satisfy the first requirement we show that a natural choice for $\boldsymbol{\tau}^1$ is the zero vector. The zero vector, $\boldsymbol{\tau}^1 = \mathbf{0}$, is indeed a feasible dual solution and the value of the dual function at $\mathbf{0}$ is, $\mathcal{D}(\mathbf{0}) = -k\, G(\mathbf{0}) = k \inf_{\boldsymbol{\omega}\in\Omega} F(\boldsymbol{\omega})$. Since F serves as a measure of "complexity" of vectors in Ω we enforce the requirement that the minimum of F over the vectors in Ω is zero. From this requirement we get that $\mathcal{D}(\mathbf{0}) = 0$, and therefore, by setting $\boldsymbol{\tau}^1 = \mathbf{0}$, we satisfy the requirement $\mathcal{D}(\boldsymbol{\tau}^1) = 0$. The second important property of the dual function given by Eq. (5) is that if for all triplets (i,r,s) such that $i > t$ we have $\tau_{i,r,s} = 0$ then $\mathcal{D}(\boldsymbol{\tau})$ does not depend on *yet to be seen* examples $(\mathbf{x}^{t+1}, Y^{t+1}), \ldots, (\mathbf{x}^m, Y^m)$. This fact is true since the dependence of $\mathcal{D}$ on examples is expressed solely through the vectors $\boldsymbol{\theta}_y$ (see Eq. (6)) and $\boldsymbol{\theta}_y$ is a linear combination of the examples with coefficients $\tau_{i,r,s}$. Therefore, if $\tau_{i,r,s} = 0$ for all $i > t$ then $\boldsymbol{\theta}_y$ is independent of $\mathbf{x}^i$ for all $i > t$. This simple property allows us to devise different online update procedures which increase sufficiently the dual objective function.

3 Derived Online Updates

In this section we present three different online update schemes that are based on the same principle of ascending the dual by modifying solely the dual variables corresponding to the tth trial. We start by setting $\boldsymbol{\tau}^1 = \mathbf{0}$ which results in a zero value for $\mathcal{D}(\boldsymbol{\tau}^1)$. For $t = 1$ we have that $\tau^t_{i,r,s} = 0$ for all $i \ge t$. We keep ensuring that the property holds for all $t \in [m]$. At the beginning of trial t, we construct a primal solution, $\bar{\boldsymbol{\omega}}^t$, based on $\boldsymbol{\tau}^t$ as follows. For simplicity, let us assume that the function G is differentiable and denote its gradient by g. We first use $\boldsymbol{\tau}^t$ for defining the set of vectors $\{\boldsymbol{\theta}_1^t, \ldots, \boldsymbol{\theta}_k^t\}$ as in Eq. (6). We then define $\boldsymbol{\omega}_y^t = g(\boldsymbol{\theta}_y^t)$ for all $y \in [k]$. Note that since $\tau_{i,r,s}^t = 0$ for all $i \geq t$ we have that $\boldsymbol{\theta}_y^t$ and $\boldsymbol{\omega}_y^t$ are independent of the examples $(\mathbf{x}^t, Y^t), \ldots, (\mathbf{x}^m, Y^m)$. Next, we use the set of vectors $\bar{\boldsymbol{\omega}}^t = \{\boldsymbol{\omega}_1^t, \ldots, \boldsymbol{\omega}_k^t\}$ for predicting the label ranking $\boldsymbol{\rho}^t$. Finally, after receiving the feedback Y^t, we find a new dual solution $\boldsymbol{\tau}^{t+1}$ by setting $\tau_{t,r,s}^{t+1}$ where for all $i \geq t+1$ the variables $\tau_{i,r,s}^{t+1}$ are kept at zero. While the three update schemes described in the sequel are based on this approach they vary in their complexity. As we have shown before, an online learning algorithm is (λ, C)–competitive, if on all trials $t \in \mathcal{M}$ we ensure a minimal increase in the dual $\mathcal{D}(\boldsymbol{\tau}^{t+1}) - \mathcal{D}(\boldsymbol{\tau}^t) \geq \lambda$ while on the rest of the trials we ensure that the dual is non-decreasing. Despite the varying complexity of the updates, all the three of them satisfy these conditions and achieve the same mistake bound. However, the added complexity does result in improved empirical performance (see Sec. 5).

Update I: fixed-size dual ascent w.r.t. a single constraint The first update we consider makes a simple predefined change to one variable of $\boldsymbol{\tau}$ at the end of each erroneous trial. Formally, if on trial t the algorithm did not make a prediction mistake we do not change $\boldsymbol{\tau}$ at all and set $\boldsymbol{\tau}^{t+1} = \boldsymbol{\tau}^t$. If there was a prediction error we let,

$$(r', s') \ = \ \underset{(r,s) \in E^t}{\operatorname{argmin}} \ \langle \boldsymbol{\omega}_{r'}^t - \boldsymbol{\omega}_{s'}^t, \mathbf{x}^t \rangle \ . \qquad (7)$$

That is, the pair (r', s') designates the labels which mostly violate the required preference constraints. Since there was a prediction mistake, we get that $\langle \boldsymbol{\omega}_{r'}^t - \boldsymbol{\omega}_{s'}^t, \mathbf{x}^t \rangle \leq 0$. We now set the (t, r', s') element of $\boldsymbol{\tau}$ to C and leave the rest of the elements intact. Formally, for $t \in \mathcal{M}$ the new vector $\boldsymbol{\tau}^{t+1}$ is set as follows,

$$\tau_{i,r,s}^{t+1} \ = \ \begin{cases} C & \text{if } (i,r,s) = (t,r',s') \\ \tau_{i,r,s}^t & \text{otherwise} \end{cases} \qquad (8)$$

This form of update implies that the components of $\boldsymbol{\tau}$ are either zero or C. Using the definition of $\boldsymbol{\theta}_y$ given in Eq. (6) we get that the corresponding update of $\boldsymbol{\theta}_y$ is,

$$\begin{aligned} &\boldsymbol{\theta}_{r'}^{t+1} = \boldsymbol{\theta}_{r'}^t + C\,x^t \ \ , \quad \boldsymbol{\theta}_{s'}^{t+1} = \boldsymbol{\theta}_{s'}^t - C\,x^t \ \ , \\ &\forall y \in \mathcal{Y} - \{r', s'\} : \ \boldsymbol{\theta}_y^{t+1} = \boldsymbol{\theta}_y^t \ \ . \end{aligned}$$

By construction, if $t \notin \mathcal{M}$ then $\mathcal{D}(\boldsymbol{\tau}^{t+1}) - \mathcal{D}(\boldsymbol{\tau}^t) = 0$. The lemma below gives sufficient conditions for which the increase in the dual objective on erroneous trials is strictly positive.

Lemma 1 *Let $\boldsymbol{\tau}^t \in S$ be a dual solution such that $\tau_{i,r,s}^t = 0$ for all $i \geq t$. Assume that $t \in \mathcal{M}$ and let $\boldsymbol{\tau}^{t+1}$ be as defined in Eq. (8). Assume in addition that G is twice differentiable with a Hessian H which satisfies the condition that $\langle \mathbf{x}^t, H(\boldsymbol{\theta})\mathbf{x}^t \rangle \leq 1/2$ for all $\boldsymbol{\theta}$. Then the increase in the dual, $\mathcal{D}(\boldsymbol{\tau}^{t+1}) - \mathcal{D}(\boldsymbol{\tau}^t)$, for $t \in \mathcal{M}$ is at least, $\gamma\,C - \frac{1}{2}\,C^2$.*

The proof is given in Appendix A. In summary, we have shown that $\mathcal{D}(\boldsymbol{\tau}^1) = 0$, $\mathcal{D}(\boldsymbol{\tau}^{t+1}) - \mathcal{D}(\boldsymbol{\tau}^t) \geq 0$ for all t, and $\mathcal{D}(\boldsymbol{\tau}^{t+1}) - \mathcal{D}(\boldsymbol{\tau}^t) \geq \lambda$ for $t \in \mathcal{M}$ where $\lambda = \gamma\,C - \frac{1}{2}\,C^2$. Therefore, the resulting online learning algorithm is (λ, C)–competitive.

Update II: optimal dual ascent w.r.t. a single constraint The previous update scheme modifies $\boldsymbol{\tau}$ only on trials for which there was a prediction mistake ($t \in \mathcal{M}$). The update is performed by setting $\tau_{t,r',s'}$ to C and keeping the rest of the variables intact. We now enhance this update in two ways. First, note that while setting $\tau_{t,r',s'}^{t+1}$ to C guarantees a sufficient increase in the dual, there might be other values of $\tau_{t,r',s'}^{t+1}$ which might lead to larger increases of the dual objective. Furthermore, we can also update $\boldsymbol{\tau}$ on trials on which the prediction was correct so long as the dual does not decrease. Our second update sets $\tau_{t,r',s'}^{t+1}$ to be the value which results in the largest increase in the dual objective. Formally, we set the dual variables on the next trial to be the solution of the following,

$$\begin{aligned} \boldsymbol{\tau}^{t+1} = \ \ &\underset{\boldsymbol{\tau} \in S}{\operatorname{argmax}} \ \ \mathcal{D}(\boldsymbol{\tau}) \\ &\text{s.t.} \ \ \forall (i,r,s) \neq (t,r',s'), \ \tau_{i,r,s} = \tau_{i,r,s}^t \ \ , \end{aligned} \qquad (9)$$

where (r', s') are as defined in Eq. (7). By construction, the increase in the dual due to the update given in Eq. (9) is guaranteed to be at least as large as the increase due to the previous update from Eq. (8). Thus, this update scheme results in an online algorithm which is also (λ, C)–competitive with λ again being equal to $\gamma\,C - \frac{1}{2}\,C^2$.

Update III: optimal dual ascent w.r.t multiple constraints The two updates above were restricted to modifying a *single* element of $\boldsymbol{\tau}$ and were thus been based on a single constraint of the primal problem. the third update scheme potentially modifies all the dual variables of the current example. Formally, we define $\boldsymbol{\tau}^{t+1}$ to be the solution of the following optimization problem,

$$\begin{aligned} \boldsymbol{\tau}^{t+1} = \ \ &\underset{\boldsymbol{\tau} \in S}{\operatorname{argmax}} \, \mathcal{D}(\boldsymbol{\tau}) \\ &\text{s.t.} \ \ \forall i \neq t, \ \forall (r,s) \in E^i, \ \tau_{i,r,s} = \tau_{i,r,s}^t \ \ . \end{aligned} \qquad (10)$$

The increase in the dual due to this update is no smaller than the increase due to the update from Eq. (8). Thus, this update scheme results in an online algorithm which is also (λ, C)–competitive with λ again being equal to $\gamma\,C - \frac{1}{2}\,C^2$.

4 Efficient Implementations

Update II and III described above are given in an implicit form as a solution for reduced optimization problems as described by Eq. (9) and Eq. (10). In this section we describe efficient implementations of the two updates. For update II we derive analytic solutions for two popular choices of F while for update III we describe a general solver based on an interior point (IP) method which exploits the structure of the label ranking problem. The result is a specialized algorithm which is more efficient than general IP methods.

An efficient implementation of Update II When $\mathbf{F}(\bar{\boldsymbol{\omega}}) = \sum_r F(\boldsymbol{\omega}_r)$ where $F(\boldsymbol{\omega}_r) = \frac{1}{2}\|\boldsymbol{\omega}_r\|^2$, standard use of Lagrange multipliers yields that $\tau_{t,r',s'}^{t+1}$ is the minimum between C and $(\gamma + \langle \boldsymbol{\omega}_{s'}^t - \boldsymbol{\omega}_{r'}^t, \mathbf{x}^t \rangle)/(2\|\mathbf{x}^t\|^2)$. We would like to note that this form of update was suggested and analyzed by several authors for the simple case of binary classification [7] and multiclass problems [2]. In the following we show that the update can be utilized with the less studied case in which $F(\boldsymbol{\omega}_r) = \sum_{j=1}^n w_{r,j} \log(w_{r,j}/(1/n))$. To devise an analytic solution, we further assume that each instance $\mathbf{x}^t$ is a binary vector in $\{0,1\}^n$. While this assumption seems restrictive, many text processing applications use term appearances as features which distill to binary vectors. In this case the conjugate of F is $G(\boldsymbol{\theta}) = \log\left(\sum_{j=1}^n \exp(\theta_j)\right) - \log(n)$. Upon selecting the label pair (r,s), the change in the dual due to update II is a scalar function in $\tau_{t,r,s}^{t+1}$ which we simply abbreviate by τ. Omitting terms which do not depend on $\tau_{t,r,s}^{t+1}$ this change in the dual amounts to, $\Delta_t = \gamma\tau - G(\boldsymbol{\theta}_r^t + \tau\,\mathbf{x}^t) - G(\boldsymbol{\theta}_s^t - \tau\,\mathbf{x}^t)$. Since the original dual objective is concave in its dual variables, the change in the dual is also a concave function in τ. Furthermore, τ resides in the compact interval $[0, C]$ and thus there exists a unique value of τ which maximizes the increase in the dual. To find this optimal value we introduce the following auxiliary functions, $q_r = \frac{1}{Z_r}\sum_{j:x_j=1} e^{\theta_{r,j}}$; $Z_r = \sum_{j=1}^n e^{\theta_{r,j}}$ and $q_s = \frac{1}{Z_s}\sum_{j:x_j=1} e^{\theta_{s,j}}$; $Z_s = \sum_{j=1}^n e^{\theta_{s,j}}$. Equipped with these definitions we now take the derivative of Δ_t with respect to τ and equate it to zero to get that,

$$\gamma - \frac{q_r e^{\tau}}{q_r e^{\tau} + (1-q_r)} + \frac{q_s e^{-\tau}}{q_s e^{-\tau} + (1-q_s)} = 0 \; .$$

Defining $\beta = e^{\tau}$ we get the quadratic equation, $\beta^2 q_r(1-q_s)(1-\gamma) - \beta\gamma(q_r q_s + (1-q_r)(1-q_s)) - q_s(1-q_r)(1+\gamma) = 0$, Since β must be non-negative, the minimum between the positive root of the above equation and C gives the optimal value for β. From β we obtain τ by setting $\tau = \log(\beta)$.

An Efficient Implementation of Update III While the first two update schemes use only a *single* variable to form the update, the third update scheme employs the *entire* set of variables associated with the example. Recall that each example is associated with $|E^i|$ dual variables. Thus, the optimization problem given in Eq. (10) is over $|E^i|$ dual variables which can be on the order of k^2. To obtain an efficient update we derive an equivalent, more compact, optimization problem which has exactly k variables. The compact problem is

$$\begin{aligned} \max_{\alpha_{t,1},\ldots,\alpha_{t,k}} \quad & \gamma \textstyle\sum_{y \in Y^t} \alpha_{t,y} - \sum_{y=1}^k G(\boldsymbol{\theta}_y^t + \alpha_{t,y}\mathbf{x}^t) \\ \text{s.t.} \quad & \sum_{y=1}^k \alpha_{t,y} = 0 \; , \; \sum_{y \in Y^t} \alpha_{t,y} \le C \; , \qquad (11) \\ & \forall y \in Y^t : \; \alpha_{t,y} \ge 0 \; , \; \forall y \notin Y^t : \; \alpha_{t,y} \le 0 \end{aligned}$$

In Appendix B we prove that the problem given in Eq. (11) is equivalent to the problem given in Eq. (10). A rather complex algorithm for solving the compact problem in the special case where the complexity function is $F(\boldsymbol{\omega}_r) = \frac{1}{2}\|\boldsymbol{\omega}_r\|^2$ was presented in [11]. Here we present an efficient primal-dual interior point algorithm (PDIP) for solving the compact optimization problem which is applicable to a larger family of complexity functions. We describe the PDIP algorithm for a slightly more general optimization problem which still exploits the structure of the problem and leads to a very efficient PDIP algorithm. Let $\{f_r | f_r : \mathbb{R} \to \mathbb{R}\}_{r=1}^d$ be a set of d twice differentiable functions from and denote by $\{f_r'\}$ and $\{f_r''\}$ their first and second derivatives. Let $\mathbf{p}$ and $\mathbf{q}$ be two vectors in $\mathbb{R}^d$, A be a $2 \times d$ matrix, and $\mathbf{b}$ a two dimensional vector over $\mathbb{R}$. Instead of the original problem defined by Eq. (11), we work with the following minimization problem,

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^d} \sum_{r=1}^d f_r(\alpha_r) \text{ s.t. } A\boldsymbol{\alpha} = \mathbf{b}, \; \forall r \; p_r\alpha_r \le q_r \; . \qquad (12)$$

It is easy to verify that the problem defined by Eq. (11) can be reformatted and described as an instance of the problem defined by Eq. (12).

To motivate the derivation of the PDIP algorithm, let us first note that the dual of Eq. (12) is the problem $\max_{\boldsymbol{\lambda} \in \mathbb{R}_+^d, \boldsymbol{\nu} \in \mathbb{R}^2} \mathcal{D}(\boldsymbol{\lambda}, \boldsymbol{\nu})$ where $\mathcal{D}(\boldsymbol{\lambda}, \boldsymbol{\nu})$ is

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^d} \sum_{r=1}^d f_r(\alpha_r) + \sum_{r=1}^d \lambda_r (p_r\alpha_r - q_r) + \langle \boldsymbol{\nu}, (A\boldsymbol{\alpha} - \mathbf{b}) \rangle \; . \qquad (13)$$

Denote by $\mathcal{P}(\boldsymbol{\alpha})$ the objective function of the problem in Eq. (12). As the name implies, the PDIP algorithm maintains strictly feasible primal $(\boldsymbol{\alpha})$ and dual $(\boldsymbol{\lambda}, \boldsymbol{\nu})$ solutions at all times. (To remind the reader, a strictly feasible solution of a given problem satisfies all the constraints of the problem, where each inequality constraint holds with strict inequality.) Assume that we have on hand a strictly feasible primal-dual solution $(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \boldsymbol{\nu})$. We now define the following function, $\eta(\boldsymbol{\alpha}, \boldsymbol{\lambda}) = \sum_{r=1}^d \lambda_r (q_r - p_r\alpha_r)$. We next show that $\eta(\boldsymbol{\alpha}, \boldsymbol{\lambda})$ is a lower bound on the duality gap of our primal-dual solution. The definition of $\mathcal{D}(\boldsymbol{\lambda}, \boldsymbol{\nu})$ implies that,

$$\begin{aligned} \mathcal{D}(\boldsymbol{\lambda}, \boldsymbol{\nu}) &\le \sum_{r=1}^d \left(f_r(\alpha_r) + \lambda_r(p_r\alpha_r - q_r)\right) + \langle \boldsymbol{\nu}, A\boldsymbol{\alpha} - \mathbf{b} \rangle \\ &= \mathcal{P}(\boldsymbol{\alpha}) + \eta(\boldsymbol{\alpha}, \boldsymbol{\lambda}) \; , \qquad (14) \end{aligned}$$

where the second equality is due to the fact that $\boldsymbol{\alpha}$ is a feasible dual solution, thus $A\boldsymbol{\alpha} = \mathbf{b}$. Therefore, the duality gap is bounded below by

$$\mathcal{P}(\boldsymbol{\alpha}) - \mathcal{D}(\boldsymbol{\lambda}, \boldsymbol{\nu}) \ \geq\ \eta(\boldsymbol{\alpha}, \boldsymbol{\lambda}) \ . \tag{15}$$

Moreover, if

$$\forall r \in [d],\ f'_r(\alpha_r) + \lambda_r p_r + \nu_1 A_{1,r} + \nu_2 A_{2,r} \ = \ 0 \ , \tag{16}$$

then $\boldsymbol{\alpha}$ attains the minimum of Eq. (13). Therefore, both Eq. (14) and Eq. (15) hold with equality. In this case, $\eta(\boldsymbol{\alpha}, \boldsymbol{\lambda})$ amounts to be the duality gap of the primal-dual solution $(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \boldsymbol{\nu})$.

The PDIP algorithm is an iterative procedure where on each iteration it finds a new strictly feasible primal-dual solution. The primary goal of the update is to decrease the duality gap. To do so, we use the fact that Eq. (15) establishes $\eta(\boldsymbol{\alpha}, \boldsymbol{\lambda})$ as a lower bound on the duality gap. Thus, the main goal of the update is to decrease $\eta(\boldsymbol{\alpha}, \boldsymbol{\lambda})$ on each iteration while ensuring that the actual duality gap, $\mathcal{P}(\boldsymbol{\alpha}) - \mathcal{D}(\boldsymbol{\lambda}, \boldsymbol{\nu})$, stays close to $\eta(\boldsymbol{\alpha}, \boldsymbol{\lambda})$ as much as possible. Additionally, we need to make sure that the new primal-dual solution is also strictly feasible. We are now ready to describe the core update of the PDIP algorithm. Let us denote by $(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \boldsymbol{\nu})$ the current primal-dual solution of the algorithm. The new primal-dual solution is obtained from the current solution by finding a step-size parameter, $s \in (0, 1)$ for a triplet $(\Delta\boldsymbol{\alpha}, \Delta\boldsymbol{\lambda}, \Delta\boldsymbol{\nu})$ and the update itself takes the form $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} + s\Delta\boldsymbol{\alpha}$, $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + s\Delta\boldsymbol{\lambda}$, and $\boldsymbol{\nu} \leftarrow \boldsymbol{\nu} + s\Delta\boldsymbol{\nu}$. To compute the triplet $(\Delta\boldsymbol{\alpha}, \Delta\boldsymbol{\lambda}, \Delta\boldsymbol{\nu})$ we linearize each summand of $\eta(\boldsymbol{\alpha} + \Delta\boldsymbol{\alpha}, \boldsymbol{\lambda} + \Delta\boldsymbol{\lambda})$ using a first order Taylor approximation and get,

$$\begin{aligned}(\lambda_r + \Delta\lambda_r)\,(q_r - p_r(\alpha_r + \Delta\alpha_r)) \ &\approx \\ &(\lambda_r + \Delta\lambda_r)(q_r - p_r\alpha_r) - \lambda_r p_r\,\Delta\alpha_r \ .\end{aligned}$$

We require that the value of η for the new solution is approximately a fraction of the value at the current solution. This is achieved by solving the following set of linear equalities in $\Delta\alpha_r$ and $\Delta\lambda_r$,

$$\forall r \in [d],\ (\lambda_r + \Delta\lambda_r)(q_r - p_r\alpha_r) - \lambda_r p_r\,\Delta\alpha_r = 0.1\,\tfrac{\eta(\boldsymbol{\alpha}, \boldsymbol{\lambda})}{d} \ . \tag{17}$$

The choice of the contraction constant 0.1 was set empirically. Assuming that the above set of equations hold, then $\eta(\boldsymbol{\alpha} + \Delta\boldsymbol{\alpha}, \boldsymbol{\lambda} + \Delta\boldsymbol{\lambda}) \ \approx\ 0.1\,\eta(\boldsymbol{\alpha}, \boldsymbol{\lambda})$. To recap, solving the system of linear equations given by Eq. (17) serves as a proxy for achieving a substantial decrease in η. Next, we need to make sure that η at the new parameters provides a rather tight lower bound. We do so by making sure that the linearization of the left hand side of Eq. (16) is approximately zero by casting the following set of linear equations, to Eq. (16),

$$\begin{aligned}\forall r \in [d],\ \ & f'_r(\alpha_r) + f''_r(\alpha_r)\Delta\alpha_r + (\lambda_r + \Delta\lambda_r)p_r + \\ & (\nu_1 + \Delta\nu_1)A_{1,r} + (\nu_2 + \Delta\nu_2)A_{2,r} \ = \ 0 \ .\end{aligned} \tag{18}$$

Solving Eq. (18) helps us in tightening the lower bound on the duality gap given in Eq. (14). Last, we need to make sure that the new set of parameters is indeed a feasible primal solution by requiring the equality $A(\boldsymbol{\alpha} + \Delta\boldsymbol{\alpha}) = \mathbf{b}$ to hold. The triplet $(\Delta\boldsymbol{\alpha}, \Delta\boldsymbol{\lambda}, \Delta\boldsymbol{\nu})$ is thus found by finding the solution of all the sets of linear equations described above. The step size s is found by a backtracking search (see for instance pp. 612-613 in [1]).

There exists both theoretical and empirical evidence that a PDIP algorithm reaches the optimal solution (within computer accuracy) after $O(\sqrt{d})$ iterations [1, 6, 9]. On each iteration we need to solve a set of $2d + 2$ linear equations. A direct implementation would require $O(d^3)$ operations for each iteration of the PDIP algorithm. However, as we now show, we can utilize the structure of the problem to solve the set of linear equations in linear time. Thus, the complexity of update III is $O(d\sqrt{d}) = O(k\sqrt{k})$. To obtain an efficient solver, we first eliminate the variables $\Delta\boldsymbol{\lambda}$ by rewriting Eq. (17) as

$$\forall r,\ (\lambda_r + \Delta\lambda_r) \ = \ \tfrac{\lambda_r p_r}{q_r - p_r\alpha_r}\,\Delta\alpha_r + \tfrac{0.1\eta(\boldsymbol{\alpha}, \boldsymbol{\lambda})}{d(q_r - p_r\alpha_r)} \ \ , \tag{19}$$

and substituting the above into Eq. (18). We now define $u_r = -f'_r(\alpha_r) - \frac{0.1\eta(\boldsymbol{\alpha}, \boldsymbol{\lambda})}{d(q_r - p_r\alpha_r)} - \nu_1 A_{1,r} - \nu_2 A_{2,r}$, and $z_r = f''_r(\alpha_r) + \lambda_r p_r/(q_r - p_r\alpha_r)$, and rewrite Eq. (18)

$$\forall r \in [d],\ z_r\,\Delta\alpha_r = u_r + A_{1,r}\Delta\nu_1 + A_{2,r}\Delta\nu_2 \ . \tag{20}$$

Finally, we rewrite the set of two equalities $A(\boldsymbol{\alpha} + \Delta\boldsymbol{\alpha}) = \mathbf{b}$ as $A\Delta\boldsymbol{\alpha} = \mathbf{0}$. Substituing $\Delta\alpha_r$ with the right hand side of Eq. (20) in the linear set of equalities $A\Delta\boldsymbol{\alpha} = \mathbf{0}$, we obtain a system of 2 linear equations in 2 variables which can be solved in constant time. From the solution we obtain $\Delta\boldsymbol{\nu}$ and then compute $\Delta\boldsymbol{\alpha}$ as described in Eq. (20). From $\Delta\boldsymbol{\alpha}$ we now compute $\Delta\boldsymbol{\lambda}$ using Eq. (19). The overall complexity of the procedure of assigning new values to the primal-dual feasible solution is thus $O(d)$.

5 Experiments

In this section we present experimental results that demonstrate different aspects of our proposed algorithms. Our experiments compare the three updates given in Sec. 2 using two complexity functions. The first is the squared norm as a complexity function $F(\boldsymbol{\omega}) = \frac{1}{2}\|\boldsymbol{\omega}\|^2$ with the domain $\Omega = \mathbb{R}^n$ and the second is the entropy function $F(\boldsymbol{\omega}) = \sum_{j=1}^{n} \omega_j \log(\omega_j) + \log(n)$ with the domain $\Omega = \{\boldsymbol{\omega} : \omega_j = 0, \sum_j \omega_j = 1\}$. We would like to note that using update I with the first complexity function yields an algorithm which was previously proposed and studied in [3] while using update II with the same complexity function yields the PA algorithm described in [2]. We experimented with the Enron email dataset (available from http://www.cs.umass.edu/∼ronb/datasets/enron_flat.tar.gz). The task is to automatically classify email messages into

Table 1: The average number of online mistakes for different algorithms on seven users from the Enron datasets.

			$F(\boldsymbol{\omega}) = \frac{1}{2}\lVert\boldsymbol{\omega}\rVert^2$			$F(\boldsymbol{\omega}) = \sum_{j=1}^{n} \omega_j \log(n\omega_j)$		
username	$\lvert\mathcal{Y}\rvert$	m	update I	update II	update III	update I	update II	update III
beck-s	101	1971	58.5	55.2	51.9	54.0	50.2	47.1
farmer-d	25	3672	29.5	23.3	22.7	27.6	22.6	22.0
kaminski-v	41	4477	50.2	44.5	41.9	46.7	42.9	40.0
kitchen-l	47	4015	48.2	41.9	40.4	41.9	38.3	36.0
lokay-m	11	2489	24.9	19.1	18.4	24.0	18.7	18.2
sanders-r	30	1188	31.7	28.3	27.2	28.3	24.2	23.4
williams-w3	18	2769	5.0	4.5	4.4	4.2	3.4	3.1

user defined folders. Thus, the instances in this dataset are email messages while the set of classes is the email folders. Note that our online setting naturally captures the essence of this email classification task. We represented each email message as a binary vector $\mathbf{x} \in \{0,1\}^n$ with a coordinate for each word, so that $x_i = 1$ if the word corresponding to the index i appears in the email message and zero otherwise. We ran the various algorithms on sequences of email messages from 7 users. For update II we used the closed form solution derived in Sec. 4 and for update III we used the PDIP algorithm. We found out in our experiments that the number of iterations required by the PDIP algorithm never exceeded 15. The performance of the different algorithms on the datasets is summarized in Table 1. It is apparent that regardless of the complexity function used, update III consistently outperforms update II which in turn consistently outperforms update I. However, the improvement of update II over update I is more significant than the improvement of update III over update II. Comparing the two complexity functions we note that regardless of the update used, the complexity function based on the entropy consistently outperforms the complexity function based on the squared norm. Note that when using update I with the squared norm as a complexity function we obtain an adaptation of the Perceptron algorithm for the label ranking task while when using update I with the entropy complexity function we obtain an adaptation of the EG algorithm [8]. The superiority of the entropy-based complexity function over the squared norm was underscored in [8] for regression and classification problems.

References

[1] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[2] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive aggressive algorithms. *Journal of Machine Learning Research*, 7, Mar 2006.

[3] K. Crammer and Y. Singer. A new family of online algorithms for category ranking. *Jornal of Machine Learning Research*, 3:1025–1058, 2003.

[4] K. Crammer and Y. Singer. Loss bounds for online category ranking. In *COLT*, 2005.

[5] A. Elisseeff and J. Weston. A kernel method for multi-labeled classification. In *NIPS*, 2001.

[6] A. Forsgren, P. E. Gill, and M. H. Wright. Interior methods for nonlinear optimization. *SIAM Review*, 44(4), 2002.

[7] M. Herbster. Learning additive models online with fast evaluating kernels. In *COLT*, pages 444–460, 2001.

[8] J. Kivinen and M. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–64, January 1997.

[9] Y. Nesterov and A. Nemirovski. *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM Studies in Applied Mathematics, 1994.

[10] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):1–40, 1999.

[11] S. Shalev-Shwartz and Y. Singer. Efficient learning of label ranking by soft projections onto polyhedra. *Journal of Machine Learning Research*, 7:1567–1599, 2006.

[12] S. Shalev-Shwartz and Y. Singer. Online learning meets optimization in the dual. In *COLT*, 2006.

[13] S. Shalev-Shwartz and Y. Singer. Convex repeated games and fenchel duality. In *NIPS*, 2006.

[14] V. N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

A Proof of Lemma 1

Proof Denote $\Delta_t = \mathcal{D}(\boldsymbol{\tau}^{t+1}) - \mathcal{D}(\boldsymbol{\tau}^t)$. From the definition of $\boldsymbol{\tau}^{t+1}$ we have that,

$$\begin{aligned}\Delta_t \;=\; & \gamma\, C - G(\boldsymbol{\theta}^t_{r'} + C\,\mathbf{x}^t) + G(\boldsymbol{\theta}^t_{r'}) \\ & - G(\boldsymbol{\theta}^t_{s'} - C\,\mathbf{x}^t) + G(\boldsymbol{\theta}^t_{s'}) \;\; .\end{aligned} \tag{21}$$

Using Taylor expansion of G around $\boldsymbol{\theta}^t_{r'}$ we get that there exists $\boldsymbol{\theta}$ for which,

$$\begin{aligned}G(\boldsymbol{\theta}^t_{r'} + C\mathbf{x}^t) &\le G(\boldsymbol{\theta}^t_{r'}) + C\langle \mathbf{x}^t, g\big(\boldsymbol{\theta}^t_{r'}\big)\rangle + \frac{C^2\langle \mathbf{x}^t, H(\boldsymbol{\theta})\mathbf{x}^t\rangle}{2} \\ &\le G(\boldsymbol{\theta}^t_{r'}) + C\,\langle \mathbf{x}^t, \boldsymbol{\omega}^t_{r'}\rangle + \frac{1}{4}\,C^2 \;\; ,\end{aligned} \tag{22}$$

where for the last inequality we used the fact that if G is differentiable then $\boldsymbol{\omega}^t_{r'} = g\big(\boldsymbol{\theta}^t_{r'}\big)$ and our assumption that $\langle \mathbf{x}^t, H(\boldsymbol{\theta})\,\mathbf{x}^t\rangle \le 1/2$. Similarly, using Taylor expansion of G around $\boldsymbol{\theta}^t_{s'}$ we get that there exists $\boldsymbol{\theta}'$ for which, $G(\boldsymbol{\theta}^t_{s'} - C\,\mathbf{x}^t) \le G(\boldsymbol{\theta}^t_{s'}) - C\,\langle \mathbf{x}^t, \boldsymbol{\omega}^t_{s'}\rangle + \frac{1}{4}\,C^2$. Plugging this inequality and Eq. (22) into Eq. (21) gives that $\Delta_t \ge \gamma\, C + C\,\langle \boldsymbol{\omega}^t_{r'} - \boldsymbol{\omega}^t_{s'}, \mathbf{x}^t\rangle - \frac{1}{2}\,C^2$. Recall that the choice of (r', s') implies that $\langle \boldsymbol{\omega}^t_{r'} - \boldsymbol{\omega}^t_{s'}, \mathbf{x}^t\rangle \le 0$. We therefore get that $\Delta_t \ge \gamma\, C - \frac{1}{2}\,C^2$. ■

B The Reduced Dual Problem

In this section we derive a reduced dual problem which is equivalent to the dual problem given in Eq. (5). Using the definition of the hinge-loss from Eq. (1) we can rewrite the primal problem at the right-hand side of Eq. (3) as

$$\begin{aligned} &\inf_{\bar{\boldsymbol{\omega}}\in\Omega^k,\boldsymbol{\xi}\geq\mathbf{0}} \quad \sum_{r=1}^{k} F(\boldsymbol{\omega}_r) + C\sum_{i=1}^{m}\xi_i \\ &\text{s.t. } \forall i\in[m] \quad , \quad \forall (r,s)\in E^i,\ \ \langle \boldsymbol{\omega}_r - \boldsymbol{\omega}_s, \mathbf{x}^i\rangle \geq \gamma - \xi_i \ . \end{aligned} \qquad (23)$$

The core idea for deriving the compact representation is to introduce virtual variables, one for each example. Each variable acts as a threshold for separating the predictions for the labels in Y^i from the predictions for the rest of the labels. The complicating factor in proving the equivalence of the primal problems is due to the fact that ξ_i might be strictly positive. Let $\boldsymbol{b}\in\mathbb{R}^m$ denote the vector of virtual thresholds, then the more compact optimization problem is defined as follows,

$$\begin{aligned} &\inf_{\bar{\boldsymbol{\omega}}\in\Omega^k,\boldsymbol{b},\boldsymbol{\xi}\geq\mathbf{0}} \sum_{r=1}^{k} F(\boldsymbol{\omega}_r) + C\sum_{i=1}^{m}\xi_i \\ &\text{s.t. } \forall i\in[m] \ \ \forall (r,s)\in E^i : \\ &\qquad\qquad \langle \boldsymbol{\omega}_r, \mathbf{x}^i\rangle \geq b_i + \gamma/2 - \xi_i/2\ ; \\ &\qquad\qquad \langle \boldsymbol{\omega}_s, \mathbf{x}^i\rangle \leq b_i - \gamma/2 + \xi_i/2\ \ . \end{aligned} \qquad (24)$$

Since the objective function of Eq. (24) and Eq. (23) are identical and $\boldsymbol{b}$ has no effect on the objective function, but rather on the constraints, it suffices to show that for any feasible solution $(\bar{\boldsymbol{\omega}},\boldsymbol{\xi})$ of Eq. (24) there exists a feasible solution $(\bar{\boldsymbol{\omega}},\mathbf{b},\boldsymbol{\xi})$ of Eq. (23) and vice versa.

Let $(\bar{\boldsymbol{\omega}},\mathbf{b},\boldsymbol{\xi})$ be a feasible solution of Eq. (24). Then, for any pair of labels $r\in Y^i$ and $s\notin Y^i$ we get that,

$$\langle \boldsymbol{\omega}_r - \boldsymbol{\omega}_s, \mathbf{x}^i\rangle \geq \gamma/2 + b_i - \xi_i/2 - (b_i - \gamma/2 + \xi_i/2) = \gamma - \xi\ \ .$$

Therefore, $(\bar{\boldsymbol{\omega}},\boldsymbol{\xi})$ is a feasible solution of Eq. (23). Proving that if $(\bar{\boldsymbol{\omega}},\boldsymbol{\xi})$ is a feasible solution of Eq. (23) then there exists $\boldsymbol{b}$ such that $(\bar{\boldsymbol{\omega}},\boldsymbol{b},\boldsymbol{\xi})$ is a feasible solution of Eq. (24) is a bit more complex to show. We do so by first defining the following two variables for each $i\in[m]$,

$$\begin{aligned} \bar{b}_i &= \min_{r\in Y^i}\langle \boldsymbol{\omega}_r, \mathbf{x}^i\rangle - \gamma/2 + \xi_i/2\ ; \\ \underline{b}_i &= \max_{s\notin Y^i}\langle \boldsymbol{\omega}_s, \mathbf{x}^i\rangle + \gamma/2 - \xi_i/2\ \ . \end{aligned} \qquad (25)$$

Let j and l denote the indices of the labels which attain, respectively, the minimum and maximum of the problems defined by Eq. (25). Then, by construction we get that,

$$\bar{b}_i - \underline{b}_i = \langle \boldsymbol{\omega}_j - \boldsymbol{\omega}_l, \mathbf{x}^i\rangle - \gamma + \xi_i \geq 0\ \ ,$$

where the last inequality is due to feasibility of the solution $(\bar{\boldsymbol{\omega}},\boldsymbol{\xi})$ with respect to the problem defined by Eq. (23). We now define $b_i = (\bar{b}_i + \underline{b}_i)/2$ which immediately implies that $\underline{b}_i \leq b_i \leq \bar{b}_i$. We therefore get that for any label $r\in Y^i$ the following inequality hold,

$$\begin{aligned} \langle \boldsymbol{\omega}_r, \mathbf{x}^i\rangle \ \geq\ \langle \boldsymbol{\omega}_j, \mathbf{x}^i\rangle \quad &\geq \quad \bar{b}_i + \gamma/2 - \xi_i/2 \\ &\geq \quad b_i + \gamma/2 - \xi_i/2\ \ , \end{aligned}$$

and similarly for any label $s\notin Y^i$ we get,

$$\begin{aligned} \langle \boldsymbol{\omega}_s, \mathbf{x}^i\rangle \ \leq\ \langle \boldsymbol{\omega}_l, \mathbf{x}^i\rangle \quad &\leq \quad \underline{b}_i - \gamma/2 + \xi_i/2 \\ &\leq \quad b_i - \gamma/2 + \xi_i/2\ \ . \end{aligned}$$

We have thus established a feasible solution $(\bar{\boldsymbol{\omega}},\boldsymbol{b},\boldsymbol{\xi})$ as required.

We next derive the dual of the more compact problem defined by Eq. (24). We now associate a Lagrange multiplier with each constraint and then follow the same line of derivation used to obtain the dual of the original problem. We briefly review this derivation. The Lagrangian of the compact problem defined by Eq. (24) is, $\mathcal{L}(\bar{\boldsymbol{\omega}},\boldsymbol{b},\boldsymbol{\xi},\boldsymbol{\alpha}) = \sum_r F(\boldsymbol{\omega}_r) + C\sum_i \xi_i + \sum_i\sum_{r\in Y^i}\alpha_{i,r}(b_i + \gamma/2 - \xi_i/2 - \langle \boldsymbol{\omega}_r, \mathbf{x}_i\rangle) + \sum_i\sum_{s\notin Y^i}\alpha_{i,s}(\langle \boldsymbol{\omega}_r, \mathbf{x}_i\rangle - b_i + \gamma/2 - \xi_i/2)$. Analogous to the original problem the dual is now defined to be, $\mathcal{D}(\boldsymbol{\alpha}) = \inf_{\bar{\boldsymbol{\omega}}\in\Omega^k,\boldsymbol{b},\boldsymbol{\xi}\geq\mathbf{0}} \mathcal{L}(\bar{\boldsymbol{\omega}},\boldsymbol{b},\boldsymbol{\xi},\boldsymbol{\alpha})$. We now overload our notation and redefine the following vector,

$$\boldsymbol{\theta}_y \ = \ \sum_{i:y\in Y^i}\alpha_{i,y}\mathbf{x}^i - \sum_{i:y\notin Y^i}\alpha_{i,y}\mathbf{x}^i \ \ . \qquad (26)$$

Taking the derivative of the Lagrangian with respect to each b_i and equating it to zero gives the following constraint

$$\forall i\in[m],\ \sum_{r\in Y^i}\alpha_{i,r} - \sum_{s\notin Y^i}\alpha_{i,s} = 0\ \ . \qquad (27)$$

Analogous to the constraint that $\sum_{(r,s)}\tau_{i,r,s}\leq C$, by taking the derivative of the Lagrangian with respect to each ξ_i and equating it to zero, we now obtain that $\sum_{r\in Y^i}\alpha_{i,r}\leq C$. Let us now depart from the standard notation and redefine $\alpha_{i,s}$ to be $-\alpha_{i,s}$ for all $s\notin Y^i$ and for all i. Eq. (27) distills to the constraint $\sum_{y=1}^{k}\alpha_{i,y}=0$ and finally the dual of the compact form distills to the following constraint optimization problem,

$$\begin{aligned} &\max_{\boldsymbol{\alpha}} \ \ \gamma\sum_{i=1}^{m}\sum_{y\in Y^i}\alpha_{i,y} - \sum_{y=1}^{k} G(\boldsymbol{\theta}_y) \\ &\text{s.t. } \forall i\in[m],\ \sum_{y=1}^{k}\alpha_{i,y} = 0\ ,\ \sum_{y\in Y^i}\alpha_{i,y}\leq C\ , \\ &\qquad \forall y\in Y^i:\ \alpha_{i,y}\geq 0\ ,\ \forall y\notin Y^i:\ \alpha_{i,y}\leq 0 \end{aligned} \ \ . \qquad (28)$$

We denote the reduced dual objective function by $\mathcal{D}(\boldsymbol{\alpha})$. Finally note that the optimization problem given in Eq. (10) can be rewritten as the problem of maximizing $\mathcal{D}(\boldsymbol{\alpha})$ over the variables $\alpha_{t,1},\ldots,\alpha_{t,k}$.