Predtron: A Family of Online Algorithms for General Prediction Problems

Prateek Jain
Microsoft Research, INDIA
prajain@microsoft.com

Nagarajan Natarajan
University of Texas at Austin, USA
naga86@cs.utexas.edu

Ambuj Tewari
University of Michigan, Ann Arbor, USA
tewaria@umich.edu

Abstract

Modern prediction problems arising in multilabel learning and learning to rank pose unique challenges to the classical theory of supervised learning. These problems have large prediction and label spaces of a combinatorial nature and involve sophisticated loss functions. We offer a general framework to derive mistake driven online algorithms and associated loss bounds. The key ingredients in our framework are a general loss function, a general vector space representation of predictions, and a notion of margin with respect to a general norm. Our general algorithm, Predtron, yields the perceptron algorithm and its variants when instantiated on classic problems such as binary classification, multiclass classification, ordinal regression, and multilabel classification. For multilabel ranking and subset ranking, we derive novel algorithms, notions of margins, and loss bounds. A simulation study confirms the behavior predicted by our bounds and demonstrates the flexibility of the design choices in our framework.

1 Introduction

Classical supervised learning problems, such as binary and multiclass classification, share a number of characteristics. The prediction space (the space in which the learner makes predictions) is often the same as the label space (the space from which the learner receives supervision). Because directly learning discrete valued prediction functions is hard, one learns real-valued or vector-valued functions. These functions generate continuous predictions that are converted into discrete ones via simple mappings, e.g., via the 'sign' function (binary classification) or the 'argmax' function (multiclass classification). Also, the most commonly used loss function is simple, viz. the 0-1 loss.

In contrast, modern prediction problems, such as multilabel learning, multilabel ranking, and subset ranking do not share these characteristics. In order to handle these problems, we need a more general framework that offers more flexibility. First, it should allow for the possibility of having different label space and prediction space. Second, it should allow practitioners to use creative, new ways to map continuous, vector-valued predictions to discrete ones. Third, it should permit the use of general loss functions.

Extensions of the theory of classical supervised learning to modern predictions problems have begun. For example, the work on calibration dimension [1] can be viewed as extending one aspect of the theory, viz. that of calibrated surrogates and consistent algorithms based on convex optimization. This paper deals with the extension of another interesting part of classical supervised learning: mistake driven algorithms such as perceptron (resp. winnow) and their analyses in terms of ℓ_2 (resp. ℓ_1) margins [2, Section 7.3].

We make a number of contributions. First, we provide a general framework (Section 2) whose ingredients include an arbitrary loss function and an arbitrary representation of discrete predictions in a continuous space. The framework is abstract enough to be of general applicability but it offers enough mathematical structure so that we can derive a general online algorithm, Predtron (Algorithm 1), along with an associated loss bound (Theorem 1) under an abstract margin condition (Section 2.2). Second, we show that our framework unifies several perception-like algorithms for classical problems such as binary classification, multiclass classification, ordinal regression, and multilabel classification (Section 3). Even for these classical problems, we get some new results, for example, when the loss function treats labels asymmetrically or when there exists a 'reject' option in classification. Third, we apply our framework to two modern prediction problems: subset ranking (Section 4) and multilabel ranking (Section 5). In both of these problems, the prediction space (rankings) is different from the supervision space (set of labels or vector of relevance scores). For these two problems, we propose interesting, novel notions of correct prediction with a margin and derive mistake bounds under a loss derived from NDCG, a ranking measure that pays more attention to the performance at the top of a ranked list. Fourth, our techniques based on online convex optimization (OCO) can effortlessly incorporate notions of margins w.r.t. non-Euclidean norms, such as ℓ_1 norm, group norm, and trace norm. Such flexibility is important in modern prediction problems where the learned parameter can be a high dimensional vector or a large matrix with low group or trace norm. Finally, we test our theory in a simulation study (Section 6) dealing with the subset ranking problem showing how our framework can be adapted to a specific prediction problem. We investigate different margin notions as we vary two key design choices in our abstract framework: the map used to convert continuous predictions into discrete ones, and the choice of the norm used in the definition of margin.

Related Work. Our general algorithm is related to the perceptron and online gradient descent algorithms used in structured prediction [3, 4]. But, to the best of knowledge, our emphasis on keeping label and prediction spaces possibly distinct, our use of a general representation of predictions, and our investigation of generalized notions of margins are all novel. The use of simplex coding in multiclass problems [5] inspired the use of maximum similarity/minimum distance decoding to obtain discrete predictions from continuous ones. Our proofs use results about Online Gradient Descent and Online Mirror Descent from the Online Convex Optimization literature [6].

2 Framework and Main Result

The key ingredients in classic supervised learning are an input space, an output space and a loss function. In this paper, the input space $\mathcal{X} \in \mathbb{R}^p$ will always be some subset of a finite dimensional Euclidean space. Our algorithms maintain prediction functions as a linear combination of the seen inputs. As a result, they easily kernelize and the theory extends, in a straightforward way to the case when the input space is a, possibly infinite dimensional, reproducing kernel Hilbert space (RKHS).

2.1 Labels, Prediction, and Scores

We will distinguish between the label space and the prediction space. The former is the space where the training labels come from whereas the latter is the space where the learning algorithm has to make predictions in. Both spaces will be assumed to be finite. Therefore, without any loss of generality, we can identify the label space with $[\ell] = \{1, \ldots, \ell\}$ and the prediction space with $[k]$ where ℓ, k are positive, but perhaps very large, integers. A given loss function $\mathbf{L} : [k] \times [\ell] \to \mathbb{R}_+$ maps a prediction $\sigma \in [k]$ and a label $y \in [\ell]$ to a non-negative loss $\mathbf{L}(\sigma, y)$. The loss $\mathbf{L}$ can equivalently be thought of as a $k \times \ell$ matrix with loss values as entries. Define the set of correct predictions for a label y as $\Sigma_y = \{\sigma_y \in [k] \ : \ \mathbf{L}(\sigma_y, y) = 0\}$. We assume that, for every label y, the set Σ_y is non-empty. That is, every column of the loss matrix has a zero entry. Also, let $c_{\mathbf{L}} = \min_{\mathbf{L}(\sigma,y)>0} \ \mathbf{L}(\sigma, y)$ and $C_{\mathbf{L}} = \max_{\sigma,y} \ \mathbf{L}(\sigma, y)$ be the minimum (non-zero) and maximum entries in the loss matrix.

In an online setting, the learner will see a stream of examples $(X_\tau, Y_\tau) \in \mathcal{X} \times [\ell]$. Learner will predict scores using a linear predictor $W \in \mathbb{R}^{d \times p}$. However, the predicted scores WX_τ will be in $\mathbb{R}^d$, not in the prediction space $[k]$. So, we need a function $\mathrm{pred} : \mathbb{R}^d \to [k]$ to convert scores into actual predictions. We will assume that there is a unique representation $\mathrm{rep}(\sigma) \in \mathbb{R}^d$ of each prediction σ such that $\| \operatorname{rep}(\sigma)\|_2 = 1$ for all σ. Given this, a natural transformation of scores into prediction is given by the following *maximum similarity decoding*:

$$\operatorname{pred}(t) \in \operatorname*{argmax}_{\sigma \in [k]} \ \langle \operatorname{rep}(\sigma), t \rangle \ , \tag{1}$$

where ties in the "argmax" can be broken arbitrarily. There are some nice consequences of the definition of pred above. First, because $\| \operatorname{rep}(\sigma)\|_2 = 1$, maximum similarity decoding is equivalent to *nearest neighbor decoding*: $\operatorname{pred}(t) \in \operatorname{argmin}_\sigma \| \operatorname{rep}(\sigma) - t\|_2$. Second, we have a homogeneity property: $\operatorname{pred}(ct) = \operatorname{pred}(t)$ if $c > 0$. Third, rep serves as an "inverse" of pred in the following sense. We have, $\operatorname{pred}(\operatorname{rep}(\sigma)) = \sigma$ for all σ. Moreover, $\operatorname{rep}(\operatorname{pred}(t))$ is more similar to t than the representation of any other prediction σ:

$$\forall t \in \mathbb{R}^d, \sigma \in [k], \ \langle \operatorname{rep}(\operatorname{pred}(t)), t\rangle \geq \langle \operatorname{rep}(\sigma), t\rangle \, .$$

In view of these facts, we will use $\operatorname{pred}^{-1}(\sigma)$ and $\operatorname{rep}(\sigma)$ interchangeably. Using pred, the loss function L can be extended to a function defined on $\mathbb{R}^d \times [k]$ as:

$$\mathbf{L}(t, y) = \mathbf{L}(\operatorname{pred}(t), y).$$

With a little abuse of notation, we will continue to denote this new function also by $\mathbf{L}$.

2.2 Margins

We say that *a score t is compatible with a label y* if the set of σ's that achieve the maximum in the definition (1) of pred is exactly Σ_y. That is, $\operatorname{argmax}_{\sigma \in [k]} \ \langle \operatorname{pred}^{-1}(\sigma), t\rangle = \Sigma_y$. Hence, for any $\sigma_y \in \Sigma_y, \sigma \notin \Sigma_y$, we have $\langle \operatorname{pred}^{-1}(\sigma_y), t\rangle > \langle \operatorname{pred}^{-1}(\sigma), t\rangle$. The notion of margin makes this requirement stronger. We say that a score t *has a margin* $\gamma > 0$ *on label* y, iff t is compatible with y and

$$\forall \sigma_y \in \Sigma_y, \sigma \notin \Sigma_y, \ \langle \operatorname{pred}^{-1}(\sigma_y), t\rangle \geq \langle \operatorname{pred}^{-1}(\sigma), t\rangle + \gamma$$

Note that margin scales with t: if t has margin γ on y then ct has margin $c\gamma$ on y for any positive c. If we are using linear predictions $t = WX$, we say that W *has margin* γ *on* (X, y) iff $t = WX$ has margin γ on y. We say that W *has margin* γ *on a dataset* $(X_1, y_1), \ldots, (X_n, y_n)$ iff W has margin γ on (X_τ, y_τ) for all $\tau \in [n]$. Finally, a dataset $(X_1, y_1), \ldots, (X_n, y_n)$ is said to be *linearly separable with margin* γ if there is a unit norm[1] $W^\star$ such that $W^\star$ has margin γ on $(X_1, y_1), \ldots, (X_n, y_n)$.

2.3 Algorithm

Just like the classic perceptron algorithm, our generalized perceptron algorithm (Algorithm 1) is mistake driven. That is, it only updates on round when a mistake, i.e., a non-zero loss, is incurred. On a mistake round, it makes a rank-one update of the form $W_{\tau+1} = W_\tau - g_\tau \cdot X_\tau^\top$ where $g_\tau \in \mathbb{R}^d, X_\tau \in \mathbb{R}^p$. Therefore, W_τ always has a representation of the form $\sum_i g_i X_i^\top$. The prediction on a fresh input X is given by $\sum_i g_i \langle X_i, X\rangle$ which means the algorithm, just like the original perceptron, can be kernelized.

We will give a loss bound for the algorithm using tools from Online Convex Optimization (OCO). Define the function $\phi : \mathbb{R}^d \times [\ell] \to \mathbb{R}$ as

$$\phi(t, y) = \max_{\sigma \in [k]} \ \left(\mathbf{L}(\sigma, y) - \langle \operatorname{pred}^{-1}(\sigma_y) - \operatorname{pred}^{-1}(\sigma), t\rangle \right) \tag{2}$$

where $\sigma_y \in \Sigma_y$ is an arbitrary member of Σ_y. For any y, $\phi(\cdot, y)$ is a point-wise maximum of linear functions and hence convex. Also, ϕ is non-negative: choose $\sigma = \sigma_y$ to lower bound the maximum. The inner product part vanishes and the loss $\mathbf{L}(\sigma_y, y)$ vanishes too because $\sigma_y \in \Sigma_y$. Given the definition of ϕ, Algorithm 1 can be described succinctly as follows. At round τ, if $\mathbf{L}(W_\tau X_\tau, Y_\tau) > 0$, then $W_{\tau+1} = W_\tau - \eta \nabla_W \phi(W X_\tau, Y_\tau)$, otherwise $W_{\tau+1} = W_\tau$.

[1] Here, we mean that the Frobenius norm $\|W^\star\|_F$ equals 1. Of course, the notion of margin can be generalized to any norm including the entry-based ℓ_1 norm $\|W\|_1$ and the spectrum-based ℓ_1 norm $\|W\|_{S(1)}$ (also called the nuclear or trace norm). See Appendix B.2.

Algorithm 1 Predtron: Extension of the Perceptron Algorithm to General Prediction Problems

```
$W_1 \leftarrow \mathbf{0}$
for $\tau = 1, 2, \ldots$ do
    Receive $X_\tau \in \mathbb{R}^p$
    Predict $\sigma_\tau = \mathrm{pred}(W_\tau X_\tau) \in [k]$
    Receive label $y_\tau \in [\ell]$
    if $\mathbf{L}(\sigma_\tau, y_\tau) > 0$ then
        $(t, y) = (W_\tau X_\tau, y_\tau)$
        $\tilde{\sigma}_\tau = \mathrm{argmax}_{\sigma \in [k]} \left(\mathbf{L}(\sigma, y) - \langle \mathrm{pred}^{-1}(\sigma_y) - \mathrm{pred}^{-1}(\sigma), t \rangle\right) \in [k]$
        $\nabla_\tau = (\mathrm{pred}^{-1}(\tilde{\sigma}_\tau) - \mathrm{pred}^{-1}(\sigma_y)) \cdot X_\tau^\top \in \mathbb{R}^{d \times p}$
        $W_{\tau+1} = W_\tau - \eta \nabla_\tau$
    else
        $W_{\tau+1} = W_\tau$
    end if
end for
```

Theorem 1. *Suppose the dataset $(X_1, y_1), \ldots, (X_n, y_n)$ is linearly separable with margin γ. Then the sequence W_τ generated by Algorithm 1 with $\eta = c_{\mathbf{L}}/(4R^2)$ satisfies the loss bound,*

$$\sum_{\tau=1}^{n} \mathbf{L}(W_\tau X_\tau, y_\tau) \leq \frac{4R^2 C_{\mathbf{L}}^2}{c_{\mathbf{L}} \gamma^2}$$

where $\|X_\tau\|_2 \leq R$ for all τ.

Note that the bound above assumes perfect linear separability. However, just the classic perceptron, the bound will degrade gracefully when the best linear predictor does not have enough margin on the data set.

The Predtron algorithm has some interesting variants, two of which we consider in the appendix. A loss driven version, Predtron.LD, enjoys a loss bound that gets rid of the $C_{\mathbf{L}}/c_{\mathbf{L}}$ factor in the bound above. A version, Predtron.Link, that uses link functions to deal with margins defined with respect to non-Euclidean norms is also considered.

3 Relationship to Existing Results

It is useful to discuss a few concrete applications of the abstract framework introduced in the last section. Several existing loss bounds can be readily derived by applying our bound for the generalized perceptron algorithm in Theorem 1. In some cases, our framework yields a different algorithm than existing counterparts, yet admitting identical loss bounds, up to constants.

Binary Classification. We begin with the classical perceptron algorithm for binary classification (i.e., $\ell = 2$) [7]: $\mathbf{L}_{\text{0-1}}(\sigma, y) = 1$ if $\sigma \neq y$ or 0 otherwise. Letting $\mathrm{rep}(\sigma)$ be $+1$ for the positive class and -1 for the negative class, predictor vector $W_\tau \in \mathbb{R}^{1 \times p}$, and thus $\mathrm{pred}(t) = \mathrm{sign}(t)$, Algorithm 1 reduces to the original perceptron algorithm; Theorem 1 yields identical mistake bound on a linearly separable dataset with margin γ (if the classical margin is γ, ours works out to be 2γ), i.e. $\sum_{\tau=1}^{n} \mathbf{L}_{\text{0-1}}(W_\tau X_\tau, y_\tau) \leq \frac{R^2}{\gamma^2}$. We can also easily incorporate asymmetric losses. Let $\mathbf{L}_\alpha(\sigma, y) = \alpha_y$, if $\sigma \neq y$ and 0 otherwise. We then have the following result.

Corollary 2. *Consider the perceptron with weighted loss $\mathbf{L}_\alpha$. Assume $\alpha_1 \geq \alpha_2$ without loss of generality. Then the sequence W_τ generated by Algorithm 1 satisfies the weighted mistake bound,*

$$\sum_{\tau=1}^{n} \mathbf{L}_\alpha(W_\tau X_\tau, y_\tau) \leq \frac{4R^2 \alpha_1^2}{\alpha_2^2 \gamma^2}.$$

We are not aware of such results for weighted loss. Previous work [8] studies perceptrons with uneven margins, and the loss bound there only implies a bound on the unweighted loss: $\sum_{\tau=1}^{n} \mathbf{L}_{\text{0-1}}(t_\tau, y_\tau)$. In a technical note, Rätsch and Kivinen [9] provide a mistake bound of the form (without proof): $\sum_{\tau=1}^{n} \mathbf{L}_\alpha(W_\tau X_\tau, y_\tau) \le \frac{R^2}{4\gamma^2}$, but for the specific choice of weights $\alpha_1 = a^2$ and $\alpha_2 = (1-a)^2$ for any $a \in [0,1]$.

Another interesting extension is obtained by allowing the predictions to have a REJECT option. Define $\mathbf{L}_{\text{REJ}}(\text{REJECT}, y) = \beta_y$ and $\mathbf{L}_{\text{REJ}}(\sigma, y) = \mathbf{L}_{\text{0-1}}(\sigma, y)$ otherwise. Assume $1 \ge \beta_1 \ge \beta_2 > 0$ without loss of generality. Choosing the standard basis vectors in $\mathbb{R}^2$ to be $\text{rep}(\sigma)$ for the positive and the negative classes, and $\text{rep}(\text{REJECT}) = \frac{1}{\sqrt{2}} \sum_{\sigma \in \{1,2\}} \text{rep}(\sigma)$, we obtain $\sum_{\tau=1}^{n} \mathbf{L}_{\text{REJ}}(W_\tau X_\tau, y_\tau) \le \frac{4R^2\beta_1^2}{\gamma^2\beta_2^2}$ (See Appendix C.1).

Multiclass Classification. Each instance is assigned exactly one of m classes (i.e., $\ell = m$). Extending binary classification, we choose the standard basis vectors in $\mathbb{R}^m$ to be $\text{rep}(\sigma)$ for the m classes. The learner predicts score $t \in \mathbb{R}^m$ using the predictor $W \in \mathbb{R}^{m \times p}$. So, $\text{pred}(t) = \text{argmax}_i\, t_i$. Let w_j denote the jth row of W (corresponding to label j). The definition of margin becomes:

$$\langle w_y, X \rangle - \max_{j \ne y} \langle w_j, X \rangle \ge \gamma$$

which is identical to the multiclass margin studied earlier [10]. For the multiclass 0-1 loss $\mathbf{L}_{\text{0-1}}$, we recover their bound, up to constants[2]. Moreover, our surrogate ϕ for $\mathbf{L}_{\text{0-1}}$:

$$\phi(t, y) = \max\big(0, 1 + \max_{\sigma \ne y} t_\sigma - t_y\big),$$

matches the multiclass extension of the Hinge loss studied by [11]. Finally, note that it is straightforward to obtain loss bounds for multiclass perceptron with REJECT option by naturally extending the definitions of rep and $\mathbf{L}_{\text{REJ}}$ for the binary case.

Ordinal Regression. The goal is to assign ordinal classes (such as ratings) to a set of objects $\{X_1, X_2, \dots\}$ described by their features $X_i \in \mathbb{R}^p$. In many cases, precise rating information may not be available, but only their relative ranks; i.e., the observations consist of object-rank pairs (X_τ, y_τ) where $y_\tau \in [\ell]$. $\mathcal{Y}$ is totally-ordered with "$>$" relation, which in turn induces a partial ordering on the objects (X_j is preferred to $X_{j'}$ if $y_j > y_{j'}$, X_j and $X_{j'}$ are not comparable if $y_j = y_{j'}$). For the ranking loss $\mathbf{L}(\sigma, y) = |\sigma - y|$, the PRank perceptron algorithm [12] enjoys the bound $\sum_{\tau=1}^{n} \mathbf{L}(\tau_\tau, y_\tau) \le (\ell - 1)(R^2 + 1)/\tilde{\gamma}^2$, where $\tilde{\gamma}$ is a certain rank margin. By a reduction to multi-class classification with ℓ classes, Algorithm 1 achieves the loss bound $4(\ell-1)^2R^2/\gamma^2$ (albeit, for a different margin γ).

Multilabel Classification. This setting generalizes multiclass classification in that instances are assigned subsets of m classes rather than unique classes, i.e., $\ell = 2^m$. The loss function $\mathbf{L}$ of interest may dictate the choice of rep and in turn pred. For example, consider the following subset losses that treat labels as well as predictions as subsets: (i) Subset 0-1 loss: $\mathbf{L}_{\text{IsErr}}(\sigma, y) = 1$ if $\sigma = y$ or 0 otherwise; (ii) Hamming loss: $\mathbf{L}_{\text{Ham}}(\sigma, y) = |\sigma \cup y| - |\sigma \cap y|$, and (ii) Error set size: $\mathbf{L}_{\text{ErrSetSize}}(\sigma, y) = \big|\{(r, s) \in y \times ([m] \setminus y) : r \notin \sigma, s \in \sigma\}\big|$. A natural choice of rep then is the subset indicator vector in $\{+1, -1\}^d$, where $d = m = \log \ell$, which can be expressed as $\text{rep}(\sigma) = \frac{1}{\sqrt{m}}\big(\sum_{j \in \sigma} e_j - \sum_{j \notin \sigma} e_j\big)$ (where e_j's are the standard basis vectors in $\mathbb{R}^m$). The learner predicts score $t \in \mathbb{R}^m$ using a matrix $W \in \mathbb{R}^{m \times p}$. Note that $\text{pred}(t) = \text{sign}(t)$, where sign is applied component-wise. The number of predictions is 2^m, but we show in Appendix C.2 that the surrogate (2) and its gradient can be efficiently computed for all of the above losses.

4 Subset Ranking

In subset ranking [13], the task is to learn to rank a number of documents in order of their relevance to a query. We will assume, for simplicity, that the number of documents per query is constant that we denote by m. The input space is a subset of $\mathbb{R}^{m \times p_0}$ that we can identify with $\mathbb{R}^p$ for $p = mp_0$. Each row of an input matrix corresponds to a p_0-dimensional feature vector derived jointly using the query

[2] Perceptron algorithm in [10] is based on a slightly different loss defined as $\mathbf{L}_{\text{ErrSet}}(t, y) = 1$ if $|\{r \ne y : t_r \ge t_y\}| > 0$ or 0 otherwise (where $t = WX$). This loss upper bounds $\mathbf{L}_{\text{0-1}}$ (because of the way ties are handled, there can be rounds when $\mathbf{L}_{\text{0-1}}$ is 0, but $\mathbf{L}_{\text{ErrSet}}$ is 1).

and one of the documents associated with it. The predictions σ are all $m!$ permutations of degree m. The most natural (but by no means the only one) representation of permutations is to set $\text{rep}(\sigma) = -\sigma/Z$ where $\sigma(i)$ is the position of the document i in the predicted ranking and the normalization Z ensures that $\text{rep}(\sigma)$ is a unit vector. Note that the dimension d of this representation is equal to m. The minus sign in this representation ensures that $\text{pred}(t)$ outputs a permutation that corresponds to sorting the entries of t in *decreasing* order, a common convention in existing work. A more general representation is obtained by setting $\text{rep}(\sigma) = f(\sigma)/Z$ where $f : \mathbb{R} \to \mathbb{R}$ is a strictly decreasing real valued function that is applied entry-wise to σ. The normalization $Z = \sqrt{\sum_{i=1}^{m} f^2(i)}$ ensures that $\| \text{rep}(\sigma)\|_2 = 1$. To convert an input matrix $X \in \mathbb{R}^p$ ($p = mp_0$) into a score vector $t \in \mathbb{R}^m$, it seems that we need to learn a matrix $W \in \mathbb{R}^{m \times mp_0}$. However, a natural permutation invariance requirement (if the documents associated are presented in a permuted fashion, the output scores should also get permuted in the same way) reduces the dimensionality of W to p_0 (see, e.g., [14] for more details). Thus, given a vector $w \in \mathbb{R}^{p_0}$ we get the score vector as $t = Xw$. The label space consists of relevance score vectors $y \in \{0, 1, \ldots, Y_{\max}\}^m$ where $Y_{\max}$ is typically between 1 and 4 (yielding 2 to 5 grades of relevance). Note that the prediction space (of size $k = m!$) is different from the label space (of size $\ell = (Y_{\max} + 1)^m$).

A variety of loss functions have been used in subset ranking. For multigraded relevance judgments, a very popular choice is NDCG which is defined as $NDCG(\sigma, y) = \big(\sum_{i=1}^{m} \frac{2^{y(i)-1}}{\log_2(1+\sigma(i))}\big)/Z(y)$ where $Z(y)$ is a normalization constant ensuring NDCG stays bounded by 1. To convert it into a loss we define $\mathbf{L}_{\text{NDCG}} = 1 - NDCG$. Note that any permutation that sorts y in decreasing order gets zero $\mathbf{L}_{\text{NDCG}}$. One might worry that the computation of the surrogate defined in (2) and its gradient might require an enumeration of $m!$ permutations. The next lemma allays such a concern.

Lemma 3. *When $\mathbf{L} = \mathbf{L}_{\text{NDCG}}$ and $\text{rep}(\sigma)$ is chosen as above, the computation of the surrogate* (2)*, as well as its gradient, can be reduced to solving a linear assignment problem and hence can be done in $O(m^3)$ time.*

We now give a result explaining what it means for a score vector t to have a margin γ on y when we use a representation of the form described above. Without loss of generality, we may assume that y is sorted in decreasing order of relevance judgements.

Lemma 4. *Suppose $\text{rep}(\sigma) = f(\sigma)/Z$ for a strictly decreasing function $f : \mathbb{R} \to \mathbb{R}$ and $Z = \sqrt{\sum_{i=1}^{m} f^2(i)}$. Let y be a non-constant relevance judgement vector sorted in decreasing order. Suppose $i_1 < i_2, \ldots < i_N, N \geq 1$ are the positions where the relevance drops by a grade or more (i.e., $y(i_j) < y(i_j - 1)$). Then t has a margin γ on y iff t is compatible with y and, for $j \in [N]$,*

$$t_{i_j - 1} \geq t_{i_j} + \frac{\gamma Z}{f(i_j - 1) - f(i_j)}$$

where we define $i_0 = 1, i_{N+1} = m + 1$ to handle boundary cases.

Note that if we choose $f(i) = -i^\alpha, \alpha > 1$ then $f(i_j - 1) - f(i_j) = O(i_j^{\alpha-1})$ for large i_j. In that case, the margin condition above requires less separation between documents with different relevance scores down the list (when viewed in decreasing order of relevance scores) than at the top of the list. We end this section with a loss bound for $\mathbf{L}_{\text{NDCG}}$ under a margin condition.

Corollary 5. *Suppose $\mathbf{L} = \mathbf{L}_{\text{NDCG}}$ and $\text{rep}(\sigma)$ is as in Lemma 4. Then, assuming the dataset is linearly separable with margin γ, the sequence generated by Algorithm 1 with line 9 replaced by*

$$\nabla_\tau = X_\tau^\top (\text{pred}^{-1}(\tilde{\sigma}_\tau) - \text{pred}^{-1}(\sigma_y)) \in \mathbb{R}^{p_0 \times 1}$$

satisfies

$$\sum_{\tau=1}^{n} \mathbf{L}_{\text{NDCG}}(X_\tau w_\tau, y_\tau) \leq \frac{2^{Y_{\max}+3} \cdot m^2 \log_2^2(2m) \cdot R^2}{\gamma^2}$$

where $\|X_\tau\|_{\text{op}} \leq R$.

Note that the result above uses the standard ℓ_2-norm based notion of margin. Imagine a subset ranking problem, where only a small number of features are relevant. It is therefore natural to consider a notion of margin where the weight vector that ranks everything perfectly has low group ℓ_1 norm, instead of low ℓ_2 norm. The ℓ_1 margin also appears in the analysis of AdaBoost [2, Definition

6.2]. We can use a special case of a more general algorithm given in the appendix (Appendix B.2, Algorithm 3). Specifically, we replace line 10 with the step $w_{\tau+1} = (\nabla\psi)^{-1}(\nabla\psi(w_\tau) - \nabla_\tau)$ where $\psi(w) = \frac{1}{2}\|w\|_r^2$. We set $r = \log(p_0)/(\log(p_0) - 1)$. The mapping $\nabla\psi$ and its inverse can both be easily computed (see, e.g., [6, p. 145]).

Corollary 6. *Suppose* $\mathbf{L} = \mathbf{L}_{\text{NDCG}}$ *and* $\text{rep}(\sigma)$ *is as in Lemma 4. Then, assuming the dataset is linearly separable with margin* γ *by a unit* ℓ_1 *norm* $w^\star$ *(*$\|w^\star\|_1 = 1$*), the sequence generated by Algorithm 3 with* ψ *chosen as above (and line 9 modified as in Corollary 5), satisfies*

$$\sum_{\tau=1}^{n} \mathbf{L}_{\text{NDCG}}(X_\tau w_\tau, y_\tau) \leq \frac{9 \cdot 2^{Y_{\max}+3} \cdot m^2 \log_2^2(2m) \cdot R^2 \cdot \log p_0}{\gamma^2}$$

where $\max_{j=1,\ldots,p_o} \|X_{\tau,j}\|_2 \leq R$ *and* $X_{\tau,j}$ *denotes the jth column of* X_τ*.*

5 Multilabel Ranking

As discussed in Section 3, in multilabel *classification*, both prediction space and label space are $\{0,1\}^m$ with sizes $k = \ell = 2^m$. In multilabel *ranking*, however, the learner has to output rankings as predictions. So, as in the previous section, we have $k = m!$ since the prediction σ can be any one of $m!$ permutations of the labels. As before, we choose $\text{rep}(\sigma) = f(\sigma)/Z$ and hence $d = m$. However, unlike the previous section, the input is no longer a matrix but a vector $X \in \mathbb{R}^p$. A prediction $t \in \mathbb{R}^d$ is obtained as WX where $W \in \mathbb{R}^{m\times p}$. Note the contrast with the last section: there, inputs are matrices and a weight vector is learned; here, inputs are vectors and a weight matrix is learned. Since we output rankings, it is reasonable to use a loss that takes positions of labels into account. We can use $\mathbf{L} = \mathbf{L}_{\text{NDCG}}$. Algorithm 1 now immediately applies. Lemma 3 already showed that is efficiently implementable. We have the following straightforward corollary.

Corollary 7. *Suppose* $\mathbf{L} = \mathbf{L}_{\text{NDCG}}$ *and* $\text{rep}(\sigma)$ *is as in Lemma 4. Then, assuming the dataset is linearly separable with margin* γ*, the sequence generated by Algorithm 1 satisfies*

$$\sum_{\tau=1}^{n} \mathbf{L}_{\text{NDCG}}(X_\tau w_\tau, y_\tau) \leq \frac{2^{Y_{\max}+3} \cdot m^2 \log_2^2(2m) \cdot R^2}{\gamma^2}$$

where $\|X_\tau\|_2 \leq R$*.*

The bound above matches the corresponding bound, up to loss specific constants, for the multiclass multilabel perceptron (MMP) algorithm studied by [15]. The definition of margin by [15] for MMP is different from ours since their algorithms are designed *specifically for multilabel ranking*. Just like them, we can also consider other losses, e.g., precision at top K positions. Another perceptron style algorithm for multilabel ranking adopts a pairwise approach of comparing two labels at a time [16]. However, no loss bounds are derived.

The result above uses the standard Frobenius norm based margin. Imagine a multilabel problem, where only a small number of features are relevant across all labels. Then, it is natural to consider a notion of margin where the matrix that ranks everything perfectly has low group $(2,1)$ norm, instead of low Frobenius norm, where $\|W\|_{2,1} = \sum_{j=1}^{p} \|W_j\|_2$ (W_j denotes a column of W). We again use a special case of Algorithm 3 (Appendix B.2). Specifically, we replace line 10 with the step $W_{\tau+1} = (\nabla\psi)^{-1}(\nabla\psi(W_\tau) - \nabla_\tau)$ where $\psi(W) = \frac{1}{2}\|W\|_{2,r}^2$. Recall that the group $(2,r)$-norm is the ℓ_r norm of the ℓ_2 norm of the columns of W. We set $r = \log(p)/(\log(p) - 1)$. The mapping $\nabla\psi$ and its inverse can both be easily computed (see, e.g., [17, Eq. (2)]).

Corollary 8. *Suppose* $\mathbf{L} = \mathbf{L}_{\text{NDCG}}$ *and* $\text{rep}(\sigma)$ *is as in Lemma 4. Then, assuming the dataset is linearly separable with margin* γ *by a unit group norm* $W^\star$ *(*$\|W^\star\|_{2,1} = 1$*), the sequence generated by Algorithm 3 with* ψ *chosen as above, satisfies*

$$\sum_{\tau=1}^{n} \mathbf{L}_{\text{NDCG}}(X_\tau w_\tau, y_\tau) \leq \frac{9 \cdot 2^{Y_{\max}+3} \cdot m^2 \log_2^2(2m) \cdot R^2 \cdot \log p}{\gamma^2}$$

where $\|X_\tau\|_\infty \leq R$*.*

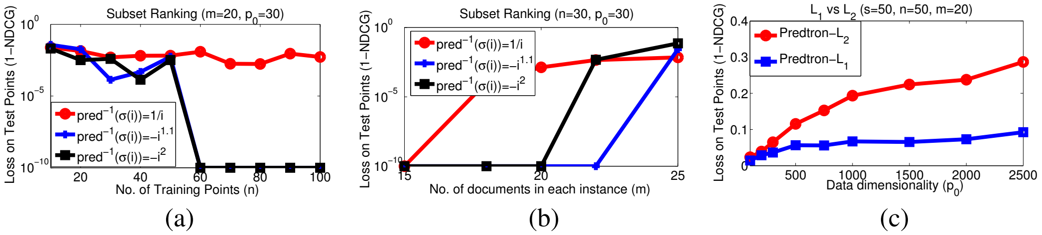

Figure 1: Subset Ranking: NDCG loss for different pred^{-1} choices with varying n (Plot (a)) and m (Plot (b)). As predicted by Lemmas 4 and 5, $\mathrm{pred}^{-1}(\sigma_i) = -i^{1.1}$ is more accurate than $1/i$. (c): L_1 vs L_2 margin. $\mathbf{L}_{\mathrm{NDCG}}$ for two different Predtron algorithms based on L_1 and L_2 margin. Data is generated using L_1 margin notion but with varying sparsity of the optimal scoring function w^*.

6 Experiments

We now present simulation results to demonstrate the application of our proposed Predtron framework to subset ranking. We also demonstrate that empirical results match the trend predicted by our error bounds, hence hinting at tightness of our (upper) bounds. Due to lack of space, we focus only on the subset ranking problem. Also, we would like to stress that we do not claim that the basic version of Predtron itself (with $\eta = 1$) provides a state-of-the-art ranker. Instead, we wish to demonstrate the applicability and flexibility of our framework in a controlled setting.

We generated n data points $X_\tau \in \mathbb{R}^{m \times p_0}$ using a Gaussian distribution with independent rows. The ith row of X_τ represents a document and is sampled from a spherical Gaussian centered at μ_i. We selected a $w^* \in \mathbb{R}^{p_0}$ and also a set of thresholds $[\zeta_1, \ldots, \zeta_{m+1}]$ to generate relevance scores; we set $\zeta_j = \frac{1}{j}, \forall 2 \leq j \leq m$ and $\zeta_1 = +\infty$ and $\zeta_{m+1} = -\infty$. We set relevance score $y_\tau(i)$ of the ith document in the τth document-set as: $y_\tau(i) = m - j$ iff $\zeta_{j+1} \leq \langle X_\tau(i), w^* \rangle \leq \zeta_j$. That is, $y_\tau(i) \in [m-1]$.

We measure performance of a given method using the NDCG loss $\mathbf{L}_{\mathrm{NDCG}}$ defined in Section 4. Note that $\mathbf{L}_{\mathrm{NDCG}}$ is less sensitive to errors in predictions for the less relevant documents in the list. On the other hand, our selection of thresholds ζ_i's implies that the *gap* between scores of lower-ranked documents is very small compared to the higher-ranked ones, and hence chances of making mistakes lower down the list is higher.

Figure 1 (a) shows $\mathbf{L}_{\mathrm{NDCG}}$ (on a test set) for our Predtron algorithm (see Section 4) but with different pred^{-1} functions. For $\mathrm{pred}^{-1}(\sigma(i)) = f_2(\sigma) = -i^{1.1}$, $f_2(i-1) - f_2(i)$ is monotonically increasing with i. On the other hand, for $\mathrm{pred}^{-1}(\sigma(i)) = f_1(\sigma) = 1/i$, $f_1(i-1) - f_1(i)$ is monotonically decreasing with i. Lemma 4 shows that the mistake bound (in terms of $\mathbf{L}_{\mathrm{NDCG}}$) of Predtron is better when pred^{-1} function is selected to be $f_2(\sigma(i)) = -i^{1.1}$ (as well as for $f_3(\sigma(i)) = -i^2$) instead of $f_1(\sigma(i)) = 1/i$. Clearly, Figure 1 (a) empirically validates this mistake bound with $\mathbf{L}_{\mathrm{NDCG}}$ going to almost 0 for f_2 and f_3 with just 60 training points, while f_1 based Predtron has large loss even with $n = 100$ training points.

Next, we fix the number of training instances to be $n = 30$ and vary the number of documents m. As the gap between ζ_i's decreases for larger i, increasing m implies reducing the margin. Naturally, Predtron with the above mentioned inverse functions has monotonically increasing loss (see Figure 1 (b)). However, f_2 and f_3 provide zero-loss solutions for larger m when compared to f_1.

Finally, we conduct an experiment to show that by selecting appropriate notion of margin, Predtron can obtain more accurate solutions. To this end, we generate data from $[-1, 1]^{p_0}$ and select a sparse w^*. Now, Predtron with ℓ_2-margin notion, i.e., standard gradient descent has $\sqrt{p_0}$ dependency in the error bounds while the ℓ_1-margin (see Corollary 6) has only $s \log(p_0)$ dependence. This error dependency is also revealed by Figure 1 (c), where increasing p_0 with fixed s leads to minor increase in the loss for ℓ_1-based Predtron but leads to significantly higher loss for ℓ_2-based Predtron.

Acknowledgments

A. Tewari acknowledges the support of NSF under grant IIS-1319810.

References

[1] Harish G. Ramaswamy and Shivani Agarwal. Classification calibration dimension for general multiclass losses. In *Advances in Neural Information Processing Systems*, pages 2078–2086, 2012.

[2] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2012.

[3] Michael Collins. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 1–8, 2002.

[4] Nathan D. Ratliff, J Andrew Bagnell, and Martin Zinkevich. (Approximate) subgradient methods for structured prediction. In *International Conference on Artificial Intelligence and Statistics*, pages 380–387, 2007.

[5] Youssef Mroueh, Tomaso Poggio, Lorenzo Rosasco, and Jean-Jeacques Slotine. Multiclass learning with simplex coding. In *Advances in Neural Information Processing Systems*, pages 2789–2797, 2012.

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

[7] Albert B.J. Novikoff. On convergence proofs on perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata*, volume 12, pages 615–622, 1962.

[8] Yaoyong Li, Hugo Zaragoza, Ralf Herbrich, John Shawe-Taylor, and Jaz S. Kandola. The perceptron algorithm with uneven margins. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 379–386, 2002.

[9] Gunnar Ratsch and Jyrki Kivinen. Extended classification with modified Perceptron, 2002. Presented at the NIPS 2002 Workshop: Beyond Classification and Regression: Learning Rankings, Preferences, Equality Predicates, and Other Structures; abstract available at `http://www.cs.cornell.edu/people/tj/ranklearn/raetsch_kivinen.pdf`.

[10] Koby Crammer and Yoram Singer. Ultraconservative online algorithms for multiclass problems. *The Journal of Machine Learning Research*, 3:951–991, 2003.

[11] Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research*, 2:265–292, 2002.

[12] Koby Crammer and Yoram Singer. Pranking with ranking. *Advances in Neural Information Procession Systems*, 14:641–647, 2002.

[13] David Cossock and Tong Zhang. Statistical analysis of bayes optimal subset ranking. *IEEE Transactions on Information Theory*, 54(11):5140–5154, 2008.

[14] Ambuj Tewari and Sougata Chaudhuri. Generalization error bounds for learning to rank: Does the length of document lists matter? In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *JMLR Workshop and Conference Proceedings*, 2015.

[15] Koby Crammer and Yoram Singer. A family of additive online algorithms for category ranking. *The Journal of Machine Learning Research*, 3:1025–1058, 2003.

[16] Eneldo Loza Mencía and Johannes Furnkranz. Pairwise learning of multilabel classifications with perceptrons. In *IEEE International Joint Conference on Neural Networks*, pages 2899–2906, 2008.

[17] Sham M. Kakade, Shai Shalev-Shwartz, and Ambuj Tewari. Regularization techniques for learning with matrices. *Journal of Machine Learning Research*, 13:1865–1890, 2012.

A Proof of Theorem 1

Before we prove the main result, we provide a couple of useful lemmas. The first shows that $\phi(t, y)$ is an upper bound on $\mathbf{L}(t, y)$.

Lemma 9. *For any $t \in \mathbb{R}^d, y \in [\ell]$, we have $\phi(t, y) \geq \mathbf{L}(t, y)$.*

Proof. We have,

$$\begin{aligned}
&\max_{\sigma \in [k]} \left(\mathbf{L}(\sigma, y) - \left\langle \operatorname{pred}^{-1}(\sigma_y) - \operatorname{pred}^{-1}(\sigma), t\right\rangle\right) \\
&\geq \mathbf{L}(\operatorname{pred}(t), y) - \left\langle \operatorname{pred}^{-1}(\sigma_y) - \operatorname{pred}^{-1}(\operatorname{pred}(t)), t\right\rangle \\
&\geq \mathbf{L}(t, y).
\end{aligned}$$

Note that the last step is by the similarity maximization property of $\operatorname{pred}^{-1}(\operatorname{pred}(t))$. □

The next lemma proves a key self-bounding property of the derivative of $\phi(t, y)$, w.r.t. t, that is crucial for the analysis of the generalized perceptron algorithm to go through.

Lemma 10. *Fix a $t \in \mathbb{R}^d, y \in [\ell]$ such that $\mathbf{L}(t, y) > 0$. Then, we have*

$$\|\nabla_t \phi(t, y)\|_2^2 \leq \frac{4}{c_{\mathbf{L}}} \mathbf{L}(t, y)$$

Proof. Note that $\nabla_t \phi(t, y)$ is

$$\operatorname{pred}^{-1}(\tilde{\sigma}_{t,y}) - \operatorname{pred}^{-1}(\sigma_y)$$

where

$$\tilde{\sigma}_{t,y} = \operatorname*{argmax}_{\sigma \in [k]} \left(\mathbf{L}(\sigma, y) - \left\langle \operatorname{pred}^{-1}(\sigma_y) - \operatorname{pred}^{-1}(\sigma), t\right\rangle\right)$$

So, $\|\nabla_t \phi(t, y)\|_2^2 \leq (\|\operatorname{pred}^{-1}(\tilde{\sigma}_{t,y})\|_2 + \|\operatorname{pred}^{-1}(\sigma_y)\|_2)^2 \leq 4$. On the other hand, on a mistake round, $\mathbf{L}(t, y) \geq c_{\mathbf{L}}$. □

Finally, the lemma below states that if t gets large enough margin on a label y then $\phi(t, y)$ is zero.

Lemma 11. *If $t \in \mathbb{R}^d$ has margin $\gamma \geq C_{\mathbf{L}}$ on y, then $\phi(t, y) = 0$.*

Proof. Note that for any $\sigma \notin \Sigma_y$,

$$\mathbf{L}(\sigma, y) - \left\langle \operatorname{pred}^{-1}(\sigma_y) - \operatorname{pred}^{-1}(\sigma), t\right\rangle \leq \mathbf{L}(\sigma, y) - \gamma \leq \mathbf{L}(\sigma, y) - C_{\mathbf{L}} \leq 0$$

For any $\sigma'_y \in \Sigma_y$, we have $\mathbf{L}(\sigma'_y, y) = 0$ and

$$\left\langle \operatorname{pred}^{-1}(\sigma_y), t\right\rangle = \left\langle \operatorname{pred}^{-1}(\sigma'_y), t\right\rangle = \max_{\sigma} \left\langle \operatorname{pred}^{-1}(\sigma), t\right\rangle.$$

Therefore, $\phi(t, y) = 0$. □

Now we have all the ingredients to prove the main result.

Proof of Theorem 1. Recall that a mistake round is one where $\mathbf{L}(W_\tau X_\tau, Y_\tau) > 0$. Define the following sequence of convex functions:

$$f_\tau(W) = \begin{cases} 0 & \text{on non-mistake round} \\ W \mapsto \phi(W X_\tau, Y_\tau) & \text{on mistake round} \end{cases}$$

Consider online gradient descent (OGD) updates: $W_{\tau+1} = W_\tau - \eta \nabla f_\tau(W_\tau)$. Standard OGD analysis (see, e.g., [6, Eq. (2.15)]) implies that, for any W (we will deal with the issue of choosing η shortly):

$$\sum_{\tau=1}^{n} f_\tau(W_\tau) \leq \sum_{\tau=1}^{n} f_\tau(W) + \frac{\eta}{2} \sum_{\tau=1}^{n} \|\nabla_\tau\|_F^2 + \frac{\|W\|_F^2}{2\eta} \tag{3}$$

where $\nabla_\tau = \nabla_W f_\tau(W_\tau)$.

On non-mistake rounds, the gradient as well as loss, are both zero. On mistake rounds, the gradient is

$$\nabla_\tau = \nabla_W f_\tau(W_\tau) = \nabla_t \phi(W_\tau X_\tau, Y_\tau) X_\tau^\top$$

and therefore

$$\|\nabla_\tau\|_F^2 \leq \frac{4}{c_{\mathbf{L}}} R^2 \mathbf{L}(W_\tau X_\tau, Y_\tau)$$

by the self-bounding property (Lemma 10) and boundedness of X_τ. Therefore, we have

$$\frac{\eta}{2} \sum_{\tau=1}^{n} \|\nabla_\tau\|_F^2 \leq \frac{2\eta R^2}{c_{\mathbf{L}}} \sum_{\tau=1}^{n} \mathbf{L}(W_\tau X_\tau, Y_\tau) \tag{4}$$

On non-mistake rounds, f_τ as well as loss, are both zero. On mistake rounds,

$$f_\tau(W_\tau) = \phi(W_\tau X_\tau, Y_\tau) \geq \mathbf{L}(W_\tau X_\tau, Y_\tau)$$

by upper bound property of ϕ (Lemma 9). So we also have

$$\sum_{\tau=1}^{n} \mathbf{L}(W_\tau X_\tau, Y_\tau) \leq \sum_{\tau=1}^{n} f_\tau(W_\tau) \tag{5}$$

Now plugging in (5) and (4) into (3), we get

$$\sum_{\tau=1}^{n} \mathbf{L}(W_\tau X_\tau, Y_\tau) \leq \sum_{\tau=1}^{n} f_\tau(W) + \frac{2\eta R^2}{c_{\mathbf{L}}} \sum_{\tau=1}^{n} \mathbf{L}(W_\tau X_\tau, Y_\tau) + \frac{\|W\|_F^2}{2\eta}$$

By assumption, the sequence (X_τ, y_τ) is linearly separable with margin γ. That is, there exists a $W^\star$ with margin γ on (X_τ, y_τ). By the scaling property of margin, this means that $W = C_{\mathbf{L}} W^\star/\gamma$ has margin $C_{\mathbf{L}}$ on (X_τ, y_τ). For this W, by Lemma 11, we have $\sum_{\tau=1}^{n} f_\tau(W) = 0$. Since $\|W\|_F^2 \leq C_{\mathbf{L}}^2/\gamma^2$, we have the bound

$$\left(1 - \frac{2\eta R^2}{c_{\mathbf{L}}}\right) \sum_{\tau=1}^{n} \mathbf{L}(W_\tau X_\tau, Y_\tau) \leq \frac{C_{\mathbf{L}}^2}{2\gamma^2 \eta}$$

and choosing $\eta = c_{\mathbf{L}}/(4R^2)$ gives the bound

$$\sum_{\tau=1}^{n} \mathbf{L}(W_\tau X_\tau, Y_\tau) \leq \frac{4R^2 C_{\mathbf{L}}^2}{c_{\mathbf{L}} \gamma^2}$$

□

B Algorithm Variants

We provide two variants of Predtron. First, we present Predtron.LD, a loss driven version that uses a surrogate that is not dependent on the loss but incorporates the loss in the stepsize. Then, we present Predtron.Link, a version that allows for margin to be defined w.r.t. an arbitrary norm and uses an appropriate link function in its updates.

B.1 Choice of Surrogate

Consider using the surrogate:

$$\phi_1(t, y) = \max\{0, 1 + \max_{\sigma \notin \Sigma_y} \langle \mathrm{pred}^{-1}(\sigma), t\rangle - \langle \mathrm{pred}^{-1}(\sigma_y), t\rangle\}$$

For any y, $\phi_1(t, y)$ is obviously non-negative and convex in t. Moreover, when a mistake is made this surrogate is at least 1.

Lemma 12. *Suppose t, y are such that $\mathbf{L}(t, y) > 0$. Then $\phi_1(t, y) \geq 1$.*

Proof. Since $\mathbf{L}(t, y) > 0$, there exists $\sigma \notin \Sigma_y$ such that $\langle \mathrm{pred}^{-1}(\sigma), t \rangle = \max_{\sigma'} \langle \mathrm{pred}^{-1}(\sigma'), t \rangle$. Therefore, we have

$$\max_{\sigma \notin \Sigma_y} \langle \mathrm{pred}^{-1}(\sigma), t \rangle \geq \langle \mathrm{pred}^{-1}(\sigma_y), t \rangle$$

and therefore $\phi_1(t, y) \geq 1$. □

The surrogate ϕ_1 is also zero given large enough margin.

Lemma 13. *If $t \in \mathbb{R}^d$ has margin $\gamma \geq 1$ on y, then $\phi_1(t, y) = 0$.*

Proof. Note that for any $\sigma \notin \Sigma_y$,

$$1 - \langle \mathrm{pred}^{-1}(\sigma_y) - \mathrm{pred}^{-1}(\sigma), t \rangle \leq 1 - \gamma \leq 0.$$

Therefore,

$$1 + \max_{\sigma \notin \Sigma_y} \langle \mathrm{pred}^{-1}(\sigma), t \rangle - \langle \mathrm{pred}^{-1}(\sigma_y), t \rangle \leq 0$$

and hence $\phi_1(t, y) = 0$. □

Algorithm 2 Predtron.LD: A Loss Driven Version of Predtron

```
$W_1 \leftarrow \mathbf{0}$
for $\tau = 1, 2, \ldots$ do
    Receive $X_\tau \in \mathbb{R}^p$
    Predict $\sigma_\tau = \mathrm{pred}(W_\tau X_\tau) \in [k]$
    Receive label $y_\tau \in [\ell]$
    if $\mathbf{L}(\sigma_\tau, y_\tau) > 0$ then
        $(t, y) = (W_\tau X_\tau, y_\tau)$
        $\tilde{\sigma}_\tau = \mathrm{argmax}_{\sigma \notin \Sigma_y} \langle \mathrm{pred}^{-1}(\sigma), t \rangle \in [k]$
        $\nabla_\tau = (\mathrm{pred}^{-1}(\tilde{\sigma}_\tau) - \mathrm{pred}^{-1}(\sigma_y)) \cdot X_\tau^\top \in \mathbb{R}^{d \times p}$
        $W_{\tau+1} = W_\tau - \eta\, \mathbf{L}(\sigma_\tau, y_\tau) \cdot \nabla_\tau$
    else
        $W_{\tau+1} = W_\tau$
    end if
end for
```

Theorem 14. *Suppose the dataset $(X_1, y_1), \ldots, (X_n, y_n)$ is linearly separable with margin γ. Then the sequence W_τ generated by Algorithm 2 with $\eta = 1/(4C_{\mathbf{L}}R^2)$ satisfies the loss bound*

$$\sum_{\tau=1}^{n} \mathbf{L}(W_\tau X_\tau, y_\tau) \leq \frac{4R^2 C_{\mathbf{L}}}{\gamma^2}$$

where $\|X_\tau\|_2 \leq R$ for all τ.

Proof. As before, let a mistake round be one where $\mathbf{L}(W_\tau X_\tau, Y_\tau) > 0$. Let $\mathbf{L}_\tau = \mathbf{L}(W_\tau X_\tau, y_\tau)$. Define the following sequence of convex functions:

$$f_\tau(W) = \mathbf{L}_\tau \cdot \phi_1(W X_\tau, y_\tau).$$

Algorithm 2 is simply running online gradient descent (OGD) updates: $W_{\tau+1} = W_\tau - \nabla f_\tau(W_\tau)$. This is trivial to see for non-mistake round. On mistake rounds, observe that the outer maximum in the definition of ϕ_1 is not achieved at 0 and hence the gradient is given by

$$\nabla_t \phi(t, y) = \mathrm{pred}^{-1}(\tilde{\sigma}) - \mathrm{pred}^{-1}(\sigma_y)$$

where

$$\tilde{\sigma} = \underset{\sigma \notin \Sigma_y}{\mathrm{argmax}} \langle \mathrm{pred}^{-1}(\sigma), t \rangle .$$

Setting $\nabla_\tau = \nabla_W f_\tau(W_\tau)$ and using standard OGD analysis (see, e.g., [6, Eq. (2.15)]) we get that, for any W:

$$\begin{aligned}
\sum_{\tau=1}^n f_\tau(W_\tau) &\le \sum_{\tau=1}^n f_\tau(W) + \frac{\eta}{2}\sum_{\tau=1}^n \|\nabla_\tau\|_F^2 + \frac{\|W\|_F^2}{2\eta} \\
&\le \sum_{\tau=1}^n f_\tau(W) + \frac{\eta}{2}\sum_{\tau=1}^n \mathbf{L}_\tau^2 \cdot \|\nabla_t \phi_1(X_\tau W_\tau, y_\tau)\|_2^2 \cdot \|X_\tau\|_2^2 + \frac{\|W\|_F^2}{2\eta} \\
&\le \sum_{\tau=1}^n f_\tau(W) + \frac{\eta}{2}\sum_{\tau=1}^n C_{\mathbf{L}} \cdot \mathbf{L}_\tau \cdot 4 \cdot R^2 + \frac{\|W\|_F^2}{2\eta} \\
&= \sum_{\tau=1}^n f_\tau(W) + 2\eta C_{\mathbf{L}} R^2 \sum_{\tau=1}^n \mathbf{L}_\tau + \frac{\|W\|_F^2}{2\eta}.
\end{aligned}$$

By Lemma 12, we know that $\mathbf{L}_\tau \le f_\tau(W_\tau)$. Further, by assumption, the sequence (X_τ, y_τ) is linearly separable with margin γ. That is, there exists a $W^\star$ with margin γ on (X_τ, y_τ). By the scaling property of margin, this means that $W = W^\star/\gamma$ has margin 1 on (X_τ, y_τ). For this W, by Lemma 13, we have $\sum_{\tau=1}^n f_\tau(W) = 0$. Since $\|W\|_F^2 \le 1/\gamma^2$, we have the bound

$$\left(1 - 2\eta C_{\mathbf{L}} R^2\right) \sum_{\tau=1}^n \mathbf{L}_\tau \le \frac{1}{2\gamma^2 \eta}$$

and choosing $\eta = 1/(4 C_{\mathbf{L}} R^2)$ gives the bound

$$\sum_{\tau=1}^n \mathbf{L}(W_\tau X_\tau, Y_\tau) \le \frac{4R^2 C_{\mathbf{L}}}{\gamma^2}.$$

□

B.2 Choice of Norm

Algorithm 3 Predtron.Link: Predtron with a Link Function $(\nabla\psi)^{-1}$

```
$\Theta_1 \leftarrow \mathbf{0}; W_1 = (\nabla\psi)^{-1}(\Theta_1)$
for $\tau = 1, 2, \ldots$ do
    Receive $X_\tau \in \mathbb{R}^p$
    Predict $\sigma_\tau = \mathrm{pred}(W_\tau X_\tau) \in [k]$
    Receive label $y_\tau \in [\ell]$
    if $\mathbf{L}(\sigma_\tau, y_\tau) > 0$ then
        $(t, y) = (W_\tau X_\tau, y_\tau)$
        $\tilde{\sigma}_\tau = \mathrm{argmax}_{\sigma \in [k]} \left(\mathbf{L}(\sigma, y) - \langle \mathrm{pred}^{-1}(\sigma_y) - \mathrm{pred}^{-1}(\sigma), t \rangle\right) \in [k]$
        $\nabla_\tau = (\mathrm{pred}^{-1}(\tilde{\sigma}_\tau) - \mathrm{pred}^{-1}(\sigma_y)) \cdot X_\tau^\top \in \mathbb{R}^{d \times p}$
        $\Theta_{\tau+1} = \Theta_\tau - \eta\,\nabla_\tau; W_{\tau+1} = (\nabla\psi)^{-1}(\Theta_{\tau+1})$
    else
        $\Theta_{\tau+1} = \Theta_\tau; W_{\tau+1} = W_\tau$
    end if
end for
```

Let $\|\cdot\|$ be a norm and $\psi(W) = \frac{1}{2}\|W\|^2$ be α-strongly convex w.r.t. $\|\cdot\|$. Note that $\psi(\mathbf{0}) = 0$ and $\nabla\psi(cW) = c\nabla\psi(W)$ for $c > 0$. Consider Algorithm 3, a version of Algorithm 1 that uses inverse of the mapping $\nabla\psi$ to generate iterates. Since ψ is strongly convex, the mapping $\nabla\psi$ is indeed invertible. Let $\|\cdot\|_\star$ be the norm dual to $\|\cdot\|$. Since the gradients ∇_τ are rank one, we need one additional property. Suppose there exists a norm $|\!|\!|\cdot|\!|\!|$ such that, for any $u \in \mathbb{R}^d, v \in \mathbb{R}^p$, we have

$$\|uv^\top\|_\star \le \|u\|_2 \cdot |\!|\!|v|\!|\!| \tag{6}$$

We can now prove a loss bound for Algorithm 3.

Theorem 15. *Suppose $\psi, \|\cdot\|, \|\cdot\|_\star, |||\cdot|||$ are as above. In particular, let ψ be α-strongly convex w.r.t. $\|\cdot\|$. Suppose the dataset $(X_1, y_1), \ldots, (X_n, y_n)$ is linearly separable by a unit norm $W^\star$ ($\|W^\star\| = 1$), by margin γ. Then, Algorithm 3 with $\eta = \alpha c_{\mathbf{L}}/(4R^2)$ satisfies the loss bound*

$$\sum_{\tau=1}^{n} \mathbf{L}(W_\tau X_\tau, Y_\tau) \leq \frac{4R^2 C_{\mathbf{L}}^2}{\alpha c_{\mathbf{L}} \gamma^2}$$

where $|||X_\tau||| \leq R$.

Proof. Recall that a mistake round is one where $\mathbf{L}(W_\tau X_\tau, Y_\tau) > 0$. Define the following sequence of convex functions:

$$f_\tau(W) = \begin{cases} 0 & \text{on non-mistake round} \\ W \mapsto \phi(W X_\tau, Y_\tau) & \text{on mistake round} \end{cases}$$

Consider online mirror descent (OMD) updates: $\nabla\psi(W_{\tau+1}) = \nabla\psi(W_\tau) - \eta \nabla f_\tau(W_\tau)$. Standard OMD analysis (see, e.g., [6, Theorem 2.21]) implies that, for any W (we will deal with the issue of choosing η shortly):

$$\sum_{\tau=1}^{n} f_\tau(W_\tau) \leq \sum_{\tau=1}^{n} f_\tau(W) + \frac{\eta}{2\alpha} \sum_{\tau=1}^{n} \|\nabla_\tau\|_\star^2 + \frac{\|W\|^2}{2\eta} \tag{7}$$

where $\nabla_\tau = \nabla_W f_\tau(W_\tau)$.

On non-mistake rounds, the gradient as well as loss, are both zero. On mistake rounds, the gradient is

$$\nabla_\tau = \nabla_W f_\tau(W_\tau) = \nabla_t \phi(W_\tau X_\tau, Y_\tau) X_\tau^\top$$

and therefore

$$\|\nabla_\tau\|_\star^2 \leq \|\nabla_t \phi(W_\tau X_\tau, Y_\tau)\|_2^2 \cdot |||X_\tau|||^2 \leq \frac{4}{c_{\mathbf{L}}} R^2 \mathbf{L}(W_\tau X_\tau, Y_\tau)$$

by the inequality (6), the self-bounding property (Lemma 10), and boundedness of X_τ. Therefore, we have

$$\frac{\eta}{2\alpha} \sum_{\tau=1}^{n} \|\nabla_\tau\|_F^2 \leq \frac{2\eta R^2}{\alpha c_{\mathbf{L}}} \sum_{\tau=1}^{n} \mathbf{L}(W_\tau X_\tau, Y_\tau) \tag{8}$$

On non-mistake rounds, f_τ as well as loss, are both zero. On mistake rounds,

$$f_\tau(W_\tau) = \phi(W_\tau X_\tau, Y_\tau) \geq \mathbf{L}(W_\tau X_\tau, Y_\tau)$$

by upper bound property of ϕ (Lemma 9). So we also have

$$\sum_{\tau=1}^{n} \mathbf{L}(W_\tau X_\tau, Y_\tau) \leq \sum_{\tau=1}^{n} f_\tau(W_\tau) \tag{9}$$

Now plugging in (9) and (8) into (7), we get

$$\sum_{\tau=1}^{n} \mathbf{L}(W_\tau X_\tau, Y_\tau) \leq \sum_{\tau=1}^{n} f_\tau(W) + \frac{2\eta R^2}{\alpha c_{\mathbf{L}}} \sum_{\tau=1}^{n} \mathbf{L}(W_\tau X_\tau, Y_\tau) + \frac{\|W\|^2}{2\eta}$$

By assumption, the sequence (X_τ, y_τ) is linearly separable with margin γ. That is, there exists a $W^\star$ with margin γ on (X_τ, y_τ). By the scaling property of margin, this means that $W = C_{\mathbf{L}} W^\star/\gamma$ has margin $C_{\mathbf{L}}$ on (X_τ, y_τ). For this W, by Lemma 11, we have $\sum_{\tau=1}^{n} f_\tau(W) = 0$. Since $\|W\|^2 \leq C_{\mathbf{L}}^2/\gamma^2$, we have the bound

$$\left(1 - \frac{2\eta R^2}{\alpha c_{\mathbf{L}}}\right) \sum_{\tau=1}^{n} \mathbf{L}(W_\tau X_\tau, Y_\tau) \leq \frac{C_{\mathbf{L}}^2}{2\gamma^2 \eta}$$

and choosing $\eta = \alpha c_{\mathbf{L}}/(4R^2)$ gives the bound

$$\sum_{\tau=1}^{n} \mathbf{L}(W_\tau X_\tau, Y_\tau) \leq \frac{4R^2 C_{\mathbf{L}}^2}{\alpha c_{\mathbf{L}} \gamma^2}$$

□

C Details of Results in Section 3

C.1 Classification with REJECT option

We note that the separability requirement to allow REJECT option is more stringent than the standard classification in the following sense. If a dataset is linearly separable with margin γ for the standard classification, it may no longer be linearly separable with the same margin γ if we allow REJECT option. The other way, however, holds true. This is observed by examining the definition of margin requirement in Section 2.2. Consider $1 \geq \beta_1 \geq \beta_2 > 0$. Then, a score $t \in \mathbb{R}^2$ has a margin $\gamma > 0$ on label $y = 1$, iff $\mathrm{pred}(t) = 1$ and $t_1 \geq \max\left(t_2, \frac{t_1+t_2}{\sqrt{2}}\right) + \gamma$, and a margin $\gamma > 0$ on label $y = 2$, iff $\mathrm{pred}(t) = 2$ and $t_2 \geq \max\left(t_1, \frac{t_1+t_2}{\sqrt{2}}\right) + \gamma$. For the case $\beta_2 = 0$ (i.e. instance of class 2 can be predicted as REJECT without penalty), then a score $t \in \mathbb{R}^2$ has a margin $\gamma > 0$ on label $y = 2$, iff $\mathrm{pred}(t) \in \{2, \text{REJECT}\}$ and $t_1 \leq \min\left(t_2, \frac{t_1+t_2}{\sqrt{2}}\right) - \gamma$.

C.2 Subset losses for Multilabel learning

For a given y and t, using the definition of rep in Section 3 for multilabel learning, the surrogate (2) for a given loss $\mathbf{L}$ can be expressed as:

$$\phi(t, y) = -\left\langle \mathrm{pred}^{-1}(\sigma_y), t\right\rangle + \max_{\mathbf{v} \in \{+1,-1\}^m} \mathbf{L}(\mathbf{v}, y) + \langle \mathbf{v}, t\rangle .$$

Clearly, we can compute the surrogate (and its gradient) efficiently if we can compute the max efficiently. Define the indicator function $I(P) = 1$ if predicate P is true or 0 otherwise. Let:

$$a = \sum_i I(v_i = 1) I(i \in y), \qquad b = \sum_i I(v_i = 1) I(i \notin y),$$
$$c = \sum_i I(v_i = -1) I(i \in y), \qquad d = \sum_i I(v_i = -1) I(i \notin y).$$

In the following, we show that the max in the surrogate can be computed in time $O(m^2)$, for any loss which can be expressed as a function of a, b, c and d, i.e.

$$\max_{\mathbf{v} \in \{+1,-1\}^m} f(a, b, c, d) + \langle \mathbf{v}, t\rangle .$$

The three subset losses listed in main text take this form: $\mathbf{L}_{\text{IsErr}}(\mathbf{v}, y) = I(b = 0)I(c = 0)$, $\mathbf{L}_{\text{Ham}}(\mathbf{v}, y) = b + c$, and $\mathbf{L}_{\text{ErrSetSize}}(\mathbf{v}, y) = bc$. The key idea is that though the max itself is over 2^m quantities, there are only $O(m^2)$ possible values for $C = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ — note that fixing a (where $0 \leq a \leq m$) also fixes $c = |y| - a$, and similarly fixing d (where $0 \leq d \leq m$) also fixes $b = m - |y| - d$. For any fixed C (i.e. fixing a, b, c and d), let $\mathcal{V}_{abcd}$ denote the set of vectors $\mathbf{v}$ that yield C. We can compute $\max_{\mathbf{v} \in \mathcal{V}_C} \langle \mathbf{v}, t\rangle$ in closed form, because the objective to be maximized is linear. Let $\mathcal{I}_{pos}$ denote the classes in y sorted in decreasing order of t. Let $\mathcal{I}_{neg}$ denote the classes *not* in y sorted in decreasing order of t. Now, for $\mathbf{v}$ to be optimal, we set the values corresponding to the first a indices in $\mathcal{I}_{pos}$ to 1, the remaining indices in $\mathcal{I}_{pos}$ to -1, the last d indices in $\mathcal{I}_{neg}$ to -1, and the remaining indices in $\mathcal{I}_{neg}$ to $+1$.

The procedure can be implemented with two `for` loops, where in the innermost `for` loop, we will set $\mathbf{v}_C^*$ that maximizes $\max_{\mathbf{v} \in \mathcal{V}_C} \langle \mathbf{v}, t\rangle$, compute $J_C = f(a, b, c, d) + \langle \mathbf{v}_C^*, t\rangle$ and keep track of the best J_C so far. Finally we note that faster implementations can be obtained for specific functions $f(a, b, c, d)$.

D Proofs for Results in Section 4

Proof of Lemma 3. Both the surrogate as well as its gradient (w.r.t. t) can be computed if we can compute

$$\tilde{\sigma}_{t,y} = \operatorname*{argmax}_{\sigma \in [k]} \left(\mathbf{L}(\sigma, y) - \left\langle \mathrm{pred}^{-1}(\sigma_y) - \mathrm{pred}^{-1}(\sigma), t\right\rangle\right)$$

Let $\mathbf{L}(\sigma, y)$ be a loss derived from an NDCG type gain function. That is, let

$$\mathbf{L}(\sigma, y) = 1 - \frac{1}{W(y)} \sum_{i=1}^{m} \frac{F(y(i))}{G(\sigma(i))}$$

for some monotonically increasing functions F, G and

$$W(r) = \max_{\sigma} \sum_{i=1}^{m} \frac{F(y(i))}{G(\sigma(i))}.$$

Note that $W(r)$ can be computed easily by sorting y. Since $\mathrm{rep}(\sigma) = \mathrm{pred}^{-1}(\sigma) = f(\sigma)/Z$, where $Z = \sqrt{\sum_i f^2(i)}$, we have,

$$\begin{aligned}
\tilde{\sigma}_{t,y} &= \operatorname*{argmax}_{\sigma \in [k]} \left(\mathbf{L}(\sigma, y) - \left\langle \mathrm{pred}^{-1}(\sigma_y) - \mathrm{pred}^{-1}(\sigma), t \right\rangle \right) \\
&= \operatorname*{argmax}_{\sigma \in [k]} \left(\mathbf{L}(\sigma, y) + \left\langle \mathrm{pred}^{-1}(\sigma), t \right\rangle \right) \\
&= \operatorname*{argmax}_{\sigma \in [k]} \left(1 - \frac{1}{W(y)} \sum_{i=1}^{m} \frac{F(y(i))}{G(\sigma(i))} + \frac{1}{Z} \sum_{i=1}^{m} f(\sigma(i)) t_i \right) \\
&= \operatorname*{argmax}_{\sigma \in [k]} \left(\sum_{i=1}^{m} \frac{-F(y(i))}{W(y) G(\sigma(i))} + \frac{f(\sigma(i)) t_i}{Z} \right).
\end{aligned}$$

This is a linear assignment problem where the cost $C(i, j)$ of assigning item i to position j is

$$C(i, j) = \frac{-F(y(i))}{W(y) G(j)} + \frac{f(j) t_i}{Z}$$

which can be solved, e.g., using the $O(m^3)$ time complexity Hungarian algorithm (also known as Munkres' algorithm). □

Proof of Lemma 4. Note that y is sorted in decreasing order with relevance grade changes at positions $i_1, \ldots, i_N$. That is, the entries of y obey the following ordering:

$$y(1) = \ldots = y(i_1 - 1) > y(i_1) = \ldots = y(i_2 - 1) > y(i_2) = [\ldots] = y(i_N - 1) > y(i_N) = \ldots = y(m)$$

Define $N + 1$ sets $G_j = \{i_j, \ldots, i_{j+1} - 1\}$ for $j \in \{0, \ldots, k\}$ where $i_0 = 1$ and $i_{N+1} = m + 1$ to handle boundary cases. If t has margin γ on y, it has to be first of all compatible with y. So entries of t in group $j - 1$ should be equal to each other and larger than the entries in group j:

$$t_{i_{j-1}} = \ldots = t_{i_j - 1} > t_{i_j} = \ldots = t_{i_{j+1} - 1}$$

for $j \in [N]$. Moreover, we should have

$$\min_{\sigma \in \Sigma_y} \langle f(\sigma), t \rangle \geq \max_{\sigma' \notin \Sigma_y} \langle f(\sigma), t \rangle + \gamma'$$

where $\gamma' = \gamma Z$ and Z is the normalization needed so that $f(\sigma)/Z$ is a unit vector.

Note that $\sigma' \notin \Sigma_y$ means that there is at least one "bad" pair (i, i') such that $y(i) > y(i')$ (and hence $t_i > t_{i'}$) and yet $\sigma'(i) > \sigma'(i')$. We now claim that the maximum on the RHS is achieved at a $\sigma' \notin \Sigma_y$ with *exactly one* such bad pair. This is because, if we swap a bad pair in σ'' to get a new σ' then $\langle f(\sigma''), t \rangle < \langle f(\sigma'), t \rangle$. So we can eliminate all bad pairs but one. This keeps us outside of Σ_y and increases $\langle f(\sigma'), t \rangle$. We further claim that if there is exactly one bad pair (i, i') then i, i' have to be in adjacent groups with one of them right next to a group boundary. This is because, otherwise, there will have to be bad pairs other than (i, i').

Let the bad pair be in groups $j - 1$ and j, i.e. $i \in G_{j-1}, i' \in G_j$ and $\sigma(i) = i', \sigma(i') = i$. The margin condition then says that,

$$f(i) t_{i_j - 1} + f(i') t_{i_j} \geq f(i') t_{i_j - 1} + f(i) t_{i_j} + \gamma'$$

which means

$$t_{i_j - 1} - t_{i_j} \geq \max_{i \in G_{j-1}, i' \in G_j} \frac{\gamma'}{f(i) - f(i')}$$

Since f is strictly decreasing, the worst case is when $i = i_j - 1, i' = i_j$. This proves the lemma. □

Proof of Corollary 5. The condition $\|X_\tau\|_{\mathrm{op}}$ arises because the form of ∇_τ in the subset ranking setting implies $\|\nabla_\tau\|_2 \le 2\,\|X_\tau\|_{\mathrm{op}}$. For $\mathbf{L} = \mathbf{L}_{\mathrm{NDCG}}$, $C_{\mathbf{L}} = 1$ and $c_{\mathbf{L}}$ can be computed as follows. Let y be sorted in decreasing order of its entries. Minimum non-zero loss occurs if the last two documents are relevant and irrelevant get ranked incorrectly (errors higher up in the ranking will only incur more loss). So the minimum possible non-zero loss for a given y is

$$
\begin{aligned}
\frac{\frac{2^{y(m-1)}-1}{\log_2(1+m-1)} + \frac{2^{y(m)}-1}{\log_2(1+m)} - \left(\frac{2^{y(m-1)}-1}{\log_2(1+m)} + \frac{2^{y(m)}-1}{\log_2(1+m-1)}\right)}{\sum_{i=1}^{m} \frac{2^{y(i)}-1}{\log_2(1+i)}} &\ge \frac{\frac{1}{\log_2 m} - \frac{1}{\log_2(m+1)}}{(2^{Y_{\max}}-1)\sum_{i=1}^{m}\frac{1}{\log_2(1+i)}} \\
&= \frac{\frac{\log_2(1+1/m)}{\log_2 m \cdot \log_2(m+1)}}{(2^{Y_{\max}}-1)\sum_{i=1}^{m}\frac{1}{\log_2(1+i)}} \\
&\ge \frac{\frac{1}{2m\cdot\log_2^2(m+1)}}{(2^{Y_{\max}}-1)\cdot\frac{m}{\log_2 2}} \\
&= \frac{1}{2(2^{Y_{\max}}-1)m^2\log_2^2(m+1)}.
\end{aligned}
$$

Therefore, the bound in Theorem 1 becomes

$$
\frac{4C_{\mathbf{L}}^2 R^2}{c_{\mathbf{L}}\gamma^2} \le \frac{2^{Y_{\max}+3} m^2 \log_2^2(2m) R^2}{\gamma^2}.
$$

□

Proof of Corollary 6. We use Theorem 15 with $\psi(w) = \|w\|_r^2$ where $r = \log p_0/(\log p_0 - 1)$. For such an r, $\|w\|_r \le \|w\|_1 \le 3\|w\|_r$. Also, $\|\cdot\| = \|\cdot\|_r$, $\|\cdot\|_\star = \|\cdot\|_q$ where $q = \log p_0$. Note that ψ is $(r-1)$-strongly convex w.r.t. $\|\cdot\|_r$ (see, e.g., [17]). Since

$$
\|X^\top u\|_q \le 3\|X^\top u\|_\infty \le 3\|X^\top\|_{2\to\infty}\|u\|_2
$$

the norm $|\!|\!|\cdot|\!|\!|$ is simply $3\|X^\top\|_{2\to\infty}$ where $\|X^\top\|_{2\to\infty} = \max_{j=1,\ldots,p}\|X_j\|_2$ (X_j's are columns of X). Since $\|w^\star\|_r \le \|w^\star\|_1$ there obviously exists a unit ℓ_r norm vector that has margin at least γ on the dataset. The bound in Theorem 15 then becomes

$$
\frac{4(3R)^2 C_{\mathbf{L}}^2}{(r-1)c_{\mathbf{L}}\gamma^2} \le \frac{36R^2 \log p_0 C_{\mathbf{L}}^2}{c_{\mathbf{L}}\gamma^2}.
$$

Corollary nows follows by using the bounds for $C_{\mathbf{L}}, c_{\mathbf{L}}$ from proof of Corollary 5. □

E Proofs for Results in Section 5

Proof of Corollary 7. Corollary follows immediately from Theorem 1 and the $C_{\mathbf{L}}, c_{\mathbf{L}}$ calculations in the proof of Corollary 5. □

Proof of Corollary 8. We use Theorem 15 with $\psi(w) = \|W\|_{2,r}^2$ where $r = \log p/(\log p - 1)$. For such an r, $\|W\|_{2,r} \le \|W\|_{2,1} \le 3\|W\|_{2,r}$. Also, $\|\cdot\| = \|\cdot\|_{2,r}$, $\|\cdot\|_\star = \|\cdot\|_{2,q}$ where $q = \log p$. Note that ψ is $(r-1)$-strongly convex w.r.t. $\|\cdot\|_{2,r}$ (see, e.g., [17]). Since

$$
\|uv^\top\|_{2,q} \le 3\|uv^\top\|_{2,\infty} = 3\|u\|_2\|v\|_\infty
$$

the norm $|\!|\!|\cdot|\!|\!|$ is simply $3\|\cdot\|_\infty$. Since $\|W^\star\|_{2,r} \le \|W^\star\|_{2,1}$ there obviously exists a unit group $(2,r)$-norm matrix that has margin at least γ on the dataset. The bound in Theorem 15 then becomes

$$
\frac{4(3R)^2 C_{\mathbf{L}}^2}{(r-1)c_{\mathbf{L}}\gamma^2} \le \frac{36R^2 \log p\, C_{\mathbf{L}}^2}{c_{\mathbf{L}}\gamma^2}.
$$

Corollary nows follows by using the bounds for $C_{\mathbf{L}}, c_{\mathbf{L}}$ from proof of Corollary 5. □