Surrogate Functions for Maximizing Precision at the Top

Purushottam Kar T-PURKAR@MICROSOFT.COM
Microsoft Research, INDIA

Harikrishna Narasimhan[*] HARIKRISHNA@CSA.IISC.ERNET.IN
Indian Institute of Science, Bangalore, INDIA

Prateek Jain PRAJAIN@MICROSOFT.COM
Microsoft Research, INDIA

Abstract

The problem of maximizing precision at the top of a ranked list, often dubbed Precision@k (prec@k), finds relevance in myriad learning applications such as ranking, multi-label classification, and learning with severe label imbalance. However, despite its popularity, there exist significant gaps in our understanding of this problem and its associated performance measure.

The most notable of these is the lack of a convex upper bounding surrogate for prec@k. We also lack scalable perceptron and stochastic gradient descent algorithms for optimizing this performance measure. In this paper we make key contributions in these directions. At the heart of our results is a family of truly upper bounding surrogates for prec@k. These surrogates are motivated in a principled manner and enjoy attractive properties such as consistency to prec@k under various natural margin/noise conditions.

These surrogates are then used to design a class of novel perceptron algorithms for optimizing prec@k with provable mistake bounds. We also devise scalable stochastic gradient descent style methods for this problem with provable convergence bounds. Our proofs rely on novel uniform convergence bounds which require an in-depth analysis of the structural properties of prec@k and its surrogates. We conclude with experimental results comparing our algorithms with state-of-the-art cutting plane and stochastic gradient algorithms for maximizing prec@k.

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37. Copyright 2015 by the author(s).

[*]Part of the work was done while H.N. was an intern at MSR India.

1. Introduction

Ranking a given set of points or labels according to their relevance forms the core of several real-life learning systems. For instance, in classification problems with a rare-class as is the case in spam/anomaly detection, the goal is to rank the given emails/events according to their likelihood of being from the rare-class (spam/anomaly). Similarly, in multi-label classification problems, the goal is to rank the labels according to their likelihood of being relevant to a data point (Tsoumakas & Katakis, 2007).

The ranking of items at the top is of utmost importance in these applications and several performance measures, such as Precision@k, Average Precision and NDCG have been designed to promote accuracy at top of ranked lists. Of these, the Precision@k (prec@k) measure is especially popular in a variety of domains. Informally, prec@k counts the number of relevant items in the top-k positions of a ranked list and is widely used in domains such as binary classification (Joachims, 2005), multi-label classification (Prabhu & Varma, 2014) and ranking (Le & Smola, 2007).

Given its popularity, prec@k has received attention from algorithmic, as well as learning theoretic perspectives. However, there remain specific deficiencies in our understanding of this performance measure. In fact, to the best of our knowledge, there is only one known convex surrogate function for prec@k, namely, the *struct-SVM* surrogate due to (Joachims, 2005) which, as we reveal in this work, is not an upper bound on prec@k in general, and need not recover an optimal ranking even in strictly separable settings.

Our aim in this paper is to develop efficient algorithms for optimizing prec@k for ranking problems with binary relevance levels. Since the intractability of binary classification in the agnostic setting (Guruswami & Raghavendra, 2009) extends to prec@k, our goal would be to exploit natural notions of *benign-ness* usually observed in natural distributions to overcome such intractability results.

1.1. Our Contributions

We make several contributions in this paper that both, give deeper insight into the prec@k performance measure, as well as provide scalable techniques for optimizing it.

Precision@k margin: motivated by the success of margin-based frameworks in classification settings, we develop a family of margin conditions appropriate for the prec@k problem. Recall that the prec@k performance measure counts the number of relevant items at the top k positions of a ranked list. The simplest of our margin notions, that we call the *weak* (k, γ)*-margin*, is said to be present if a privileged set of k relevant items can be separated from all irrelevant items by a margin of γ. This is the least restrictive margin condition that allows for a perfect ranking w.r.t prec@k. Notably, it is much less restrictive than the binary classification notion of margin which requires all relevant items to be separable from all irrelevant items by a certain margin. We also propose two other notions of margin suited to our perceptron algorithms.

Surrogate functions for prec@k: we design a family of three novel surrogates for the prec@k performance measure. Our surrogates satisfy two key properties. Firstly they always upper bound the prec@k performance measure so that optimizing them promotes better performance w.r.t prec@k. Secondly, these surrogates satisfy *conditional consistency* in that they are consistent w.r.t. prec@k under some noise condition. We show that there exists a one-one relationship between the three prec@k margin conditions mentioned earlier and these three surrogates so that each surrogate is consistent w.r.t. prec@k under one of the margin conditions. Moreover, our discussion reveals that the three surrogates, as well as the three margin conditions, lie in a concise hierarchy.

Perceptron and SGD algorithms: using insights gained from the previous analyses, we design two perceptron-style algorithms for optimizing prec@k. Our algorithms can be shown to be a natural extension of the classical perceptron algorithm for binary classification (Rosenblatt, 1958). Indeed, akin to the classical perceptron, both our algorithms enjoy mistake bounds that reduce to crisp convergence bounds under the margin conditions mentioned earlier. We also design a mini-batch-style stochastic gradient descent algorithm for optimizing prec@k.

Learning theory: in order to prove convergence bounds for the SGD algorithm, and online-to-batch conversion bounds for our perceptron algorithms, we further study prec@k and its surrogates and prove uniform convergence bounds for the same. These are novel results and require an in-depth analysis into the involved structure of the prec@k performance measure and its surrogates. However, with these results in hand, we are able to establish crisp convergence bounds for the SGD algorithm, as well as generalization bounds for our perceptron algorithms.

Paper Organization: Section 2 presents the problem formulation and sets up the notation. Section 3 introduces three novel surrogates and margin conditions for prec@k and reveals the interplay between these with respect to consistency to prec@k. Section 4 presents two perceptron algorithms for prec@k and their mistake bounds, as well as a mini-batch SGD-based algorithm. Section 5 discusses uniform convergence bounds for our surrogates and their application to convergence and online-to-batch conversion bounds for our the perceptron and SGD-style algorithms. We conclude with empirical results in Section 6.

1.2. Related Work

There has been much work in the last decade in designing algorithms for bipartite ranking problems. While the earlier methods for this problem, such as RankSVM, focused on optimizing pair-wise ranking accuracy (Herbrich et al., 2000; Joachims, 2002; Freund et al., 2003; Burges et al., 2005), of late, there has been enormous interest in performance measures that promote good ranking performance at the top portion of the ranked list, and in ranking methods that directly optimize these measures (Clémençon & Vayatis, 2007; Rudin, 2009; Agarwal, 2011; Boyd et al., 2012; Narasimhan & Agarwal, 2013a;b; Li et al., 2014).

In this work, we focus on one such evaluation measure – Precision@k, which is widely used in practice. The only prior algorithms that we are aware of that directly optimize this performance measure are a structural SVM based cutting plane method due to (Joachims, 2005), and an efficient stochastic implementation of the same due to (Kar et al., 2014). However, as pointed out earlier, the convex surrogate used in these methods is not well-suited for prec@k.

It is also important to note that the bipartite ranking setting considered in this work is different from other popular forms of ranking such as subset or list-wise ranking settings, which arise in several information retrieval applications, where again there has been much work in optimizing performance measures that emphasize on accuracy at the top (e.g. NDCG) (Valizadegan et al., 2009; Cao et al., 2007; Yue et al., 2007; Le & Smola, 2007; Chakrabarti et al., 2008; Yun et al., 2014). There has also been some recent work on perceptron style ranking methods for list-wise ranking problems (Chaudhuri & Tewari, 2014), but these methods are tailored to optimize the NDCG and MAP measures, which are different from the prec@k measure that we consider here. Other less related works include online ranking algorithms for optimizing ranking measures in an adversarial setting with limited feedback (Chaudhuri & Tewari, 2015).

2. Problem Formulation and Notation

We will be presented with a set of labeled points $(\mathbf{x}_i, y_i), \ldots, (\mathbf{x}_n, y_n)$, where $\mathbf{x}_i \in \mathcal{X}$ and $y_i \in \{0,1\}$. We shall use $\mathbf{X}$ to denote the entire dataset, $\mathbf{X}_+$ and $\mathbf{X}_-$ to denote the set of positive and negatively (null) labeled points, and $\mathbf{y} \in \{0,1\}^n$ to denote the label vector. $\mathbf{z} = (\mathbf{x}, y)$ shall denote a labeled data point. Our results readily extend to multi-label and ranking settings but for sake of simplicity, we focus only on bipartite ranking problems, where the goal is to rank (a subset of) positive examples above the negative ones.

Given n labeled data points $\mathbf{z}_1, \ldots, \mathbf{z}_n$ and a scoring function $s : \mathcal{X} \to \mathbb{R}$, let $\sigma_s \in S_n$ be the permutation that sorts points according to the scores given by s i.e. $s(\mathbf{x}_{\sigma_s(i)}) \geq s(\mathbf{x}_{\sigma_s(j)})$ for $i \leq j$. The Precision@k measure for this scoring function can then be expressed as:

$$\text{prec@k}(s;\ \mathbf{z}_1, \ldots, \mathbf{z}_n) = \sum_{i=1}^{k} (1 - \mathbf{y}_{\sigma_s(i)}). \qquad (1)$$

Note that the above is a "loss" version of the performance measure which penalizes any top-k ranked data points that have a null label. For simplicity, we will use the abbreviated notation $\text{prec@k}(s) := \text{prec@k}(s;\ \mathbf{z}_1, \ldots, \mathbf{z}_n)$. We will also use the shorthand $s_i = s(\mathbf{x}_i)$. For any label vectors $\mathbf{y}', \mathbf{y}'' \in \{0,1\}^n$, we define

$$\begin{aligned} \Delta(\mathbf{y}', \mathbf{y}'') &= \sum_{i=1}^{n} (1 - \mathbf{y}'_i)\mathbf{y}''_i, \\ K(\mathbf{y}', \mathbf{y}'') &= \sum_{i=1}^{n} \mathbf{y}'_i \mathbf{y}''_i. \end{aligned} \qquad (2)$$

Let $n_+(\mathbf{y}') = K(\mathbf{y}', \mathbf{y}') = \|\mathbf{y}'\|_1$ denote the number of positives in the label vector $\mathbf{y}'$ and $n_+ = n_+(\mathbf{y})$ denote the number of actual positives. Let $\mathbf{y}^{(s,k)}$ be the label vector that assigns the label 1 only to the top k ranked items according to the scoring function s. That is, $\mathbf{y}_i^{(s,k)} = 1$ if if $\sigma_s^{-1}(i) \leq k$ and 0 otherwise. It is easy to verify that for any scoring function s, $\Delta(\mathbf{y}, \mathbf{y}^{(s,k)}) = \text{prec@k}(s)$.

3. A Family of Novel Surrogates for prec@k

As prec@k is a non-convex loss function that is hard to optimize directly, it is natural to seek surrogate functions that act as a good proxy for prec@k. There will be two properties that we shall desire of such a surrogate:

1. **Upper Bounding Property**: the surrogate should *upper bound* the prec@k loss function, so that minimizing the surrogate promotes small prec@k loss.
2. **Conditional Consistency**: under some regularity assumptions, optimizing the surrogate should yield an optimal solution for prec@k as well.

Motivated by the above requirements, we develop a family of surrogates which upper bound the prec@k loss function and are consistent to it under certain margin/noise conditions. We note that the results of (Calauzènes et al., 2012) that negate the possibility of consistent convex surrogates for ranking performance measures do not apply to our results since they are neither stated for prec@k, nor do they negate the possibility of conditional consistency.

It is notable that the seminal work of (Joachims, 2005) did propose a convex surrogate for prec@k, that we refer to as $\ell^{\text{struct}}_{\text{prec@k}}(\cdot)$. However, as the discussion below shows, this surrogate is not even an upper bound on prec@k let alone be consistent to it. Understanding the reasons for the failure of this surrogate would be crucial in designing our own.

3.1. The Curious Case of $\ell^{\text{struct}}_{\text{prec@k}}(\cdot)$

The $\ell^{\text{struct}}_{\text{prec@k}}(\cdot)$ surrogate is a part of a broad class of surrogates called *struct-SVM* surrogates that are designed for structured output prediction problems that can have exponentially large output spaces (Joachims, 2005). Given a set of n labeled data points, $\ell^{\text{struct}}_{\text{prec@k}}(\cdot)$ is defined as

$$\max_{\substack{\hat{\mathbf{y}} \in \{0,1\}^n \\ \|\hat{\mathbf{y}}\|_1 = k}} \left\{ \Delta(\mathbf{y}, \hat{\mathbf{y}}) + \sum_{i=1}^{n} (\hat{\mathbf{y}}_i - \mathbf{y}_i)\, s_i \right\}. \qquad (3)$$

The above surrogate penalizes a scoring function if there exists a set of k points with large scores (i.e. the second term is large) which are actually negatives (i.e. the first term is large). However, since the candidate labeling $\hat{\mathbf{y}}$ is restricted to labeling just k points as positive whereas the true label vector $\mathbf{y}$ has n_+ positives, in cases where $n_+ > k$, a non-optimal candidate labeling $\hat{\mathbf{y}}$ can exploit the remaining $n_+ - k$ labels to hide the high scoring negative points, thus confusing the surrogate function. This indicates that this surrogate may not be an upper bound to prec@k. We refer the reader to Appendix A for an explicit example where, not only does this surrogate not upper bound prec@k, but more importantly, minimizing $\ell^{\text{struct}}_{\text{prec@k}}(\cdot)$ does not produce a model that is optimal for prec@k, even in separable settings where all positives points are separated from negatives by a margin.

In the sequel, we shall propose three surrogates, all of which are consistent with prec@k under various noise/margin conditions. The surrogates, as well as the noise conditions, will be shown to form a hierarchy.

3.2. The Ramp Surrogate $\ell^{\text{ramp}}_{\text{prec@k}}(\cdot)$

The key to maximizing prec@k in a bipartite ranking setting is to select a subset of k relevant items and rank them at the top k positions. This can happen iff the *top ranked* k relevant items are not outranked by any irrelevant item. Thus, a surrogate must penalize a scoring function that assigns scores to irrelevant items that are higher than those of the top ranked relevant items. Our *ramp* surrogate $\ell^{\text{ramp}}_{\text{prec@k}}(s)$ implicitly encodes this strategy:

$$\max_{\|\hat{\mathbf{y}}\|_1=k} \left\{ \Delta(\mathbf{y},\hat{\mathbf{y}}) + \sum_{i=1}^n \hat{\mathbf{y}}_i s_i \right\} - \underbrace{\max_{\substack{\|\tilde{\mathbf{y}}\|_1=k \\ K(\mathbf{y},\tilde{\mathbf{y}})=k}} \sum_{i=1}^n \tilde{\mathbf{y}}_i s_i}_{(P)}. \quad (4)$$

The term (P) contains the sum of scores of the k highest scoring positives. Note that $\ell^{\text{ramp}}_{\text{prec@k}}(\cdot)$ is similar to the "ramp" losses for binary classification (Do et al., 2008). We now show that $\ell^{\text{ramp}}_{\text{prec@k}}(\cdot)$ is indeed an upper bounding surrogate for prec@k.

Claim 1. *For any $k \leq n_+$ and scoring function s, we have $\ell^{ramp}_{\text{prec@k}}(s) \geq \text{prec@k}(s)$. Moreover, if $\ell^{ramp}_{\text{prec@k}}(s) \leq \xi$ for a given scoring function s, then there necessarily exists a set $S \subset [n]$ of size at most k such that for all $\|\hat{\mathbf{y}}\|_1 = k$, we have $\sum_{i\in S} s_i \geq \sum_{i=1}^n \hat{\mathbf{y}}_i s_i + \Delta(\mathbf{y},\hat{\mathbf{y}}) - \xi$.*

Proofs for this section are deferred to Appendix B. We can show that this surrogate is conditionally consistent as well. To do so, we introduce the notion of *weak (k,γ)-margin.*

Definition 2 (*Weak* (k,γ)-margin)**.** *A set of n labeled data points satisfies the* weak (k,γ)*-margin condition if for some scoring function s and set $S_+ \subseteq \mathbf{X}_+$ of size k,*

$$\min_{i\in S_+} s_i - \max_{j:\mathbf{y}_j=0} s_j \geq \gamma.$$

Moreover, we say that the function s realizes *this margin. We abbreviate the* weak $(k,1)$*-margin condition as simply the* weak *k-margin condition.*

Clearly, a dataset has a *weak* (k,γ)-margin iff there exist some k positive points that substantially outrank all negatives. Note that this notion of margin is strictly weaker than the standard notion of margin for binary classification as it allows all but those k positives to be completely mingled with the negatives. Moreover, this seems to be one of the most natural notions of margin for prec@k. The following lemma establishes that $\ell^{\text{ramp}}_{\text{prec@k}}(\cdot)$ is indeed consistent w.r.t. prec@k under the *weak* k-margin condition.

Claim 3. *For any scoring function s that realizes the* weak *k-margin over a dataset, $\ell^{ramp}_{\text{prec@k}}(s) = \text{prec@k}(s) = 0$.*

This suggests that $\ell^{\text{ramp}}_{\text{prec@k}}(\cdot)$ is not only a tight surrogate, but tight at the optimal scoring function, i.e. prec@k$(s) = 0$; this along with upper bounding property implies consistency. However, it is also a non-convex function due to the term (P). To obtain convex surrogates, we perform relaxations on this term by first rewriting it as follows:

$$(P) = \sum_{i=1}^n \mathbf{y}_i s_i - \underbrace{\min_{\substack{\tilde{\mathbf{y}}\preceq\mathbf{y} \\ \|\tilde{\mathbf{y}}\|_1=n_+-k}} \sum_{i=1}^n \tilde{\mathbf{y}}_i s_i}_{(Q)}, \quad (5)$$

where $\tilde{\mathbf{y}} \preceq \mathbf{y}$ implies that $\mathbf{y}_i = 0 \Rightarrow \tilde{\mathbf{y}}_i = 0$. Thus, to convexify the surrogate $\ell^{\text{ramp}}_{\text{prec@k}}(\cdot)$, we need to design a convex upper bound on (Q). Notice that the term (Q) contains the sum of the scores of the $n_+ - k$ lowest ranked positive data points. This can be readily upper bounded in several ways which give us different surrogate functions.

3.3. The Max Surrogate $\ell^{\textbf{max}}_{\textbf{prec@k}}(\cdot)$

An immediate convex upper bound on (Q) is obtained by replacing the sum of scores of the $n_+ - k$ lowest ranked positives with those of the highest ranked ones as follows: $(Q) \leq \max_{\substack{\tilde{\mathbf{y}}\preceq(1-\hat{\mathbf{y}})\cdot\mathbf{y} \\ \|\tilde{\mathbf{y}}\|_1=n_+-k}} \sum_{i=1}^n \tilde{\mathbf{y}}_i s_i$, which gives us the $\ell^{\text{max}}_{\text{prec@k}}(s)$ surrogate defined below:

$$\max_{\|\hat{\mathbf{y}}\|_1=k} \left\{ \Delta(\mathbf{y},\hat{\mathbf{y}}) + \sum_{i=1}^n (\hat{\mathbf{y}}_i - \mathbf{y}_i)s_i + \max_{\substack{\tilde{\mathbf{y}}\preceq(1-\hat{\mathbf{y}})\cdot\mathbf{y} \\ \|\tilde{\mathbf{y}}\|_1=n_+-k}} \sum_{i=1}^n \tilde{\mathbf{y}}_i s_i \right\}. \quad (6)$$

The above surrogate, being a point-wise maximum over convex functions, is convex, as well as an upper bound on prec@k(s) since it upper bounds $\ell^{\text{ramp}}_{\text{prec@k}}(s)$. This surrogate can also be shown to be consistent w.r.t. prec@k under the *strong* γ-margin condition defined below for $\gamma = 1$.

Definition 4 (*Strong* γ-margin)**.** *A set of n labeled data points satisfies the γ-strong margin condition if for some scoring function s,* $\min_{i:\mathbf{y}_i=1} s_i - \max_{j:\mathbf{y}_j=0} s_j \geq \gamma$.

We notice that the strong margin condition is actually the standard notion of binary classification margin and hence much stronger than the *weak* (k,γ)-margin condition. It also does not incorporate any elements of the prec@k problem. This leads us to look for tighter convex relaxations to the term (Q) that we do below.

3.4. The Avg Surrogate $\ell^{\textbf{avg}}_{\textbf{prec@k}}(\cdot)$

A tighter upper bound on (Q) can be obtained by replacing (Q) by the average score of the false negatives. Define $C(\hat{\mathbf{y}}) = \frac{n_+ - K(\mathbf{y},\hat{\mathbf{y}})}{n_+ - k}$ and consider the relaxation $(Q) \leq \frac{1}{C(\hat{\mathbf{y}})} \sum_{i=1}^n (1-\hat{\mathbf{y}}_i)\mathbf{y}_i s_i$. Combining this with (4), we get a new convex surrogate $\ell^{\text{avg}}_{\text{prec@k}}(s)$ defined as:

$$\max_{\|\hat{\mathbf{y}}\|_1=k} \left\{ \Delta(\mathbf{y},\hat{\mathbf{y}}) + \sum_{i=1}^n s_i(\hat{\mathbf{y}}_i - \mathbf{y}_i) + \frac{1}{C(\hat{\mathbf{y}})} \sum_{i=1}^n (1-\hat{\mathbf{y}}_i)\mathbf{y}_i s_i \right\}. \quad (7)$$

We refer the reader to Appendix B.4 for a proof that $\ell^{\text{avg}}_{\text{prec@k}}(\cdot)$ is an upper bounding surrogate. It is notable that for $k = n_+$ (i.e. for the PRBEP measure), the surrogate $\ell^{\text{avg}}_{\text{prec@k}}(\cdot)$ recovers Joachims' original surrogate $\ell^{\text{struct}}_{\text{prec@k}}(\cdot)$. To establish conditional consistency of this surrogate, consider the following notion of margin:

Definition 5 ((k,γ)-margin)**.** *A set of n labeled data points satisfies the (k,γ)-margin condition if for some scoring*

$$\begin{tikzcd}
\text{prec@k}(s) \leq & \ell^{\text{ramp}}_{\text{prec@k}}(s) \arrow[r, phantom, "\leq"] & \ell^{\text{avg}}_{\text{prec@k}}(s) \arrow[r, phantom, "\leq"] & \ell^{\text{max}}_{\text{prec@k}}(s) \\
& \text{weak } (k,\gamma) \text{ margin} \arrow[u, Rightarrow] \arrow[r, phantom, "\supseteq"] & (k,\gamma) \text{ margin} \arrow[u, Rightarrow] \arrow[r, phantom, "\supseteq"] & \text{strong } \gamma \text{ margin} \arrow[u, Rightarrow]
\end{tikzcd}$$

Figure 1. A hierarchy among the three surrogates for prec@k and the corresponding margin conditions for conditional consistency.

Algorithm 1 PERCEPTRON@K-AVG

```
Input: Batch length $b$
1:  $\mathbf{w}^0 \leftarrow \mathbf{0}, t \leftarrow 0$
2:  while stream not exhausted do
3:      $t \leftarrow t + 1$
4:      Receive $b$ data points $\mathbf{X}_t = [\mathbf{x}_t^1, \ldots, \mathbf{x}_t^b]$, $\mathbf{y}_t \in \{0,1\}^b$
5:      Calculate $s_t = \mathbf{w}^{t-1}\mathbf{X}_t$ and let $\hat{\mathbf{y}}_t = \mathbf{y}^{(s_t,k)}$
6:      $\Delta_t \leftarrow \Delta(\mathbf{y}_t, \hat{\mathbf{y}}_t)$
7:      if $\Delta_t = 0$ then
8:          $\mathbf{w}^t \leftarrow \mathbf{w}^{t-1}$
9:      else
10:         $D_t \leftarrow \frac{\Delta_t}{\|\mathbf{y}_t\|_1 - K(\mathbf{y}_t,\hat{\mathbf{y}}_t)}$
11:         $\mathbf{w}^t \leftarrow \mathbf{w}^{t-1} - \sum_{i\in[b]}(1-\mathbf{y}_i)\hat{\mathbf{y}}_i \cdot \mathbf{x}_t^i$   {false pos.}
12:         $\mathbf{w}^t \leftarrow \mathbf{w}^t + D_t \cdot \sum_{i\in[b]}(1-\hat{\mathbf{y}}_i)\mathbf{y}_i \cdot \mathbf{x}_t^i$   {false neg.}
13:     end if
14: end while
15: return $\mathbf{w}^t$
```

Algorithm 2 PERCEPTRON@K-MAX

```
10:         $S_t \leftarrow \text{FN}(s, \Delta_t)$
11:         $\mathbf{w}^t \leftarrow \mathbf{w}^{t-1} - \sum_{i\in[b]}(1-\mathbf{y}_i)\hat{\mathbf{y}}_i \cdot \mathbf{x}_t^i$   {false pos.}
12:         $\mathbf{w}^t \leftarrow \mathbf{w}^t + \sum_{i\in S_t} \mathbf{x}_t^i$   {top ranked false neg.}
```

function s, we have, for all sets $S_+ \subseteq \mathbf{X}_+$ of size $n_+ - k + 1$,

$$\frac{1}{n_+ - k + 1}\sum_{i\in S_+} s_i - \max_{j:\mathbf{y}_j=0} s_j \geq \gamma.$$

Moreover, we say that the function s realizes *this margin. We abbreviate the $(k,1)$-margin condition as simply the k-margin condition.*

We can now establish the consistency of $\ell^{\text{avg}}_{\text{prec@k}}(\cdot)$ under the k-margin condition. See Appendix B.5 for a proof.

Claim 6. *For any scoring function s that realizes the k-margin over a dataset, $\ell^{avg}_{\text{prec@k}}(s) = \text{prec@k}(s) = 0$.*

We note that the (k,γ)-margin condition is strictly weaker than the *strong* γ-margin condition (Definition 4) since it still allows a non negligible fraction of the positive points to be assigned a lower score than those assigned to negatives. On the other hand, the (k,γ)-margin condition is strictly stronger than the *weak* (k,γ)-margin condition (Definition 2). The weak k-margin condition only requires one set of k-positives to be separated from the negatives, whereas the above margin condition at least requires the average of *all* positives to be separated from the negatives.

As Figure 1 demonstrates, the three surrogates presented above, as well as their corresponding margin conditions, fall in a neat hierarchy. We will now use these surrogates to formulate two perceptron algorithms with mistake bounds with respect to these margin conditions.

4. Perceptron & SGD Algorithms for prec@k

We now present perceptron-style algorithms for maximizing the prec@k performance measure in bipartite ranking settings. Our algorithms work with a stream of binary labeled points and process them in *mini-batches* of a predetermined size b. Mini-batch methods have recently gained popularity and have been used to optimize ranking loss functions such as $\ell^{\text{struct}}_{\text{prec@k}}(\cdot)$ as well (Kar et al., 2014). It is useful to note that the requirement for mini-batches goes away in ranking and multi-label classification settings, for our algorithms can be applied to individual data points in those settings (e.g. individual queries in ranking settings).

At every time instant t, our algorithms receive a batch of b points $\mathbf{X}_t = [\mathbf{x}_t^1, \ldots, \mathbf{x}_t^b]$ and rank these points using the existing model. Let Δ_t denote the prec@k loss (equation 1) at time t. If $\Delta_t = 0$ i.e. all top k ranks are occupied by positive points, then the model is not updated. Otherwise, the model is updated using the false positives and negatives. For sake of simplicity, we will only look at linear models in this paper. Depending on the kind of updates we make, we get two variants of the perceptron rule for prec@k.

Our first algorithm, PERCEPTRON@K-AVG, updates the model using a combination of all the false positives and negatives (see Algorithm 1). The effect of the update is a very natural one – it explicitly boosts the scores of the positive points that failed to reach the top ranks, and attenuates the scores of the negative points that got very high scores. It is interesting to note that in the limiting case of $k = 1$ and unit batch length (i.e. $b = 1$), the PERCEPTRON@K-AVG update reduces to that of the standard perceptron algorithm (Rosenblatt, 1958; Minsky & Papert, 1988) for the choice $\hat{\mathbf{y}}_t = \text{sign}(s_t)$. Thus, our algorithm can be seen as a natural extension of the classical perceptron algorithm.

The next lemma establishes that, similar to the classical perceptron (Novikoff, 1962), PERCEPTRON@K-AVG also enjoys a mistake bound. Our mistake bound is stated in the most general agnostic setting with the hinge loss function replaced with our surrogate $\ell^{\text{avg}}_{\text{prec@k}}(s)$. All proofs in this section are deferred to Appendix C.

Theorem 7. *Suppose $\|\mathbf{x}_t^i\| \leq R$ for all t, i. Let $\Delta_T^C = \sum_{t=1}^{T}\Delta_t$ be the cumulative mistake value observed when Algorithm 1 is executed for T batches. Also, for any $\mathbf{w}$, let*

Algorithm 3 SGD@K-AVG

```
Input: Batch length $b$, step lengths $\eta_t$, feasible set $\mathcal{W}$
Output: A model $\bar{\mathbf{w}} \in \mathcal{W}$
1: $\mathbf{w}^0 \leftarrow \mathbf{0}, t \leftarrow 0$
2: while stream not exhausted do
3:     $t \leftarrow t+1$
4:     Receive $b$ data points $\mathbf{X}_t = [\mathbf{x}_t^1, \ldots, \mathbf{x}_t^b], \mathbf{y}_t \in \{0,1\}^b$
5:     Set $\mathbf{g}_t \in \partial_{\mathbf{w}} \ell^{\text{avg}}_{\text{prec@k}}(\mathbf{w}_{t-1}; \mathbf{X}_t, \mathbf{y}_t)$    {See Algorithm 4}
6:     $\mathbf{w}_t \leftarrow \Pi_{\mathcal{W}}[\mathbf{w}_{t-1} - \eta_t \cdot \mathbf{g}_t]$    {project onto set $\mathcal{W}$}
7: end while
8: return $\bar{\mathbf{w}} = \frac{1}{t}\sum_{\tau=1}^{t} \mathbf{w}_\tau$
```

Algorithm 4 Subgradient calculation for $\ell^{\text{avg}}_{\text{prec@k}}(\cdot)$

```
Input: A model $\mathbf{w}_{\text{in}}$, $n$ data points $\mathbf{X}, \mathbf{y}$, parameter $k$
Output: A subgradient $\mathbf{g} \in \partial_{\mathbf{w}} \ell^{\text{avg}}_{\text{prec@k}}(\mathbf{w}_{\text{in}}; \mathbf{X}, \mathbf{y})$
1: Sort pos. and neg. points separately in dec. order of scores
   assigned by $\mathbf{w}_{\text{in}}$ i.e. $s_1^+ \geq \ldots \geq s_{n_+}^+$ and $s_1^- \geq \ldots \geq s_{n_-}^-$
2: for $k' = 0 \to k$ do
3:     $D_{k'} \leftarrow \frac{k-k'}{n_+ - k'}$
4:     $\Delta_{k'} \leftarrow k - k' - D_{k'} \sum_{i=k'+1}^{n_+} s_i^+ + \sum_{i=1}^{k-k'} s_i^-$
5:     $\mathbf{g}_{k'} \leftarrow \sum_{i=1}^{k-k'} \mathbf{x}_i^- - D_{k'} \sum_{i=k'+1}^{n_+} \mathbf{x}_i^+$
6: end for
7: $k^* \leftarrow \arg\max_{k'} \Delta_{k'}$
8: return $\mathbf{g}_{k^*}$
```

$\hat{\mathcal{L}}_T^{avg}(\mathbf{w}) = \sum_{t=1}^T \ell^{avg}_{\text{prec@k}}(\mathbf{w}; \mathbf{X}_t, \mathbf{y}_t)$. *Then we have*

$$\Delta_T^C \leq \min_{\mathbf{w}} \left(\|\mathbf{w}\| \cdot R \cdot \sqrt{4k} + \sqrt{\hat{\mathcal{L}}_T^{avg}(\mathbf{w})} \right)^2.$$

Similar to the classical perceptron mistake bound (Novikoff, 1962), the above bound can also be reduced to a simpler convergence bound in *separable settings*.

Corollary 8. *Suppose a unit norm $\mathbf{w}^*$ exists such that the scoring function $s : \mathbf{x} \mapsto \mathbf{x}^\top \mathbf{w}^*$ realizes the (k, γ)-margin condition for all the batches, then Algorithm 1 guarantees the mistake bound: $\Delta_T^C \leq \frac{4kR^2}{\gamma^2}$.*

The above result assures that, as datasets become "easier" in the sense that their (k, γ)-margin becomes larger, PERCEPTRON@K-AVG will converge to an optimal hyperplane at a faster rate. It is important to note there that the (k, γ)-margin condition is strictly weaker than the standard classification margin condition. Hence for several datasets, PERCEPTRON@K-AVG might be able to find a perfect ranking while at the same time, it might be impossible for standard binary classification techniques to find any reasonable classifier in poly-time (Guruswami & Raghavendra, 2009).

We note that PERCEPTRON@K-AVG performs updates with all the false negatives in the mini-batches. This raises the question as to whether sparser updates are possible as such updates would be slightly faster as well as, in high dimensional settings, ensure that the model is sparser. To this end we design the PERCEPTRON@K-MAX algorithm (Algorithm 2). PERCEPTRON@K-MAX differs from PERCEPTRON@K-AVG in that it performs updates using only a few of the *top ranked* false negatives. More specifically, for any scoring function s and $m > 0$, define:

$$\text{FN}(s, m) = \underset{S \subset \mathbf{X}_t^+, |S| = m}{\arg\max} \sum_{i \in S} \left(1 - \mathbf{y}_i^{(s,k)}\right) \mathbf{y}_i s_i$$

as the set of the m top ranked false negatives. PERCEPTRON@K-MAX makes updates only for false positives in the set $\text{FN}(s, \Delta_t)$. Note that Δ_t can significantly smaller than the total number of false negatives if $k \ll n_+$. PERCEPTRON@K-MAX also enjoys a mistake bound but with respect to the $\ell^{\text{max}}_{\text{prec@k}}(\cdot)$ surrogate.

Theorem 9. *Suppose $\|\mathbf{x}_t^i\| \leq R$ for all t, i. Let $\Delta_T^C = \sum_{t=1}^T \Delta_t$ be the cumulative observed mistake value when Algorithm 2 is executed for T batches. Also, for any $\mathbf{w}$, let $\hat{\mathcal{L}}_T^{max}(\mathbf{w}) = \sum_{t=1}^T \ell^{max}_{\text{prec@k}}(\mathbf{w}; \mathbf{X}_t, \mathbf{y}_t)$. Then we have*

$$\Delta_T^C \leq \min_{\mathbf{w}} \left(\|\mathbf{w}\| \cdot R \cdot \sqrt{4k} + \sqrt{\hat{\mathcal{L}}_T^{max}(\mathbf{w})} \right)^2.$$

Similar to PERCEPTRON@K-AVG, we can give a simplified mistake bound in situations where the separability condition specified by Definition 4 is satisfied.

Corollary 10. *Suppose a unit norm $\mathbf{w}^*$ exists such that the scoring function $s : \mathbf{x} \mapsto \mathbf{x}^\top \mathbf{w}^*$ realizes the* strong *γ-margin condition for all the batches, then Algorithm 2 guarantees the mistake bound: $\Delta_T^C \leq \frac{4kR^2}{\gamma^2}$.*

As the *strong* γ-margin condition is exactly the same as the standard notion of margin for binary classification, the above bound is no stronger than the one for the classical perceptron. However, in practice, we observe that PERCEPTRON@K-MAX at times outperforms even PERCEPTRON@K-AVG, even though the latter has a tighter mistake bound. This suggests that our analysis of PERCEPTRON@K-MAX might not be optimal and fails to exploit latent structures that might be present in the data.

Stochastic Gradient Descent for Optimizing prec@k. We now extend our algorithmic repertoire to include a stochastic gradient descent (SGD) algorithm for the prec@k performance measure. SGD methods are known to be very successful at optimizing large-scale empirical risk minimization (ERM) problems as they require only a few passes over the data to achieve optimal statistical accuracy.

However, SGD methods typically require access to cheap gradient estimates which are difficult to obtain for non-additive performance measures such as prec@k. This has been noticed before by (Kar et al., 2014; Narasimhan et al., 2015) who propose to use mini-batch methods to overcome this problem (Kar et al., 2014). By combining the

$\ell^{\text{avg}}_{\text{prec@k}}(\cdot)$ surrogate with mini-batch-style processing, we design SGD@K-AVG (Algorithm 3), a scalable SGD algorithm for optimizing prec@k. The algorithm uses mini-batches to update the current model using gradient descent steps. The subgradient calculation for this surrogate turns out to be non-trivial and is detailed in Algorithm 4.

The task of analyzing this algorithm is made non-trivial by the fact that the gradient estimates available to SGD@K-AVG via Algorithm 4 are far from being unbiased. The luxury of having unbiased gradient estimates is crucially exploited by standard SGD analyses but unfortunately, unavailable to us. To overcome this hurdle, we propose a uniform convergence based proof that, in some sense, bounds the bias in the gradient estimates.

In the following section, we present this, and many other generalization and online-to-batch conversion bounds with applications to our perceptron and SGD algorithms.

5. Generalization Bounds

In this section, we discuss novel uniform convergence (UC) bounds for our proposed surrogates. We will use these UC bounds along with the mistake bounds in Theorems 7 and 9 to prove two key results – 1) online-to-batch conversion bounds for the PERCEPTRON@K-AVG and PERCEPTRON@K-MAX algorithms and, 2) a convergence guarantee for the SGD@K-AVG algorithm.

To better present our generalization and convergence bounds, we use normalized versions of prec@k and the surrogates. To do so we write $k = \kappa \cdot n_+$ for some $\kappa \in (0, 1]$ and define, for any scoring function s, its prec@κ loss as:

$$\text{prec@}\kappa(s;\ \mathbf{z}_1, \ldots, \mathbf{z}_n) = \frac{1}{\kappa n_+}\Delta(\mathbf{y}, \mathbf{y}^{(s,\kappa n_+)}).$$

We will also normalize the surrogate functions $\ell^{\text{ramp}}_{\text{prec@}\kappa}(\cdot)$, $\ell^{\text{max}}_{\text{prec@}\kappa}(\cdot)$, and $\ell^{\text{avg}}_{\text{prec@}\kappa}(\cdot)$ by dividing by $k = \kappa \cdot n_+$.

Definition 11 (Uniform Convergence). *A performance measure $\Psi : \mathcal{W} \times (\mathcal{X} \times \{0, 1\})^n \mapsto \mathbb{R}_+$ exhibits uniform convergence with respect to a set of predictors $\mathcal{W}$ if for some $\alpha(b, \delta) = poly\left(\frac{1}{b}, \log\frac{1}{\delta}\right)$, for a sample $\hat{\mathbf{z}}_1, \ldots, \hat{\mathbf{z}}_b$ of size b chosen i.i.d. (or uniformly without replacement) from an arbitrary population $\mathbf{z}_1, \ldots, \mathbf{z}_n$, we have w.p. $1 - \delta$,*

$$\sup_{\mathbf{w}\in\mathcal{W}} |\Psi(\mathbf{w};\ \mathbf{z}_1, \ldots, \mathbf{z}_n) - \Psi(\mathbf{w};\ \hat{\mathbf{z}}_1, \ldots, \hat{\mathbf{z}}_b)| \leq \alpha(b, \delta)$$

We now state our UC bounds for prec@κ and its surrogates. We refer the reader to Appendix D for proofs.

Theorem 12. *The loss function* prec@$\kappa(\cdot)$*, as well as the surrogates $\ell^{ramp}_{\text{prec@}\kappa}(\cdot)$, $\ell^{avg}_{\text{prec@}\kappa}(\cdot)$ and $\ell^{max}_{\text{prec@}\kappa}(\cdot)$, all exhibit uniform convergence at the rate $\alpha(b, \delta) = \mathcal{O}\left(\sqrt{\frac{1}{b}\log\frac{1}{\delta}}\right)$.*

Recently, (Kar et al., 2014) also established a similar result for the $\ell^{\text{struct}}_{\text{prec@k}}(\cdot)$ surrogate. However, a very different proof technique is required to establish similar results for $\ell^{\text{max}}_{\text{prec@}\kappa}(\cdot)$ and $\ell^{\text{avg}}_{\text{prec@}\kappa}(\cdot)$, partly necessitated by the terms in these surrogates which depend, in a complicated manner, on the positives predicted by the candidate labeling $\hat{\mathbf{y}}$. Nevertheless, the above results allow us to establish strong online-to-batch conversion bounds for PERCEPTRON@K-AVG and PERCEPTRON@K-MAX, as well as convergence rates for the SGD@K-AVG method. In the following we shall assume that our data streams are composed of points chosen i.i.d. (or u.w.r.) from some fixed population $\mathcal{Z}$.

Theorem 13. *Suppose an algorithm, when fed a random stream of data points, in T batches of length b each, generates an ensemble of models $\mathbf{w}_1, \ldots, \mathbf{w}_T$ which together suffer a cumulative mistake value of Δ^C_T. Then, with probability at least $1 - \delta$, we have*

$$\frac{1}{T}\sum_{t=1}^{T}\text{prec@}\kappa(\mathbf{w}^t; \mathcal{Z}) \leq \frac{\Delta^C_T}{bT} + \mathcal{O}\left(\sqrt{\frac{1}{b}\log\frac{T}{\delta}}\right).$$

The proof of this theorem follows from Theorem 12 which guarantees that w.p. $1 - \delta$, prec@$\kappa(\mathbf{w}^t; \mathcal{Z}) \leq \Delta_t/b + \mathcal{O}\left(\sqrt{\frac{1}{b}\log\frac{1}{\delta}}\right)$ for all t. Combining this with the mistake bound from Theorem 7 ensures the following generalization guarantee for the ensemble generated by Algorithm 1.

Corollary 14. *Let $\mathbf{w}^1, \ldots, \mathbf{w}^T$ be the ensemble of classifiers returned by the* PERCEPTRON@K-AVG *algorithm on a random stream of data points and batch length b. Then, with probability at least $1 - \delta$, for any $\mathbf{w}^*$ we have*

$$\frac{1}{T}\sum_{t=1}^{T}\text{prec@}\kappa(\mathbf{w}^t; \mathcal{Z}) \leq \left(\sqrt{\ell^{avg}_{\text{prec@}\kappa}(\mathbf{w}^*; \mathcal{Z})} + C\right)^2,$$

where $C = \mathcal{O}\left(\|\mathbf{w}^*\|\,R\sqrt{\frac{1}{T}} + \sqrt[4]{\frac{1}{b}\log\frac{T}{\delta}}\right)$.

A similar statement holds for the PERCEPTRON@K-MAX algorithm with respect to the $\ell^{\text{max}}_{\text{prec@}\kappa}(\cdot)$ surrogate as well. Using the results from Theorem 12, we can also establish the convergence rate of the SGD@K-AVG algorithm.

Theorem 15. *Let $\bar{\mathbf{w}}$ be the model returned by Algorithm 3 when executed on a stream with T batches of length b. Then with probability at least $1 - \delta$, for any $\mathbf{w}^* \in \mathcal{W}$, we have*

$$\ell^{avg}_{\text{prec@}\kappa}(\bar{\mathbf{w}}; \mathcal{Z}) \leq \ell^{avg}_{\text{prec@}\kappa}(\mathbf{w}^*; \mathcal{Z}) + \mathcal{O}\left(\sqrt{\frac{1}{b}\log\frac{T}{\delta}}\right) + \mathcal{O}\left(\sqrt{\frac{1}{T}}\right)$$

The proof of this Theorem can be found in Appendix E.

6. Experiments

We shall now evaluate our methods on several benchmark datasets for binary classification problems with a rare-class.

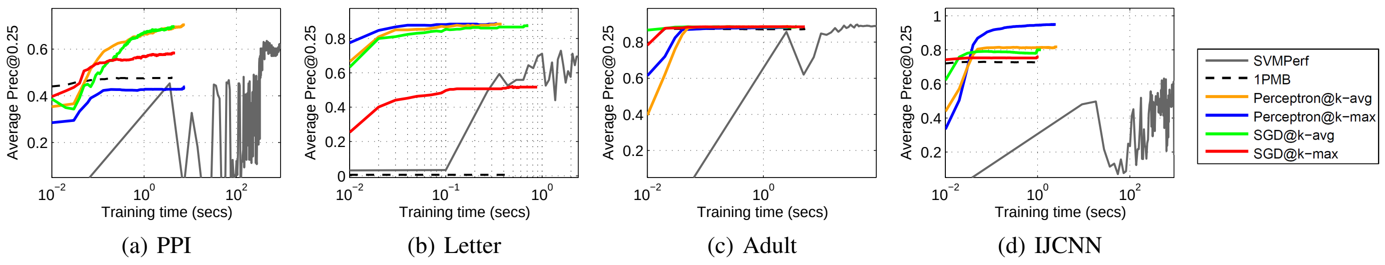

Figure 2. A comparison of the proposed perceptron and SGD based methods with baseline methods (SVMPerf and **1PMB**) on prec@0.25 maximization tasks. PERCEPTRON@K-AVG and SGD@K-AVG (both based on $\ell^{\text{avg}}_{\text{prec@k}}(\cdot)$) are the most consistent methods across tasks.

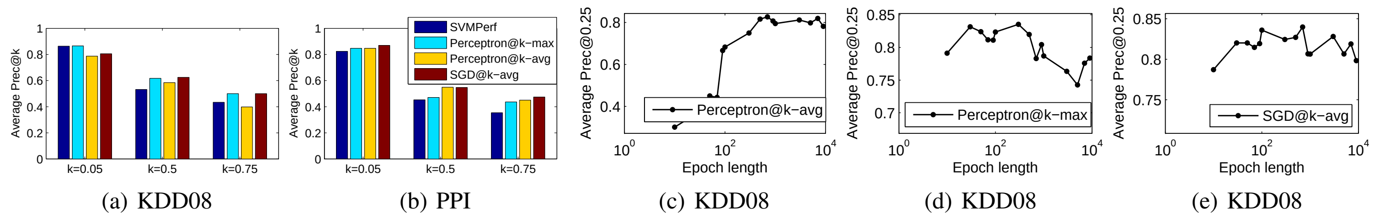

Figure 3. (a), (b): A comparison of different methods on optimizing prec@κ for different values of κ. (c), (d), (e): The performance of the proposed perceptron and SGD methods on prec@0.25 maximization tasks with varying batch lengths b.

Datasets: We evaluated our methods on 7 publicly available benchmark datasets: a) PPI, b) KDD Cup 2008, c) Letter, d) Adult, e) IJCNN, f) Covertype, and g) Cod-RNA. All datasets exhibit moderate to severe label imbalance with the KDD Cup dataset having just 0.61% positives.

Methods: We compared both perceptron algorithms, SGD@K-AVG, as well as an SGD solver for the $\ell^{\text{max}}_{\text{prec@k}}(\cdot)$ surrogate, with the cutting plane-based SVMPerf solver of (Joachims, 2005), and the stochastic **1PMB** solver of (Kar et al., 2014). The perceptron and SGD methods were given a maximum of 25 passes over the data with a batch length of 500. All methods were implemented in C. We used 70% of the data for training and the rest for testing. All results are averaged over 5 random train-test splits.

Our experiments reveal three interesting insights into the problem of prec@k maximization – 1) using tighter surrogates for optimization routines is indeed beneficial, 2) the presence of a stochastic solver cannot always compensate for the use of a suboptimal surrogate, and 3) mini-batch techniques, applied with perceptron or SGD-style methods, can offer rapid convergence to accurate models.

We first timed all the methods on prec@κ maximization tasks for $\kappa = 0.25$ on various datasets (see Figure 2). Of all the methods, the cutting plane method (SVMPerf) was found to be the most expensive computationally. On the other hand, the perceptron and stochastic gradient methods, which make frequent but cheap updates, were much faster at identifying accurate solutions.

We also observed that PERCEPTRON@K-AVG and SGD@K-AVG, which are based on the tight $\ell^{\text{avg}}_{\text{prec@k}}(\cdot)$ surrogate, were the most consistent at converging to accurate solutions whereas PERCEPTRON@K-MAX and SGD@K-MAX, which are based on the loose $\ell^{\text{max}}_{\text{prec@k}}(\cdot)$ surrogate, showed large deviations in performance across tasks. Also, **1PMB** and SVMPerf, which are based on the non upper-bounding $\ell^{\text{struct}}_{\text{prec@k}}(\cdot)$ surrogate, were frequently found to converge to suboptimal solutions.

The effect of working with a tight surrogate is also clear from Figure 3 (a), (b) where the algorithms working with our novel surrogates were found to consistently outperform the SVMPerf method which works with the $\ell^{\text{struct}}_{\text{prec@k}}(\cdot)$ surrogate. For these experiments, SVMPerf was allowed a runtime of up to $50\times$ of what was given to our methods after which it was terminated.

Finally, to establish the stability of our algorithms, we ran, both the perceptron, as well as the SGD algorithms with varying batch lengths (see Figure 3 (c)-(e)). We found the algorithms to be relatively stable to the setting of the batch length. To put things in perspective, all methods registered a relative variation of less than 5% in accuracies across batch lengths spanning an order of magnitude or more. We present additional experimental results in Appendix F.

Acknowledgments

HN thanks support from a Google India PhD Fellowship.

References

Agarwal, S. The Infinite Push: A new support vector ranking algorithm that directly optimizes accuracy at the absolute top of the list. In *11th SIAM International Conference on Data Mining (SDM)*, pp. 839–850, 2011.

Boucheron, Stphane, Lugosi, Gbor, and Bousquet, Olivier. Concentration inequalities. In *Advanced Lectures in Machine Learning*, pp. 208–240. Springer, 2004.

Boyd, Stephen, Cortes, Corinna, Mohri, Mehryar, and Radovanovic, Ana. Accuracy at the top. In *26th Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 953–961, 2012.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G. Learning to rank using gradient descent. In *22nd International Conference on Machine Learning (ICML)*, pp. 89–96, 2005.

Calauzènes, Clément, Usunier, Nicolas, and Gallinari, Patrick. On the (Non-)existence of Convex, Calibrated Surrogate Losses for Ranking. In *26th Annual Conference on Neural Information Processing Systems (NIPS)*, 2012.

Cao, Zhe, Qin, Tao, Liu, Tie-Yan, Tsai, Ming-Feng, and Li, Hang. Learning to rank: from pairwise approach to listwise approach. In *24th International Conference on Machine learning (ICML)*, pp. 129–136. ACM, 2007.

Chakrabarti, Soumen, Khanna, Rajiv, Sawant, Uma, and Bhattacharyya, Chiru. Structured Learning for Non-Smooth Ranking Losses. In *14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2008.

Chaudhuri, Sougata and Tewari, Ambuj. Perceptron-like algorithms and generalization bounds for learning to rank. *CoRR*, abs/1405.0591, 2014.

Chaudhuri, Sougata and Tewari, Ambuj. Online ranking with top-1 feedback. In *18th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

Clémençon, Stéphan and Vayatis, Nicolas. Ranking the best instances. *The Journal of Machine Learning Research*, 8:2671–2699, 2007.

Do, Chuong B., Le, Quoc, Teo, Choon Hui, Chapelle, Olivier, and Smola, Alex. Tighter Bounds for Structured Estimation. In *22nd Annual Conference on Neural Information Processing Systems (NIPS)*, 2008.

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

Guruswami, Venkatesan and Raghavendra, Prasad. Hardness of learning halfspaces with noise. *SIAM J. Comput.*, 39(2):742–765, 2009.

Herbrich, R., Graepel, T., and Obermayer, K. Large margin rank boundaries for ordinal regression. In Smola, A., Bartlett, P., Schoelkopf, B., and Schuurmans, D. (eds.), *Advances in Large Margin Classifiers*, pp. 115–132. MIT Press, 2000.

Joachims, T. Optimizing search engines using clickthrough data. In *8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 133–142, 2002.

Joachims, Thorsten. A Support Vector Method for Multivariate Performance Measures. In *22nd International Conference on Machine Learning (ICML)*, 2005.

Kar, Purushottam, Narasimhan, Harikrishna, and Jain, Prateek. Online and stochastic gradient methods for non-decomposable loss functions. In *28th Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 694–702, 2014.

Le, Quoc V. and Smola, Alexander J. Direct optimization of ranking measures. *arXiv preprint arXiv:0704.3359*, 2007.

Li, Nan, Jin, Rong, and Zhou, Zhi-Hua. Top rank optimization in linear time. In *28th Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 1502–1510, 2014.

Minsky, Marvin Lee and Papert, Seymour. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, 1988. ISBN 0262631113.

Narasimhan, Harikrishna and Agarwal, Shivani. A Structural SVM Based Approach for Optimizing Partial AUC. In *30th International Conference on Machine Learning (ICML)*, 2013a.

Narasimhan, Harikrishna and Agarwal, Shivani. $\text{SVM}_{\text{pAUC}}^{\text{tight}}$: A New Support Vector Method for Optimizing Partial AUC Based on a Tight Convex Upper Bound. In *19th ACM SIGKDD Conference on Knowledge, Discovery and Data Mining (KDD)*, 2013b.

Narasimhan, Harikrishna, Kar, Purushottam, and Jain, Prateek. Optimizing Non-decomposable Performance Measures: A Tale of Two Classes. In *32nd International Conference on Machine Learning (ICML)*, 2015.

Novikoff, A.B.J. On convergence proofs on perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata*, volume 12, pp. 615–622, 1962.

Prabhu, Yashoteja and Varma, Manik. Fastxml: a fast, accurate and stable tree-classifier for extreme multi-label learning. In *20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD*, pp. 263–272, 2014.

Rosenblatt, Frank. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

Rudin, C. The p-norm push: A simple convex ranking algorithm that concentrates at the top of the list. *Journal of Machine Learning Research*, 10:2233–2271, 2009.

Tsoumakas, Grigorios and Katakis, Ioannis. Multi-Label Classification: An Overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13, 2007.

Valizadegan, Hamed, Jin, Rong, Zhang, Ruofei, and Mao, Jianchang. Learning to rank by optimizing NDCG measure. In *26th Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 1883–1891, 2009.

Yue, Y., Finley, T., Radlinski, F., and Joachims, T. A support vector method for optimizing average precision. In *30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 271–278, 2007.

Yun, Hyokun, Raman, Parameswaran, and Vishwanathan, S. Ranking via robust binary classification. In *28th Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 2582–2590, 2014.

Zhang, Tong. Covering Number Bounds of Certain Regularized Linear Function Classes. *Journal of Machine Learning Research*, 2:527–550, 2002.