Subset Ranking Using Regression

David Cossock[1] and Tong Zhang[2]

[1] Yahoo Inc., Santa Clara, CA, USA
dcossock@yahoo-inc.com
[2] Yahoo Inc., New York City, USA
tzhang@yahoo-inc.com

Abstract. We study the subset ranking problem, motivated by its important application in web-search. In this context, we consider the standard DCG criterion (discounted cumulated gain) that measures the quality of items near the top of the rank-list. Similar to error minimization for binary classification, the DCG criterion leads to a non-convex optimization problem that can be NP-hard. Therefore a computationally more tractable approach is needed. We present bounds that relate the approximate optimization of DCG to the approximate minimization of certain regression errors. These bounds justify the use of convex learning formulations for solving the subset ranking problem. The resulting estimation methods are not conventional, in that we focus on the estimation quality in the top-portion of the rank-list. We further investigate the generalization ability of these formulations. Under appropriate conditions, the consistency of the estimation schemes with respect to the DCG metric can be derived.

1 Introduction

We consider the general ranking problem, where a computer system is required to rank a set of items based on a given input. In such applications, the system often needs to present only a few top ranked items to the user. Therefore the quality of the system output is determined by the performance near the top of its rank-list.

Ranking is especially important in electronic commerce and internet, where personalization and information based decision making is critical to the success of such business. The decision making process can often be posed as a problem of selecting top candidates from a set of potential alternatives, leading to a conditional ranking problem. For example, in a recommender system, the computer is asked to choose a few items a user is most likely to buy based on the user's profile and buying history. The selected items will then be presented to the user as recommendations. Another important example that affects millions of people everyday is the internet search problem, where the user presents a query to the search engine, and the search engine then selects a few web-pages that are most relevant to the query from the whole web. The quality of a search engine is largely determined by the top-ranked results the search engine can display on the first page. Internet search is the main motivation of this theoretical study,

G. Lugosi and H.U. Simon (Eds.): COLT 2006, LNAI 4005, pp. 605–619, 2006.

although the model presented here can be useful for many other applications. For example, another ranking problem is ad placement in a web-page (either search result, or some content page) according to revenue-generating potential.

Since for search and many other ranking problems, we are only interested in the quality of the top choices, the evaluation of the system output is different from many traditional error metrics such as classification error. In this setting, a useful figure of merit should focus on the top portion of the rank-list. To our knowledge, this particular characteristic of ranking problems has not been carefully explored in earlier studies. The purpose of this paper is to develop some theoretical results for converting a ranking problem into convex optimization problems that can be efficiently solved. The resulting formulation focuses on the quality of the top ranked results. The theory can be regarded as an extension of related theory for convex risk minimization formulations for classification, which has drawn much attention recently in the statistical learning literature[1, 2, 3, 4, 5, 6].

We organize the paper as follows. Section 2 introduces the subset ranking problem. We define two ranking metrics: one is the DCG measure which we focus on in this paper, and the other is a measure that counts the number of correctly ranked pairs. The latter has been studied recently by several authors. Section 3 contains the main theoretical results in this paper, where we show that the approximate minimization of certain regression errors lead to the approximate optimization of the ranking metrics defined earlier. This implies that asymptotically the non-convex ranking problem can be solved using regression methods that are convex. Section 4 presents the regression learning formulation derived from the theoretical results in Section 3. Similar methods are currently used to optimize Yahoo's production search engine. Section 5 studies the generalization ability of regression learning, where we focus on an L_1-boosting approach. Together with earlier theoretical results, we can establish the consistency of regression based ranking under appropriate conditions.

2 The Subset Ranking Problem

We first describe the abstract version of our subset ranking model, and then use web-search as a concrete example for this model.

2.1 Problem Definition

Let $\mathcal{X}$ be the space of observable features, and $\mathcal{Z}$ be the space of variables that are not necessarily directly used in the deployed system. Denote by $\mathcal{S}$ the set of all finite subsets of $\mathcal{X}$ that may possibly contain elements that are redundant. Let y be a non-negative real-valued variable that corresponds to the quality of $x \in \mathcal{X}$. Assume also that we are given a (measurable) feature-map F that takes each $z \in \mathcal{Z}$, and produces a finite subset $F(z) = S = \{x_1, \ldots, x_m\} \in \mathcal{S}$. Note that the order of the items in the set is of no importance; the numerical subscripts are for notational purpose only, so that permutations can be more conveniently defined.

In subset ranking, we randomly draw a variable $z \in \mathcal{Z}$ according to some underlying distribution on $\mathcal{Z}$. We then create a finite subset $F(z) = S = \{x_1, \ldots, x_m\} \in \mathcal{S}$ consisting of feature vectors x_j in $\mathcal{X}$, and at the same time, a set of grades $\{y_j\} = \{y_1, \ldots, y_m\}$ such that for each j, y_j corresponds to x_j. Whether the size of the set m should be a random variable has no importance in our analysis. In this paper we assume that it is fixed for simplicity.

Based on the observed subset $S = \{x_1, \ldots, x_m\}$, the system is required to output an ordering (ranking) of the items in the set. Using our notation, this ordering can be represented as a permutation $J = [j_1, \ldots, j_m]$ of $[1, \ldots, m]$. Our goal is to produce a permutation such that y_{j_i} is in decreasing order for $i = 1, \ldots, m$. Given the grades $y_j (j = 1, \ldots, m)$, the quality of the rank-list $J = [j_1, \ldots, j_m]$ is measured by the following weighted sum:

$$\mathbf{DCG}(J, [y_j]) = \sum_{i=1}^{m} c_i y_{j_i},$$

where $\{c_i\}$ is a pre-defined sequence of non-increasing non-negative discount factors that are independent of S. This metric, described in [7] as DCG (discounted cumulated gain), is one of the main metrics widely used in the evaluation of internet search systems, including the production system of Yahoo and that of Microsoft [8]. A typical choice of c_i is to set $c_i = 1/\log(1 + i)$ when $i \leq k$ and $c_i = 0$ when $i > k$ for some k. By choosing a decaying sequence of c_i, this measure focuses on the quality of the top portion of the rank-list.

Our goal is to train a ranking function r that can take a subset $S \in \mathcal{S}$ as input, and produce an output permutation $J = r(S)$ such that the expected DCG is as large as possible:

$$\mathbf{DCG}(r) = \mathbf{E}_S \, \mathbf{DCG}(r, S), \tag{1}$$

where

$$\mathbf{DCG}(r, S) = \sum_{i=1}^{m} c_i \mathbf{E}_{y_{j_i} | (x_{j_i}, S)} \; y_{j_i}. \tag{2}$$

An alternative ranking metric is the weighted total of correctly ranked pairs minus incorrectly ranked pairs:

$$\mathbf{T}(J, [y_j]) = \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{i'=i+1}^{m} (y_{j_i} - y_{j_{i'}}).$$

If the output label y_i takes binary-values, and the subset $S = \mathcal{X}$ is global (we may assume that it is finite), then this metric is known to be equivalent to AUC (area under ROC) up to a scaling, and related to the Mann-Whitney-Wilcoxon statistics [9]. In the literature, theoretical analysis has focused mainly on global ranking (that is, the set S we observe is $\mathcal{X}$) and the $\mathbf{T}$-criterion (for example, see [10, 11, 12, 13]). However, such a model is inadequate for many practical ranking problems including web-search. Although we pay special attention to the DCG metric, we shall also include some analysis of the $\mathbf{T}$ criterion for completeness.

Similar to (1) and (2), we can define the following quantities:

$$\mathbf{T}(r) = \mathbf{E}_S \, \mathbf{T}(r, S), \tag{3}$$

where

$$\mathbf{T}(r, S) = \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{i'=i+1}^{m} (\mathbf{E}_{y_{j_i}|(x_{j_i}, S)} \, y_{j_i} - \mathbf{E}_{y_{j_{i'}}|(x_{j_{i'}}, S)} \, y_{j_{i'}}). \tag{4}$$

Similar to the concept of Bayes classifier in classification, we can define the Bayes ranking function that optimizes the **DCG** and **T** measures. Based on the conditional formulations in (2) and (4), we have the following result:

Theorem 1. *Given a set $S \in \mathcal{S}$, for each $x_j \in S$, we define the Bayes-scoring function as*

$$f_B(x_j, S) = \mathbf{E}_{y_j|(x_j, S)} \, y_j$$

An optimal Bayes ranking function $r_B(S)$ that maximizes (4) returns a rank list $J = [j_1, \ldots, j_m]$ such that $f_B(x_{j_i}, S)$ is in descending order: $f_B(x_{j_1}, S) \geq f_B(x_{j_2}, S) \geq \cdots \geq f_B(x_{j_m}, S)$. An optimal Bayes ranking function $r_B(S)$ that maximizes (2) returns a rank list $J = [j_1, \ldots, j_m]$ such that $c_k > c_{k'}$ implies that $f_B(x_{j_k}, S) > f_B(x_{j_{k'}}, S)$.

Proof. Consider any $k, k' \in \{1, \ldots, m\}$. Define $J' = [j'_1, \ldots, j'_m]$, where $j'_i = j_i$ when $i \neq k, k'$, and $j'_k = j_{k'}$, and $j'_{k'} = j_k$.

We consider the **T**-criterion first, and let $k' = k + 1$. It is easy to check that $\mathbf{T}(J', S) - \mathbf{T}(J, S) = 4(f_B(x_{j_{k+1}}, S) - f_B(x_{j_k}, S))/m(m-1)$. Therefore $\mathbf{T}(J', S) \leq \mathbf{T}(J, S)$ implies that $f_B(x_{j_{k+1}}, S) \leq f_B(x_{j_k}, S)$.

Now consider the **DCG**-criterion. We have $\mathbf{DCG}(J', S) - \mathbf{DCG}(J, S) = (c_k - c_{k'})(f_B(x_{j_{k'}}, S) - f_B(x_{j_k}, S))$. Now $c_k > c_{k'}$ and $\mathbf{DCG}(J', S) \leq \mathbf{DCG}(J, S)$ implies $f_B(x_{j_k}, S) \geq f_B(x_{j_{k'}}, S)$. □

2.2 Web-Search Example

The subset ranking model can be applied to the web-search problem, where the user submits a query q, and expects the search engine to return a rank-list of web-pages $\{p_j\}$ such that a more relevant page is placed before a less relevant page. In a typical internet search engine, the system takes a query and uses a simple ranking formula for the initial filtering, which limits the set of web-pages to an initial pool $\{p_j\}$ of size m (e.g., $m = 100000$).

After this initial ranking, the system go through a more complicated second stage ranking process, which reorders the pool. This critical stage is the focus of this paper. At this step, the system takes the query q, and possible information from additional resources, to generate a feature vector x_j for each page p_j in the initial pool. The feature vector can encode various types of information such as the length of query q, the position of p_j in the initial pool, the number of query terms that match the title of p_j, the number of query terms that match the body of p_j, etc. The set of all possible feature vectors x_j is $\mathcal{X}$. The ranking algorithm

only observes a list of feature vectors $\{x_1, \ldots, x_m\}$ with each $x_j \in \mathcal{X}$. A human editor is presented with a pair (q, p_j) and assigns a score s_j on a scale, e.g., $1-5$ (least relevant to highly relevant). The corresponding target value y_j is defined as a transformation of y_j,[1] which maps the grade into the interval $[0, 1]$. Another possible choice of y_j is to normalize it by multiplying each y_j by a factor such that the optimal DCG is no more than one.

2.3 Set Dependent Features

Due to the dependency of conditional probability of y on S, and thus the optimal ranking function on S, subset ranking becomes a very difficult problem when m is large. In general, without further assumptions, the optimal Bayes ranking function rank the items using the Bayes scoring function $f_B(x, S)$ for each $x \in S$.

If the size m of S is small, then we may simply represent S as a feature vector $[x_1, \ldots, x_m]$ (although this may not be the best representation), so that we can learn a function of the form $f_B(x_j, S) = f([x_j, x_1, \ldots, x_m])$. In the general case when m is large, this approach is not practical. Instead of using the whole set S as a feature, we have to project S into a lower dimensional space using a feature map $g(\cdot)$, so that $f_B(x, g(S)) \approx f(x, g(S))$. Note that the information of $g(S)$ can be incorporated into x (this can be achieved by simply redefining x as $[x, g(S)]$), so that $f_B(x, S)$ can be approximated by a function of the form $f(x)$.

Definition 1. *If for every $S \in \mathcal{S}$ and $x, x' \in S$, we have*

$$f_B(x, S) > f_B(x', S) \quad \text{if and only if} \quad f(x) > f(x'),$$

then we say that f is an optimal rank preserving function.

An optimal rank preserving function may not exist for casual feature representations. As a simple example, we assume that $\mathcal{X} = \{a, b, c\}$ has three elements, with $m = 2$, $c_1 = 1$ and $c_2 = 0$ in the DCG definition. We observe $\{y_1 = 1, y_2 = 0\}$ for the set $\{x_1 = a, x_2 = b\}$, $\{y_1 = 1, y_2 = 0\}$ for the set $\{x_1 = b, x_2 = c\}$, $\{y_1 = 1, y_2 = 0\}$ for the set $\{x_1 = c, x_2 = a\}$. If an optimal rank preserving function f exists, then by definition we have: $f(a) > f(b)$, $f(b) > f(c)$, and $f(c) > f(a)$. This is impossible. The following result gives a sufficient condition for the existence of optimal rank preserving function.

Proposition 1. *Assume that for each x_j, we observe $y_j = n(S)y'_j$ where $n(S)$ is a normalization factor that may depend on S, and $\{y'_j\}$ is a set of random variables that satisfy:*

$$P(\{y'_j\}|S) = \mathbf{E}_\xi \prod_{j=1}^{m} P(y'_j|x_j, \xi),$$

where ξ is a hidden random variable independent of S. Then $\mathbf{E}_{y'_j|(x_j,S)}\, y'_j = \mathbf{E}_{y'_j|x_j}\, y'_j$. That is, the conditional expectation $f(x) = \mathbf{E}_{y'|x}\, y'$ is an optimal rank preserving function.

[1] For example, the formula $(2^{s_j} - 1)/(2^5 - 1)$ is used in [8]. Yahoo uses a different transformation based on empirical user surveys.

This result justifies using an appropriately defined feature function to remove set-dependency. If y'_j is a deterministic function of x_j and ξ, then the result always holds, which implies set-independent conditional expectation is optimal. We also note that the optimality of conditional expectation as the scoring function does not require that the grade y' to be independent of S.

In web-search, the model in Proposition 1 has a natural interpretation. Consider a pool of human editors indexed by ξ. For each query q, we randomly pick an editor ξ to grade the set of pages p_j to be ranked, and assume that the grade the editor gives to each page p_j depends only on the pair $x_j = (q, p_j)$.

In the literature, various methods for solving ranking problems have been proposed. The most relevant model in the statistics literature is ordinal regression, which was adapted to large margin methods in [14]. In machine learning, the focus was on pair-wise preference learning, where one learns a scoring function $f(x)$ so that pair-wise rank-orders are preserved. For example, this idea was adopted in the Microsoft system [8]. Proposition 1 (and discussion thereafter) suggests that regression based learning of the conditional expectation $\mathbf{E}_{y|x}\, y$ is asymptotically optimal under some assumptions that are reasonable. Moreover, as discussed earlier in this section, in the regression based approach, one may always introduce set-dependent features through a feature map $g(S)$. Due to these advantages, we shall focus on regression based methods in this paper.

2.4 Relationship to Multi-category Classification

The subset ranking problem is a generalization of multi-category classification. In this case, we observe an input x_0, and are interested in classifying it into one of the m classes. Let the output value be $k \in \{1, \ldots, m\}$. We encode the input x_0 into m feature vectors $\{x_1, \ldots, x_m\}$, where $x_i = [0, \ldots, 0, x_0, 0, \ldots, 0]$ with the i-th component being x_0, and the other components are zeros. We then encode the output k into m values $\{y_j\}$ such that $y_k = 1$ and $y_j = 0$ for $j \neq k$. In this setting, we try to find a scoring function f such that $f(x_k) > f(x_j)$ for $j \neq k$. Consider the DCG criterion with $c_1 = 1$ and $c_j = 0$ when $j > 1$. Then the classification error is given by the corresponding DCG.

Given any multi-category classification algorithm, one may use it to solve subset ranking as follows. Consider a sample S as input, and a set of outputs $\{y_j\}$. We randomly draw k from 1 to m according to the distribution $y_k / \sum_j y_j$. We form another sample with S as input, and $\{y'_j\}$ as output (where $y'_k = 1$, and $y'_j = 0$ when $j \neq k$). This changes the problem formulation into multi-category classification. Since this transformation does not change the order of conditional expectation $\mathbf{E}_{y_j|(x_j,S)} y_j$, it does not change the optimal Bayes ranking function. Therefore a multi-category classification solver that estimates conditional probability can be used to solve the subset ranking problem. The regression method we investigate in this paper is related to the one-versus-all approach.

3 Convex Surrogate Bounds

The subset ranking problem defined in Section 2 is combinatorial in nature, which is very difficult to solve. This section provides some theoretical results that relate the optimization of the ranking metrics defined in Section 2 to the minimization of some regression errors, which allow us to design appropriate convex learning formulations to solve the ranking problem efficiently.

A scoring function $f(x, S)$ maps each $x \in S$ to a real valued score. It induces a ranking function r_f, which ranks elements $\{x_j\}$ of S in descending order of $f(x_j)$. We are interested in bounding the DCG performance of r_f compared with that of f_B. This can be regarded as extensions of Theorem 1 that motivate regression based learning.

Theorem 2. *Let $f(x, S)$ be a real-valued scoring function, which induces a ranking function r_f. We have the following relationship for each $S = \{x_1, \ldots, x_m\}$:*

$$\mathbf{DCG}(r_B, S) - \mathbf{DCG}(r_f, S) \leq \left(2\sum_{i=1}^{m} c_i^2\right)^{1/2} \left(\sum_{j=1}^{m} (f(x_j, S) - f_B(x_j, S))^2\right)^{1/2}.$$

Proof. Let $S = \{x_1, \ldots, x_m\}$. Let $r_f(S) = J = [j_1, \ldots, j_m]$, and let $J^{-1} = [\ell_1, \ldots, \ell_m]$ be its inverse permutation. Similarly, let $r_B(S) = J_B = [j_1^*, \ldots, j_m^*]$, and let $J_B^{-1} = [\ell_1^*, \ldots, \ell_m^*]$ be its inverse permutation. We have

$$\begin{aligned}
\mathbf{DCG}(r_f, S) =& \sum_{i=1}^{m} c_i f_B(x_{j_i}, S) = \sum_{i=1}^{m} c_{\ell_i} f_B(x_i, S) \\
=& \sum_{i=1}^{m} c_{\ell_i} f(x_i, S) + \sum_{i=1}^{m} c_{\ell_i} (f_B(x_i, S) - f(x_i, S)) \\
\geq& \sum_{i=1}^{m} c_{\ell_i^*} f(x_i, S) + \sum_{i=1}^{m} c_{\ell_i} (f_B(x_i, S) - f(x_i, S)) \\
=& \sum_{i=1}^{m} c_{\ell_i^*} f_B(x_i, S) + \sum_{i=1}^{m} c_{\ell_i^*} (f(x_i, S) - f_B(x_i, S)) \\
&+ \sum_{i=1}^{m} c_{\ell_i} (f_B(x_i, S) - f(x_i, S)) \\
\geq& \mathbf{DCG}(r_B, S) - \sum_{i=1}^{m} c_{\ell_i} (f(x_i, S) - f_B(x_i, S))_+ \\
&- \sum_{i=1}^{m} c_{\ell_i^*} (f_B(x_i, S) - f(x_i, S))_+ \\
\geq& \mathbf{DCG}(r_B, S) - \left(2\sum_{i=1}^{m} c_i^2\right)^{1/2} \left(\sum_{j=1}^{m} (f(x_j, S) - f_B(x_j, S))^2\right)^{1/2}.
\end{aligned}$$

where we used the notation $(z)_+ = \max(0, z)$. □

The above theorem shows that the **DCG** criterion can be bounded through regression error. If regression error goes to zero, then the resulting ranking converges to the optimal DCG. Similarly, we can show the following result for the **T** criterion.

Theorem 3. *Let $f(x, S)$ be a real-valued scoring function, which induces a ranking function r_f. We have the following relationship for each $S = \{x_1, \ldots, x_m\}$:*

$$\mathbf{T}(r_B, S) - \mathbf{T}(r_f, S) \leq \frac{4}{\sqrt{m}} \left(\sum_{j=1}^{m} (f(x_j, S) - f_B(x_j, S))^2 \right)^{1/2}.$$

Proof. Let $S = \{x_1, \ldots, x_m\}$. Let $r_f(S) = J = [j_1, \ldots, j_m]$, and let $r_B(S) = J_B = [j_1^*, \ldots, j_m^*]$. We have

$$\begin{aligned}
&\mathbf{T}(r_f, S) \\
=&\frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{i'=i+1}^{m} (f_B(x_{j_i}, S) - f_B(x_{j_{i'}}, S)) \\
\geq&\frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{i'=i+1}^{m} (f(x_{j_i}, S) - f(x_{j_{i'}}, S)) - \frac{2}{m} \sum_{i=1}^{m} |f(x_{j_i}, S) - f_B(x_{j_i}, S)| \\
\geq&\frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{i'=i+1}^{m} (f(x_{j_i^*}, S) - f(x_{j_{i'}^*}, S)) - \frac{2}{m} \sum_{i=1}^{m} |f(x_{j_i}, S) - f_B(x_{j_i}, S)| \\
\geq&\frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{i'=i+1}^{m} (f_B(x_{j_i^*}, S) - f_B(x_{j_{i'}^*}, S)) - \frac{4}{m} \sum_{i=1}^{m} |f(x_{j_i}, S) - f_B(x_{j_i}, S)| \\
=&\mathbf{T}(r_B, S) - \frac{4}{m} \sum_{i=1}^{m} |f(x_{j_i}, S) - f_B(x_{j_{i'}}, S)| \\
\geq&\mathbf{T}(r_B, S) - \frac{4}{\sqrt{m}} \left(\sum_{i=1}^{m} (f(x_{j_i}, S) - f_B(x_{j_{i'}}, S))^2 \right)^{1/2}.
\end{aligned}$$

□

The above approximation bounds imply that least square regression can be used to learn the optimal ranking functions. The approximation error converges to zero when f converges to f_B in L_2. However, in general, requiring f to converge to f_B in L_2 is not necessary. More importantly, in real applications, we are often only interested in the top portion of the rank-list. Our bounds should reflect this practical consideration. In the following, we develop a more refined bound for the DCG metric, which will be used to motivate practical learning methods in the next section.

Theorem 4. *Let $f(x, S)$ be a real-valued scoring function, which induces a ranking function r_f. Given $S = \{x_1, \ldots, x_m\}$, let the optimal ranking order be $J_B = [j_1^*, \ldots, j_m^*]$, where $f_B(x_{j_i^*})$ is arranged in non-increasing order. Assume that $c_i = 0$ for all $i > k$. Then we have the following relationship for all $\gamma \in (0, 1)$, $u > 0$ and subset $K \subset \{1, \ldots, m\}$ that contains $j_1^*, \ldots, j_k^*$:*

$$\begin{aligned}
&\mathbf{DCG}(r_B, S) - \mathbf{DCG}(r_f, S) \\
\leq & C(\gamma, u) \left(\sum_{j \in K} (f(x_j, S) - f_B(x_j, S))^2 + u \sup_{j \notin K} (f(x_j, S) - f'_B(x_j, S))_+^2 \right)^{1/2},
\end{aligned}$$

where $(z)_+ = \max(z, 0)$, *and*

$$C(\gamma, u) = \frac{1}{1-\gamma} \sqrt{2 \sum_{i=1}^{k} c_i^2 + \frac{\left(\sum_{i=1}^{k} c_i\right)^2}{u}}, \; f'_B(x_j) = f_B(x_j) + \gamma (f_B(x_{j_k^*}) - f_B(x_j))_+.$$

Proof. Let $S = \{x_1, \ldots, x_m\}$. Let $r_f(S) = J = [j_1, \ldots, j_m]$, and let $J^{-1} = [\ell_1, \ldots, \ell_m]$ be its inverse permutation. Similarly, let $J_B^{-1} = [\ell_1^*, \ldots, \ell_m^*]$ be the inverse permutation of $r_B(S) = J_B = [j_1^*, \ldots, j_m^*]$. Let $M = f_B(x_{j_k^*})$, we have

$$\begin{aligned}
&\mathbf{DCG}(r_B, S) - \mathbf{DCG}(r_f, S) \\
=& \sum_{i=1}^{m} c_i ((f_B(x_{j_i^*}, S) - M) - (f_B(x_{j_i}, S) - M)) \\
=& \sum_{i=1}^{m} c_i ((f_B(x_{j_i^*}, S) - M) - (f_B(x_{j_i}, S) - M)_+) + \sum_{i=1}^{m} c_i (M - f_B(x_{j_i}, S))_+ \\
\leq & \frac{1}{1-\gamma} \left[\sum_{i=1}^{m} c_i ((f_B(x_{j_i^*}, S) - M) - (f'_B(x_{j_i}, S) - M)_+) + \sum_{i=1}^{m} c_i (M - f'_B(x_{j_i}, S))_+ \right] \\
=& \frac{1}{1-\gamma} \left(\sum_{i=1}^{m} c_i f_B(x_{j_i^*}, S) - \sum_{i=1}^{m} c_i f'_B(x_{j_i}, S) \right) \\
\leq & \frac{1}{1-\gamma} \left(\sum_{i=1}^{m} c_i (f_B(x_{j_i^*}, S) - f(x_{j_i^*}, S)) - \sum_{i=1}^{m} c_i (f'_B(x_{j_i}, S) - f(x_{j_i}, S)) \right) \\
\leq & \frac{1}{1-\gamma} \left(\sum_{i=1}^{m} c_i (f_B(x_{j_i^*}, S) - f(x_{j_i^*}, S))_+ + \sum_{i=1}^{m} c_i (f(x_{j_i}, S) - f'_B(x_{j_i}, S))_+ \right) \\
\leq & \frac{1}{1-\gamma} \left((\sum_{i=1}^{k} c_i^2)^{1/2} \left[(\sum_{j \in K} (f_B(x_j, S) - f(x_j, S))_+^2)^{1/2} + (\sum_{j \in K} (f(x_{j_i}, S) \right. \right. \\
& \left. \left. - f'_B(x_{j_i}, S))_+^2)^{1/2} \right] + (\sum_{i=1}^{k} c_i) \sup_{j \notin K} (f(x_j, S) - f'_B(x_j, S))_+ \right) \\
\leq & \frac{1}{1-\gamma} \left(\sqrt{2 \sum_{i=1}^{k} c_i^2 \sum_{j \in K} (f_B(x_j, S) - f(x_j, S))^2} + \sum_{i=1}^{k} c_i \sup_{j \notin K} (f(x_j, S) - f'_B(x_j, S))_+ \right).
\end{aligned}$$

Note that in the above derivation, Cauchy-Schwartz inequality has been applied multiple times. From the last inequality, we can apply the Schwartz inequality (again) to obtain the desired bound. □

Intuitively, the bound says the following: we should estimate the top ranked items using least squares. For the other items, we do not have to make very accurate estimation of their conditional expectations. The DCG score will not be affect as long as we do not over-estimate their conditional expectations to such a degree that some of these items are near the top of the rank-list.

4 Regression Based Learning

Motivated by the analysis in Section 3, we consider regression based training method to solve the DCG optimization problem. We shall not discuss the implementation details for modeling the function $f(x, S)$, which is beyond the scope of this paper. One simple model is to assume a form $f(x, S) = f(x)$. Section 2.3 discussed the validity of such models. For example, this model is reasonable if we assume that for each $x \in S$, and the corresponding y, we have: $\mathbf{E}_{y|(x,S)}y = \mathbf{E}_{y|x}y$ (see Proposition 1).

Let $\mathcal{F}$ be a function space that contains functions $\mathcal{X} \times \mathcal{S} \to R$. We draw n sets $S_1, \ldots, S_n$ randomly, where $S_i = \{x_{i,1}, \ldots, x_{i,m}\}$, with the corresponding grades $\{y_{i,j}\}_j = \{y_{i,1}, \ldots, y_{i,m}\}$. Based on Theorem 2, a simple regression based approach can be used to solve the ranking problem:

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \left[\sum_{j=1}^{m} (f(x_{i,j}, S_i) - y_{i,j})^2 \right].$$

However, this direct regression method is not appropriate for large scale ranking problems such as web-search, for which there are many items to rank but only the top ranked pages are important. This is because the method pays equal attention to relevant and irrelevant pages. In reality, one should pay more attention to the top-ranked (relevant) pages. The grades of lower rank pages do not need to be estimated accurately, as long as we do not over-estimate them so that these pages appear in the top ranked positions.

The above mentioned intuition can be captured by Theorem 4, which motivates the following alternative training method:

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} L(f, S_i, \{y_{i,j}\}_j), \tag{5}$$

where for $S = \{x_1, \ldots, x_m\}$, with the corresponding $\{y_j\}_j$, we have

$$\begin{aligned} & L(f, S, \{y_j\}_j) \\ = & \sum_{j=1}^{m} w(x_j, S)(f(x_j, S) - y_j)^2 + u \sup_j w'(x_j, S)(f(x_j, S) - \delta(x_j, S))_+^2, \end{aligned} \tag{6}$$

where u is a non-negative parameter. A variation of this method is used to optimize the production system of Yahoo's internet search engine. The detailed

implementation and parameter choices are trade secrets of Yahoo, which we cannot completely disclose here[2]. It is also irrelevant for the purpose of this paper. However, in the following, we shall briefly explain the intuition behind (6) using Theorem 4, and some practical considerations.

The weight function $w(x_j, S)$ is chosen so that it focuses only on the most important examples (the weight is set to zero for pages that we know are irrelevant). This part of the formulation corresponds to the first part of the bound in Theorem 4 (in that case, we choose $w(x_j, S)$ to be one for the top part of the example with index set K, and zero otherwise). The specific choice of the weight function is not important for the purpose of this paper. In the second part of the formulation, we choose $w'(x_j, S)$ so that it focuses on the examples not covered by $w(x_j, S)$. In particular, it only covers those data points x_j that are low-ranked with high confidence. We choose $\delta(x_j, S)$ to be a small threshold that can be regarded as a lower bound of $f'_B(x_j)$ in Theorem 4, such as $\gamma f_B(x^*_k)$. An important observation is that although m is often very large, the number of points so that $w(x_j, S)$ is nonzero is often small. Moreover, $(f(x_j, S) - \delta(x_j, S))_+$ is not zero only when $f(x_j, S) \geq \delta(x_j, S)$. In practice the number of these points is usually small (that is, most irrelevant pages will be predicted as irrelevant). Therefore the formulation completely ignores those low-ranked data points such that $f(x_j, S) \leq \delta(x_j, S)$. This makes the algorithm computationally efficient even when m is large. The analogy here is support vector machines, where only the support vectors are useful in the learning formulation. One can completely ignore samples corresponding to non support vectors.

In the practical implementation of (6), we can use an iterative refinement scheme, where we start with a small number of samples to be included in the first part of (6), and then put the low-ranked points into the second part of (6) only when their ranking scores exceed $\delta(x_j, S)$. In fact, one may also put these points into the first part of (6), so that the second part always has zero values (which makes the implementation simpler). In this sense, the formulation in (6) suggests a selective sampling scheme, in which we pay special attention to important and highly ranked data points, while completely ignoring most of the low ranked data points. In this regard, with appropriately chosen $w(x, S)$, the second part of (6) can be completely ignored.

The empirical risk minimization method in (5) approximately minimizes the following criterion:

$$Q(f) = \mathbf{E}_S L(f, S), \tag{7}$$

where

$$\begin{aligned} L(f, S) &= \mathbf{E}_{\{y_j\}_j|S} L(f, S, \{y_j\}_j) \\ &= \sum_{j=1}^{m} w(x_j, S) \mathbf{E}_{y_j|(x_j,S)} \left(f(x_j, S) - y_j\right)^2 + u \sup_j w'(x_j, S)(f(x_j, S) - \delta(x_j, S))_+^2. \end{aligned}$$

The following theorem shows that under appropriate assumptions, approximate minimization of (7) leads to the approximate optimization of DCG.

[2] Some aspects of the implementation were covered in [15].

Theorem 5. *Assume that $c_i = 0$ for all $i > k$. Assume the following conditions hold for each $S = \{x_1, \ldots, x_m\}$:*

- *Let the optimal ranking order be $J_B = [j_1^*, \ldots, j_m^*]$, where $f_B(x_{j_i^*})$ is arranged in non-increasing order.*
- *There exists $\gamma \in [0, 1)$ such that $\delta(x_j, S) \leq \gamma f_B(x_{j_k^*}, S)$.*
- *For all $f_B(x_j, S) > \delta(x_j, S)$, we have $w(x_j, S) \geq 1$.*
- *Let $w'(x_j, S) = I(w(x_j, S) < 1)$.*

Then the following results hold:

- *A function f_* minimizes (7) if $f_*(x_j, S) = f_B(x_j, S)$ when $w(x_j, S) > 0$ and $f_*(x_j, S) \leq \delta(x_j, S)$ otherwise.*
- *For all f, let r_f be the induced ranking function. Let r_B be the optimal Bayes ranking function, we have:*

$$\mathbf{DCG}(r_f) - \mathbf{DCG}(r_B) \leq C(\gamma, u)(Q(f) - Q(f_*))^{1/2}.$$

Proof. Note that if $f_B(x_j, S) > \delta(x_j, S)$, then $w(x_j, S) \geq 1$ and $w'(x_j, S)$. Therefore the minimizer $f_*(x_j, S)$ should minimize $\mathbf{E}_{y_j|(x_j,S)}(f(x_j, S) - y_j)^2$, achieved at $f_*(x_j, S) = f_B(x_j, S)$. If $f_B(x_j, S) \leq \delta(x_j, S)$, then there are two cases:

- $w(x_j, S) > 0$, $f_*(x_j, S)$ should minimize $\mathbf{E}_{y_j|(x_j,S)}(f(x_j, S) - y_j)^2$, achieved at $f_*(x_j, S) = f_B(x_j, S)$.
- $w(x_j, S) = 0$, $f_*(x_j, S)$ should minimize $\mathbf{E}_{y_j|(x_j,S)}(f(x_j, S) - \delta(x_j, S))_+^2$, achieved at $f_*(x_j, S) \leq \delta(x_j, S)$.

This proves the first claim.

For each S, denote by K the set of x_j such that $w'(x_j, S) = 0$. The second claim follows from the following derivation:

$$\begin{aligned}
&Q(f) - Q(f_*) \\
=&\mathbf{E}_S(L(f, S) - L(f_*, S)) \\
=&\mathbf{E}_S\left[\sum_{j=1}^{k} w(x_j, S)(f(x_j, S) - f_B(x_j, S))^2 + u \sup_j w'(x_j, S)(f(x_j, S) - \delta(x_j, S))_+^2\right] \\
\geq&\mathbf{E}_S\left[\sum_{j \in K} (f_B(x_j, S) - f(x_j, S))_+^2 + u \sup_{j \notin K}(f(x_j, S) - \delta(x_j, S))_+^2\right] \\
\geq&\mathbf{E}_S(\mathbf{DCG}(r_B, S) - \mathbf{DCG}(r_f, S))^2 C(\gamma, u)^{-2} \\
\geq&(\mathbf{DCG}(r_B) - \mathbf{DCG}(r_f))^2 C(\gamma, u)^{-2}.
\end{aligned}$$

Note that the second inequality follows from Theorem 4. □

5 Generalization Analysis

In this section, we analyze the generalization performance of an L_1-boosting method, similar to [16, 2, 17]. Yahoo's machine learning ranking system employs the closely related gradient boosting method in [18], which can be similarly analyzed.

Consider a class of so-called *weak learners* $\mathcal{H}$, consisting of binary functions $\mathcal{X} \times \mathcal{S} \to \{0,1\}$, with a finite VC-dimension $\mathrm{VC}(\mathcal{H})$. We define for all $\beta \geq 0$: $\mathrm{CO}_\beta(\mathcal{H}) = \{f : \ f(x) = \sum_{i=1}^t \alpha_i h_i(x), \ \sum_{i=1}^t |\alpha_i| \leq \beta, \ h_i \in \mathcal{H}\}$. We are interested in algorithms that select a hypothesis from $\mathrm{CO}_\beta(\mathcal{H})$. Similar to Section 4, we use $(S_i, \{y_{i,j}\}_j)$ to indicate a sample point indexed by i. Note that for each sample i, we shall not assume that $y_{i,j}$ are independently generated for different j.

The following result is a simplified uniform convergence bound for the empirical risk minimization method in (5).

Theorem 6. *Assume that grades $y \in [0,1]$. Consider $\beta > 1$, and let $\hat{f}$ be the estimator defined in (5), with $\mathcal{F} = \mathrm{CO}_\beta(\mathcal{H})$. Then we have*

$$\mathbf{E}_{\{S_i,\{y_{i,j}\}_j\}_{i=1}^n} Q(\hat{f}) \leq \inf_{f \in \mathrm{CO}_\beta(\mathcal{H})} Q(f) + C\beta^2 \sqrt{\frac{W \cdot \mathrm{VC}(\mathcal{H})}{n}},$$

where C is a universal constant and

$$W = \mathbf{E}_S \left[\sum_{j=1}^m w(x_j, S) + u \sup_j w'(x_j, S) \right]^2.$$

Due to the limitation of space, we shall skip the proof, which is an adaptation of the standard Rademacher complexity analysis to our setting. Here we have paid special attention to the properties of (5). In particular, the quantity W is usually much smaller than m, which is large for web-search applications. The point we'd like to emphasize here is that even though the number m is large, the estimation complexity is only affected by the top-portion of the rank-list. If the estimation of the bottom ranked items is relatively easy (as is generally the case), then the learning complexity does not depend on the majority of items near the bottom of the rank-list.

We can combine Theorem 5 and Theorem 6, giving the following bound:

Theorem 7. *Suppose the conditions in Theorem 5 and Theorem 6 hold with f_* minimizing (7). We have*

$$\begin{aligned}
&\mathbf{E}_{\{S_i,\{y_{i,j}\}_j\}_{i=1}^n} \mathbf{DCG}(r_{\hat{f}}) \\
\leq &\mathbf{DCG}(r_B) + C(\gamma, u) \left[\inf_{f \in \mathrm{CO}_\beta(\mathcal{H})} Q(f) - Q(f_*) + C\beta^2 \sqrt{\frac{W \cdot \mathrm{VC}(\mathcal{H})}{n}} \right]^{1/2}.
\end{aligned}$$

Proof. From Theorem 5, we obtain

$$\begin{aligned}
&\mathbf{E}_{\{S_i,\{y_{i,j}\}_j\}_{i=1}^n} \mathbf{DCG}(r_{\hat{f}}) - \mathbf{DCG}(r_B) \\
\leq &C(\gamma, u) \mathbf{E}_{\{S_i,\{y_{i,j}\}_j\}_{i=1}^n} (Q(\hat{f}) - Q(f_*))^{1/2} \\
\leq &C(\gamma, u) (\mathbf{E}_{\{S_i,\{y_{i,j}\}_j\}_{i=1}^n} Q(\hat{f}) - Q(f_*))^{1/2}.
\end{aligned}$$

Now by applying Theorem 6, we obtain the desired bound. □

The theorem implies that if $Q(f_*) = \lim_{\beta\to\infty} \inf_{f\in \mathrm{CO}_\beta(\mathcal{H})} Q(f)$, then as $n \to \infty$, we can let $\beta \to \infty$ and $\beta^2/\sqrt{n} \to 0$ so that the second term on the right hand side vanishes in the large sample limit. Therefore asymptotically, we can achieve the optimal DCG score. This implies the consistency of regression based learning methods for the DCG criterion.

6 Conclusion

This paper considers the subset ranking problem, motivated by the web-search application. We investigated the DCG criterion that emphasizes the quality of the top-ranked items, and derived bounds that relate the optimization of DCG scores to the minimization of convex regression errors. These bounds can be used to motivate regression based methods that focus on the top-portion of the rank-list. In addition to conceptual advantages, these methods have significant computational advantages over standard regression methods because only a small number of items contribute to the solution. This means that they are computationally efficient to solve. As we have commented, the implementation of these methods can be achieved through appropriate selective sampling procedures. Moreover, we showed that the generalization performance of the system does not depend on m. Instead, it only depends on the estimation quality of the top ranked items. Again this is important for practical applications.

Results obtained here are closely related to the theoretical analysis for solving classification methods using convex optimization formulations. Our theoretical results show that the regression approach provides a solid basis for solving the subset ranking problem. The practical value of such methods is also significant. In Yahoo's case, substantial improvement of DCG has been achieved after the deployment of machine learning based ranking system. At the time of this writing, the system performance is already on par with the competitions, while further improvements are expected in the future.

Although the DCG criterion is difficult to optimize directly, it is a natural metric for ranking. The investigation of convex surrogate formulations provides a systematic approach to developing efficient machine learning methods for solving this difficult problem. We shall point out that the convex surrogate bounds proved in this paper are still quite loose. Therefore by deriving tighter bounds and developing better understanding of the ranking problem, we may obtain improved machine learning methods in the future.

References

1. Bartlett, P., Jordan, M., McAuliffe, J.: Convexity, classification, and risk bounds. Technical Report 638, Statistics Department, University of California, Berkeley (2003) to appear in JASA.
2. Lugosi, G., Vayatis, N.: On the Bayes-risk consistency of regularized boosting methods. The Annals of Statistics **32** (2004) 30–55 with discussion.
3. Zhang, T.: Statistical behavior and consistency of classification methods based on convex risk minimization. The Annals of Statistics **32** (2004) 56–85 with discussion.

4. Zhang, T.: Statistical analysis of some multi-category large margin classification methods. Journal of Machine Learning Research **5** (2004) 1225–1251
5. Steinwart, I.: Support vector machines are universally consistent. J. Complexity **18** (2002) 768–791
6. Tewari, A., Bartlett, P.: On the consistency of multiclass classification methods. In: COLT. (2005)
7. Jarvelin, K., Kekalainen, J.: IR evaluation methods for retrieving highly relevant documents. In: SIGIR'00. (2000) 41–48
8. Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.: Learning to rank using gradient descent. In: ICML'05. (2005)
9. Hanley, J., McNeil, B.: The meaning and use of the Area under a Receiver Operating Characetristic (ROC) curve. Radiology (1982) 29–36
10. Agarwal, S., Graepel, T., Herbrich, R., Har-Peled, S., Roth, D.: Generalization bounds for the area under the ROC curve. Journal of Machine Learning Research **6** (2005) 393–425
11. Agarwal, S., Roth, D.: Learnability of bipartite ranking functions. In: Proceedings of the 18th Annual Conference on Learning Theory. (2005)
12. Clemencon, S., Lugosi, G., Vayatis, N.: Ranking and scoring using empirical risk minimization. In: COLT'05. (2005)
13. Rosset, S.: Model selection via the AUC. In: ICML'04. (2004)
14. Herbrich, R., Graepel, T., Obermayer, K.: Large margin rank boundaries for ordinal regression. In A. Smola, P. Bartlett, B.S., Schuurmans, D., eds.: Advances in Large Margin Classifiers. MIT Press (2000) 115–132
15. Cossock, D.: Method and apparatus for machine learning a document relevance function. US patent application, 20040215606 (2003)
16. Blanchard, G., Lugosi, G., Vayatis, N.: On the rate of convergence of regularized boosting classifiers. Journal of Machine Learning Research **4** (2003) 861–894
17. Mannor, S., Meir, R., Zhang, T.: Greedy algorithms for classification - consistency, convergence rates, and adaptivity. Journal of Machine Learning Research **4** (2003) 713–741
18. Friedman, J.: Greedy function approximation: A gradient boosting machine. The Annals of Statistics **29** (2001) 1189–1232