Statistical Analysis of Bayes Optimal Subset Ranking

David Cossock
Yahoo Inc., Santa Clara, CA, USA
dcossock@yahoo-inc.com
Tong Zhang
Yahoo Inc., New York City, USA
tzhang@yahoo-inc.com

Abstract

The ranking problem has become increasingly important in modern applications of statistical methods in automated decision making systems. In particular, we consider a formulation of the statistical ranking problem which we call subset ranking, and focus on the DCG (discounted cumulated gain) criterion that measures the quality of items near the top of the rank-list. Similar to error minimization for binary classification, direct optimization of natural ranking criteria such as DCG leads to a non-convex optimization problems 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 many internet applications, where personalization and information based decision making are 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, 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 characteristic of practical ranking problems has not been carefully explored in earlier studies (except for a recent paper [23], which also touched on this issue). 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 [4, 18, 29, 28, 24, 25].

We organize the paper as follows. Section 2 discusses earlier work in statistics and machine learning on global and pair-wise ranking. Section 3 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 in the context of pair-wise preference learning. Section 4 investigates the relationship of subset ranking and global ranking. Section 5 introduces some basic estimation methods for ranking. This paper focuses on the least squares regression based formulation. Section 6 contains the main theoretical results in this paper, where we show that the approximate minimization of certain regression errors leads 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 7 presents the regression learning formulation derived from the theoretical results in Section 6. Similar methods are currently used to optimize Yahoo's production search engine. Section 8 studies the generalization ability of regression learning, where we focus on the L_2-regularization approach. Together with earlier theoretical results, we can establish the consistency of regression based ranking under appropriate conditions.

2 Ranking and Pair-wise Preference Learning

The traditional prediction problem in statistical machine learning assumes that we observe an input vector $q \in \mathcal{Q}$, so as to predict an unobserved output $p \in \mathcal{P}$. However, in a ranking problem, if we assume $\mathcal{P} = \{1, \ldots, m\}$ contains m possible values, then instead of predicting a value in $\mathcal{P}$, we predict a permutation of $\mathcal{P}$ that gives an optimal ordering of $\mathcal{P}$. That is, if we denote by $\mathcal{P}!$ the set of permutations of $\mathcal{P}$, then the goal is to predict an output in $\mathcal{P}!$. There are two fundamental issues: first, how to measure the quality of ranking; second, how to learn a good ranking procedure from historical data.

At the first sight, it may seem that we can simply cast the ranking problem as an ordinary prediction problem where the output space becomes $\mathcal{P}!$. However, the number of permutations in $\mathcal{P}!$ is $m!$, which can be extremely large even for small m. Therefore it is not practical to solve the ranking problem directly without imposing certain structures on the search space. Moreover, in practice, given a training point $q \in \mathcal{Q}$, we are generally not given an optimal permutation in $\mathcal{P}!$ as the observed output. Instead, we may observe another form of output that typically infers the optimal ranking but may contain extra information as well. The training procedure should take advantage of such information.

A common idea to generate optimal permutation in $\mathcal{P}!$ is to use a scoring function that takes a pair (q, p) in $\mathcal{Q} \times \mathcal{P}$, and maps it to a real valued number $r(q, p)$. For each q, the predicted permutation in $\mathcal{P}!$ induced by this scoring function is defined as the ordering of $p \in \mathcal{P}$ sorted with non-increasing value $r(q, p)$. This is the method we will focus on in this paper.

Although the ranking problem have received considerable interests in machine learning recently due to its important applications in modern automated information processing systems, the problem has not been extensively studied in the traditional statistical literature. A relevant statistical model is *ordinal regression* [20]. In this model, we are still interested in predicting a single output. We redefine the input space as $\mathcal{X} = \mathcal{Q} \times \mathcal{P}$, and for each x, we observe an output value $y \in \mathcal{Y}$. Moreover, we assume that the values in $\mathcal{Y} = \{1, \ldots, L\}$ are ordered, and the cumulative probability $P(y \leq j|x)$ $(j = 1, \ldots, L)$ has the form $\gamma(P(y \leq j|x)) = \theta_j + g_\beta(x)$. In this model, both $\gamma(\cdot)$ and $g_\beta(\cdot)$ have known functional forms, and θ and β are model parameters.

Note that the ordinal regression model induces a stochastic preference relationship on the input space $\mathcal{X}$. Consider two samples (x_1, y_1) and (x_2, y_2) on $\mathcal{X} \times \mathcal{Y}$. We say $x_1 \prec x_2$ if and only if $y_1 < y_2$. This is a classification problem that takes a pair of input x_1 and x_2 and tries to predict whether $x_1 \prec x_2$ or not (that is, whether the corresponding outputs satisfy $y_1 < y_2$ or not). In this formulation, the optimal prediction rule to minimize classification error is induced by the ordering of $g_\beta(x)$ on $\mathcal{X}$ because if $g_\beta(x_1) < g_\beta(x_2)$, than $P(y_1 < y_2) > 0.5$ (based on the ordinal regression model), which is consistent with the Bayes rule. Motivated by this observation, an SVM ranking method is proposed in [16]. The idea is to reformulate ordinal regression as a model to learn preference relationship on the input space $\mathcal{X}$, which can be learned using pair-wise classification. Given the parameter $\hat{\beta}$ learned from training data, the scoring function is simply $r(q, p) = g_{\hat{\beta}}(x)$.

The pair-wise preference learning model becomes a major trend for ranking in the machine learning literature. For example, in addition to SVM, a similar method based on AdaBoost is proposed in [13]. The idea was also used in optimizing the Microsoft web-search system [7].

A number of researchers worked on the theoretical analysis of ranking, using the pair-wise ranking model. The criterion is to minimize the error of pair-wise preference prediction when we draw two pairs x_1 and x_2 randomly from the input space $\mathcal{X}$. That is, given a scoring function $g : \mathcal{X} \to R$, the ranking loss is:

$$\begin{aligned} &\mathbf{E}_{(X_1,Y_1)}\mathbf{E}_{(X_2,Y_2)}[I(Y_1 < Y_2)I(g(X_1) \geq g(X_2)) + I(Y_1 > Y_2)I(g(X_1) \leq g(X_2))] \\ =&\mathbf{E}_{X_1,X_2}[P(Y_1 < Y_2|X_1, X_2)I(g(X_1) \geq g(X_2)) + P(Y_1 > Y_2|X_1, X_2)I(g(X_1) \leq g(X_2))], \end{aligned} \tag{1}$$

where $I(\cdot)$ denotes the indicator function. For binary output $y = 0, 1$, it is known that this metric is equivalent to the AUC measure (area under ROC) for binary classifiers up to a scaling, and it is closely related to the Mann-Whitney-Wilcoxon statistics [15]. In the literature, theoretical analysis has focused mainly on this ranking criterion (for example, see [1, 2, 9, 22]).

The pair-wise preference learning model has some limitations. First, although the criterion in (1) measures the global pair-wise ranking quality, it is not the best metric to evaluate practical ranking systems. Note that in most applications, a system does not need to rank all data-pairs, but only a subset of them each time. Moreover, typically only the top few positions of the rank-list is of importance. Another issue with the pair-wise preference learning model is that the scoring function is usually learned by minimizing a convex relaxation of the pair-wise classification error, similar to large margin classification. However, if the preference relationship is stochastic, then an important question that should be addressed is whether such a learning algorithm leads to a Bayes optimal ranking function in the large sample limit. Unfortunately this is difficult to analyze for

general risk minimization formulations if the decision rule is induced by a single-variable scoring function of the form $r(x)$.

The problem of Bayes optimality in the pair-wise learning model was partially investigated in [9], but with a decision rule of a general form $r(x_1, x_2)$: we predict $x_1 \prec x_2$ if $r(x_1, x_2) < 0$. To our knowledge, this method is not widely used in practice because a naive application can lead to contradiction: we may predict $r(x_1, x_2) < 0$, $r(x_2, x_3) < 0$, and $r(x_3, x_1) < 0$. Therefore in order to use such a method effectively for ranking, there needs to be a mechanism to resolve such contradiction. For example, one possibility is to define a scoring function $f(x) = \sum_{x'} r(x, x')$, and rank the data accordingly. Another possibility is to use a sorting method (such as quick-sort) directly with the comparison function given by $r(x_1, x_2)$. However, in order to show that such contradiction resolution methods are well behaved asymptotically, it is necessary to analyze the corresponding error. We are not aware of any study on such error analysis.

3 Subset Ranking Model

The global pair-wise preference learning model in Section 2 has some limitations. In this paper, we shall describe a model more relevant to practical ranking systems such as web-search. We will first describe the model, and then use search as an example to illustrate it.

3.1 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 purposes 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 $i = 1, \ldots, m$. In practical applications, each available position i can be associated with a weight c_i that measures the importance of that position. Now, given the grades $y_j (j = 1, \ldots, m)$, a very natural measure of the rank-list $J = [j_1, \ldots, j_m]$'s quality is the following weighted sum:

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

We assume that $\{c_i\}$ is a pre-defined sequence of non-increasing non-negative discount factors that may or may not depend on S. This metric, described in [17] as DCG (discounted cumulated gain), is one of the main metrics used in the evaluation of internet search systems, including the

production system of Yahoo and that of Microsoft [7]. In this context, a typical choice of c_i is to set $c_i = 1/\log(1+i)$ when $i \le k$ and $c_i = 0$ when $i > k$ for some k. One may also use other choices, such as letting c_i be the probability of user looking at (or clicking) the result at position i.

Although introduced in the context of web-search, the DCG criterion is clearly natural for many other ranking applications such as recommender systems. Moreover, by choosing a decaying sequence of c_i, this measure naturally focuses on the quality of the top portion of the rank-list. This is in contrast with the pair-wise error criterion in (1), which does not distinguishing top portion of the rank-list from the bottom portion.

For the DCG criterion, 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:

$$\mathbf{DCG}(r) = \mathbf{E}_S \, \mathbf{DCG}(r, S), \tag{2}$$

where

$$\mathbf{DCG}(r, S) = \sum_{i=1}^{m} c_i \mathbf{E}_{y_{j_i}|(x_{j_i}, S)} \; y_{j_i}. \tag{3}$$

The global pair-wise preference learning metric (1) can be adapted to the subset ranking setting. We may consider the following 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 equivalent to (1). Although we pay special attention to the DCG metric, we shall also include some analysis of the $\mathbf{T}$ criterion for completeness.

Similar to (2) and (3), we can define the following quantities:

$$\mathbf{T}(r) = \mathbf{E}_S \, \mathbf{T}(r, S), \tag{4}$$

where

$$\mathbf{T}(r, S) = \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{i'=i+1}^{m} (\mathbf{E}_{y_{j_i}|(x_{j_i}, S)} \; y_{j_i} - \mathbf{E}_{y_{j_{i'}}|(x_{j_{i'}}, S)} \; y_{j_{i'}}). \tag{5}$$

Similar to the concept of Bayes classifier in classification, we can define the Bayes ranking function that optimizes the $\mathbf{DCG}$ and $\mathbf{T}$ measures. Based on the conditional formulations in (3) and (5), 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 (5) returns a rank list $J = [j_1, \ldots, j_m]$ such that $f_B(x_{j_i}, S)$ is in descending order: $f_B(x_{j_1}, S) \ge f_B(x_{j_2}, S) \ge \cdots \ge f_B(x_{j_m}, S)$. An optimal Bayes ranking function $r_B(S)$ that maximizes (3) 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 $\mathbf{T}$-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 $\mathbf{DCG}$-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 $f_B(x_{j_k}, S) \geq f_B(x_{j_{k'}}, S)$. ■

The result indicates that the optimal ranking can be induced by a single variable ranking function of the form $r(x, S) : \mathcal{X} \times \mathcal{S} \to R$ where $x \in S$.

3.2 Web-search example

As an example of the subset ranking model, we consider the web-search problem. In this application, a 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 goes 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.

4 Some Computational Aspects of Subset Ranking

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

The explicit S dependency of $f_B(x, S)$ is one of the differences that distinguish subset ranking from global ranking. 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 $f_B(x_j, S) = f([x_j, x_1, \ldots, x_m])$. Therefore by redefining $\tilde{x}_j = [x_j, x_1, \ldots, x_m] \in \mathcal{X}^{m+1}$, we can remove the subset dependency by embedding the original problem into a higher dimensional space. In the general case when m is large, this approach is not practical. Instead of using the

[1] For example, the formula $(2^{s_j} - 1)/(2^5 - 1)$ is used in [7]. Yahoo uses a different transformation based on empirical user surveys.

whole set S as a feature, we can project S into a lower dimensional space using a feature map $g(\cdot)$, so that $f_B(x, g(S)) \approx f(x, g(S))$. By introducing such a set dependent feature vector $g(S)$, we can remove the set dependency by incorporating $g(S)$ into x: this can be achieved by simply redefining x as $\tilde{x} = [x, g(S)]$. In this way, $f_B(x, S)$ can be approximated by a function of the form $f(\tilde{x})$.

If the subsets are identical, then subset ranking is equivalent to global ranking. In the more general case where subsets are not identical, the reduction from set-dependent local ranking into set-independent global ranking can be complicated if we do not assume any underlying structures of the problem (we shall discuss such a structure later). However, one may ask the question that if we only use a set-independent function of the form $f(x)$ as the scoring function, how well it can approximate the Bayes scoring function $f_B(x, S)$, and whether it is easy to compute such a function $f(x)$.

If the subsets are disjoint (or nearly disjoint), then the effect of $f_B(x, S)$ can be achieved by a global scoring function of the form $f(x)$ exactly (or almost exactly) because x determines S. This can be a good approximation for practical problems, where the feature vectors for different subsets (e.g. queries in web-search) usually do not overlap.

If the subsets overlap significantly but not exactly the same, the problem can be computationally difficult. To see this, we may consider for simplicity that $\mathcal{X}$ is finite, and each subset only contains two elements, and one is preferred over the other (deterministically). Now in the subset learning model, such a preference relationship $x \prec x'$ of two elements $x, x' \in \mathcal{X}$ can be denoted by a directed edge from x to x'. In this setting, to find a global scoring function that approximates the optimal set dependent Bayes scoring rule is equivalent to finding a maximum subgraph that is acyclic. In general, this problem is computationally difficult, and known to be NP-hard (an application of similar arguments in ranking can be found in [12, 3]) as well as APX-hard [11]: the class APX consists of problems having an approximation to with $1+c$ of the optimum for some c. A polynomial time approximation scheme (PTAS) is an algorithm which runs in polynomial time in the instance size (but not necessarily poly$(1/\epsilon)$) and returns a solution approximate to within $1+\epsilon$ for any given $\epsilon > 0$. If any APX-hard problem admits a PTAS then P=NP.

The above argument implies that without any assumption, the reduction of the set-dependent Bayes optimal scoring function $f_B(x, S)$ to a set independent function of the form $f(x)$ is difficult. If we are able to incorporate appropriate set dependent feature into x or if the sets do not overlap significantly, then this is computationally feasible. In the ideal case, we can introduce the following definition.

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

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

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

Clearly, an optimal rank preserving function may not always exist (without using set-dependent features). 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 $f(c) > f(a)$, which is impossible.

Under appropriate assumptions, the optimal rank preserving function exists. The following result provides a sufficient condition.

Proposition 1 *Assume that for each x_j, we observe $y_j = a(S)y'_j + b(S)$ where $a(S) \neq 0$ and $b(S)$ are normalization/shifting factors 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.

Proof Observe that $\mathbf{E}_{y_j|(x_j,S)} y_j = a(S)\mathbf{E}_{y'_j|(x_j,S)} y'_j + b(S)$. Therefore the scoring functions $\mathbf{E}_{y_j|(x_j,S)} y_j$ and $\mathbf{E}_{y'_j|(x_j,S)} y'_j$ lead to identical ranking. Moreover,

$$\mathbf{E}_{y'_j|(x_j,S)}\, y'_j = \mathbf{E}_\xi \int y'_j d\prod_{i=1}^{m} P(y'_i|x_i, \xi) = \mathbf{E}_\xi \int y'_j dP(y'_j|x_j, \xi) = \int y'_j dP(y'_j|x_j) = \mathbf{E}_{y'_j|x_j}\, y'_j.$$

This proves the claim. ■

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 the optimality of set-independent conditional expectation. In this case, the optimal global scoring rule gives the optimal Bayes rule for subset ranking. We also note that this equivalence 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 this case, Proposition 1 can be applied to show that the features x_j are sufficient to determine the optimal Bayes rule.

Proposition 1 (and discussion there-after) suggests that regression based learning of the conditional expectation $\mathbf{E}_{y|x}\, y$ is asymptotically optimal under some assumptions that are reasonable. We call a method that learns such conditional expectation $\mathbf{E}_{y|x}\, y$ or its transformation *regression based approach*, which is different from the pair-wise preference learning methods used in the early work. There are two advantages for using regression: first, the computational complexity is at most $O(m)$ (it can be sub-linear in m with appropriate importance subsampling schemes) instead of $O(m^2)$; second, we are able to prove the consistency of such methods under reasonable assumptions. As discussed at the end of Section 2, this issue is more complicated for pair-wise methods. Furthermore, as we will discuss in the next section, some advantages of pair-wise learning can be incorporated into the regression approach by using set-dependent features.

5 Risk Minimization based Estimation Methods

From the previous section, we know that the optimal scoring function is the conditional expectation of the grades y. We investigate some basic estimation methods for conditional expectation learning.

5.1 Relation to multi-category classification

The subset ranking problem is a generalization of multi-category classification. In the latter 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 the subset ranking problem as follows. Consider a sample $S = [x_1, \ldots, x_m]$ 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 then form another sample with weight $\sum_j y_j$, which has the vector $\bar{S} = [x_1, \ldots, x_m]$ (where order is important) as input, and label $y' = k \in \{1, \ldots, m\}$ as output. This changes the problem formulation into multi-category classification. Since the conditional expectation can be expressed as

$$\mathbf{E}_{y_k|(x_k,S)}\, y_k = P(y' = k|S)\, \mathbf{E}_{\{y_j\}|S} \sum_j y_j,$$

the order induced by the scoring function $\mathbf{E}_{y_k|(x_k,S)}\, y_k$ is the same as that induced by $P(y' = k|S)$. Therefore a multi-category classification solver that estimates conditional probability can be used to solve the subset ranking problem. In particular, if we consider a risk minimization based multi-category classification solver for m-class problem [28, 25] of the following form:

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{n} \Phi(f(X_i), Y_i),$$

where (X_i, Y_i) are training points with $Y_i \in \{1, \ldots, m\}$, $\mathcal{F}$ is a vector function class that takes values in R^m, and Φ is some risk functional. Then for ranking with training points $(\bar{S}_i, \{y_{i,1}, \ldots, y_{i,m}\})$ and $\bar{S}_i = [x_{i,1}, \ldots, x_{i,m}]$, the corresponding learning method becomes

$$\hat{f} = \arg\min_{f \in \bar{\mathcal{F}}} \sum_{i=1}^{n} \sum_{j=1}^{m} y_{i,j} \Phi(f(\bar{S}_i), j),$$

where the function space $\bar{\mathcal{F}}$ contains a subset of functions $\{f(\bar{S}) : \mathcal{X}^m \to R^m\}$ of the form

$$f(\bar{S}) = [f(x_1, S), \ldots, f(x_m, S)], \quad \text{and } S = \{x_1, \ldots, x_m\} \text{ is unordered set.}$$

An example would be maximum entropy (multi-category logistic regression) which has the following loss function $\Phi(f(\bar{S}), j) = -f(x_j, S) + \ln \sum_{k=1}^{m} e^{f(x_k,S)}$.

5.2 Regression based learning

Since in ranking problems $y_{i,j}$ can take values other than 0 or 1, we can have more general formulations than multi-category classification. In particular, we may consider variations of the following regression based learning method to train a scoring function in $\mathcal{F} \subset \{\mathcal{X} \times \mathcal{S} \to R\}$:

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{n} \sum_{j=1}^{m} \phi(f(x_{i,j}, S_i), y_{i,j}), \quad S_i = \{x_{i,1}, \ldots, x_{i,m}\} \in \mathcal{S}, \tag{6}$$

where we assume that

$$\phi(a,b) = \phi_0(a) + \phi_1(a)b + \phi_2(b).$$

The estimation formulation is decoupled for each element $x_{i,j}$ in a subset S_i, which makes the problem easier to solve. In this method, each training point $((x_{i,j}, S_i), y_{i,j})$ is treated as a single sample (for $i = 1, \ldots, n$ and $j = 1, \ldots, m$). The population version of the risk function is:

$$\mathbf{E}_S \sum_{x \in S} \left[\phi_0(f(x,S)) + \phi_1(f(x,S))\mathbf{E}_{y|(x,S)} y + \mathbf{E}_{y|(x,S)}\phi_2(y).\right]$$

This implies that the optimal population solution is a function that minimizes

$$\phi_0(f(x,S)) + \phi_1(f(x,S))\mathbf{E}_{y|(x,S)} y,$$

which is a function of $\mathbf{E}_{y|(x,S)} y$. Therefore the estimation method in (6) leads to an estimator of conditional expectation with a reasonable choice of $\phi_0(\cdot)$ and $\phi_1(\cdot)$.

A simple example is the least squares method, where we pick $\phi_0(a) = a^2$, $\phi_1(a) = -2a$ and $\phi_2(b) = b^2$. That is, the learning method (6) becomes least squares estimation:

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{n} \sum_{j=1}^{m} (f(x_{i,j}, S_i) - y_{i,j})^2. \tag{7}$$

This method, and some essential variations which we will introduce later, will be the focus of our analysis.

It was shown in [8] that the only loss function with conditional expectation as the minimizer (for an arbitrary conditional distribution of y) is least squares. However, for practical purposes, we only need to estimate a monotonic transformation of the conditional expectation. For this purpose, we can have additional loss functions of the form (6). In particular, let $\phi_0(a)$ be an arbitrary convex function such that $\phi_0'(a)$ is a monotone increasing function of a, then we may simply take the function $\phi(a,b) = \phi_0(a) - ab$ in (6). The optimal population solution is uniquely determined by $\phi_0'(f(x,S)) = \mathbf{E}_{y|(x,S)} y$. A simple example is $\phi_0(a) = a^4/4$ such that the population optimal solution is $f(x,S) = (\mathbf{E}_{y|(x,S)} y)^{1/3}$. Clearly such a transformation does not affect ranking. Moreover, in many ranking problems, the range of y is bounded. It is known that additional loss functions can be used for computing the conditional expectation. As a simple example, if we assume that $y \in [0,1]$, then the following modified least squares can be used:

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{n} \sum_{j=1}^{m} \left[(1 - y_{i,j}) \max(0, f(x_{i,j}, S_i))^2 + y_{i,j} \max(0, 1 - f(x_{i,j}, S_i))^2\right]. \tag{8}$$

One may replace this with other loss functions used for binary classification that estimate conditional probability, such as those discussed in [29]. Although such general formulations might be interesting for certain applications, advantages over the simpler least squares loss of (7) are not completely certain, and they are more complicated to deal with. Therefore we will not consider such general formulations in this paper, but rather focus on adapting the least squares method in (7) to the ranking problems. As we shall see, non-trivial modifications of (7) are necessary to optimize system performance near the top of rank-list.

5.3 Pair-wise preference learning

A popular idea in the recent machine learning literature is to pose the ranking problem as a pair-wise preference relationship learning problem (see Section 2). Using this idea, the scoring function for subset ranking can be trained by the following method:

$$\hat{f} = \arg\min_{f\in\mathcal{F}} \sum_{i=1}^{n} \sum_{(j,j')\in E_i} \phi(f(x_{i,j}, S_i), f(x_{i,j'}, S_i); y_{i,j}, y_{i,j'}), \tag{9}$$

where each E_i is a subset of $\{1,\ldots,m\}\times\{1,\ldots,m\}$ such that $y_{i,j} < y_{i,j'}$. For example, we may use a non-increasing monotone function ϕ_0 and let $\phi(a_1,a_2;b_1,b_2) = \phi_0((a_2-a_1)-(b_2-b_1))$ or $\phi(a_1,a_2;b_1,b_2) = (b_2-b_1)\phi_0(a_2-a_1)$. Example loss functions include SVM loss $\phi_0(x) = \max(0,1-x)$ and AdaBoost loss $\phi_0(x) = \exp(-x)$ (see [13, 16, 23]).

The approach works well if the ranking problem is noise-free (that is, $y_{i,j}$ is deterministic). However, one difficulty with this approach is that if $y_{i,j}$ is stochastic, then the corresponding population estimator from (9) may not be Bayes optimal, unless a more complicated scheme such as [9] is used. It will be interesting to investigate the error of such an approach, but the analysis is beyond the scope of this paper.

One argument used by the advocates of the pair-wise learning formulation is that we do not have to learn an absolute grade judgment (or its expectation), but rather only the relative judgment that one item is better than another. In essence, this means that for each subset S, if we shift each judgment by a constant, the ranking is not affected. If invariance with respect to a set-dependent judgment shift is a desirable property, then it can be incorporated into the regression based model [26]. For example, similar to Proposition 1, we may introduce an explicit set dependent shift feature (which is rank-preserving) into (6):

$$\hat{f} = \arg\min_{f\in\mathcal{F}} \sum_{i=1}^{n} \min_{b_i\in R} \sum_{j=1}^{m} \phi(f(x_{i,j}, S_i) + b_i, y_{i,j}).$$

In particular, for least squares, we have the following method:

$$\hat{f} = \arg\min_{f\in\mathcal{F}} \sum_{i=1}^{n} \min_{b_i\in R} \sum_{j=1}^{m} (f(x_{i,j}, S_i) + b_i - y_{i,j})^2. \tag{10}$$

More generally, we may introduce more sophisticated set dependent features and hierarchical models into the regression formulation, and obtain effects that may not even be easily incorporated into pair-wise models.

6 Convex Surrogate Bounds

The subset ranking problem defined in Section 3 is combinatorial in nature, which is very difficult to solve. Since the optimal Bayes ranking rule is given by conditional expectation, in Section 5, we discussed various formulations to estimate the conditional expectation. In particular, we are interested in least squares regression based methods. In this context, a natural question to ask is if a scoring function approximately minimizes regression error, how well it can optimize ranking metrics such as DCG or $\mathbf{T}$. This section provides some theoretical results that relate the optimization of

the ranking metrics defined in Section 3 to the minimization of regression errors. This allows us to design appropriate convex learning formulations that improve the simple least squares methods in (7) and (10).

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. Consider pair $p, q \in [1, \infty]$ such that $1/p + 1/q = 1$. We have the following relationship for each $S = \{x_1, \ldots, x_m\}$:*

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

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

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

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

The above theorem shows that the DCG criterion can be bounded through regression error. Although the theorem applies to any arbitrary pair of p and q such that $1/p + 1/q = 1$, the most useful case is with $p = q = 2$. This is because in this case, the problem of minimizing $\sum_{j=1}^{m}(f(x_j, S) - f_B(x_j, S))^2$ can be directly achieved using least squares regression in (7). 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. Assume that the coefficients c_i in the DCG criterion decay fast, so that $\sum_i c_i$ is bounded (independent of m). In this case, we may pick $p = 1$ and $q = \infty$ in Theorem 2. If $\sup_j |f(x_j, S) - f_B(x_j, S)|$ is small, then we obtain a better bound than the least squares error bound with $p = q = 1/2$ which depends on m.

However, we cannot ensure that $\sup_j |f(x_j, S) - f_B(x_j, S)|$ is small using the simple least squares estimation in (7). Therefore in the following, we develop a more refined bound for the DCG metric, which will then be used to motivate practical learning methods that improve on the simple least squares method.

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 $\gamma \in (0, 1)$, $p, q \geq 1$ such that $1/p + 1/q = 1$, $u > 0$, and subset $K \subset \{1, \ldots, m\}$ that contains $j_1^*, \ldots, j_k^*$:*

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

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

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

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

$$(M - f_B(x_{j_i}, S))_+ \leq \frac{1}{1-\gamma} (M - f'_B(x_{j_i}, S))_+.$$

Moreover, since $\sum_{i=1}^m c_i ((f_B(x_{j_i^*}, S) - M) - (f_B(x_{j_i}, S) - M)_+) \geq 0$, we have

$$\sum_{i=1}^{m} c_i ((f_B(x_{j_i^*}, S) - M) - (f_B(x_{j_i}, S) - M)_+) \leq \frac{1}{1-\gamma} \sum_{i=1}^{m} c_i ((f_B(x_{j_i^*}, S) - M) - (f_B(x_{j_i}, S) - M)_+).$$

Therefore

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

Note that in the above derivation, Hölder's inequality has been applied to obtain the last two inequalities. From the last inequality, we can apply the Hölder's inequality again to obtain the desired bound. ■

The easiest way to interpret this bound is still to take $p = q = 1/2$. 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 affected 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. This point is a very important difference between this bound and Theorem 2 which assumes that we estimate the conditional expectation uniformly well.

The bound in Theorem 4 can still be refined. However, the resulting inequalities will become more complicated. Therefore we will not include such bounds in this paper. Similar to Theorem 4, such refined bounds show that we do not have to estimate conditional expectation uniformly well. We present a simple example as illustration.

Proposition 2 *Consider $m = 3$ and $S = \{x_1, x_2, x_3\}$. Let $c_1 = 2, c_2 = 1, c_3 = 0$, and $f_B(x_1, S) = f_B(x_2, S) = 1, f_B(x_3, S) = 0$. Let $f(x, S)$ be a real-valued scoring function, which induces a ranking function r_f. Then*

$$\mathbf{DCG}(r_B, S) - \mathbf{DCG}(r_f, S) \leq 2|f(x_3, S) - f_B(x_3, S)| + |f(x_1, S) - f_B(x_1, S)| + |f(x_2, S) - f_B(x_2, S)|.$$

The coefficients on the right hand side cannot be improved.

Proof Note that f is suboptimal only when either $f(x_3, S) \geq f(x_1, S)$ or when $f(x_3, S) \geq f(x_2, S)$. This gives the following bound:

$$\begin{aligned}
&\mathbf{DCG}(r_B, S) - \mathbf{DCG}(r_f, S) \\
\leq& I(f(x_3, S) \geq f(x_1, S)) + I(f(x_3, S) \geq f(x_2, S)) \\
\leq& I(|f(x_3, S) - f_B(x_3, S)| + |f(x_1, S) - f_B(x_1, S)| \geq 1) \\
&\quad + I(|f(x_3, S) - f_B(x_3, S)| + |f(x_2, S) - f_B(x_2, S)| \geq 1) \\
\leq& 2|f(x_3, S) - f_B(x_3, S)| + |f(x_1, S) - f_B(x_1, S)| + |f(x_2, S) - f_B(x_2, S)|.
\end{aligned}$$

To see that the coefficients cannot be improved, we simply note that the bound is tight when either $f(x_1, S) = f(x_2, S) = f(x_3, S) = 1$, or when $f(x_1, S) = 1$ and $f(x_2, S) = f(x_3, S) = 0$, or when $f(x_2, S) = 1$ and $f(x_1, S) = f(x_3, S) = 0$. ■

The Proposition implies that not all errors should be weighted equally: in the example, getting x_3 right is more important than getting x_1 or x_2 right. Conceptually, Theorem 4 and Proposition 2 show the following:

- Since we are interested in the top portion of the rank-list, we only need to estimate the top rated items accurately, while preventing the bottom items from being over-estimated (the conditional expectations don't have to be estimated accurately).
- For ranking purposes, some points are more important than other points. Therefore we should bias our learning method to produce more accurate conditional expectation estimation at the more important points.

7 Importance Weighted Regression

The key message from the analysis in Section 6 is that we do not have to estimate the conditional expectations equally well for all items. In particular, since we are interested in the top portion of the rank-list, Theorem 4 implies that we need to estimate the top portion more accurately than the bottom portion.

Motivated by this analysis, we consider a regression based training method to solve the DCG optimization problem but weight different points differently according to their importance. 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 4 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, the simple least squares regression (7) can be used to solve the subset ranking problem. However, this direct regression method is not adequate for many practical problems such as web-search, for which there are many items to rank (that is, m is large) 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 and Proposition 2, which motivate the following alternative training method:

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} L(f, S_i, \{y_{i,j}\}_j), \tag{11}$$

where for $S = \{x_1, \ldots, x_m\}$, with the corresponding $\{y_j\}_j$, we have the following importance weighted regression loss:

$$L(f, S, \{y_j\}_j) = \sum_{j=1}^{m} w(x_j, S)(f(x_j, S) - y_j)^2 + u \sup_j w'(x_j, S)(f(x_j, S) - \delta(x_j, S))_+^2, \tag{12}$$

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 (12) using Theorem 4, and some practical considerations.

The weight function $w(x_j, S)$ in (12) 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 usefulness of non-uniform weighting is also demonstrated in Proposition 2. The specific choice of the weight function requires various engineering considerations that are not important for the purpose of this paper. In general, if there are many items with similar grades, then it is beneficial to give each of the similar items a smaller weight. In the second part of (12), 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 learning procedure 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 (12), we can use an iterative refinement scheme, where we start with a small number of samples to be included in the first part of (12), and then put the

[2]Some aspects of the implementation were covered in [10].

low-ranked points into the second part of (12) only when their ranking scores exceed $\delta(x_j, S)$. In fact, one may also put these points into the first part of (12), so that the second part always has zero values (which makes the implementation simpler). In this sense, the formulation in (12) 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 (12) can be completely ignored.

The empirical risk minimization method in (11) approximately minimizes the following criterion:

$$Q(f) = \mathbf{E}_S L(f, S), \tag{13}$$

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 (13) leads to the approximate optimization of DCG.

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 (13) 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. ■

8 Generalization Analysis

In this section, we analyze the generalization performance of (11). The analysis depends on the underlying function class $\mathcal{F}$. In the literature, one often employs a linear function class with appropriate regularization condition, such as L_1 or L_2 regularization for the linear weight coefficients. Yahoo's machine learning ranking system employs the gradient boosting method described in [14], which is closely related to L_1 regularization, analyzed in [5, 18, 19]. Although the consistency of boosting for the standard least squares regression is known (for example, see [6, 30]), such analysis does not deal with the situation that m is large and thus is not suitable for analyzing the ranking problem considered in this paper.

In this section, we will consider linear function class with L_2 regularization, which is closely related to kernel methods. We employ a relatively simple stability analysis which is suitable for L_2 regularization. Our result does not depend on m explicitly, which is important for large scale ranking problems such as web-search. Although similar results can be obtained for L_1 regularization or gradient boosting, the analysis will become much more complicated.

For L_2 regularization, we consider a feature map $\psi : \mathcal{X} \times \mathcal{S} \to \mathcal{H}$, where $\mathcal{H}$ is a vector space. We denote by $w^T v$ the L_2 inner product of w and v in $\mathcal{H}$. The function class $\mathcal{F}$ considered here is of the following form:

$$\{\beta^T \psi(x,S);\ \beta \in \mathcal{H}, \beta^T\beta \leq A^2\} \subset \mathcal{X} \times \mathcal{S} \to R, \tag{14}$$

where the complexity is controlled by L_2 regularization of the weight vector $\beta^T\beta \leq A^2$. We use $(S_i = \{x_{i,1}, \ldots, x_{i,m}\}, \{y_{i,j}\}_j)$ to indicate a sample point indexed by i. Note that for each sample i, we do not need to assume that $y_{i,j}$ are independently generated for different j. Using (14), the importance weighted regression in (11) becomes the following regularized empirical risk

minimization method:

$$
\begin{aligned}
f_{\hat{\beta}}(x,S) =& \hat{\beta}^T\psi(x,S),\\
\hat{\beta} =& \arg\min_{\beta\in\mathcal{H}}\left[\frac{1}{n}\sum_{i=1}^{n} L(\beta,S_i,\{y_{i,j}\}_j)+\lambda\beta^T\beta\right], \qquad (15)\\
L(\beta,S,\{y_j\}_j) =& \sum_{j=1}^{m} w(x_j,S)(\beta^T\psi(x_j,S)-y_j)^2 + u\sup_j w'(x_j,S)(\beta^T\psi(x_j,S)-\delta(x_j,S))_+^2.
\end{aligned}
$$

In this method, we replace the hard regularization in (14) with tuning parameter A by soft regularization with tuning parameter λ, which is computationally more convenient.

The following result is an expected generalization bound for the L_2-regularized empirical risk minimization method (15), which uses the stability analysis in [27]. The proof is in Appendix A.

Theorem 6 *Let $M=\sup_{x,S}\|\psi(x,S)\|_2$ and $W=\sup_S[\sum_{x_j\in S} w(x_j,S)+u\sup_{x_j\in S} w'(x_j,S)]$. Let $f_{\hat{\beta}}$ be the estimator defined in (15). Then we have*

$$
\mathbf{E}_{\{S_i,\{y_{i,j}\}_j\}_{i=1}^n} Q(f_{\hat{\beta}}) \le \left(1+\frac{WM^2}{\sqrt{2}\lambda n}\right)^2 \inf_{\beta\in\mathcal{H}}[Q(f_\beta)+\lambda\beta^T\beta].
$$

We have paid special attention to the properties of (15). 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 lowest 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 (13). Let $\hat{f}=f_{\hat{\beta}}$, we have*

$$
\mathbf{E}_{\{S_i,\{y_{i,j}\}_j\}_{i=1}^n}\mathbf{DCG}(r_{\hat{f}}) \le \mathbf{DCG}(r_B) + C(\gamma,u)\left[\left(1+\frac{WM^2}{\sqrt{2}\lambda n}\right)^2 \inf_{\beta\in\mathcal{H}}(Q(f_\beta)+\lambda\beta^T\beta)-Q(f_*)\right]^{1/2}.
$$

Proof From Theorem 5, we obtain

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

The second inequality is a consequence of Jensen's inequality. Now by applying Theorem 6, we obtain the desired bound. ■

The theorem implies that if $Q(f_*)=\inf_{\beta\in\mathcal{H}} Q(f_\beta)$, then as $n\to\infty$, we can let $\lambda\to 0$ and $\lambda n\to\infty$ 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.

9 Conclusion

Ranking problems have many important real-word applications. Although various formulations have been investigated in the literature, most theoretical results are concerned with global ranking using the pair-wise AUC criterion. Motivated by applications such as web-search, we introduced the subset ranking problem, and focus on the DCG criterion that measures the quality of the top-ranked items.

We derived bounds that relate the optimization of DCG scores to the minimization of convex regression errors. In our analysis, it is essential to weight samples differently according to their importance. These bounds are used to motivate modifications of least squares regression 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. 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 a machine learning based ranking system.

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. This paper shows that with appropriate features, importance sample weighted regression methods can produce the optimal scoring function in the large sample limit. It will be interesting to investigate other methods such as pair-wise based learning using similar analysis.

A Proof of Theorem 6

We shall introduce the following notation: let $Z_n = \{(S_i, \{y_{i,j}\}_j) : i = 1, \ldots, n\}$. Let $\hat{\beta}(Z_n)$ be the solution of (15) and $\hat{\beta}(Z_{n+1})$ be the solution using training data Z_{n+1}:

$$\hat{\beta}(Z_{n+1}) = \arg\min_{\beta \in \mathcal{H}} \left[\frac{1}{n} \sum_{i=1}^{n+1} L(\beta, S_i, \{y_{i,j}\}_j) + \lambda \beta^T \beta \right].$$

We have the following stability lemma in [27], which can be stated with our notation as:

Lemma 1 *The following inequality holds:*

$$\|\hat{\beta}(Z_n) - \hat{\beta}(Z_{n+1})\|_2 \leq \frac{1}{2\lambda n} \left\| \frac{\partial}{\partial \beta} L(\hat{\beta}(Z_{n+1}), S_{n+1}, \{y_{n+1,j}\}_j) \right\|_2,$$

where $\frac{\partial}{\partial \beta} L(\beta, S, \{y_j\}_j)$ *denotes a subgradient of* L *with respect to* β.

Note that from simple subgradient algebra in [21], we know that a subgradient of $\sup_j L_j(\beta)$ for a convex function $L_j(\beta)$ can be written as $\sum_j \alpha_j \partial L_j(\beta)/\partial\beta$, where $\sum_j \alpha_j \leq 1$ and $\alpha_j \geq 0$. Therefore we can find $\alpha_j \geq 0$ and $\sum_j \alpha_j \leq 1$ such that

$$\begin{aligned}
&\left\| \frac{\partial}{\partial\beta} L(\hat{\beta}(Z_{n+1}), S_{n+1}, \{y_{n+1,j}\}_j) \right\|_2^2 \\
=&2 \left\| \sum_{j=1}^m w(x_{n+1,j}, S_{n+1})(\beta^T \psi(x_{n+1,j}, S_{n+1}) - y_{n+1,j})\psi(x_{n+1,j}, S_{n+1}) \right. \\
&\left. + u \sum_{j=1}^m \alpha_j w'(x_{n+1,j}, S_{n+1})(\beta^T \psi(x_{n+1,j}, S_{n+1}) - \delta(x_{n+1,j}, S_{n+1}))_+ \psi(x_{n+1,j}, S_{n+1}) \right\|_2^2 \\
\leq& 2L(\hat{\beta}(Z_{n+1}), S_{n+1}, \{y_{n+1,j}\}_j) \left(\sum_{j=1}^m (w(x_{n+1,j}, S_{n+1}) + u\alpha_j w'(x_{n+1,j}, S_{n+1})) \|\psi(x_{n+1,j}, S_{n+1})\|_2^2 \right) \\
\leq& 2L(\hat{\beta}(Z_{n+1}), S_{n+1}, \{y_{n+1,j}\}_j) \left(\sum_{j=1}^m w(x_{n+1,j}, S_{n+1}) + u \sup_j w'(x_{n+1,j}, S_{n+1}) \right) M^2,
\end{aligned}$$

where the first inequality in the derivation is a direct application of Cauchy-Schwartz inequality. Now applying Lemma 1 with $\delta\beta = \hat{\beta}(Z_n) - \hat{\beta}(Z_{n+1})$, and use the inequality $(a+b)^2 \leq (1+s)a^2 + (1+s^{-1})b^2$ (where $s > 0$ is an arbitrary real number), we have

$$\begin{aligned}
&L(\hat{\beta}(Z_n), S_{n+1}, \{y_{n+1,j}\}_j) \\
=&L(\hat{\beta}(Z_{n+1}) + \delta\beta, S_{n+1}, \{y_{n+1,j}\}_j) \\
\leq&L(\hat{\beta}(Z_{n+1}) + \delta\beta, S_{n+1}, \{y_{n+1,j}\}_j) \\
\leq&(1+s)L(\hat{\beta}(Z_{n+1}), S_{n+1}, \{y_{n+1,j}\}_j) + (1+s^{-1})W(S_{n+1})\|\delta\beta\|_2^2 M^2 \\
\leq&(1+s)L(\hat{\beta}(Z_{n+1}), S_{n+1}, \{y_{n+1,j}\}_j) + (1+s^{-1})W(S_{n+1}) \frac{M^2}{4\lambda^2 n^2} \left\| \frac{\partial}{\partial\beta} L(\hat{\beta}(Z_{n+1}), S_{n+1}, \{y_{n+1,j}\}_j) \right\|_2^2 \\
\leq&L(\hat{\beta}(Z_{n+1}), S_{n+1}, \{y_{n+1,j}\}_j) \left[(1+s) + (1+s^{-1})W(S_{n+1})^2 \frac{M^4}{2\lambda^2 n^2} \right],
\end{aligned}$$

where we define

$$W(S) = \sum_{x_j \in S} w(x_j, S) + u \sup_{x_j \in S} w'(x_j, S).$$

By optimizing over s, we obtain

$$L(\hat{\beta}(Z_n), S_{n+1}, \{y_{n+1,j}\}_j) \leq \left(1 + \frac{W(S_{n+1})M^2}{\sqrt{2}\lambda n} \right)^2 L(\hat{\beta}(Z_{n+1}), S_{n+1}, \{y_{n+1,j}\}_j).$$

Now denote by $Z_{n+1}^{(i)}$ the training data obtained from Z_{n+1} by removing the i-th datum $(S_i, \{y_{i,j}\}_j)$, and let $\hat{\beta}(Z_{n+1}^{(i)})$ be the solution of (15) with Z_n replaced by $Z_{n+1}^{(i)}$, then we have:

$$\begin{aligned}\sum_{i=1}^{n+1} L(\hat{\beta}(Z_{n+1}^{(i)}), S_i, \{y_{i,j}\}_j) &\leq \left(1 + \frac{WM^2}{\sqrt{2}\lambda n}\right)^2 \sum_{i=1}^{n+1} L(\hat{\beta}(Z_{n+1}), S_i, \{y_{i,j}\}_j) \\ &\leq \left(1 + \frac{WM^2}{\sqrt{2}\lambda n}\right)^2 \inf_{\beta \in \mathcal{H}} \left[\sum_{i=1}^{n+1} L(\beta, S_i, \{y_{i,j}\}_j) + \lambda n \beta^T \beta\right].\end{aligned}$$

To obtain the desired bound, we simply take expectation with respect to Z_{n+1} on both sides.

References

[1] Shivani Agarwal, Thore Graepel, Ralf Herbrich, Sariel Har-Peled, and Dan Roth. Generalization bounds for the area under the ROC curve. *Journal of Machine Learning Research*, 6:393–425, 2005.

[2] Shivani Agarwal and Dan Roth. Learnability of bipartite ranking functions. In *Proceedings of the 18th Annual Conference on Learning Theory*, 2005.

[3] Noga Alon. Ranking tournaments. *SIAM J. Discrete Math.*, page 137.

[4] P.L. Bartlett, M.I. Jordan, and J.D. McAuliffe. Convexity, classification, and risk bounds. Technical Report 638, Statistics Department, University of California, Berkeley, 2003. to appear in JASA.

[5] Gilles Blanchard, Gabor Lugosi, and Nicolas Vayatis. On the rate of convergence of regularized boosting classifiers. *Journal of Machine Learning Research*, 4:861–894, 2003.

[6] Peter Bühlmann. Boosting for high-dimensional linear models. *Annals of Statistics*, 34:559–583, 2006.

[7] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *ICML'05*, 2005.

[8] Andrea Caponnetto. A note on the role of squared loss in regression. Technical report, CBCL, Massachusetts Institute of Technology, 2005. http://cbcl.mit.edu/projects/cbcl/publications/ps/caponnetto-squareloss-6-05.pdf.

[9] S. Clemencon, G. Lugosi, and N. Vayatis. Ranking and scoring using empirical risk minimization. In *COLT'05*, 2005.

[10] David Cossock. Method and apparatus for machine learning a document relevance function. US patent application, 20040215606, 2003.

[11] P. Crescenzi and V. Kann. A compendium of np optimization problems. Technical Report SI/RR-95/02, Dipartimento di Scienze dell'Informazione, Universit di Roma "La Sapienza", 1995. updated at http://www.nada.kth.se/∼viggo/wwwcompendium/wwwcompendium.html.

[12] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation revisited. In *Proceedings of WWW10*, 2001.

[13] Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *JMLR*, 4:933–969, 2003.

[14] J. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29:1189–1232, 2001.

[15] J.A. Hanley and B.J. McNeil. The meaning and use of the Area under a Receiver Operating Characetristic (ROC) curve. *Radiology*, pages 29–36, 1982.

[16] R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. In B. Schölkopf A. Smola, P. Bartlett and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 115–132. MIT Press, 2000.

[17] Kalervo Jarvelin and Jaana Kekalainen. IR evaluation methods for retrieving highly relevant documents. In *SIGIR'00*, pages 41–48, 2000.

[18] G. Lugosi and N. Vayatis. On the Bayes-risk consistency of regularized boosting methods. *The Annals of Statistics*, 32:30–55, 2004. with discussion.

[19] Shie Mannor, Ron Meir, and Tong Zhang. Greedy algorithms for classification - consistency, convergence rates, and adaptivity. *Journal of Machine Learning Research*, 4:713–741, 2003.

[20] P. McCullagh and J. A. Nelder. *Generalized linear models*. Chapman & Hall, London, 1989.

[21] R. Tyrrell Rockafellar. *Convex analysis*. Princeton University Press, Princeton, NJ, 1970.

[22] Saharon Rosset. Model selection via the AUC. In *ICML'04*, 2004.

[23] Cynthia Rudin. Ranking with a p-norm push. In *COLT 06*, 2006.

[24] Ingo Steinwart. Support vector machines are universally consistent. *J. Complexity*, 18:768–791, 2002.

[25] Ambuj Tewari and Peter Bartlett. On the consistency of multiclass classification methods. In *COLT*, 2005.

[26] Z. Zha, Z. Zheng, H. Fu, and G. Sun. Incorporating query difference for learning retrieval functions in information retrieval. In *SIGIR*, 2006.

[27] Tong Zhang. Leave-one-out bounds for kernel methods. *Neural Computation*, 15:1397–1437, 2003.

[28] Tong Zhang. Statistical analysis of some multi-category large margin classification methods. *Journal of Machine Learning Research*, 5:1225–1251, 2004.

[29] Tong Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *The Annals of Statistics*, 32:56–85, 2004. with discussion.

[30] Tong Zhang and Bin Yu. Boosting with early stopping: Convergence and consistency. *The Annals of Statistics*, 33:1538–1579, 2005.