A new class of ranking functions for DCG-like evaluation metrics using conditional probability models

October 29, 2010

Eunho Yang
Dept. of Computer Science
University of Texas at Austin
eunho@cs.utexas.edu

Pradeep Ravikumar
Dept. of Computer Science
University of Texas at Austin
pradeepr@cs.utexas.edu

Matthew Lease
School of Information
University of Texas at Austin
ml@ischool.utexas.edu

ABSTRACT

In the context of *learning to rank* for information retrieval [15], we study a general class of “DCG-like” ranking loss functions which include DCG [13] and approximate ERR [6] as specific cases. We then study the Bayes optimal ranking function for this class, which is a function of the conditional distribution of graded document relevance levels. Our main contribution is a novel class of ranking functions building upon the Bayes optimality result. Specifically, our ranking function class uses the conditional expectation of a function of the document features with respect to an ostensible multiclass logistic regression model. We empirically evaluate our proposals with respect to NDCG on seven standard learning to rank datasets [16] and three different surrogate evaluation metrics. Results show our proposed ranking functions improve accuracy across datasets and surrogate evaluation metrics, achieving 10.6% improvement on average and as high as 25% in the best case.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval Model; H.4.m [**Information Systems**]: Miscellaneous—*Machine learning*

General Terms

Algorithms, Experimentation, Theory

Keywords

ranking function, machine learning, optimization, listwise-preference

1. INTRODUCTION

Ranking. Ranking a set of instances by their relative relevance arises in many contemporary problems, such as collaborative filtering, text mining and information retrieval (IR). We are interested in a particular formulation of this problem, natural in IR, where the ranking is at the resolution of a data item such as a query. Each query has a list of documents, and the task is to rank these documents in the order of relevance to the query. While classical methods for IR were traditionally tuned through exhaustive experimentation [21], there has been a surge of recent work in estimating statistical models for IR. Indeed *Learning to Rank* [15] has emerged as a key problem at the intersection of machine learning (ML) and IR. The recent advent of standard datasets such as LETOR (LEarning TO Rank) [16] have particularly enabled the testing and development of machine learning driven IR methods. In such datasets, in the training set, the documents for each query are typically represented as feature vectors derived from the query-document pairs, and are annotated with relevance scores indicating the relative preference of the document in the list for that query. Given any new query, the goal is to rank its documents in an order that best respects their relevance scores according to some ranking evaluation measure. User studies have motivated specific ranking evaluation measures such as Mean Average Precision (MAP) [3], Expected Reciprocal Rank [6] and the popular Normalized Discounted Cumulative Gain (NDCG) [13].

DCG-like measures. In this paper, we study *DCG-like* evaluation measures, which evaluates the ranking of the entire list of documents by penalizing errors in higher ranked documents more strongly. While easy to *evaluate*, this is nonetheless a difficult measure to directly use for *training* a ranking model. A broad line of work has thus focused on breaking the ranking problem down into pointwise and pairwise problems [5]. In the *pointwise* approach, the ranking problem is viewed as a regression or classification problem of predicting the specific relevance score for any document [28]. The hope is that by minimizing some measure of the difference between each document’s true relevance level and the model’s estimate for it, the *listwise* NDCG measure of the ranking of the entire list of the documents would in turn be minimized. Examples include mean squared error, and ordinal regression. In the *pairwise* approach on the other hand, the ranking problem is reduced to the binary classification task of predicting the more relevant document amongst pairs of documents. Note that the training data for such an approach would only need pairwise relative preferences, which is easier to obtain than manually annotated listwise relevance judgements, for instance using query log click-through data [14]. But on the other hand, such training instances of document pairs and their pairwise relative relevances are typically not iid which impairs test performance.

Listwise approaches. The main caveat with such approaches is that they are ill-suited to evaluation measures as NDCG, ERR etc. which are typically *listwise*: that is their evaluation is a function of the entire list of ranked documents. Cao et al. [5], Xia et al. [27] in particular note that methods based on *listwise* loss functions outperform their pointwise and pairwise counterparts. One class of such *listwise* approaches attempt to optimize the NDCG (and such) evaluation measures directly using heuristics [4, 28, 24, 26, 25]. An-other class of *listwise* approaches optimize surrogate listwise loss functions instead [18, 5, 27]. The IR evaluation measures evaluate the goodness of fit of a ranking of a set of documents to a vector of their relevance scores. Instead of a ranking of documents, one could consider as input to the IR evaluation measure, a vector of real-valued scores so that its sorted order corresponds to the desired ranking (Note that the relevance scores themselves are such an idealized real-valued score vector). Surrogate ranking loss functions such as Cross entropy, Cosine, etc. too take as input a real-valued *score vector*, and a vector of relevance judgements, but still provide a smooth (i.e. differentiable) measure the goodness of fit of the score vector to the relevance judgements.

Ranking functions. This leads to the second ingredient in such surrogate loss based learning to rank methods: a ranking or score function, that takes as input a document-query feature vector, and outputs a real-valued score. Given a surrogate loss, and a parameterized ranking function, one could estimate these parameters by minimizing the surrogate loss over the training set.

The main contribution of our paper is a novel set of ranking functions that are motivated by the Bayes optimal ranking function. The Bayes optimal ranking function for a given surrogate loss is that ranking function that would achieve the minimum loss if there were infinitely many observations. We use a particular formulation of a class of *DCG-like* IR evaluation measures that are based on the popular discounted cumulative gain (DCG) metric [13] and also approximates the more recent expected reciprocal rank (ERR) metric [6]. Recent results have studied *Bayes optimality* for pointwise and pairwise ranking loss functions ([9] and [8] respectively), and for zero-one listwise loss Cao et al. [5]. The Bayes optimal ranking function for the *DCG-like* evaluation function class follows naturally from these results, and can be expressed quite simply as a function of the conditional distribution of graded document relevance levels (which corroborates with Robertson [20]'s principle). Such a result can be used in two ways: one could first estimate the conditional distribution of the relevance levels given the documents, and then estimate the empirical estimate of the Bayes optimal ranking function; as we discuss further §4.2. The other way of using the Bayes optimal ranking function, which is the main focus here, is to use these to develop a novel class of ranking functions.

To further characterize our perceived contribution, we believe the value of our proposal lies not in providing a new, single method that performs well, but rather that we describe a novel *class* of ranking functions that could be used in conjunction with many different surrogate loss functions, including new loss functions yet to be proposed. As such, what we propose is a *meta-method* with applicability beyond the three loss functions evaluated here.

As we show in our results section §6, our novel ranking functions consistently perform as well or better than standard linear ranking function across the different surrogate loss functions considered. In particular, we achieve 10.6% improvement on average across datasets, and as much as 25.2% improvement (on HP2003). In addition to achieving these gains, we further show (informally) these gains are achieved with low-risk. While reporting only averages can mask a method's high variance (i.e. many or large losses on some datasets being offset by strong gains on a few), we see accuracy decreased in only 3 of the 63 cases considered (4.8%), while statistically significant improvement is achieved far more often (in 23/63 cases, or 36.5%). Thus our proposal represents both a safe and more effective alternative to the baseline approach.

We also refer the interested reader to additional follow-on work by the authors [19, 29].

2. RANKING: SETUP AND NOTATION

Let m be the number of documents for each query. Let $\bar{\mathcal{X}}$ be the space of the feature vectors in which the documents are represented (typically derived from the query-document pairs). Let $\bar{\mathcal{R}} \subseteq \mathbb{R}$ be the space of the relevance scores each document receives. Thus for any query, we have a list $\mathbf{X} = (X_1, \ldots, X_m) \in \mathcal{X} := \bar{\mathcal{X}}^m$ of document feature vectors, and a corresponding list $\mathbf{R} = (R_1, \ldots, R_m) \in \mathcal{R} := \bar{\mathcal{R}}^m$ of document relevance scores. The dataset consists of n $(\mathbf{X}_i, \mathbf{R}_i)$ pairs which we assume to be drawn iid from some distribution over $\mathcal{X} \times \mathcal{R}$. We assume that $\bar{\mathcal{R}} := \{1, \ldots, K\}$, so that each document can receive a score in any of K relevance levels.

A permutation π is a bijection from $[m]$ to $[m]$. We interpret $\pi(i)$ as "the position of document i". Thus, according to π, the documents $\mathbf{x} = (x_1, \ldots, x_m)$ should be ordered as

$$(x_{\pi^{-1}(1)}, \ldots, x_{\pi^{-1}(i)}, \ldots, x_{\pi^{-1}(m)})$$

Let $\mathcal{S}_m$ be the set of all such degree m permutations. A listwise ranking evaluation metric measures the goodness of fit of any candidate ranking to the corresponding relevance scores, so that it is a map $\ell : \mathcal{S}_m \times \mathcal{R} \to \mathbb{R}$.

3. DCG-LIKE EVALUATION METRICS

We are interested in the specific class "DCG-like" evaluation metrics:

DEFINITION 1 (DCG-LIKE EVALUATION METRICS).

$$\ell_{\mathrm{NDCG}}(\pi, \mathbf{r}) = -\alpha \sum_{j=1}^{m} \frac{G(r_j)}{F(\pi(j))}, \tag{1}$$

where $G : \mathcal{R} \to \mathbb{R}_+$ is a monotonically increasing function of the relevance judgments, and $F : \mathbb{R} \to \mathbb{R}_+$ is also a monotonically increasing function, and $\alpha > 0$ is some constant.

Discounted Cumulative Gain: $G(r) = 2^r - 1$, and $D(l) = \log(1 + l)$. We note that a specific normalized variant, NDCG [13], has a normalization factor α that is not constant and depends on the relevance scores $\mathbf{r}$, so that it does not fall exactly into this family as stated. But we note that the development could be extended to such relevance dependent normalizations as well.

Expected Reciprocal Rank: ERR is a listwise evaluation metric proposed recently by Chapelle et al. [6] and is motivated by the cascade user browsing model. A user is modeled as progressively browsing through the documents in the ranked order, and proceeding to lower ranked documents only if the previous ranked documents were not satisfactorily relevant. The ERR evaluation metric then measures the goodness of fit of the ranking as the expected user effort:

$$\mathrm{ERR}(\pi, \mathbf{r}) := \sum_{j=1}^{m} \frac{1}{j} G(r_{\pi^{-1}(j)}) \prod_{l=1}^{j-1} (1 - G(r_{\pi^{-1}(l)})), \tag{2}$$

where $G(r) = \frac{2^r - 1}{2^K}$ is a monotonically increasing function depending on relevance judgments as before. In this case, it also captures the probability that the user would be satisfied with a document with this relevance level score. We can rewrite this as

$$\mathrm{ERR}(\pi, \mathbf{r}) := \sum_{j=1}^{m} \frac{1}{\pi(j)} G(r_j) \prod_{l:\pi(l)<\pi(j)} (1 - G(r_l)). \quad (3)$$

Now consider the approximation,

$$\frac{1}{\pi_j} \prod_{l:\pi(l)<\pi_(j)} (1 - G(r_l)) \approx \left(\frac{1}{\pi_j}\right)^{\beta}, \quad (4)$$

where the parameter $\beta > 1$ captures the decay of the user satisfaction with the position. Substituting this approximation in (3), we obtain the approximate metric,

$$\widetilde{\mathrm{ERR}}(\pi, \mathbf{r}) := \sum_{j=1}^{m} \left(\frac{1}{\pi_j}\right)^{\beta} G(r_j), \quad (5)$$

which has the *DCG-like* form of (1), with $G(r) = \frac{2^r - 1}{2^K}$, and $F(j) = j^{\beta}$.

4. RANKING FUNCTIONS

A ranking function maps the query-document features to a real-valued score; so that it is a map $h : \mathcal{X} \to \mathbb{R}$. Typically, this ranking function belongs to a parameterized family $\mathcal{H} = \{h_w\}_{w \in \mathcal{W}}$, where any member of the family is specified by a set of weights $\mathbf{w}$. The *learning to rank* task then is to *learn* a ranking function $h_{\hat{w}}$ that best optimizes the evaluation metric ℓ given the training set. A typical family of ranking functions is the set of linear functions, so that $h_w(\mathbf{x}) = w^T \mathbf{x}$.

In this paper, we propose a novel class of ranking functions that are motivated by *Fisher-optimal* ranking functions which are the optimal ranking function in the infinite sample limit for any given evaluation metric ℓ.

4.1 Fisher Optimal ranking functions

Note that the first argument of evaluation metrics such as the DCG-like (1) is a permutation. With some abuse of notation we can define

$$\ell(\mathbf{s}, \mathbf{r}) = \ell(\pi_{\mathbf{s}}, \mathbf{r})$$

where $\pi_{\mathbf{s}}$ is a permutation such that $\pi_{\mathbf{s}}(j)$ is the position of s_j when elements of $\mathbf{s}$ are sorted in decreasing order of their values. Note that now the first argument is a score vector. With this notation, given any particular evaluation metric ℓ, its Bayes or Fisher optimal ranking function is defined as,

$$\hat{h} := \arg\min_{h} \mathbb{E}\left[l(h(\mathbf{X}), \mathbf{R})\right], \quad (6)$$

where $\mathbb{E}[\cdot]$ denotes expectation with respect to the true distribution over $\mathcal{X} \times \mathcal{R}$ from which the samples are drawn. The Fisher optimal ranking function is thus the *best possible* ranking function given the evaluation metric ℓ.

Such questions of *Bayes optimality* in the pointwise learning model was studied by Cossock and Zhang [9], and for the pair-wise learning model was studied by Clemencon et al. [8]. The following proposition on the Bayes or Fisher optimal ranking functions for the listwise case of DCG-like evaluation metrics (1) follows naturally from these results. We will need a notion of when the sorted order of one vector $\mathbf{s}$ is compatible with the sorted order of a given vector $\mathbf{r}$. This *assymetric* binary relation between $\mathbf{s}$ and $\mathbf{r}$ is denoted by

$$\mathbf{s} \overset{s}{\to} \mathbf{r}\ ,$$

and it holds precisely when, for all $i, j \in [m]$, $r_i > r_j$ implies $s_i > s_j$.

PROPOSITION 1. *Assume the conditional distribution of the relevance level of a document given the document features is independent of other documents in the corpus. Then the Bayes optimal ranking function for evaluation metrics of the form in* (1) *is given by*

$$h^*(\mathbf{x}) \overset{s}{\to} \{\mathbb{E}(G(r_1)|x_1), \ldots, \mathbb{E}(G(r_m)|x_m)\}. \quad (7)$$

PROOF. The expected loss of any ranking function h for the NDCG-like evaluation function (1), conditioned on $(X_1, \ldots, X_m)$, is given by

$$\mathbb{E}(l_N(h(\mathbf{x}), \mathbf{r})|\mathbf{x}) = -\alpha \sum_{j=1}^{m} \frac{\mathbb{E}(G(r_j)|x_j)}{F(h_j)}, \quad (8)$$

where h_j is shorthand for the ranking of the j-th document $h(\mathbf{x})(j)$, and where we have used the conditional independence: $P(\mathbf{r}|\mathbf{x}) = \prod_{j=1}^{m} P(r_j|x_j)$. Since F is a monotonic function, it follows that the loss is minimized as a function of $h(\mathbf{x})$ when $h(\mathbf{x})$ is set as stated in the theorem.

□

4.2 Plugin estimates for Fisher Optimal ranking functions

Proposition 1 suggests a simple strategy for learning to rank. As it shows, the Fisher optimal ranking function uses the conditional distribution $P(r_j|x_j)$ of the relevance level of a document to compute the expectation $\mathbb{E}(G(r_j)|x_j)$ for all documents $(x_1, \ldots, x_m)$ corresponding to the query. We can thus *directly* estimate this Fisher optimal ranking function by *estimating this conditional distribution* of the relevance level of a document given the document, and using it to compute the list of expectations, and sort these. This is summarized in Algorithm 1.

While it performed well in our experiments, it was still below the state of the art learning to rank methods such as Listnet Cao et al. [5]. The reason for this is that while Proposition 1 specifies the optimal ranking function in the infinite sample limit, the plugin estimate in Algorithm 4.2 need not be the optimal ranking function given a finite number of samples. This is similar to the case in classification, where thresholding conditional expectation of the response given the input is the Fisher optimal classification function, but directly estimating this conditional expectation is outperformed by other state of the art methods.

Thus, we use the intuition from Proposition 1 by proposing a *new family* of ranking functions which could then be used in any state of the art learning to rank methods.

5. NEW CLASS OF RANKING FUNCTIONS

This section contains the main thrust of our paper: a class of ranking functions motivated by proposition 1. As discussed in Section 4, given an evaluation metric ℓ such as (1), and a family of

Algorithm 1 Multiclass NDCG; Weights Maximizing Conditional Likelihood

```
Input: Samples $\{((x_1^{(i)}, \ldots, x_{m_i}^{(i)}), (r_1^{(i)}, \ldots, r_{m_i}^{(i)}))\}_{i=1}^n$.

*Training..*

Estimate the conditional distribution $\hat{P}(r|x)$ from samples, or
alternatively, a multiclass classifier $\hat{f} : \bar{\mathcal{X}} \to \bar{\mathcal{R}}$ from which con-
ditional probabilities can be estimated via calibrations as in [17].

*Testing..*

For any test query $q$, with documents $\{x_1, \ldots, x_m\}$,
compute the list of expectations, $Z = \{\hat{\mathbb{E}}_w(G(r_1)|x_1), \ldots, \hat{\mathbb{E}}_w(G(r_m)|x_m)\}$
with respect to the learned conditional distribution.

Output $\hat{h}(x_1, \ldots, x_m) = \text{sort}(Z)$.
```

ranking functions $\{h_w\}$, the *learning to rank* task consists of *learning* a ranking function $h_{\hat{w}}$ that best optimizes the evaluation metric ℓ given the training set:

$$\hat{w} \in \arg\min_w \hat{\mathbb{E}} \left[\ell(\text{sort}(h_w(\mathbf{x})), \mathbf{r}) \right], \tag{9}$$

where we use $\hat{\mathbb{E}}[\cdot]$ to denote the empirical expectation: $\hat{\mathbb{E}}(g(\mathbf{x}, \mathbf{r})) = \frac{1}{n} \sum_{i=1}^n g(x_i, r_i)$, and where we sort the real-valued output of the ranking function to obtain the ranked permutation of the documents.

As can be seen, (9) is a difficult optimization problem to solve: it is not even differentiable, and depends in a complicated fashion (after a sort) on the ranking function and hence on the weights $\mathbf{w}$.

Surrogate Evaluation Metrics. This has motivated the development of *surrogate* evaluation metrics $\phi : \mathbb{R}_m \times \mathcal{R} \to \mathbb{R}$, so that it directly takes as input a real-valued score vector, instead of a permutation as in the DCG-like evaluation metrics (1). Some examples include:

- Cosine Loss, by Qin et al. [18]:

  $$\phi(h(\mathbf{x}), \mathbf{r}) = \frac{1}{2}\left(1 - \frac{\psi(\mathbf{r})^T h(\mathbf{x})}{\|\psi(\mathbf{r})\| \|h(\mathbf{x})\|}\right), \tag{10}$$

  where $\psi : \bar{\mathcal{R}}^m \to \mathbb{R}^m$ is any mapping that respects the relevance levels in the vector $\mathbf{r}$, so that $\psi(\mathbf{r})_j > \psi(\mathbf{r})_k$ if $r_j > r_k$.

- The Cross-Entropy loss, also called Listnet, by Cao et al. [5]:

  $$\phi(h(\mathbf{x}), \mathbf{r}) = D(P(\pi|\psi(\mathbf{r})) \| P(\pi|h(\mathbf{x}))), \tag{11}$$

  where $\psi : \bar{\mathcal{R}}^m \to \mathbb{R}^m$ is a mapping from relevance levels to reals as before, $P(\pi|h(\mathbf{x})) = \prod_{i=1}^m \frac{\exp(h_{\pi^{-1}(i)})}{\sum_{k=i}^m \exp(h_{\pi^{-1}(k)})}$

Such a surrogate evaluation metric would then be used with a parameterized family of ranking function $h_w : \bar{\mathcal{X}} \to \mathbb{R}$ by optimizing the weights w with respect to the surrogate evaluation metric:

$$\hat{w} \in \arg\min_w \hat{\mathbb{E}} \left[\phi(h_w(\mathbf{x}), \mathbf{r}) \right].$$

Again, a typical family of ranking functions used is the set of linear functions, so that $h_w(\mathbf{x}) = w^T \mathbf{x}$.

5.1 Some Intuition

Consider the Fisher-optimal ranking function $h(\mathbf{x})$ given a surrogate metric ϕ:

$$h_\phi := \arg\min_h \mathbb{E} \left[\phi(h(\mathbf{x}), \mathbf{r}) \right].$$

Let P be the distribution over $\mathcal{X} \times \mathcal{R}$ and suppose $h_\phi = \mathcal{F}(P)$ for some functional $\mathcal{F}$ of the distribution P. In other words, h_ϕ is a parameter of the distribution P, and if this were an identifiable parameter (i.e. if $\mathcal{F}$ were invertible) then h_ϕ would in turn specify the distribution P. We state this loosely as $P = \mathcal{F}^{-1}(h_\phi)$.

For instance, consider boosting in the context of binary classification with label $Y \in \{-1, 1\}$ and features X. Friedman et al. [11] showed that with a surrogate loss $\phi(f(x), y) = \exp(-yf(x))$, the Fisher optimal function f_ϕ is given as $f_\phi(x) = \frac{1}{2} \log \frac{P(y=1|x)}{P(y=-1|x)}$. This in turn entails that $P(y = 1|x) = \frac{\exp(f_\phi(x))}{\exp(f_\phi(x)) + \exp(-f_\phi(x))}$.

Now suppose we use a ranking function $h_w(\mathbf{x})$ and find the optimal weights w_ϕ:

$$w_\phi := \arg\min_w \mathbb{E} \left[\phi(h_w(\mathbf{x}), \mathbf{r}) \right].$$

Assuming that this attains the Bayes optimal ϕ risk, $h_{w_\phi} = h_\phi = \mathcal{F}(P)$, so that this is equivalent to a distributional assumption on P as $\mathcal{F}^{-1}(h_{w_\phi})$. For instance in the example of classification and boosting discussed above, as Friedman et al. [11] showed, Fisher optimality of $f_w(\mathbf{x}) = w^T \mathbf{x}$ for surrogate loss function $\phi(f(x), y) = \exp(-yf(x))$ entails the distributional assumption of a logistic regression model $P(y = 1|x) = \exp(w^T x)/(1 + \exp(w^T x))$.

Armed with this intuition, we can see that it is vital to select a right ranking function that would entail a reasonable distributional assumption on P.

But can we obtain a ranking function that would entail a *specific* distributional assumption on P? Specifically, we are interested in a multiclass logistic model for the conditional distribution $P(\mathbf{r}|\mathbf{x})$ of the relevance level of a document given the document:

$$P_{\mathbf{w}}(r = j|x) = \frac{\exp(w_j^T x)}{\sum_l \exp(w_l^T x)}, \quad j \in \{1, \ldots, K\}, \tag{12}$$

for K parameter vectors $\{w_k\}_{k=1}^K$.

It can be easily verified that the ranking function $h_{\mathbf{w}}(\mathbf{x}) = \mathcal{F}(P_{\mathbf{w}})$ where $P_{\mathbf{w}}$ is the logistic regression model (12), would correspond to the logistic distributional assumption (12) on P.

5.2 Ranking functions motivated by Fisher optimality

Based on the intuition from the previous section, we then propose the following class of ranking functions:

$$h_w(\mathbf{x}) = \mathbb{E}_w(G(r)|\mathbf{x}),$$

where the expectation $\mathbb{E}_w(\cdot)$ is with respect to a logistic regression model for the conditional distribution of the relevance labels given the document features. Specifically,

$$h_{\mathbf{w}}(x) = \sum_{j=1}^{K} P_{\mathbf{w}}(r = j|x)\ G(j),$$

where $P_{\mathbf{w}}$ is the logistic probability as specified in (12).

Note that this is precisely $\mathcal{F}(P_{\mathbf{w}})$ for the functional $\mathcal{F}$ characterizing the Fisher optimal ranking function for the DCG-like evaluation metric, as derived in Proposition 1.

Thus, optimizing the weights for this ranking function given a surrogate evaluation metric ϕ would yield:

$$\min_{w} \hat{\mathbb{E}} \left[\phi(\ \{\mathbb{E}_w(G(r)|x_j)\}_{j=1}^{m}, \ \mathbf{r}) \right]. \tag{13}$$

We thus arrive at our main algorithm:

Algorithm 2 Surrogate loss with our Fisher-optimality motivated class of ranking functions

```
Input: Samples $\{((x_1^{(i)}, \ldots, x_{m_i}^{(i)}), (r_1^{(i)}, \ldots, r_{m_i}^{(i)}))\}_{i=1}^{n}$.

Training..

Estimate the weights $\hat{\mathbf{w}}$ using (13).

Testing..

For any test query $q$, with documents $\{x_1, \ldots, x_m\}$, use the ranking function with the learnt weights to obtain the scores $Z = \{\mathbb{E}_{\hat{\mathbf{w}}}(G(Y_1)|x_1), \ldots, \mathbb{E}_{\hat{\mathbf{w}}}(G(Y_m)|x_m)\}$.

Output $\hat{h}(x_1, \ldots, x_m) = \text{sort}(Z)$.
```

6. EVALUATION

This section reports empirical effectiveness of our ranking functions using three different listwise surrogate evaluation metrics:

- Cosine loss, given by Equation (10)
- Cross entropy or Listnet loss, given by Equation (11)
- Squared loss (see below)

With regard to squared loss, we consider a listwise variant given by

$$\phi(h(\mathbf{x}), \mathbf{r}) = \sum_{j=1}^{m} (r_j - h_j)^2. \tag{14}$$

For each surrogate loss metric, we take the the linear ranking function as our baseline and evaluate accuracy of our "Bayes-Optimality" motivated logistic-regression ranking function in comparison.

6.1 Experimental Setup

We evaluate the empirical effectiveness of our ranking model on 7 standard benchmark collections from LETOR 3.0 [16]. These benchmarks target four tasks over two collections: 2003-2004 TREC Web track [10] tasks of (1) Homepage finding, (2) Named page finding, and (3) Topic distillation on the .GOV collection (1.25 million page 2002 crawl of the .gov domain), as well as (4) biomedical search on the older OHSUMED collection (350,000 documents, titles and abstracts without full-text) [12]. Search accuracy of results is measured by both NDCG [13] at ranks 1, 5 and 10 to measure graded relevance at different user models cut-off ranks. LETOR includes a standard 5-fold partition of each dataset (3 training, 1 validation, and 1 test); our reported results reflect an average over the 5 test folds.

We note that the original papers proposing Listwise loss metrics employed different loss functions and techniques for optimizing those loss functions. For example, ListNet [5] used gradient descent to minimize cross-entropy loss, with the number of iterations and learning rate as parameters tuned on the validation set. On the other hand, RankCosine [18] minimized the cosine loss with the additive model. To evaluate across methods in a fair manner, we adopt the same optimization technique for all loss functions: gradient descent. In particular, we adopt the MATLAB implementation of gradient descent without any parameter tuning.

However, the conditional probability does introduce an additional parameter ϵ needing to be tuned. The role of ϵ will be explained in the following subsection. We tested values of ϵ from the set $\{0.6, 0.7, 0.8, 0.9, 1\}$ on the validation fold based on $NDCG@10$ accuracy to automatically tune the parameter. Note that the cross-entropy loss itself has a parameter α for the linear mapping in

$$P(\pi^{-1}(1) = j|\mathbf{r}) = \frac{exp(\alpha \cdot r_j)}{\sum_{k=1}^{m} exp(\alpha \cdot r_{\pi_k^{-1}})}$$

For alpha, we tested values from the set $\{0.1, 0.3, 0.5, 0.7, 1, 2, 5, 10\}$ on the validation fold similarly to select alpha automatically.

Statistical significance of results is measured using a randomization test as implemented by `ireval` in the Indri search engine [23]. Note that this significance test has been shown [22] to provide more reliable estimates of significant differences than the Wilcoxon signed rank test often used in earlier IR studies. Results achieving $p < 0.05$ are reported as significant.

6.2 Implementation issues

Several numerical issues in computation arose during our work that merit brief discussion. We describe them here, along with our solutions to them.

Weights w tending to infinity. Suppose the relevance score for a query-document feature vector x_i is $r_i = 2$ with probability one, and that the set of relevance levels $\bar{\mathcal{R}}$ is {1,2,3}. So long as the values $w_j^T x_i$ constituting the logistic probabilities are bounded, the logistic probabilities for relevance levels 1 and 3 would in turn be bounded away from zero. Thus since the surrogate metric would push these probabilities to zero, it would attempt to push the weights to be very large. As a fix, we used the intuition above to control the conditional probability directly: when performing gradient descent, if the logistic probability $P_w(r = j|x)$ was greater than ϵ, we set it to 1 and set the logistic probabilities for other relevance levels to zero; $P_w(r = k|x) = 0; k \neq j$.

Non-convexity. Since we use logistic functions of the weights in our ranking function, the net objective function for optimizing the weights is typically non-convex. The usual approach in such settings is to then perform multiple random restarts and select the local

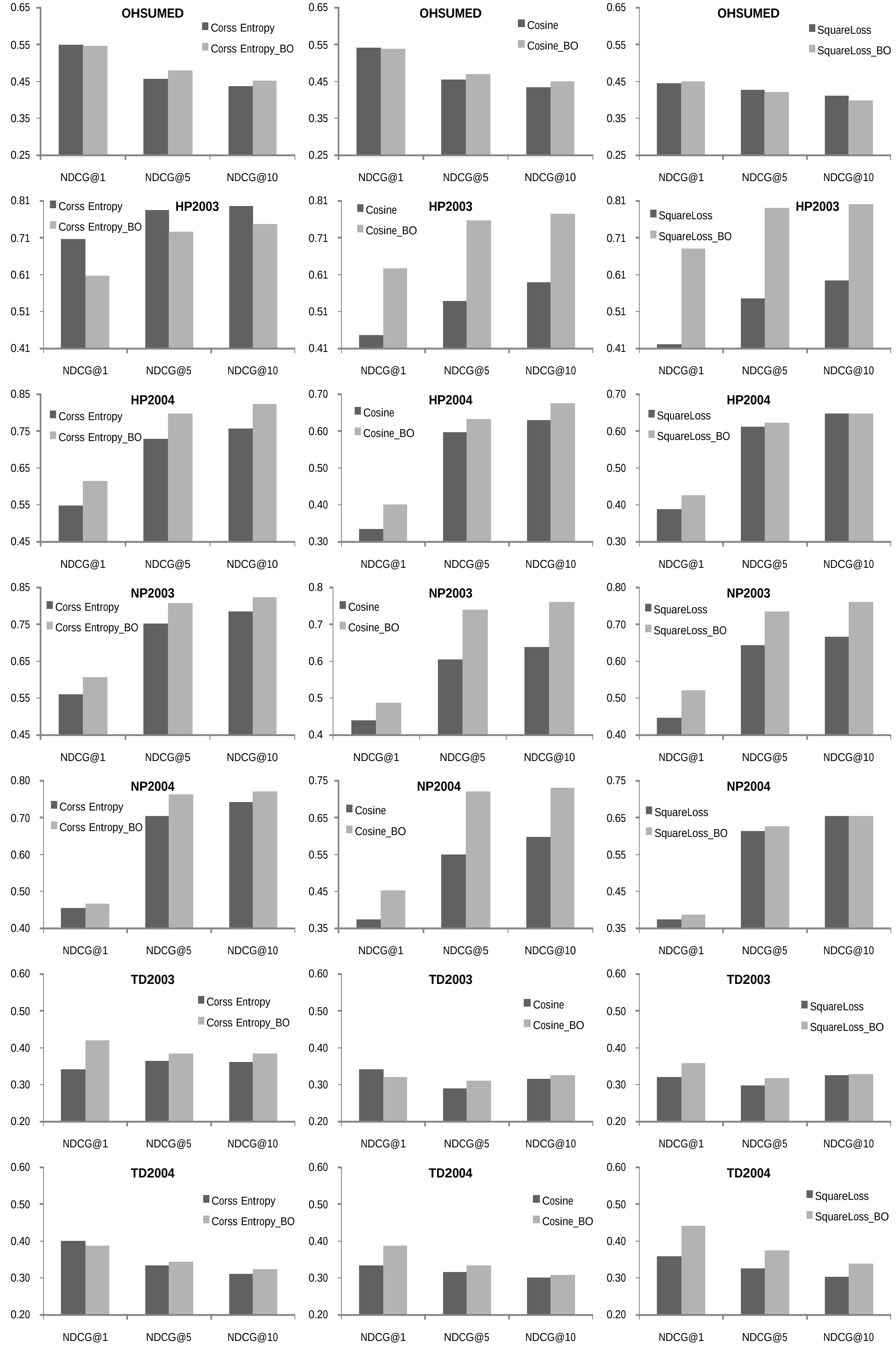

Figure 1: NDCG ranking accuracy achieved across seven LETOR [16] datasets. Detailed description of this figure is given in §6.3.

Table 1: Comparison of the "Bayes-Optimality" method to the baseline across 63 evaluation points: 7 datasets, 3 loss functions (cosine, cross-entropy, and squared), and 3 metrics (NDCG@{1,5,10}). For each case, we report whether our method performed better, same, or worse than the baseline (with statistical significance). We also report average change in relative accuracy across the 9 evaluation points for each dataset. Results show our method achieves 10.6% averge improvement overall, and on a case-by-basis basis, achieves statistically significant improvement in 23 of 63 cases (36.5%) while almost never hurting accuracy (only 3 cases).

Dataset	OHSUMED	HP2003	HP2004	NP2003	NP2004	TD2003	TD2004	Total	%
Better	3	6	2	7	3	0	2	23	36.5%
Same	6	0	7	2	6	9	7	37	58.7%
Worse	0	3	0	0	0	0	0	3	4.8%
Total	9	9	9	9	9	9	9	63	100%
Avg. change	1.2%	25.2%	8.4%	13.2%	10.6%	6.8%	8.4%	10.6%	

minimum with the least objective value. We on the other hand use a "warm-start" approach: we set the initial set of weights for the logistic probabilities based on the weights obtained for the optimal *linear ranking function* for the same surrogate evaluation metric. While the linear ranking function itself has been shown to perform well empirically. this would ensure that the weights we end up with are even better suited; as we show in the experiments below.

6.3 Results

Figure 1 presents detailed results of our emprical comparison of the proposed "Bayes-Optimality" (BO) method to the baseline method across 63 evaluation points: 7 datasets, 3 loss functions (cosine, cross-entropy, and squared), and 3 metrics (NDCG@{1,5,10}). Results of this figure are then concisely summarized in Table 1.

We see in Figure 1 the absolute NDCG ranking accuracies achieved by each method. Results on each dataset are presented as a row of three graphs corresponding to the three loss functions considered: cosine, cross-entropy, and squared (respectively). Each graph reports ranking accuracy for three metrics: NDCG@1, NDCG@5, and NDCG@10 (from left-to-right). For each metric, accuracy of the baseline is presented on the left in dark gray, with accuracy of the BO method presented on the right in light gray. The taller of the two bars indicates the better performing method and degree of difference (statistical significance is not reported in this figure but is reported in Table 1). Overall we see the proposed BO method nearly always performs as well or better, with particularly strong performance observed for most HP2003 & 2004 and NP2003 & 2004 evaluation points. The worst performance with BO is seen in using cross-entropy loss on the HP2003 dataset (the only three cases where BO performs worse). Further investigation of this abberant behavior on HP2003 only will be a subject of future work.

As mentioned above, Table 1 summarizes Figure 1 by reporting the number of cases in which the proposed BO method achieved better, same, or worse NDCG accuracy than the baseline with statistical significance (i.e. non-significant differences in accuracy are reported as "same"). Table 1 also reports average change in relative accuracy across the 9 evaluation points for each dataset.

Results show our method achieves consistent increase in NDCG accuracy across all seven datasets, averaging 10.6% improvement across datasets and achieving as high as 25.2% improvement (on HP2003). Moreover, while reporting of such averages can mask high variance (i.e. many or large losses on some datasets being offset by strong gains on a few), further analysis shows this is not true here. To the contrary, our technique demonstrates the desirable trait of being very low-risk: accuracy is reduced in only 3/63 cases (4.8%), while often achieving statistically significant improvement (in 23/63 cases, or 36.5%). Taken together, these results show our technique can be applied across a range of different datasets with strong confidence of achieving consistent improvement.

7. CONCLUSION

We studied a general class of IR evaluation metrics that includes DCG [13] and approximates ERR [6]. We showed the Bayes optimal ranking function for this class, which is the optimal ranking function in the infinite sample limit, can be expressed as a function of the conditional distribution of graded document relevance levels. We used the form of this Bayes optimal ranking function to posit a new class of *ranking functions*, which could be used in conjunction with any of the different surrogate loss functions for ranking. We tested our class of ranking functions for three such state of the art surrogate evaluation loss functions, and showed improvements in NDCG accuracies across seven LETOR datasets [16] and across the surrogate loss functions considered.

Future work will investigate further refinement and validation of our methods on two additional learning to rank datasets released in 2010 by major industrial search engine companies [1, 2, 7]. We also refer the interested reader to additional follow-on work by the authors [19, 29].

References

[1] Microsoft learning to rank datasets. `http://research.microsoft.com/mslr`.

[2] Yahoo! learning to rank grand challenge datasets. `http://learningtorankchallenge.yahoo.com`.

[3] R. Baeza-Yates and B. Ribeiro-Neto. *Modern information retrieval*. Addison Wesley, 1999.

[4] CJ Burges, QV Le, and R Ragno. Learning to Rank with Nonsmooth Cost Functions. In *Neural Information Processing Systems*, 2007.

[5] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *International Conference on Machine learning 24*, pages 129–136. ACM, 2007.

[6] O. Chapelle, D. Metzler, Y. Zhang, and P. Grinspan. Expected reciprocal rank for graded relevance. In *Conference on Information and Knowledge Management (CIKM)*, 2009.

[7] Olivier Chapelle, Yi Chang, and Tie-Yan Liu, editors. *Proceedings of the ICML 2010 Learning to Rank Challenge*

Workshop. Haifa, Israel, June 2010. To be published in JMLR: Workshop and Conference Proceedings.

[8] S. Clemencon, G. Lugosi, and N. Vayatis. Ranking and scoring using empirical risk minimization. In *Conference on Learning Theory (COLT)*, 2005.

[9] D. Cossock and T. Zhang. Statistical analysis of bayes optimal subset ranking. *IEEE Trans. Info. Theory*, 54:4140–5154, 2008.

[10] N. Craswell and D. Hawking. Overview of the TREC-2004 Web track. In *Proceedings of the 2004 Text REtreival Conference (TREC)*, 2005.

[11] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *Annals of Statistics*, 28:337–374, 2000.

[12] W. Hersh, C. Buckley, TJ Leone, and D. Hickam. OHSUMED: an interactive retrieval evaluation and new large test collection for research. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, 1994.

[13] K. Järvelin and J. Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 41–48. ACM, 2000.

[14] T. Joachims. Optimizing search engines using clickthrough data. In *Proc. of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining*, page 142, 2002.

[15] T.Y. Liu. Learning to Rank for Information Retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, 2009.

[16] T.Y. Liu, J. Xu, T. Qin, W. Xiong, and H. Li. Letor: Benchmark dataset for research on learning to rank for information retrieval. In *Proceedings of SIGIR 2007 Workshop on Learning to Rank for Information Retrieval*, pages 3–10, 2007.

[17] J. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. *Advances in Large Margin Classifiers*, pages 61–74, 1999.

[18] T. Qin, X.-D. Zhang, M.-F. Tsai, D.-S Wang, T.-Y. Liu, and H. Li. Query-level loss functions for information retrieval. *Information processing and management*, 2007.

[19] Pradeep D Ravikumar, Ambuj Tewari, and Eunho Yang. On ndcg consistency of listwise ranking methods. In *International Conference on Artificial Intelligence and Statistics*, pages 618–626, 2011.

[20] S.E. Robertson. The probability ranking principle in IR. *Journal of documentation*, 33(4):294–304, 1977.

[21] G. Salton and C. Buckley. Term Weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[22] Mark D. Smucker, James Allan, and Ben Carterette. A comparison of statistical significance tests for information retrieval evaluation. In *Proc. of CIKM*, pages 623–632, 2007.

[23] T. Strohman, D. Metzler, H. Turtle, and W.B. Croft. Indri: A language model-based search engine for complex queries. In *Proceedings of the International Conference on Intelligence Analysis*, 2004.

[24] M Taylor, J Guiver, S Robertson, and T Minka. Softrank: Optimising Non-smooth Rank Metrics. In *International Conference on Web Search and Web Data Mining*, pages 77–86. ACM, 2008.

[25] H Valizadegan, R Jin, R Zhang, and J Mao. Learning to Rank by Optimizing NDCG Measure. In *Neural Information Processing Systems*, 2010.

[26] M.N. Volkovs and R.S. Zemel. Boltzrank: Learning to maximize expected ranking gain. In *International Conference on Machine learning 26*, pages 1089–1096, 2009.

[27] F. Xia, T.-Y. Liu, J. Wang, W. Zhang, and H. Li. Listwise approach to learning to rank: theory and algorithm. In *International Conference on Machine learning 25*, pages 1192–1199, 2008.

[28] J. Xu and H. Li. Adarank: a boosting algorithm for information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, page 398. ACM, 2007.

[29] Eunho Yang, Ambuj Tewari, and Pradeep D Ravikumar. Perturbation based large margin approach for ranking. In *International Conference on Artificial Intelligence and Statistics*, pages 1358–1366, 2012.