Are Algorithms Directly Optimizing IR Measures Really Direct?

Yin He
University of Science and Technology of China
Hefei, Anhui, P.R.China
Samuel.HY@gmail.com
Tie-Yan Liu
Microsoft Research Asia
Beijing, P.R.China
tyliu@microsoft.com

Abstract

In information retrieval (IR), the objective of ranking problem is to construct and return a ranked list of relevant documents to the user. The document ranking list is demanded to satisfy user's information need as much as possible with respect to a user's query. To evaluate the goodness of the returned document ranking list, performance measures, such as Normalized Discounted Cumulative Gain (NDCG) and Mean Average Precision (MAP), are adopted.

Many learning to rank algorithms, which automatically learn ranking function through optimizing specially designed objective functions, are proposed to resolve the ranking problem. Intuitively, the IR performance measures are the ideal objective functions to be optimized to learn ranking function. However, IR performance measures, such as NDCG and MAP, are non-smooth and non-differentiable with respect to the ranking function parameter. Thus, most existing learning to rank algorithms are designed to optimize objective functions that are loosely related to the IR performance measures. As a result, such algorithms may only achieve sub-optimization of the IR performance measures even they can perform very well on optimizing their adopted objective functions. Therefore, it is highly demanded that learning to rank algorithms should be improved to be able to directly or approximately directly optimize information retrieval performance measures. To tackle the challenge of direct optimization of IR performance measures, several approaches, such as SoftRank[1] and SVM-MAP[2] are proposed. Although these algorithms can achieve good empirical performance, there are still some questions that are unclear and not yet answered: a) can ranking function learned by direct optimization of IR performance measures still perform well over unseen queries with respect to the optimized IR performance measures? b) how directly are IR performance measures optimized by the proposed approaches?

In this report, we will attempt to answer the above questions. We first point out that, under some conditions, the ranking function learned by direct optimization of IR performance measures can also perform well upon unseen queries with respect to the optimized IR performance measures. Then, to study how directly IR performance measures are optimized by previous approaches, we proposed a directness evaluate metric. Based on this metric, SoftRank is analyzed and corresponding results are presented.

I. Introduction

In information retrieval (IR), the objective of ranking problem is to construct and return a ranked list of relevant documents to the user. The document ranking list is demanded to satisfy user's information need as much as possible with respect to a user's query.

Recently, learning to rank algorithms, which automatically resolves the ranking problem with the help of supervised learning techniques, gains more and more attentions because of their good performance. In general, learning to rank algorithm consists of two processes: the training process and the testing process. The goal of the training process is to learn a ranking function that predicts the document ranking list for a given query. Usually, it is achieved by solving an optimization problem with regard to a specially designed objective function. The goal of the testing process is to evaluate the prediction goodness of the ranking function when facing unseen queries. Usually, IR performance measures such as Normalized Discount Cumulative Gain (NDCG) and Mean Average Precision (MAP) are utilized to perform the evaluation. Intuitively, the objective function, which is optimized by the training process, should be (or be directly related to) the IR performance measures themselves. However, typical IR

performance measures are non-smooth and non-differentiable with regard to ranking function parameter. Therefore, it is not easy to learn a ranking function in practical time by directly optimizing the IR performance measures. The difficulty comes from the fact that most existing optimization techniques are developed to handle smooth and differentiable cases. Therefore, most existing learning to rank algorithms only optimize objective functions that are loosely related to the IR performance measures. As a result, they may only achieve sub-optimization of the IR performance measures even they can perform very well upon optimizing their adopted objective functions. Noticing the gap between IR performance measures and actually utilized objective functions, several approaches are proposed to directly optimize IR performance measures. For example, SoftRank[1] smoothes the deterministic document score with Gaussian distribution and then optimizes the softened NDCG derived from the score distribution. SVM-MAP[2] adopts structural SVM to optimize a hinge loss that is a convex upper bound of IR performance measures. Although such algorithms can achieve good empirical performance, there are still some questions that are unclear and not yet answered: a) can ranking function learned by direct optimization of IR performance measures still perform well over unseen queries with respect to the optimized IR performance measures? b) how directly are IR performance measures optimized by the proposed approaches?

In this report, we will attempt to answer the above questions. Firstly, according to the consistency and generalization ability of empirical risk minimization (ERM), we point out that if the is the ranking function space is not very complex, when the number of training examples become infinity, the ranking function learned by directly optimizing the IR performance measures can achieve optimal performance upon unseen queries with respect to the optimized IR performance measures. In other words, direct optimization of IR performance measures is not only intuitively but also theoretically reasonable. Then, to investigate how directly IR performance measures are optimized by previous direct methods, we propose a metric to evaluate the directness of the objective functions that are actually optimized by such methods. Finally, based on the proposed directness evaluation metric, we analyze SoftRank and present the corresponding results we get.

The rest of this report is organized as follow. In section II, we describe the general framework of learning to rank algorithms. Then we introduce the formulation of SoftRank in section III. In section IV, we theoretically justify that it is reasonable to directly optimizing IR performance measures under empirical risk minimization framework. In section V a metric which can be used to evaluate the directness of direct method is proposed. Finally, based on the proposed metric, we present analyze SoftRank and present our analysis results in section VI.

II. General Framework of Learning to Rank

In this section, we describe the general framework of learning to rank. In the training process, a set of queries $Q = \{q^{(1)}, q^{(2)}, \ldots, q^{(m)}\}$ is given. Each query $q^{(i)}$ is associated with a list of documents $\mathbf{d}^{(i)} = \{d_1^{(i)}, d_2^{(i)}, \ldots, d_{n_q}^{(i)}\}$ and a ranking $\mathbf{y}^{(i)} = \{\tau^{(i)}(1), \tau^{(i)}(2), \ldots, \tau^{(i)}(n_q)\}$ over the documents, where n_q denotes the size of $\mathbf{d}^{(i)}$, $d_j^{(i)}$ denotes the j^{th} document in $\mathbf{d}^{(i)}$, $\tau \in \mathcal{T}$ is a bijection from $\{1, 2, \ldots, n_q\}$ to itself, and $\tau^{(i)}(j)$ denotes the position of document $d_j^{(i)}$ in $\mathbf{y}^{(i)}$. From each query-document pair $(q^{(i)}, d_j^{(i)})$, a feature vector $\mathbf{x}^{(i)} = \{\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \ldots, \mathbf{x}_{n_q}^{(i)}\}$ is constructed, where $\mathbf{x}_j^{(i)} = \phi(q^{(i)}, d_j^{(i)}) \in \mathcal{R}^n$. The goal of the training process is to learn a ranking function $h : \mathcal{X} \to \mathcal{Y}$ through optimizing an objective function $O(\boldsymbol{\omega}, \mathbf{x}^{(i)}, \mathbf{y}^{(i)})$. $\mathcal{X}$ is the space of $\mathbf{x}$ and $\mathcal{Y}$ is the space of all possible permutations over documents. $O(\boldsymbol{\omega}, \mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ represents the penalty for making prediction $h(\boldsymbol{\omega}, \mathbf{x}^{(i)})$ if the correct output is $\mathbf{y}^{(i)}$, where $\boldsymbol{\omega}$ is the parameter of h.

In the testing process, given a new query q associated with feature vector $\mathbf{x}$ and ranking list $\mathbf{y}$, a document ranking list $\hat{\mathbf{y}}$ is predicted by $h(\boldsymbol{\omega}, \mathbf{x})$.[1] To evaluate the goodness of the prediction of h, IR performance measures, such as NDCG and MAP, are adopted. Here, we present the definitions of NDCG:

$$\mathrm{G}(\boldsymbol{\omega}, \mathbf{x}, \mathbf{y}) = \frac{1}{G_{max}} \sum_{j=1}^{n_q} g(d_j) D(r_j), \quad g(d_j) = 2^{l_j}, \quad D(r_j) = \frac{1}{\log 2 + r_j},$$

where G_{max} is a normalization factor, l_j is the label of document d_j, and r_j is the position of document d_j in $\mathbf{y}$.

[1]We will ignore the upper script (i) when content is clear.

III. The formulation of SoftRank

SoftRank is proposed in [1] to optimize a smooth approximation of the IR performance measure NDCG. There are three key steps in the SoftRank algorithm.

The first step is to smooth the deterministic score of document $\mathbf{x}_j^{(i)}$ of query $q^{(i)}$. Originally, the score is a deterministic value outputted by the ranking function $f(\boldsymbol{\omega}, \mathbf{x}_j^{(i)}) : \mathcal{R}^n \times \mathcal{R}^n \rightarrow \mathcal{R}$. After smoothened by a Gaussian distribution whose variance is σ_s and mean is the original deterministic score, the score of document $\mathbf{x}_j^{(i)}$ is regarded as a random variable having probability density as follow.

$$p(s_j^{(i)}) = \mathcal{N}(s_j^{(i)} | f(\boldsymbol{\omega}, \mathbf{x}_j^{(i)}), \sigma_s^2)$$

The second step is to define the rank distribution of document $\mathbf{x}_j^{(i)}$. To achieve this, the probability that a document is ranked before another is deduced as follow.

$$\pi_{ij}^{(i)} = \Pr(S_i^{(i)} - S_j^{(i)} > 0) = \int_0^{\infty} \mathcal{N}\left(s | f(\boldsymbol{\omega}, \mathbf{x}_j^{(i)}) - f(\boldsymbol{\omega}, \mathbf{x}_j^{(j)}), \sigma_s^2\right) ds.$$

Then the rank distribution can be calculated from a recursive process as follow.

$$p_j^{(i)(1)}(r) = \delta(r)$$
$$p_j^{(i)(k)}(r) = p_j^{(i)(k-1)}(r-1)\pi_{kj} + p_j^{(i)(k-1)}(r)(1 - \pi_{kj})$$

where $\delta(r) = 1$ only when $r = 0$ and zero otherwise.

Third, SoftNDCG is defined as the expectation of NDCG in terms of the rank distribution.

$$\mathcal{G}(\boldsymbol{\omega}, \mathbf{x}^{(i)}, \mathbf{y}^{(i)}) = \frac{1}{G_{max}^{(i)}} \sum_{j=1}^{n_q} g(d_j^{(i)}) \sum_{r=0}^{n_q - 1} D(r) p_j^{(i)}(r)$$

In order to maximize SoftNDCG, an artificial neural network framework is utilized.

Refer to the learning to rank framework described in above section, we can know, in the training process, SoftRank learns the ranking function parameter $\boldsymbol{\omega}$ by resolving **Optimization Problem 1** presented below.

Optimization Problem 1.

$$\min_{\boldsymbol{\omega}} \frac{1}{m} \sum_{i=1}^{m} \left[1 - \mathcal{G}(\boldsymbol{\omega}, \mathbf{x}^{(i)}, \mathbf{y}^{(i)})\right]$$

In the testing process, given a new query q associated with feature vector $\mathbf{x}$ and ranking list $\mathbf{y}$, a ranking list $\hat{\mathbf{y}}$ is predicted through sorting the documents by $f(\boldsymbol{\omega}, \mathbf{x}_j)$ in descending order.

From the formulation of **Optimization Problem 1**, we can define

$$O_1(\boldsymbol{\omega}, \mathbf{x}^{(i)}, \mathbf{y}^{(i)}) = 1 - \mathcal{G}(\boldsymbol{\omega}, \mathbf{x}^{(i)}, \mathbf{y}^{(i)})$$
$$O_2(\boldsymbol{\omega}, \mathbf{x}^{(i)}, \mathbf{y}^{(i)}) = 1 - G(\boldsymbol{\omega}, \mathbf{x}^{(i)}, \mathbf{y}^{(i)}),$$

It is obviously that the objective function actually optimized and the objective function intended to be optimized by SoftRank are $\frac{1}{m}\sum_{i=1}^{m} O_1(\boldsymbol{\omega}, \mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ and $\frac{1}{m}\sum_{i=1}^{m} O_2(\boldsymbol{\omega}, \mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ respectively.

IV. Is Direct Optimization of IR Performance Measure Reasonable?

Intuitively, people think the ranking function learned by directly optimizing IR performance measures can perform well with respect to the optimized measure upon unseen queries. To theoretically justify this intuition, we should refer to **Lemma 1** as follow.

Lemma 1. *Given a training set* $\{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^{m}$*, which is sampled from distribution* $P(\mathbf{x}, \mathbf{y})$*. Let* $O(\boldsymbol{\omega}, \mathbf{x}, \mathbf{y}) : \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{R}$ *be an objective function parameterized by* $\boldsymbol{\omega}$*. Denote*

$$
\begin{aligned}
R(\boldsymbol{\omega}; O) &= \mathbb{E}_{\mathbf{X},\mathbf{Y}}[O(\boldsymbol{\omega}, \mathbf{x}, \mathbf{y})] \\
\hat{R}_m(\boldsymbol{\omega}; O) &= \frac{1}{m} \sum_{i=1}^{m} O(\boldsymbol{\omega}, \mathbf{x}^{(i)}, \mathbf{y}^{(i)}) \\
\boldsymbol{\omega}^{\star} &= \arg\min R(\boldsymbol{\omega}) \\
\hat{\boldsymbol{\omega}}_m &= \arg\min \hat{R}_m(\boldsymbol{\omega})
\end{aligned}
$$

The following inequality is ensured

$$
R(\hat{\boldsymbol{\omega}}_m; O) - R(\boldsymbol{\omega}^{\star}; O) \leq 2 \sup |R(\boldsymbol{\omega}; O) - \hat{R}_m(\boldsymbol{\omega}; O)|
$$

According to the consistency and generalization ability of empirical risk minimization, if the space of O is not very complex[2], we know when the number of training examples become infinity, $\sup |R(\boldsymbol{\omega}; O) - \hat{R}_m(\boldsymbol{\omega}; O)| \rightarrow 0$. In other words, we have $R(\hat{\boldsymbol{\omega}}_m; O) \xrightarrow{m \rightarrow \infty} R(\boldsymbol{\omega}^{\star}; O)$. As for the ranking problem, we can consider O to be directly related to the IR performance measures, such as $1 - \text{NDCG}$ or $1 - \text{MAP}$. Correspondingly, $\hat{\boldsymbol{\omega}}_m$[3] is the ranking function obtained by directly optimizing a IR performance measure over the queries in the training set. According to **Lemma 1**, we know if a ranking function is learned by directly optimizing a IR performance measure over infinity queries, the ranking function can achieve optimal performance with respect to that IR performance measure over any query. Therefore, we can believe that direct optimization IR performance measures is reasonable.

V. Directness Evaluation Metric

According to the formulation of SoftRank in **Optimization Problem 1**, we know it does not actually optimize NDCG directly. Instead, it optimizes a smooth approximation of NDCG, named as SoftNDCG, to learn the ranking function[4]. Although, the previous direct methods can achieve optimal performance in terms of their own objective functions, it is not clear whether optimal performance in terms of IR performance measures can be achieved by their learned ranking function. To answer this question, we propose a metric, as depicted in **Theorem 1**, to evaluate how directly IR performance measures is optimized when an algorithms achieve optimization of its own objective function.

Theorem 1. *For any query* $q^{(i)}$*, suppose its associated feature vector and document ranking list is* $\mathbf{x}^{(i)}$ *and* $\mathbf{y}^{(i)}$ *respectively. Denote* $O_1(\boldsymbol{\omega}, \mathbf{x}, \mathbf{y}), O_2(\boldsymbol{\omega}, \mathbf{x}, \mathbf{y}) : \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{R}$ *as two objective functions, which are parameterized by* $\boldsymbol{\omega}$*. Let*

$$
\begin{aligned}
\boldsymbol{\omega}_1 &= \arg\min \frac{1}{m} \sum_{i=1}^{m} O_1(\boldsymbol{\omega}, \mathbf{x}^{(i)}, \mathbf{y}^{(i)}) \\
\boldsymbol{\omega}_2 &= \arg\min \frac{1}{m} \sum_{i=1}^{m} O_2(\boldsymbol{\omega}, \mathbf{x}^{(i)}, \mathbf{y}^{(i)})
\end{aligned}
$$

For any $(\mathbf{x}, \mathbf{y}) \sim P(\mathbf{x}, \mathbf{y})$*, if* O_1 *and* O_2 *satisfy*

$$
\begin{aligned}
|O_1(\boldsymbol{\omega}_1, \mathbf{x}, \mathbf{y}) - O_2(\boldsymbol{\omega}_1, \mathbf{x}, \mathbf{y})| &\leq \varepsilon_1 \\
|O_1(\boldsymbol{\omega}_2, \mathbf{x}, \mathbf{y}) - O_2(\boldsymbol{\omega}_2, \mathbf{x}, \mathbf{y})| &\leq \varepsilon_2
\end{aligned}
$$

The following inequality is ensured

$$
|\mathbb{E}_{\mathbf{X},\mathbf{Y}}[O_2(\boldsymbol{\omega}_1, \mathbf{x}, \mathbf{y})] - \mathbb{E}_{\mathbf{X},\mathbf{Y}}[O_2(\boldsymbol{\omega}_2, \mathbf{x}, \mathbf{y})]| \leq 2\varepsilon_1 + \varepsilon_2
$$

[2]For example, if the space of O is finite, which is just the case in the practical optimization process since one can only try finite number of different parameters.

[3]We use $\boldsymbol{\omega}$ to represent a ranking function when context is clear

[4]Similarly, it is easy to know that SVM-MAP actually optimizes a convex upper bound of MAP to learn the ranking function.

Here we give the proof of **Theorem 1**[5].

Proof: Let From the condition of **Theorem 1**, we have:

$$O_1(\boldsymbol{\omega}_2) \geq O_1(\boldsymbol{\omega}_1)$$
$$O_2(\boldsymbol{\omega}_1) \geq O_2(\boldsymbol{\omega}_2)$$

1) If $O_1(\boldsymbol{\omega}_2) \geq O_2(\boldsymbol{\omega}_1)$

$$\begin{aligned}|O_1(\boldsymbol{\omega}_1) - O_2(\boldsymbol{\omega}_2)| &= |O_1(\boldsymbol{\omega}_1) - O_2(\boldsymbol{\omega}_1) + O_2(\boldsymbol{\omega}_1) - O_2(\boldsymbol{\omega}_2)| \\ &\leq |O_1(\boldsymbol{\omega}_1) - O_2(\boldsymbol{\omega}_1)| + |O_2(\boldsymbol{\omega}_1) - O_2(\boldsymbol{\omega}_2)| \\ &\leq \varepsilon_1 + |O_1(\boldsymbol{\omega}_2) - O_2(\boldsymbol{\omega}_2)| \\ &\leq \varepsilon_1 + \varepsilon_2\end{aligned}$$

2) If $O_1(\boldsymbol{\omega}_2) < O_2(\boldsymbol{\omega}_1)$

$$\begin{aligned}|O_1(\boldsymbol{\omega}_1) - O_2(\boldsymbol{\omega}_2)| &= |O_1(\boldsymbol{\omega}_1) - O_1(\boldsymbol{\omega}_2) + O_1(\boldsymbol{\omega}_2) - O_2(\boldsymbol{\omega}_2)| \\ &\leq |O_1(\boldsymbol{\omega}_1) - O_1(\boldsymbol{\omega}_2)| + |O_1(\boldsymbol{\omega}_2) - O_2(\boldsymbol{\omega}_2)| \\ &\leq |O_2(\boldsymbol{\omega}_1) - O_1(\boldsymbol{\omega}_1)| + \varepsilon_2 \\ &\leq \varepsilon_1 + \varepsilon_2\end{aligned}$$

In summary

$$|O_1(\boldsymbol{\omega}_1) - O_2(\boldsymbol{\omega}_2)| \leq \varepsilon_1 + \varepsilon_2$$

Therefore

$$\begin{aligned}|O_2(\boldsymbol{\omega}_1) - O_2(\boldsymbol{\omega}_2)| &= |O_2(\boldsymbol{\omega}_1) - O_1(\boldsymbol{\omega}_1) + O_1(\boldsymbol{\omega}_1) - O_2(\boldsymbol{\omega}_2)| \\ &\leq |O_2(\boldsymbol{\omega}_1) - O_1(\boldsymbol{\omega}_1)| + |O_1(\boldsymbol{\omega}_1) - O_2(\boldsymbol{\omega}_2)| \\ &\leq 2\varepsilon_1 + \varepsilon_2\end{aligned}$$

$$|\mathbb{E}_{\mathbf{X},\mathbf{Y}}[O_2(\boldsymbol{\omega}_1, \mathbf{x}, \mathbf{y})] - \mathbb{E}_{\mathbf{X},\mathbf{Y}}[O_2(\boldsymbol{\omega}_2, \mathbf{x}, \mathbf{y})]| \leq \int_{\mathbf{x}\times\mathbf{y}} |O_2(\boldsymbol{\omega}_1) - O_2(\boldsymbol{\omega}_2)| P(d\mathbf{x}, d\mathbf{y}) \leq 2\varepsilon_1 + \varepsilon_2$$

■

When considering our direct optimization problem, we can assume $O_2(\boldsymbol{\omega}, \mathbf{x}, \mathbf{y}) = 1 - E(\boldsymbol{\omega}, \mathbf{x}, \mathbf{y})$ and $O_1(\boldsymbol{\omega}, \mathbf{x}, \mathbf{y}) = 1 - \hat{E}(\boldsymbol{\omega}, \mathbf{x}, \mathbf{y})$, where E is a specified IR performance measure and $\hat{E}$ is the objective function, we call it surrogate measure, optimized by a direct optimization algorithm. $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$ correspond to two ranking functions that are obtained by optimizing $\hat{E}$ and E respectively. **Theorem 1** indicates that, for any query, if the surrogate measure $\hat{E}$ can have similar value as IR performance measure E with respect to both $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$, then $\boldsymbol{\omega}_1$ can achieve similar expected risk with respect to IR performance measure E as $\boldsymbol{\omega}_2$.

Combining **Theorem 1** and **Lemma 1**, we can know that given a big enough training set, once a surrogate measure $\hat{E}$ can satisfy the condition of **Theorem 1**, then, the ranking function learned by optimizing $\hat{E}$ over the training set can perform optimally upon any unseen query when evaluated by the corresponding IR performance measure E.

VI. Analysis of SoftRank

From above section, we know that if a surrogate measure $\hat{E}$ is related to a IR performance measure directly enough, the ranking function learned by optimizing that surrogate measure over a big enough training set is ensured to perform optimally upon unseen queries. Therefore, the directness of the surrogate measure can depict the goodness of a direct method. Taking SoftRank as an example, we study the directness of its surrogate measure SoftNDCG and present the analysis result as follow.

[5]In this proof, we simplify $O_i(\boldsymbol{\omega}, \mathbf{x}, \mathbf{y})$ as $O_i(\boldsymbol{\omega})$, where $i = 1, 2$.

Theorem 2. *Suppose a query q, its associated feature and document ranking list is $\mathbf{x}$ and $\mathbf{y}$ respectively. For any $\boldsymbol{\omega}$, let*

$$O_1(\boldsymbol{\omega}, \mathbf{x}, \mathbf{y}) = 1 - \mathcal{G}(\boldsymbol{\omega}, \mathbf{x}, \mathbf{y})$$
$$O_2(\boldsymbol{\omega}, \mathbf{x}, \mathbf{y}) = 1 - G(\boldsymbol{\omega}, \mathbf{x}, \mathbf{y}),$$

*be the objective function and its corresponding IR performance measure defined in **Optimization Problem 1**. Assume $|s_{ij}| \geq \delta$, $n_q \leq N$ and there are in total K ordered categories in the ground truth label, i.e., $\{0, \cdots, K-1\}$. Then when $\sigma_s < \frac{\delta}{2erf^{-1}\left(\sqrt{\frac{N-2}{N-1}}\right)}$, the difference between O and E satisfies:*

$$|O_1(\boldsymbol{\omega}, \mathbf{x}, \mathbf{y}) - O_2(\boldsymbol{\omega}, \mathbf{x}, \mathbf{y})| \leq 2^{K-1} \cdot N \cdot (\varepsilon_1 + \varepsilon_2)$$

where

$$\varepsilon_1 = \frac{(N-1)\sigma_s}{2\delta\sqrt{\pi}} e^{-\frac{\delta^2}{4\sigma_s^2}}, \quad \varepsilon_2 = \sqrt{\frac{\varepsilon_3(\sigma_s)}{1 - 5\varepsilon_3(\sigma_s)}} + 5\varepsilon_3(\sigma_s), \quad \varepsilon_3 = \left[1 - erf^2\left(\frac{\delta}{2\sigma_s}\right)\right]$$
$$erf(x) = \frac{2}{\sqrt{\pi}} \int_{-\infty}^{x} e^{-t^2} dt, \quad s_{ij} = s_i - s_j = f(\boldsymbol{\omega}, \mathbf{x}_i) - f(\boldsymbol{\omega}, \mathbf{x}_j), \quad f(\boldsymbol{\omega}, \mathbf{x}_j) : \mathcal{R}^n \times \mathcal{R}^n \to \mathcal{R}$$

Before giving the proof of **Theorem 2**, we first prove a lemma as below.

Lemma 2. *Assume a discrete random variable $R = 0, 1, \ldots, n-1$ follow distribution $p(r|\mu, \sigma^2)$, where μ and σ^2 is expected value and variance. Denote μ' as the closest integer to μ, μ'' as the second closest integer to μ, then*

$$Pr(R = \mu') \geq 1 - 5\sigma^2$$
$$Pr(R = \mu'') \leq 4\sigma^2$$
$$\sum_{r \neq \mu', \mu''} Pr(R = r) \leq \sigma^2$$

Proof: According to the definitions of mu_1 and μ'', we have

$$0 \leq |\mu' - \mu| \leq \frac{1}{2}$$
$$\frac{1}{2} \leq |\mu'' - \mu| \leq 1$$
$$\forall r \neq \mu', \mu'', \quad |r - \mu| \geq 1$$

According to the definition of σ^2, we have

$$\begin{aligned}
\sigma^2 &= \sum_{r=0}^{n-1} \Pr(R = r)(r - \mu)^2 \\
&= \Pr(R = \mu')(\mu' - \mu)^2 + \Pr(R = \mu'')(\mu'' - \mu)^2 \\
&+ \sum_{r \neq \mu', \mu''} \Pr(R = r)(r - \mu)^2 \\
&\geq \Pr(R = \mu')(\mu' - \mu)^2 + \frac{1}{4}\Pr(R = \mu'') + \sum_{r \neq \mu', \mu''} \Pr(R = r)
\end{aligned}$$

Therefore,

$$\begin{gathered}\Pr(R=\mu')(\mu'-\mu)^2+\Pr(R=\mu'')(\mu''-\mu)^2\geq 0\\ \Rightarrow \sum_{r\neq\mu',\mu''}\Pr\leq\sigma^2\\ \Pr(R=\mu')(\mu'-\mu)^2+\sum_{r\neq\mu',\mu''}\Pr(R=r)\geq 0\\ \Rightarrow \Pr(R=\mu'')\leq 4\sigma^2\\ \Pr(R=\mu')=1-\Pr(R=\mu'')-\sum_{r\neq\mu',\mu''}\Pr(R=r)\\ \Rightarrow \Pr(R=\mu')\geq 1-\sigma^2\end{gathered}$$

■

Then we give the proof of **Theorem 2**.

Proof: Because

$$|O_1(\boldsymbol{\omega},\mathbf{x},\mathbf{y})-O_2(\boldsymbol{\omega},\mathbf{x},\mathbf{y})|\leq 2^{K-1}N\left|\sum_{r=0}^{n_q-1}D(r)p_j(r)-D(r_j)\right|$$

$$\left|\sum_{r=0}^{n_q-1}D(r)p_j(r)-D(r_j)\right|\leq|D(\mu_j)-D(r_j)|+\left|\sum_{r=0}^{n_q-1}D(r)p_j(r)-D(\mu_j)\right|$$

$$\mu_j=\mathbb{E}[\tilde{R}_j]$$

If we can prove

$$|D(\mu_j)-D(r_j)|\leq\varepsilon_1 \tag{1}$$

$$\left|\sum_{r=0}^{n_q-1}D(r)p_j(r)-D(\mu_j)\right|\leq\varepsilon_2, \tag{2}$$

then **Theorem 2** is proved.

We first construct a random variable $\tilde{R}_{ij}$ as follow:

$$\tilde{R}_{ij}\triangleq\begin{cases}1, & \text{if } S_i>S_j\\ 0, & \text{if } S_i\leq S_j\end{cases}$$

Let $\pi_{ij}\triangleq\Pr(S_i-S_j>0)$, we have:

$$\begin{cases}\Pr(\tilde{R}_{ij}=1)=\pi_{ij}\\ \Pr(\tilde{R}_{ij}=0)=1-\pi_{ij}\end{cases}$$

We know

$$\begin{aligned}\pi_{ij}\triangleq\Pr(S_i-S_j>0)&=\int_0^{\infty}\mathcal{N}(s|s_{ij},2\sigma_s^2)ds\\ &=1-\int_{-\infty}^{0}\mathcal{N}(s|s_{ij},2\sigma_s^2)ds=1-\Phi_{s_{ij},2\sigma_s^2}(0)\\ &=1-\frac{1}{2}\left[1+\operatorname{erf}\left(\frac{0-s_{ij}}{2\sigma_s}\right)\right]=\frac{1}{2}\left[1+\operatorname{erf}\left(\frac{s_{ij}}{2\sigma_s}\right)\right]\end{aligned}$$

On the other hand, we know $\tilde{R}_j \sim p_j(r)$ can be calculated from $\tilde{R}_j = \sum_{i \neq j} \tilde{R}_{ij}$.

$$\begin{aligned}
\mu_j &\triangleq E[\tilde{R}_j] = \sum_{i \neq j} E[\tilde{R}_{ij}] \\
&= \sum_{i \neq j} \pi_{ij} = \sum_{i \neq j} \frac{1}{2}\left[1 + \text{erf}\left(\frac{s_{ij}}{2\sigma_s}\right)\right] \\
\sigma_j^2 &\triangleq Var[\tilde{R}_j] = \sum_{i \neq j} Var[\tilde{R}_{ij}] = \sum_{i \neq j} \left\{E[\tilde{R}_{ij}^2] - E^2[\tilde{R}_{ij}]\right\} \\
&= \sum_{i \neq j} \pi_{ij}(1 - \pi_{ij}) = \sum_{i \neq j} \frac{1}{4}\left[1 - \text{erf}^2\left(\frac{s_{ij}}{2\sigma_s}\right)\right]
\end{aligned}$$

Now, we first prove Eqn.(1) here. The deterministic rank of document j can be calculated by

$$r_j = \sum_{i \neq j} I[s_{ij} > 0], \quad I[s_{ij} > 0] = \begin{cases} 1 & \text{if } s_{ij} > 0 \\ 0 & \text{if } s_{ij} < 0 \end{cases}$$

The difference between r_j and μ_j is:

$$\begin{aligned}
|r_j - \mu_j| &= \left| \sum_{i \neq j} I[s_{ij} > 0] - \sum_{i \neq j} \frac{1}{2}\left[1 + \text{erf}\left(\frac{s_{ij}}{2\sigma_s}\right)\right] \right| \\
&= \left| \sum_{i \neq j, s_{ij} > 0} 1 - \frac{1}{2}\left[1 + \text{erf}\left(\frac{s_{ij}}{2\sigma_s}\right)\right] - \sum_{i \neq j, s_{ij} < 0} \frac{1}{2}\left[1 + \text{erf}\left(\frac{s_{ij}}{2\sigma_s}\right)\right] \right| \\
&= \left| \sum_{i \neq j, s_{ij} > 0} \frac{1}{2}\text{erfc}\left(\frac{s_{ij}}{2\sigma_s}\right) - \sum_{i \neq j, s_{ij} < 0} \frac{1}{2}\text{erfc}\left(\frac{-s_{ij}}{2\sigma_s}\right) \right| \\
&\leq \left| \sum_{i \neq j, s_{ij} > 0} \frac{1}{2}\text{erfc}\left(\frac{s_{ij}}{2\sigma_s}\right) + \sum_{i \neq j, s_{ij} < 0} \frac{1}{2}\text{erfc}\left(\frac{-s_{ij}}{2\sigma_s}\right) \right| \\
&= \sum_{i \neq j} \frac{1}{2}\text{erfc}\left(\frac{|s_{ij}|}{2\sigma_s}\right).
\end{aligned}$$

In above $\text{erfc}(x) = 1 - \text{erf}(x)$. In above $\text{erfc}(x) = 1 - \text{erf}(x)$. Based on the properties of Normal Distribution, we have

$$x > 0, \quad \frac{1}{2}\text{erfc}\left(\frac{x}{\sqrt{2}}\right) = 1 - \Phi(x) < \frac{1}{x}\phi(x) = \frac{1}{x}\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}.$$

Therefore,

$$\frac{1}{2}\text{erfc}\left(\frac{|s_{ij}|}{2\sigma_s}\right) < \frac{\sigma_s}{\sqrt{\pi}|s_{ij}|} e^{-\frac{s_{ij}^2}{4\sigma_s^2}}.$$

In summary, we have

$$\begin{aligned}
|r_j - \mu_j| &< \frac{1}{\sqrt{\pi}} \sum_{i \neq j} \frac{\sigma_s}{|s_{ij}|} e^{-\frac{s_{ij}^2}{4\sigma_s^2}} \leq \frac{(N-1)\sigma_s}{\delta\sqrt{\pi}} e^{-\frac{\delta^2}{4\sigma_s^2}} = 2\varepsilon_1 \\
|D(r_j) - D(\mu_j)| &= \left| \frac{1}{\log(2 + r_j)} - \frac{1}{\log(1 + \mu_j)} \right| \\
&\leq \sup_{\theta \in [0, n_q + \varepsilon_1 - 1]} \left| \frac{r_j - \mu_j}{(2 + \theta)\log^2(2 + \theta)} \right| \\
&< \varepsilon_1
\end{aligned}$$

We then prove Eqn.(2). Assume μ_j' is the closest integer to μ_j and μ_j'' is the second closest integer to μ_j. According to **Lemma 2**, we have:

$$\begin{aligned}
E[D(\tilde{R}_j)] &= \sum_{r=0}^{n_q-1} \frac{\Pr(\tilde{R}_j = r)}{\log(2+r)} \\
&= \frac{\Pr(\tilde{R}_j = \mu_j')}{\log(2+\mu_j')} + \frac{\Pr(\tilde{R}_j = \mu_j'')}{\log(2+\mu_j'')} + \sum_{r \neq \mu_j', \mu_j''} \frac{\Pr(\tilde{R}_j = r)}{\log(2+r)} \\
&\leq \frac{\Pr(\tilde{R}_j = \mu_j')}{\log(2+\mu_j')} + \Pr(\tilde{R}_j = \mu_j'') + \sum_{r \neq \mu_j', \mu_j''} \Pr(\tilde{R}_j = r) \\
&\leq \frac{\Pr(\tilde{R}_j = \mu_j')}{\log(2+\mu_j')} + 4\sigma_j^2 + \sigma_j^2 \\
&\leq \frac{1}{\log(2+\mu_j')} + 5\sigma_j^2
\end{aligned}$$

Therefore,

$$\begin{aligned}
|E[D(\tilde{R}_j)] - D(E[\tilde{R}_j])| &= \left| \sum_{r=0}^{n_q-1} \frac{\Pr(\tilde{R}_j = r)}{\log(2+r)} - \frac{1}{\log(2+\mu_j)} \right| \\
&\leq \left| \frac{1}{\log(2+\mu_j')} - \frac{1}{\log(2+\mu_j)} \right| + 5\sigma_j^2 \\
&= \frac{|\log(2+\mu_j) - \log(2+\mu_j')|}{\log(2+\mu_j) \cdot \log(2+\mu_j')} + 5\sigma_j^2 \\
&\leq |\log(2+\mu_j) - \log(2+\mu_j')| + 5\sigma_j^2
\end{aligned}$$

1) If $\mu_j \leq \mu_j'$, we have

$$\begin{aligned}
&|E[D(\tilde{R}_j)] - D(E[\tilde{R}_j])| \leq \log(2+\mu_j') - \log(2+\mu_j) + 5\sigma_j^2 \\
&= \log(1 + \frac{\mu_j' - \mu_j}{2+\mu_j}) + 5\sigma_j^2 \leq \frac{\mu_j' - \mu_j}{2+\mu_j} + 5\sigma_j^2 \\
&\leq \mu_j' - \mu_j + 5\sigma_j^2
\end{aligned}$$

2) If $\mu_j > \mu_j'$, we have

$$\begin{aligned}
&|E[D(\tilde{R}_j)] - D(E[\tilde{R}_j])| \leq \log(2+\mu_j) - \log(2+\mu_j') + 5\sigma_j^2 \\
&= \log(1 + \frac{\mu_j - \mu_j'}{2+\mu_j'}) + 5\sigma_j^2 \leq \frac{\mu_j - \mu_j'}{2+\mu_j'} + 5\sigma_j^2 \\
&\leq \mu_j - \mu_j' + 5\sigma_j^2
\end{aligned}$$

In summary

$$|E[D(\tilde{R}_j)] - D(E[\tilde{R}_j])| \leq |\mu_j - \mu_j'| + 5\sigma_j^2$$

Because

$$\begin{aligned}
\sigma_j^2 &\geq \Pr(\tilde{R}_j = \mu_j')(\mu_j' - \mu)^2 \\
\Pr(\tilde{R}_j = \mu_j') &\geq 1 - 5\sigma_j^2
\end{aligned}$$

Therefore, when $\sigma_j^2 < \frac{1}{5}$

$$|\mu_j' - \mu_j| \leq \sqrt{\frac{\sigma_j^2}{1 - 5\sigma_j^2}}$$

In summary

$$|E[D(\tilde{R}_j)] - D(E[\tilde{R}_j])| \leq \sqrt{\frac{\sigma_j^2}{1-5\sigma_j^2}} + 5\sigma_j^2$$

Because $|s_{ij}| \geq \delta > 0$ and $2 \leq n_q \leq N$, we have[6]

$$\begin{aligned}\sigma_j^2 &= \frac{1}{4}\sum_{i\neq j}\left[1-\text{erf}^2\left(\frac{s_{ij}}{2\sigma_s}\right)\right] = \frac{1}{4}\sum_{i\neq j}\left[1-\text{erf}^2\left(\frac{|s_{ij}|}{2\sigma_s}\right)\right] \\ &\leq \frac{n_q-1}{4}\left[1-\text{erf}^2\left(\frac{\delta}{2\sigma_s}\right)\right] \leq \frac{N-1}{4}\left[1-\text{erf}^2\left(\frac{\delta}{2\sigma_s}\right)\right]\end{aligned}$$

On the other hand,

$$\begin{aligned}&\frac{N-1}{4}\left[1-\text{erf}^2\left(\frac{\delta}{2\sigma_s}\right)\right] < \frac{1}{5} \Rightarrow \sigma_j^2 < \frac{1}{5} \\ &\frac{N-1}{4}\left[1-\text{erf}^2\left(\frac{\delta}{2\sigma_s}\right)\right] < \frac{1}{5} \Rightarrow \left[1-\text{erf}^2\left(\frac{\delta}{2\sigma_s}\right)\right] < \frac{4}{5(N-1)} < \frac{1}{N-1} \\ &\qquad\qquad \Rightarrow \text{erf}\left(\frac{\delta}{2\sigma_s}\right) > \sqrt{\frac{N-2}{N-1}} \\ &\qquad\qquad \Rightarrow \sigma_s < \frac{\delta}{2\text{erf}^{-1}\left(\sqrt{\frac{N-2}{N-1}}\right)}\end{aligned}$$

Therefore, when $\sigma_s < \frac{\delta}{2\text{erf}^{-1}\left(\sqrt{\frac{N-2}{N-1}}\right)}$

$$|E[D(\tilde{R}_j)] - D(E[\tilde{R}_j])| \leq \sqrt{\frac{\varepsilon_3(\sigma_s)}{1-5\varepsilon_3(\sigma_s)}} + 5\varepsilon_3(\sigma_s)$$

where $\varepsilon_3(\sigma_s) = \left[1-\text{erf}^2\left(\frac{\delta}{2\sigma_s}\right)\right]$ is a strict increasing function of σ_s. ■

From **Theorem 2**, we can see that when the parameter σ_s is small, the bound on |NDCG − SoftNDCG| will be an increasing function of σ_s. When σ_s approaches zero[7], SoftNDCG will approximate NDCG with any given accuracy, and thus the ranking function learned by maximizing SoftNDCG will have very similar test performance to that of the function learned by "directly" optimizing NDCG.

REFERENCES

[1] M. Taylor, J. Guiver, S. Robertson, and T. Minka. Softrank: optimizing non-smooth rank metrics. In *WSDM '08: Proceedings of the international conference on Web search and web data mining*, pages 77–86, New York, NY, USA, 2008. ACM.

[2] Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 271–278, New York, NY, USA, 2007. ACM.

[6]When $N = 1$, it can be proved that $E[D(\tilde{R}_j)] - D(E[\tilde{R}_j]) = 0$. However, it is not practical to assume $N = 1$, so we ignore the proof here.

[7]Note that in practice one can hardly set σ_s to be very small. Otherwise the optimization process will not be robust.