A General Approximation Framework for Direct Optimization of Information Retrieval Measures

Tao Qin, Tie-Yan Liu, Hang Li

October, 2008

Abstract

Recently direct optimization of information retrieval (IR) measures becomes a new trend in learning to rank. Several methods have been proposed and the effectiveness of them has also been empirically verified. However, theoretical justification to the algorithms was not sufficient and there were many open problems remaining. In this paper, we theoretically justify the approach of directly optimizing IR measures, and further propose a new general framework for this approach, which enjoys several theoretical advantages. The general framework, which can be used to optimize most IR measures, addresses the task by approximating the IR measures and optimizing the approximated surrogate functions. Theoretical analysis shows that a high approximation accuracy can be achieved by the approach. We take average precision (AP) and normalized discounted cumulative gains (NDCG) as examples to demonstrate how to realize the proposed framework. Experiments on benchmark datasets show that our approach is very effective when compared to existing methods. The empirical results also agree well with the theoretical results obtained in the paper.

1 Introduction

This paper is about direct optimization of IR measures in learning to rank, which is considered as one of the most important directions for the area [24]. Several methods have been developed and they can be grouped into two categories. The methods in the first category introduce upper bounds of IR measures and try to optimize the upper bounds as surrogate objective functions [23, 25, 7, 6]. The methods in the other category approximate IR measures using some smooth functions and conduct optimization on the surrogate objective functions [17, 10]. For other methods, please refer to [3, 8].

Previous studies showed that the approach of directly optimizing IR measures can achieve high performances when compared to the other approaches [23, 25, 7, 24, 17]. However, theoretical analysis was not sufficiently conducted to build solid grounds for the proposed methods.

First, it seems natural to take the direct optimization approach, but theoretical justification to it was not sufficiently provided.

Second, the relationships between the surrogate functions and the corresponding IR measures have not been sufficiently studied. This is a critical issue, because it is necessary to know whether optimizing the surrogate functions can indeed optimize the corresponding IR measures.

Third, some of the proposed surrogate functions are not easy to optimize. Existing methods have to employ complicated techniques in the optimization. For example, both SVM^{map} [25] and SVM^{ndcg} [7] use Structured SVM [19] to optimize the surrogate objective functions (i.e., AP and NDCG respectively). However, the optimization technologies are measure-specific, and thus it is not trivial how to extend them to new measures.

In this work, we propose a general direct optimization framework, which can effectively address the aforementioned three problems.

- We first investigate the theoretical foundation of the direct optimization approach. Based on the consistency theory in statistical learning, we show that when an IR measure is bounded and the function class is not very complex, directly optimizing the IR measure on a large training set can guarantee a high test performance in terms of the same IR measure. By further applying the generalization theory, we prove that under certain conditions, no other approach can outperform the approach of directly optimizing IR measures in the large sample limit.

- We then propose the general framework, which can accurately approximate any position-based IR measure, and then transform the optimization of an IR measure to that of an approximated surrogate function. The key idea is as follows. The difficulty in directly optimizing IR measures lies in that the measures are position based, and thus non-continuous and non-differentiable with respect to the score outputted by the ranking function. If we can accurately approximate the positions of documents by a continuous and differentiable function of the scores of the documents, then we will be able to approximate any position based IR measure. We give theoretical analysis which demonstrates that highly accurate approximation of a position based IR measure can be obtained and thus high test performance in ranking can be achieved.

- Taking AP and NDCG as examples, we show that it is easy to derive learning algorithms (ApproxAP and ApproxNDCG) to optimize the surrogate functions in the framework. Experimental results show that the derived algorithms can outperform existing algorithms.

The contributions of this work are as follows. We provide a theoretical justification to the direct optimization approach. We propose a general framework for direct optimization, which is applicable to any position based IR measure, theoretically justifiable, and empirically effective. Two example algorithms for optimizing AP and NDCG have also been provided.

The remainder of this paper is as follows. We start with a review on existing methods in Section 2, and then give theoretical justification to the direct optimization approach in Section 3. Section 4 sets up a general framework to approximate and optimize IR measures, and shows two examples of using this framework. Theoretical analysis about the framework is given in Section 5. Experimental results are presented in Section 6. We conclude the paper and discuss future directions in the last section.

2 Related Work

2.1 IR Measures

To evaluate the effectiveness of a ranking model, IR measures such as Precision, AP (Average Precision) [1], NDCG (Normalized Discounted Cumulative Gain) [13] and MRR (Mean Reciprocal Rank) [21] are being used. Here we review some of them.

Precision@k is a measure for evaluating top k positions of a ranked list using two levels (relevant and irrelevant) of relevance judgement:

$$\text{Pre@}k = \frac{1}{k}\sum_{j=1}^{k} r_j, \tag{1}$$

where k denotes the truncation position and

$$r_j = \begin{cases} 1 & \text{if document in } j\text{-th position is relevant,} \\ 0 & \text{otherwise,} \end{cases}$$

AP, another measure using two levels of relevance judgement, is defined on the basis of Precision:

$$\text{AP} = \frac{1}{|D_+|}\sum_{j} r_j \times \text{Pre@}j, \tag{2}$$

where $|D_+|$ denotes the number of relevant documents with respect to the query. Given a ranked list for a query, we can compute an AP for this query. Then MAP is defined as the mean of AP over a set of queries.

NDCG@k is a measure for evaluating top k positions of a ranked list using multiple levels (labels) of relevance judgment. It is defined as

$$\text{NDCG@}k = N_k^{-1}\sum_{j=1}^{k} g(r_j)d(j), \tag{3}$$

where k is the same as that in Eq (1), N_k denotes the maximum[1] of $\sum_{j=1}^{k} g(r_j)d(j)$, r_j denotes the relevance level of the document ranked at j-th position, $g(r_j)$ denotes a gain function, e.g., $g(r_j) = 2^{r_j} - 1$, and $d(j)$ denotes a discount function, e.g., $d(j) = 1/\log_2(1 + j)$.

[1] The maximum is obtained when the documents are ranked in the perfect order.

With the above specific definitions of the gain function and the discount function, NDCG@k can be reformulated as

$$\text{NDCG@}k = N_k^{-1} \sum_{j=1}^{k} \frac{2^{r_j} - 1}{\log_2(1+j)}. \tag{4}$$

If considering all the n documents for a query, we get NDCG@n, which is called as NDCG for short in this paper in the case without confusion:

$$\text{NDCG} = \text{NDCG@}n = N_n^{-1} \sum_{j=1}^{n} \frac{2^{r_j} - 1}{\log_2(1+j)}. \tag{5}$$

2.2 Learning to Rank

Learning to rank is aimed at constructing a ranking function f with training data consisting of queries and their associated documents. The function is then used in ranking, specifically, to assign a score to each document associated with a query, to sort the documents in the descending order of the scores, and to generate the final ranking list of documents for the query.

One approach in previous work takes document pairs as instances and reduces the problem of ranking to that of classification on the orders of document pairs. It then applies existing classification techniques to ranking. The methods include Ranking SVM [11, 14], RankBoost [9], RankNet [4]. See also [26, 18].

Another approach regards ranking lists as instances and conducts learning on the lists of documents. For instance, Cao *et al* proposed using a probabilistic model in the ranking learning and employing a listwise ranking algorithm called ListNet [5]. In their recent work [22], they further studied the properties of the related algorithms and derived a new algorithm based on Maximum Likelihood called ListMLE. See also [16, 3].

2.3 Direct Optimization of IR Measures

In addition to the learning to rank methods described above, people have also studied how to learn a ranking function by directly optimizing an IR measure. This new approach seems more straightforward and appealing, because what is used in evaluation is exactly an IR measure.

There are two major categories of algorithms for direct optimization of IR measures. One group of algorithms tries to optimize objective functions that are bounds of the IR measures. For example, SVM^{map} [25] minimizes a hinge loss function, which bounds 1-AP from above. SVM^{ndcg} [7, 6] minimizes a hinge loss function, which bounds 1-NDCG from above. AdaRank [23] minimizes an exponential loss function which can upper bound either 1-AP or 1-NDCG. Another group of algorithms manages to smooth the IR measures with easy-to-optimize functions. For example, SoftRank [17, 10] smooths NDCG by introducing randomness into the relevance scores of documents.

The effectiveness of these algorithms have been empirically verified. However, as mentioned in the introduction, theoretical analysis on the algorithms was not sufficiently provided. In this paper, we theoretically justify this approach and further propose a novel framework for direct optimization of IR measures.

3 Theoretical Justification

In this section, we will give a theoretical justification to the approach of directly optimizing IR measures, on the basis of the consistency theory of empirical learning process and the generalization theory in statistical machine learning. That is, if an algorithm can really directly optimize an IR measure on the training data, then the ranking function learned by the algorithm will be one of the best ranking functions one can ever obtain, in terms of the expected test performance defined by the same IR measure.

3.1 Training Performance vs. Testing Performance

Suppose that $\{q_i, i = 1, 2, \cdots, m\}$ represents m training queries and q represents a test query, sampled from the entire query space, according to an unknown probability distribution $P(q)$. We use $M(q, f)$ to denote the performance of ranking function $f \in \mathscr{F}$ with regards to query q in terms of IR measure M. Then $M(f)$ and $M_m(f)$ defined below represent the expected test performance and the empirical training performance of the ranking function f in terms of IR measure M:

$$M(f) = \int M(q, f) dP(q) \tag{6}$$

$$M_m(f) = \frac{1}{m} \sum_{i=1}^{m} M(q_i, f) \tag{7}$$

By applying Theorem 3.4 in [20], which is about the consistence of any empirical learning process, we can obtain the following theorem on the consistency of empirical learning-to-rank process.

Theorem 1. *If the ranking function space $\mathscr{F}$ is not complex*[2]*, and the IR measure $M(q, f)$ is uniformly bounded over the function space $\mathscr{F}$, then the training performance $M_m(f)$ of a learning to rank algorithm uniformly converges to the test performance $M(f)$ of it.*

$$P\left\{\sup_{f \in \mathscr{F}} |M(f) - M_m(f)| > \varepsilon\right\} \overset{m \to \infty}{\longrightarrow} 0 \tag{8}$$

[2] The complexity of a function space has its strict definition, which is beyond the scope of this paper. Please refer to Section 3.8 of [20] for more details. For example, a space containing a finite number of functions is not complex.

Since most IR measures including NDCG, MAP and Precision take values from $[0, 1]$, the corresponding $M(q, f)$ is uniformly bounded for any ranking function $f \in \mathscr{F}$. Theorem 1 implies that under certain conditions, the training performance of a ranking function will be very close to the test performance of it, when the number of training queries becomes large (i.e., $|M(f) - M_m(f)| \overset{m\to\infty}{\longrightarrow} 0$).

It is easy to understand that if an algorithm can directly optimize an IR measure on the training set, then the learned ranking function will have a high performance on the training set. Theorem 1 pushes it further by saying that the ranking function is very likely to have a high performance on test set as well, when the training set is large enough. This gives a theoretical justification to the approach of directly optimizing IR measures in learning to rank.

3.2 Direct Optimization vs. Other Methods

We can draw an even stronger conclusion from the generalization theory. That is, when the number of training queries is extremely large, the learned ranking function in direct optimization of IR measures will be the best ranking function that one can ever obtain in terms of the measures.

We use f_m to denote the ranking function in $\mathscr{F}$ with the best training performance in terms of the IR measure M, and f^* to denote the ranking function in $\mathscr{F}$ with the best testing performance, also in terms of M:

$$f_m = \arg\max_{f \in \mathscr{F}} M_m(f) \tag{9}$$

$$f^* = \arg\max_{f \in \mathscr{F}} M(f) \tag{10}$$

Then we have the following theorem based on [2]. The proof can be found in Appendix.

Theorem 2. *The difference between the testing performance of f_m and the testing performance of f^* can be bounded as below,*

$$|M(f_m) - M(f^*)| \leq 2 \sup_{f \in \mathscr{F}} |M(f) - M_m(f)|. \tag{11}$$

Combining the results above Theorem 1 and Theorem 2 yields $|M(f_m) - M(f^*)| \overset{m\to\infty}{\longrightarrow} 0$. Note that $M(f^*)$ is the best test performance one can ever obtain over the entire function space. For the ranking function learned by other methods, Theorem 2 does not necessarily hold. Therefore, it is safe to say that no other learning to rank algorithms can perform better than the approach of directly optimizing IR measures in the large sample limit.

3.3 Remarks

Theorem 1 and Theorem 2 hold only when the conditions in them are met.

- For some unbounded IR measures, such as DCG, there is no guarantee that the same conclusion holds as in Theorem 1. As a result, it is not clear whether high training performance can result in high testing performance in terms of such measures.

- Note that the two theorems hold only in the large sample limit. In practice, the number of training data is always limited. Theoretically it is difficult to analyze how a direct optimization method would perform in such situations. Note that this is the case for any learning algorithm. Therefore, we need to compare the performances of learning algorithms empirically.

- As discussed in Section 1, existing direct optimization methods try to optimize surrogate objective functions but not IR measures. In many cases the relationships between the surrogate functions and the IR measures have not been verified. Thus, it is not clear whether the existing direct optimization algorithms can outperform other methods in the large sample limit.

4 General Framework

In this section, we propose a general framework for direct optimization of IR measures. The framework is applicable to any position based IR measure, theoretical justifiable, easy to use, and empirically effective.

In the framework, we take the approach of approximating the IR measures. The framework consists of four steps:

- *Reformulating an IR measure from 'indexing by positions' to 'indexing by documents'.* The newly formulated IR measure then contains a position function and optionally a truncation function. Both functions are non-continuous and non-differentiable.

- *Approximating the position function with a smooth function of ranking scores.*

- *Approximating the truncation function with a smooth function of positions of documents.*

- *Applying an optimization technique to optimize the approximated measure (surrogate function).*

Next, for ease of explanation we take some examples to describe the steps in details. We first give some notations here.

Suppose that $\mathcal{X}$ is a set of documents for a query, and x is an element in $\mathcal{X}$. A ranking function f outputs a score s_x for each x:

$$s_x = f(x; \theta), x \in \mathcal{X}$$

where θ denotes the parameter of f. A ranked list π can be obtained by sorting the documents in descending order of their scores. We use $\pi(x)$ to denote the position of document x in the ranked list π. Given the relevance label $r(x)$ of each document x, an IR measure can be used to evaluate the goodness of π. Note that different f's will generate different π's and thus achieve different ranking performances in terms of the IR measure. The approach of direct optimization is to find an optimal f from a function class $\mathscr{F}$ by directly optimizing the performance on the data in terms of the IR measure. Further, we use $\mathbf{1}\{A\}$ to denote an indicator function:

$$\mathbf{1}\{A\} = \begin{cases} 1, & \text{if } A \text{ is true,} \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

4.1 Measure Reformulation

Most of the IR measures, for example, Precision, AP and NDCG are position based. Specifically, the summations in the definitions of IR measures are taken over positions, as can be seen in Eq. (1), (2), (3) and (4). Unfortunately, the position of a document may change during the training process, which makes the handling of the IR measures difficult. To deal with the problem, we reformulate IR measures using the indices of documents.

When indexed by documents, Precision@k in Eq. (1) can be re-written as

$$\text{Pre@}k = \frac{1}{k}\sum_{x\in\mathcal{X}} r(x)\mathbf{1}\{\pi(x) \le k\}, \tag{13}$$

where $r(x)$ equals 1 for relevant documents and 0 for irrelevant documents, and $\mathbf{1}\{\pi(x) \le k\}$ is a truncation function indicating whether document x is ranked at top k positions.

With documents as indices, AP in Eq. (2) can be re-written as,

$$\text{AP} = \frac{1}{|D_+|}\sum_{y\in\mathcal{X}} r(y) \times Pre@\pi(y). \tag{14}$$

Combining Eq. (13) and Eq. (14) yields

$$\begin{aligned} \text{AP} &= \frac{1}{|D_+|}\sum_{y\in\mathcal{X}} r(y) \times \frac{1}{\pi(y)}\sum_{x\in\mathcal{X}} r(x)\mathbf{1}\{\pi(x) \le \pi(y)\} \\ &= \frac{1}{|D_+|}\sum_{y\in\mathcal{X}}\left(\frac{r(y)}{\pi(y)} + \sum_{x\in\mathcal{X},x\neq y} r(y)r(x)\frac{\mathbf{1}\{\pi(x) < \pi(y)\}}{\pi(y)}\right) \end{aligned} \tag{15}$$

where $\mathbf{1}\{\pi(x) < \pi(y)\}$ is also a truncation function indicating whether document x is ranked before document y.

Similarly, when indexed by documents, Eq. (3) of NDCG@k can be re-written as:

$$\text{NDCG@}k = N_k^{-1}\sum_{x\in\mathcal{X}} \frac{2^{r(x)}-1}{\log_2(1+\pi(x))}\mathbf{1}\{\pi(x) \le k\}. \tag{16}$$

Here $r(x)$ is an integer. For example, $r(x) = 0$ means that document x is irrelevant to the query, and $r(x) = 4$ means that the document is very relevant to the query.

Note that NDCG does not need the truncation function,

$$\text{NDCG} = N_n^{-1} \sum_{x \in \mathcal{X}} \frac{2^{r(x)} - 1}{\log_2(1 + \pi(x))}. \tag{17}$$

The reformulated IR measures (e.g., Eq. (13), (15), (16) and (17)) contain two kinds of functions: position function $\pi(x)$ and truncation functions $\mathbf{1}\{\pi(x) < \pi(y)\}$ and $\mathbf{1}\{\pi(x) \leq k\}$. Both of them are non-continuous and non-differentiable. We will discuss how to approximate them separately in next two subsections.

4.2 Position Function Approximation

The position function can be represented as a function of ranking scores:

$$\pi(x) = 1 + \sum_{y \in \mathcal{X}, y \neq x} \mathbf{1}\{s_{x,y} < 0\}, \tag{18}$$

where $s_{x,y} = s_x - s_y$.

That is, positions can be regarded as outputs of a function of ranking scores. Unfortunately the position function is non-continuous and non-differentiable because the indicator function is so.

We want to approximate the position function to make it easy to handle. A natural way for the approximation is to approximate the indicator function $\mathbf{1}\{s_{x,y} < 0\}$ using a logistic function[3]:

$$\frac{\exp(-\alpha s_{x,y})}{1 + \exp(-\alpha s_{x,y})}, \tag{19}$$

where $\alpha > 0$ is a scaling constant.

Then we can replace $\pi(x)$ as $\hat{\pi}(x)$

$$\hat{\pi}(x) = 1 + \sum_{y \in \mathcal{X}, y \neq x} \frac{\exp(-\alpha s_{x,y})}{1 + \exp(-\alpha s_{x,y})} \tag{20}$$

while $\hat{\pi}(x)$ is a continuous and differentiable function.

Table 1 shows an example of the above position approximation process. We can see that the approximation is very accurate in this case.

[3] Note that the logistic function is a special case of sigmoid functions. In fact, we can use any other sigmoid function for this approximation, such as the ordinary arc-tangent function, the hyperbolic tangent function, and the error function. In this paper, we take the logistic function as an example, and all the derivations and conclusions can be naturally extended to other sigmoid functions.

Table 1: Examples of position approximation

document	s_x	$\pi(x)$	$\hat{\pi}(x)$ ($\alpha = 100$)
x_1	4.20074	2	2.00118
x_2	3.12378	4	4.00000
x_3	4.40918	1	1.00000
x_4	1.55258	5	5.00000
x_5	4.13330	3	2.99882

Now we can get the approximation of NDCG by simply replacing $\pi(x)$ in Eq. (17) with $\hat{\pi}(x)$:

$$\widehat{\mathrm{NDCG}} = N_n^{-1} \sum_{x \in \mathcal{X}} \frac{2^{r(x)} - 1}{\log_2(1 + \hat{\pi}(x))}. \tag{21}$$

4.3 Truncation Function Approximation

As can seen in Section 4.1, some measures have truncation functions in their definitions, such as Precision@k, AP, and NDCG@k. These measures need further approximation on the truncation functions. We will introduce in this subsection how it is done within the proposed framework. Some other measures including NDCG do not have truncation functions; In this case, the techniques introduced below can be skipped.

Due to space limitations, we take AP in the first category as an example to show how to approximate the truncation function and then approximate the measure.

To approximate AP, we need to approximate the truncation function $\mathbf{1}\{\pi(x) < \pi(y)\}$ in Eq. (15). A simple way is to use a logistic function[4]:

$$\frac{\exp(\beta(\hat{\pi}(y) - \hat{\pi}(x)))}{1 + \exp(\beta(\hat{\pi}(y) - \hat{\pi}(x)))},$$

in which $\beta > 0$ is a scaling constant.

Thus, we get the approximation of AP as follows.

$$\widehat{\mathrm{AP}} = \frac{1}{|D_+|} \sum_{y} \left(\frac{r(y)}{\hat{\pi}(y)} + \sum_{x \neq y} \frac{r(y) r(x)}{\hat{\pi}(y)} \frac{\exp(\beta(\hat{\pi}(y) - \hat{\pi}(x)))}{1 + \exp(\beta(\hat{\pi}(y) - \hat{\pi}(x)))} \right). \tag{22}$$

4.4 Surrogate Function Optimization

With the aforementioned approximation technique, the surrogate objective functions (e.g., $\widehat{\mathrm{AP}}$ and $\widehat{\mathrm{NDCG}}$) become continuous and differentiable with respect to the parameter θ in the ranking function, one can choose many optimization algorithms, e.g., the simple gradient method, to maximize them.

[4] Similarly to position approximation, we can also use other sigmoid functions.

Again we take AP and NDCG as examples to show how to perform the optimization, and call the corresponding algorithms ApproxAP and ApproxNDCG respectively. The details about the derivation of gradients of $\widehat{\text{AP}}$ and $\widehat{\text{NDCG}}$ can be found in Appendix B.2 and B.1.

The training process is shown in Algorithm 1. This process will generate T ranking functions with parameters $\theta_1, \theta_2, \cdots, \theta_T$. We usually need a validation set to select the best model for testing.

From the two examples (ApproxAP and ApproxNDCG), we can see that by using the framework, the corresponding surrogate objective function can be easily optimized by many existing optimization techniques, such as gradient methods. Measure specific optimization techniques are no longer needed.

Algorithm 1. ApproxAP (ApproxNDCG)

```
Input:
$m$ training queries, their associated documents and relevance judgments.
Number of iterations $T$;
Learning rate $\eta$.
Training:
Initialize the parameter $\theta_0$ of the ranking function $f(x;\theta)$;
For $t = 1$ to $T$ do
    Set $\theta = \theta_{t-1}$;
    Shuffle the $m$ training queries;
    For $i = 1$ to $m$ do
        Feed $i$-th training query (after shuffle) to the learning system;
        Compute the gradient $\Delta\theta$ of $\widehat{\text{AP}}$ ($\widehat{\text{NDCG}}$) with respect to $\theta$ using Eq. (44) (using Eq. (41));
        Update parameter $\theta = \theta + \eta \times \Delta\theta$;
    End for
    Set $\theta_t = \theta$.
End for
Output:
Parameters of $T$ ranking functions: $\{\theta_1, \theta_2, \cdots, \theta_T\}$.
```

5 Theoretical Analysis

As mentioned in Section 1, the relationships between the surrogate objective functions and the corresponding IR measures are not clear for the previous methods. In contrast, the relation between the approximated surrogate functions within our framework and the IR measures can be well investigated.

5.1 Position Function Approximation

The approximation of positions is a basic component in our framework. In order to approximate an IR measure, we need to approximate positions first; in order to analyze the accuracy of approximation of IR measures, we need to analyze the accuracy of approximation of positions.

Note that if $s_{x,y} = 0$ (i.e., document x and y have the same score), there will be no unique ranked list by sorting. This would bring uncertainty to IR measures. For the sake of clarity, in this paper, we assume that

$$\delta = \min_{x,y\in\mathcal{X},x\neq y} |s_{x,y}| > 0 \tag{23}$$

The following theorem shows that the position approximation in Eq. (20) can achieve very high accuracy. One can refer to the Appendix for the proof of the theorem.

Theorem 3. *Given a document collection $\mathcal{X}$ with n documents in it, for $\forall\alpha > 0$, Eq. (20) can approximate the true position with the following accuracy:*

$$|\hat{\pi}(x) - \pi(x)| < \frac{n-1}{\exp(\delta_x\alpha)+1}, \tag{24}$$

where $\delta_x = \min_{y\in\mathcal{X},y\neq x} |s_{x,y}|$.

This theorem tells us that when δ_x and α are large, the approximation will be very accurate. For example,

$$\lim_{\delta_x\alpha\to\infty} \hat{\pi}(x) = \pi(x).$$

A corollary of Theorem 3 is given below:

Corollary 4. *Given a document collection $\mathcal{X}$ with n documents in it, for $\forall\alpha > 0$, Eq. (20) can approximate the true position with an accuracy as below.*

$$\varepsilon \triangleq \max_{x\in\mathcal{X}} |\hat{\pi}(x) - \pi(x)| < \frac{n-1}{\exp(\delta\alpha)+1} \tag{25}$$

For the example in Table 1, we have an accurate approximation:

$$0.00118 = \varepsilon < \frac{5-1}{\exp(0.06744 * 100)+1} \approx 0.00471.$$

5.2 Measure Approximation

The following theorem quantifies the error in the approximation of MAP. The proof can be found in Appendix.

Theorem 5. *If the error ε of position approximation in Eq. (25) is smaller than 0.5, then we have*

$$|\widehat{AP} - AP| < \frac{1}{1+\exp(\beta(1-2\varepsilon))} \sum_{i=1}^{|D_+|} \frac{1}{i-\varepsilon} + 2\varepsilon \sum_{i=1}^{|D_+|} \frac{1}{i\cdot(i-\varepsilon)}. \tag{26}$$

The theorem indicates that when ε is small and β is large, the approximation of AP can be very accurate. In the extreme case, we have

$$\lim_{\varepsilon\to 0,\beta\to\infty} \widehat{\mathrm{AP}} = \mathrm{AP}.$$

For the example in Table 1, if setting $\beta = 100, |D_+| = 1$, we have $|\widehat{\mathrm{AP}} - \mathrm{AP}| < 0.0024$. That is, the AP approximation is very accurate in this case.

The following theorem quantifies the error in the approximation of NDCG. The proof can be found in Appendix A.4.

Theorem 6. *The approximation error of $\widehat{NDCG}$ can be bounded as*

$$|\widehat{NDCG} - NDCG| < \frac{\varepsilon}{2\ln 2}. \tag{27}$$

This theorem indicates that when ε is small, the approximation of NDCG can be very accurate. In the extreme case, we have

$$\lim_{\varepsilon\to 0} \widehat{\mathrm{NDCG}} = \mathrm{NDCG}.$$

For the example in Table 1, we have $|\widehat{\mathrm{NDCG}} - \mathrm{NDCG}| < \frac{\varepsilon}{2\ln 2} \approx 0.00085$. That is, the NDCG approximation is very accurate in this case.

From these two examples (AP and NDCG), one can see that the surrogate functions in the framework can be very accurate approximations to IR measures.

5.3 Justification of Accurate Approximation

We already show that the surrogate objective function we obtain with the framework will be very close to the original IR measure. One may argue: why do we need accurate approximation? Is there any benefit from the accurate approximation? Actually, such a high accuracy in the approximation is very important for a direct optimization method. The reasons are as follows.

As discussed in Section 3, directly optimizing IR measures will likely lead to a high test performance. One question arises here: after using the surrogate objective function, can we still have the same or similar conclusion?

Here we use $\hat{f}_m$ to indicate the ranking function in $\mathscr{F}$ with the best training performance in terms of the surrogate objective function, $\hat{M}$:

$$\hat{f}_m \quad = \quad \arg\max_{f\in\mathscr{F}} \hat{M}_m(f) \tag{28}$$

Then we have the following theorem. The proof of the theorem is very similar to that of Theorem 2. Due to space limitations, we omit the details here.

Theorem 7. *The difference between the testing performance $\hat{f}_m$ and the testing performance f^* can be bounded as below,*

$$\begin{aligned} &|M(\hat{f}_m) - M(f^*)| \leq \\ &2\sup_{f\in\mathscr{F}} |M(f) - M_m(f)| + 2\sup_{f\in\mathscr{F}} |M_m(f) - \hat{M}_m(f)|. \end{aligned} \tag{29}$$

Note that Theorem 1 implies that

$$\sup_{f\in\mathscr{F}} |M(f) - M_m(f)| \overset{m\to\infty}{\longrightarrow} 0.$$

If we further have

$$\sup_{f\in\mathscr{F}} |M_m(f) - \hat{M}_m(f)| \overset{m\to\infty}{\longrightarrow} 0,$$

then we will attain that

$$|M(\hat{f}_m) - M(f^*)| \overset{m\to\infty}{\longrightarrow} 0.$$

In other words, if the surrogate objective function is very close to the IR measure (i.e., $\sup_{f\in\mathscr{F}} |M_m(f) - \hat{M}_m(f)| \overset{m\to\infty}{\longrightarrow} 0$), then the test performance of the ranking function learned by a method of optimizing the surrogate objective function can also converge to the best test performance one can ever obtain in the large sample limit.

6 Experimental Results

We used LETOR[5] [15] in our experiments, which is a benchmark dataset developed for learning to rank research. We used the TD2003 and TD2004 datasets in LETOR to test ApproxAP and the OHSUMED datset in LETOR to test ApproxNDCG, since the first two datasets contain two-level relevance judgments and the third one contains three-level relevance judgments. We used linear ranking function for ApproxAP and ApproxNDCG since all the baseline algorithms also used linear ranking functions.

6.1 Datasets

The documents in the TD2003 and TD2004 datasets are from the “gov” collection, and the queries in them are from TREC 2003 and 2004 respectively. There are 50 queries in TD2003 and 75 queries in TD2004, with each query associated with about 1,000 documents. The relevance degrees of documents with respect to queries are offered by TREC, on two levels: *relevant* or *not relevant*. There are 44 features extracted for each query-document pair(refer to [15] for details).

The documents in the OHSUMED dataset are from the OHSUMED collection [12], which is a subset of MEDLINE. There are 106 queries, each with a number of associated documents. The relevance degrees of documents with respect to queries are provided, on three levels: *definitely relevant*, *partially relevant*, or *not relevant*. There are in total 16,140 query-document pairs with relevance judgments. Each query-document pair is represented by a 25-dimension feature vector. For the details of the features, please refer to [15].

[5] The data set can be downloaded from http://research.microsoft.com/users/LETOR/.

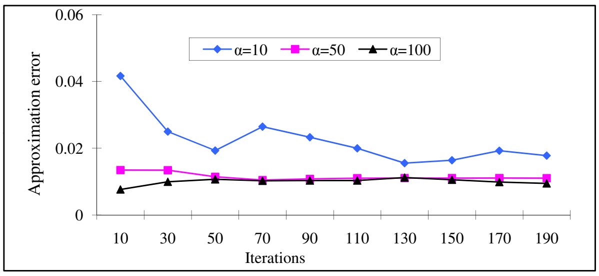

Figure 1: Accuracy of AP approximation on TD2004 dataset. This is training curve over fold 1. The x-axis is the number of iterations. Here we fix $\beta = 10$.

6.2 Accuracy of Approximating IR Measures

We evaluated the accuracy of approximations of AP and NDCG using $\widehat{\text{AP}}$ and $\widehat{\text{NDCG}}$.

As seen in Section 4.3, there are two parameters in $\widehat{\text{AP}}$, α, and β. We first fixed $\beta = 10$ and set three different values for α. Then, we applied the ApproxAP algorithm to the TD2004 dataset with three different parameters. Figure 1 shows the error ρ in the training process defined as

$$\rho = \frac{1}{|Q|} \sum_{q \in Q} |\widehat{\text{AP}}(q) - \text{AP}(q)|,$$

in which $\widehat{\text{AP}}(q)$ and $\text{AP}(q)$ mean the values of $\widehat{\text{AP}}$ and AP respectively over a query q, Q is the training query set, and $|Q|$ is the number of queries in the training set.

We can see that for all the three α values, the approximation accuracy is very high, which is more than 95%. Furthermore, when we increase α, the approximation becomes more accurate: the accuracy is higher than 98% when $\alpha = 100$.

We then fixed $\alpha = 100$ and tried different values of β. Figure 2 shows the error ρ with respect to different β values. As can bee seen, when β increases, the accuracy of the approximation also improves.

Figure 3 shows the error $\rho = \frac{1}{|Q|} \sum_{q \in Q} |\widehat{\text{NDCG}}(q) - \text{NDCG}(q)|$ with regards to different α values. We can observe similar results to those for the approximation of AP.

All these results verify the correctness of the discussions in Section 5.2, and indicate that the approximation of IR measures using our proposed framework can achieve high accuracy.

6.3 Performance of ApproxAP

In the experiments, we empirically set $T = 200, \eta = 0.01$ in Algorithm 1. We adopted the five fold cross validation as suggested in LETOR for both TD2003

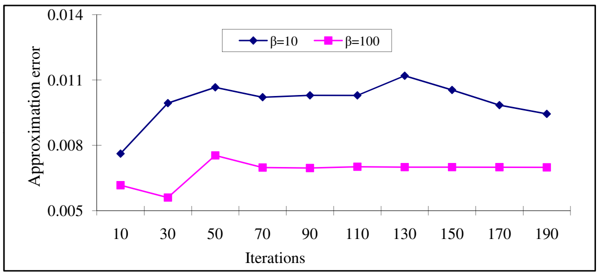

Figure 2: Accuracy of AP approximation on TD2004 dataset. This is training curve over fold 1. The x-axis is the number of iterations. Here we fix $\alpha = 100$.

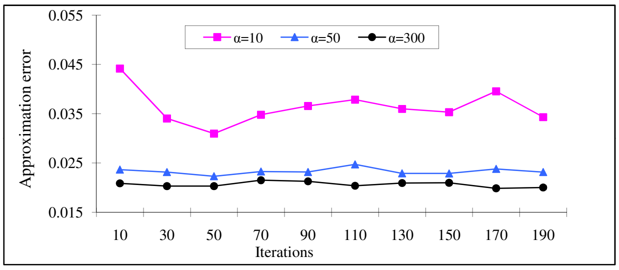

Figure 3: Accuracy of NDCG@n approximation on OHSUMED dataset. This is training curve over fold 1. The x-axis is the number of iterations.

and TD2004 datasets. For each fold, we used the validation set to select hyper parameters α and β in the ApproxAP algorithm. The detailed process is as follows.

(1) We first chose a set of α values $\{10, 20, 50, 100\}$ and a set of β values $\{1, 10, 20, 50, 100\}$.

(2) For each combination of α and β, we learned a set of ranking models (i.e., T models) from the training set. There are $20T$ models in total.

(3) We tested the performance of each model on the validation set and selected the model with the highest MAP[6] as the final model.

(4) We tested the performance of the final model on the test set.

As baselines, we used AdaRank.MAP and SVM^{map}. For AdaRank.MAP, we cite its results from the LETOR website[7]. For SVM^{map}, we used the tool from the authors[8] to produce the result. Table 2 shows average MAP for the three

[6] MAP is the mean of AP of all the queries.

[7] http://research.microsoft.com/users/LETOR/

[8] http://projects.yisongyue.com/svmmap/

Table 2: Ranking accuracy in terms of MAP

Algorithm	TD2003	TD2004
AdaRank.MAP	0.137	0.331
SVM^{map}	0.198	0.304
ApproxAP	0.233	0.350

algorithms on two datasets[9].

As can be seen, ApproxAP performs better than AdaRank.MAP and SVM^{map} on both datasets. For example, ApproxAP gets more than 15% improvement over SVM^{map} on TD2003 and more than 5% improvement over AdaRank.MAP on TD2004. Furthermore, on TD2004, AdaRank.MAP is better than SVM^{map}; On TD2003, SVM^{map} is better than AdaRank.MAP. Since ApproxAP only uses a simple gradient method for the optimization (as compared to the structured SVM and Boosting used in the two baselines), the current result clearly shows the advantage of using the proposed framework for direct optimization, and we foresee that with the use of more advanced optimization techniques, the performance of ApproxAP could be further improved.

6.4 Performance of ApproxNDCG

Similarly to the experiment on ApproxAP, we set $T = 200, \eta = 0.01$ for ApproxNDCG. We used similar strategy to select the hyper parameters α for ApproxNDCG as in ApproxAP. The minor differences are as follows. First, we chose a larger set of α values $\{10, 20, 50, 100, 150, 200, 250, 300\}$. Second, we used NDCG@$n$ for model selection instead of MAP on the validation set.

As baselines, we used AdaRank.NDCG and SoftRank. For AdaRank.NDCG, we cite its result published in the LETOR website[10]. For SoftRank, we used the tool provided by the authors to produce the experimental result.

Figure 4 shows average NDCG at position 1-10 for the three algorithms on OHSUMED. The performances in terms of NDCG@n of SoftRank, AdaRank.NDCG and ApproxNDCG are 0.6680, 0.6589 and 0.6698 respectively. As can be seen, ApproxNDCG achieves higher accuracy than SoftRank, especially for top positions. SoftRank outperforms AdaRank.NDCG. Overall, ApproxNDCG is the best of the three algorithms. This verifies the effectiveness of our proposed framework.

6.5 Discussions

There are many different measures used in the literature of IR, such as AP and NDCG. This makes the task of directly optimizing IR measures slightly com-

[9] The performance of SVM^{map} is different from that reported in [24]. We communicated with the authors; and they agreed on our result for SVM^{map} and found a bug in their experiment.

[10] http://research.microsoft.com/users/LETOR/

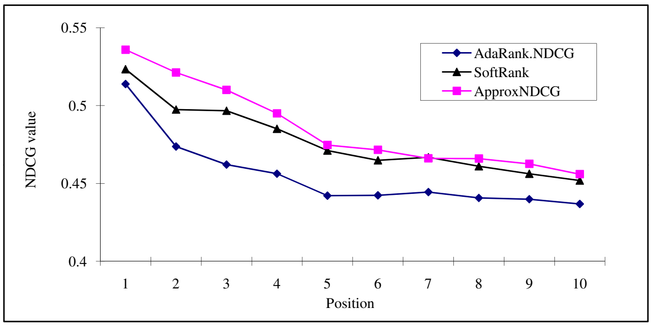

Figure 4: Ranking accuracy in terms of NDCG

plicated. For example, the following questions need to be answered in practice when one performs direct optimization of IR measures.

(a) What is the most suitable measure to optimize?

(b) Will the test performance in terms of measure M for an algorithm directly optimizing M on training set be better than that of an algorithm directly optimizing another measure M'?

Question (a) is still an open question. Generally speaking, the selection of the measure depends heavily on the specific task and data. For example, if the ground truth is given as binary judgment (relevant and irrelevant), then MAP may be a good choice; if the documents are judged with multi-level relevance degrees (e.g., Perfect, Excellent, Good, Fair and Bad), then the use of NDCG may be better.

Question (b) is also an open question. The theoretical discussion in Section 3 gives a sufficient condition for a direct optimization method to perform "almost perfectly" in the large sample limit. However, we do not know whether it is a necessary condition. As a result, it is hard to say in practice whether optimizing M is the best choice when we evaluate the performance of ranking function using M.

Note that IR measures are not independent from each other. If measure M' somewhat covers M , then it is likely that the optimization of M' on the training set can also lead to a high test performance in terms of measure M. This is related to the concept of informativeness in Robertson's presentation at SIGIR 2008 Workshop on Learning to Rank for Information Retrieval[11].

To better understand the problem raised in question (b), we performed the following experiments.

Recall that we used TD2003 and TD2004 datasets to study the performance of ApproxAP. Now we further ran the ApproxNDCG algorithm on these two datasets and compared its test performance in terms of MAP with that of ApproxAP. Similarly, we ran ApproxAP on the OHSUMED dataset and compared

[11] http://research.microsoft.com/users/LR4IR-2008/

Table 3: Ranking accuracy in terms of MAP

Algorithm	TD2003	TD2004
ApproxNDCG	0.238	0.318
ApproxAP	0.233	0.350

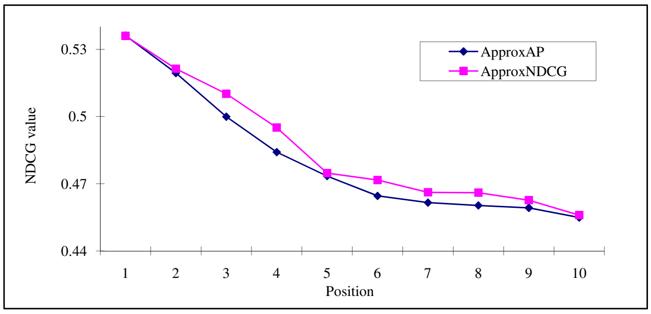

Figure 5: Ranking accuracy in terms of NDCG

its test performance in terms of NDCG with that of ApproxNDCG. As shown in Table 3, when measured with MAP, ApproxAP is better than ApproxNDCG on TD2004 and the two algorithms are comparable on TD2003. As shown in Figure 5, when measured with NDCG, ApproxAP (with NDCG@n 0.6693) is comparable with ApproxNDCG on OHSUMED. These results seem to suggest that the answer to question (b) needs further investigations.

7 Conclusions and Future Work

In this paper, we have provided theoretical justification to the approach of direct optimization of IR measures. Our analysis shows that under certain conditions, the assumption of directly optimizing IR measures is reasonable; the direct optimization approach can be one of the best approaches to learning to rank.

We have set up a general framework to approximate position based IR measures. The key part of the framework is to approximate the positions of documents by their scores. There are several advantages of this framework: 1) the way of approximating position based measures is simple yet general; 2) many existing techniques can be directly applied to the optimization and the optimization process itself is measure independent; 3) it is easy to conduct analysis on the accuracy of the approach and high approximation accuracy can be achieved by setting appropriate parameters.

We have taken AP and NDCG as examples to show how to approximate IR measures within the proposed framework, how to analyze the accuracy of the approximation, and how to derive effective learning algorithms to optimize the approximated functions. Experiments on public benchmark datasets have verified the correctness of the theoretical analysis and have proved the effectiveness of our algorithms.

There are still issues which need to be further studied.

- We have taken AP and NDCG as examples. It is worth considering other measures including Precision, NDCG@k, MRR and Kendall's τ.
- We have used simple gradient methods to optimize the approximated functions. We plan to try other optimization techniques as well.
- We have verified the effectiveness of the proposed algorithms on the LETOR datasets. We will conduct more experiments on larger datasets in the future.
- We have conducted experiments for answering question (b) in Section 6.5, but have not obtained clear conclusions. It is worth more investigations.

8 Acknowledgements

We would like to thank Mike Taylor and Vishwa Vinay for providing the tool of SoftRank and thank Yin He for his help on the experiments of SoftRank. We would also like to thank Jun Xu, Xiubo Geng, Yuting Liu and Yanyan Lan for their discussions on this work and proof-reading of this paper.

References

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval.* Addison Wesley, May 1999.

[2] O. Bousquet, S. Boucheron, and G. Lugosi. Introduction to Statistical Learning Theory. *LECTURE NOTES IN COMPUTER SCIENCE*, pages 169–207, 2004.

[3] C. Burges, R. Ragno, and Q. Le. Learning to rank with nonsmooth cost functions. In *Advances in Neural Information Processing Systems*. MIT Press, 2007.

[4] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 89–96, New York, NY, USA, 2005. ACM Press.

[5] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 129–136, New York, NY, USA, 2007. ACM Press.

[6] S. Chakrabarti, R. Khanna, U. Sawant, and C. Bhattacharyya. Structured learning for non-smooth ranking losses. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 88–96, New York, NY, USA, 2008. ACM.

[7] O. Chapelle, Q. Le, C. NICTA, and A. Smola. Large margin optimization of ranking measures. In *NIPS2007 workshop on Machine Learning for Web Search*, 2007.

[8] H. M. de Almeida, M. A. Gonçalves, M. Cristo, and P. Calado. A combined component approach for finding collection-adapted ranking functions based on genetic programming. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 399–406, New York, NY, USA, 2007. ACM.

[9] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.*, 4:933–969, 2003.

[10] J. Guiver and E. Snelson. Learning to rank with softrank and gaussian processes. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 259–266, New York, NY, USA, 2008. ACM.

[11] R. Herbrich, T. Graepel, and K. Obermayer. Support vector learning for ordinal regression. In *ICANN1999*, pages 97–102, 1999.

[12] W. Hersh, C. Buckley, T. J. Leone, and D. Hickam. Ohsumed: an interactive retrieval evaluation and new large test collection for research. In *SIGIR '94*, pages 192–201, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[13] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002.

[14] T. Joachims. Optimizing search engines using clickthrough data. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, New York, NY, USA, 2002. ACM Press.

[15] T.-Y. Liu, J. Xu, T. Qin, W.-Y. 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*, 2007.

[16] 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 & Management*, 2008.

[17] 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.

[18] M.-F. Tsai, T.-Y. Liu, T. Qin, H.-H. Chen, and W.-Y. Ma. Frank: a ranking method with fidelity loss. In *SIGIR '07*, pages 383–390, New York, NY, USA, 2007. ACM Press.

[19] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, New York, NY, USA, 2004. ACM.

[20] V. Vapnik. *Statistical learning theory.* John Wiley & Sons.

[21] E. Voorhees. The trec-8 question answering track report, 1999.

[22] F. Xia, T.-Y. Liu, J. Wang, W. Zhang, and H. Li. Listwise approach to learning to rank: theory and algorithm. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 1192–1199, New York, NY, USA, 2008. ACM.

[23] J. Xu and H. Li. Adarank: a boosting algorithm for information retrieval. In *SIGIR '07*, pages 391–398, New York, NY, USA, 2007. ACM Press.

[24] J. Xu, T.-Y. Liu, M. Lu, H. Li, and W.-Y. Ma. Directly optimizing evaluation measures in learning to rank. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 107–114, New York, NY, USA, 2008. ACM.

[25] Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. In *SIGIR '07*, pages 271–278, New York, NY, USA, 2007. ACM Press.

[26] Z. Zheng, K. Chen, G. Sun, and H. Zha. A regression framework for learning ranking functions using relative relevance judgments. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 287–294, New York, NY, USA, 2007. ACM.

A Approximation Accuracy Analysis

In the appendix, we give proofs of the major theorems in this paper.

A.1 Proof of Theorem 2

Proof. According to the definitions in Eq. (**??**) and (**??**), we have

$$M(f^*) \geq M(f_m),$$

and

$$M_m(f^*) \leq M_m(f_m).$$

Hence,

$$\begin{aligned} & |M(f_m) - M(f^*)| = -M(f_m) + M(f^*) \\ = \;& M_m(f_m) - M(f_m) + M(f^*) - M_m(f_m) \\ \leq \;& M_m(f_m) - M(f_m) + M(f^*) - M_m(f^*) \\ \leq \;& |M_m(f_m) - M(f_m)| + |M(f^*) - M_m(f^*)| \\ \leq \;& 2 \sup_{f \in \mathscr{F}} |M(f) - M_m(f)| \end{aligned}$$

□

A.2 Proof of Theorem 3

Proof. Note that

$$\begin{aligned} |\hat{\pi}(x) - \pi(x)| &= \textstyle\sum_{y \in X, y \neq x} \left(\frac{\exp(-\alpha s_{x,y})}{1+\exp(-\alpha s_{x,y})} - \mathbf{1}\{s_{x,y} < 0\} \right) \\ &\leq \textstyle\sum_{y \in X, y \neq x} \left| \frac{\exp(-\alpha s_{x,y})}{1+\exp(-\alpha s_{x,y})} - \mathbf{1}\{s_{x,y} < 0\} \right|. \end{aligned} \tag{30}$$

If we can prove that for any document $y \in \mathcal{X}$,

$$\left| \frac{\exp(-\alpha s_{x,y})}{1 + \exp(-\alpha s_{x,y})} - \mathbf{1}\{s_{x,y} < 0\} \right| < \frac{1}{\exp(\delta_x \alpha) + 1}, \tag{31}$$

then we can have

$$|\hat{\pi}(x) - \pi(x)| < \sum_{y \in X, y \neq x} \frac{1}{\exp(\delta_x \alpha) + 1} = \frac{n-1}{\exp(\delta_x \alpha) + 1}. \tag{32}$$

Now we prove the inequality (31). We consider $s_{x,y} > 0$ and $s_{x,y} < 0$ separately.

- For $s_{x,y} > 0$, from Eq. (23) we have

  $$1 + \exp(\alpha s_{x,y}) > 1 + \exp(\delta_x \alpha).$$

  Then,

  $$\frac{\exp(-\alpha s_{x,y})}{1 + \exp(-\alpha s_{x,y})} = \frac{1}{1 + \exp(\alpha s_{x,y})} < \frac{1}{1 + \exp(\delta_x \alpha)}.$$

  Note that $\mathbf{1}\{s_{x,y} < 0\} = 0$ when $s_{x,y} > 0$. Hence, for $s_{x,y} > 0$,

  $$\left| \frac{\exp(-\alpha s_{x,y})}{1 + \exp(-\alpha s_{x,y})} - \mathbf{1}\{s_{x,y} < 0\} \right| < \frac{1}{1 + \exp(\delta_x \alpha)}.$$

- For $s_{x,y} < 0$, from Eq. (23) we have

  $$1 + \exp(-\alpha s_{x,y}) > 1 + \exp(\delta_x \alpha).$$

  Note that $\mathbf{1}\{s_{x,y} < 0\} = 1$ when $s_{x,y} < 0$. Hence, for $s_{x,y} < 0$,

  $$\begin{aligned} &\left| \frac{\exp(-\alpha s_{x,y})}{1 + \exp(-\alpha s_{x,y})} - \mathbf{1}\{s_{x,y} < 0\} \right| \\ = &\frac{1}{1 + \exp(-\alpha s_{x,y})} < \frac{1}{1 + \exp(\delta_x \alpha)}. \end{aligned}$$

Combining the two cases we end up with Eq. (31). According to Eq. (32), Theorem 3 is correct. □

A.3 Proof of Theorem 5

We prove Theorem 5 about the accuracy of precision approximation.

Proof. For simplicity, we denote

$$\hat{\mathbf{1}}\{\pi(x) < \pi(y)\} = \frac{\exp(\beta(\hat{\pi}(y) - \hat{\pi}(x)))}{1 + \exp(\beta(\hat{\pi}(y) - \hat{\pi}(x)))}.$$

From Eq. (15) and (22), we have

$$\begin{aligned} |\widehat{\mathrm{MAP}} - \mathrm{MAP}| = &\left| \sum_y \frac{r(y)}{|D_+|} (\frac{1}{\hat{\pi}(y)} - \frac{1}{\pi(y)}) \right. \\ + &\left. \sum_y \sum_{x \neq y} \frac{r(y) r(x)}{|D_+|} \left(\frac{\hat{\mathbf{1}}\{\pi(x) < \pi(y)\}}{\hat{\pi}(y)} - \frac{\mathbf{1}\{\pi(x) < \pi(y)\}}{\pi(y)} \right) \right| \\ \leq &\sum_y \sum_{x \neq y} \frac{r(y) r(x)}{|D_+|} \left| \frac{\hat{\mathbf{1}}\{\pi(x) < \pi(y)\}}{\hat{\pi}(y)} - \frac{\mathbf{1}\{\pi(x) < \pi(y)\}}{\pi(y)} \right| \\ + &\sum_y \frac{r(y)}{|D_+|} \left| \frac{1}{\hat{\pi}(y)} - \frac{1}{\pi(y)} \right| \end{aligned} \tag{33}$$

Now we consider $\left| \frac{\hat{\mathbf{1}}\{\pi(x) < \pi(y)\}}{\hat{\pi}(y)} - \frac{\mathbf{1}\{\pi(x) < \pi(y)\}}{\pi(y)} \right|$ and $\left| \frac{1}{\hat{\pi}(y)} - \frac{1}{\pi(y)} \right|$ respectively.

$$\begin{aligned}
&\left|\frac{\hat{\mathbf{1}}\{\pi(x)<\pi(y)\}}{\hat{\pi}(y)}-\frac{\mathbf{1}\{\pi(x)<\pi(y)\}}{\pi(y)}\right| \\
=\ &\left|\frac{\pi(y)\hat{\mathbf{1}}\{\pi(x)<\pi(y)\}-\hat{\pi}(y)\mathbf{1}\{\pi(x)<\pi(y)\}}{\pi(y)\hat{\pi}(y)}\right| \\
=\ &\left|\frac{\pi(y)(\hat{\mathbf{1}}\{\pi(x)<\pi(y)\}-\mathbf{1}\{\pi(x)<\pi(y)\})+(\pi(y)-\hat{\pi}(y))\mathbf{1}\{\pi(x)<\pi(y)\}}{\pi(y)\hat{\pi}(y)}\right| \\
\leq\ &\left|\frac{\pi(y)(\hat{\mathbf{1}}\{\pi(x)<\pi(y)\}-\mathbf{1}\{\pi(x)<\pi(y)\})}{\pi(y)\hat{\pi}(y)}\right| \\
+\ &\left|\frac{(\pi(y)-\hat{\pi}(y))\mathbf{1}\{\pi(x)<\pi(y)\}}{\pi(y)\hat{\pi}(y)}\right| \\
\leq\ &\left|\frac{\hat{\mathbf{1}}\{\pi(x)<\pi(y)\}-\mathbf{1}\{\pi(x)<\pi(y)\}}{\hat{\pi}(y)}\right|+\frac{\varepsilon}{\hat{\pi}(y)\pi(y)} \qquad (34)
\end{aligned}$$

Similar to the derivation of Eq. (31), we can get

$$\left|\hat{\mathbf{1}}\{\pi(x)<\pi(y)\}-\mathbf{1}\{\pi(x)<\pi(y)\}\right|<\frac{1}{1+\exp(\beta(1-2\varepsilon))}. \qquad (35)$$

Combining Eq. (34) and (35), we get

$$\begin{aligned}
&\sum_y\sum_{x\neq y}\frac{r(y)r(x)}{|D_+|}\left|\frac{\hat{\mathbf{1}}\{\pi(x)<\pi(y)\}}{\hat{\pi}(y)}-\frac{\mathbf{1}\{\pi(x)<\pi(y)\}}{\pi(y)}\right| \\
\leq\ &\sum_y\sum_{x\neq y}\frac{r(y)r(x)}{|D_+|}\left(\frac{1}{\hat{\pi}(y)(1+\exp(\beta(1-2\varepsilon)))}+\frac{\varepsilon}{\hat{\pi}(y)\pi(y)}\right) \\
<\ &\sum_y r(y)\left(\frac{1}{\hat{\pi}(y)(1+\exp(\beta(1-2\varepsilon)))}+\frac{\varepsilon}{\hat{\pi}(y)\pi(y)}\right) \\
<\ &\frac{1}{1+\exp(\beta(1-2\varepsilon))}\sum_{i=1}^{|D_+|}\frac{1}{i-\varepsilon}+\varepsilon\sum_{i=1}^{|D_+|}\frac{1}{i\cdot(i-\varepsilon)} \qquad (36)
\end{aligned}$$

$$\left|\frac{1}{\hat{\pi}(y)}-\frac{1}{\pi(y)}\right|=\left|\frac{\pi(y)-\hat{\pi}(y)}{\hat{\pi}(y)\pi(y)}\right|<\frac{\varepsilon}{\hat{\pi}(y)\pi(y)} \qquad (37)$$

Then

$$\begin{aligned}
\sum_y\frac{r(y)}{|D_+|}\left|\frac{1}{\hat{\pi}(y)}-\frac{1}{\pi(y)}\right| &< \sum_y\frac{r(y)}{|D_+|}\frac{\varepsilon}{\hat{\pi}(y)\pi(y)} \\
&\leq \frac{\varepsilon}{|D_+|}\sum_{i=1}^{|D_+|}\frac{1}{i\cdot(i-\varepsilon)} \qquad (38)
\end{aligned}$$

Substitute Eq. (36) and (38) into Eq. (33), we get

$$|\widehat{\mathrm{MAP}}-\mathrm{MAP}|\frac{1}{1+\exp(\beta(1-2\varepsilon))}\sum_{i=1}^{|D_+|}\frac{1}{i-\varepsilon}+\varepsilon\frac{1+|D_+|}{|D_+|}\sum_{i=1}^{|D_+|}\frac{1}{i\cdot(i-\varepsilon)}.$$

Since $|D_+|\geq 1$, hence

$$|\widehat{\mathrm{MAP}}-\mathrm{MAP}|<\frac{1}{1+\exp(\beta(1-2\varepsilon))}\sum_{i=1}^{|D_+|}\frac{1}{i-\varepsilon}+2\varepsilon\sum_{i=1}^{|D_+|}\frac{1}{i\cdot(i-\varepsilon)}.$$

□

A.4 Proof of Theorem 6

Proof. From Eq. (17) and (21), we obtain

$$\begin{aligned}&|\widehat{\mathrm{NDCG}}-\mathrm{NDCG}|\\ &\leq N_n^{-1}\textstyle\sum_{x\in\mathcal{X}}\frac{2^{r(x)}-1}{\log_2(1+\pi(x))}\left|\frac{\log_2(1+\hat{\pi}(x))-\log_2(1+\pi(x))}{\log_2(1+\hat{\pi}(x))}\right|.\end{aligned}\tag{39}$$

Since $\frac{\partial\log_2(1+t)}{\partial t}=\frac{1}{(1+t)\ln 2}$ and $\pi(x)\geq 1$, $\hat{\pi}(x)\geq 1$, we have

$$|\log_2(1+\hat{\pi}(x))-\log_2(1+\pi(x))|<\frac{1}{2\ln 2}|\hat{\pi}(x)-\pi(x)|\leq\frac{\varepsilon}{2\ln 2}.$$

Considering that $\log_2(1+\hat{\pi}(x))>1$, we have

$$\left|\frac{\log_2(1+\hat{\pi}(x))-\log_2(1+\pi(x))}{\log_2(1+\hat{\pi}(x))}\right|<\frac{\varepsilon}{2\ln 2}.\tag{40}$$

Then Eq. (39) becomes

$$\begin{aligned}|\widehat{\mathrm{NDCG}}-\mathrm{NDCG}| &< N_n^{-1}\sum_{x\in\mathcal{X}}\frac{2^{r(x)}-1}{\log_2(1+\pi(x))}\frac{\varepsilon}{2\ln 2}\\ &= \frac{\varepsilon}{2\ln 2}\mathrm{NDCG}.\end{aligned}$$

According to the definition of NDCG, we always have NDCG ≤ 1. Hence,

$$|\widehat{\mathrm{NDCG}}-\mathrm{NDCG}|<\frac{\varepsilon}{2\ln 2}.$$

□

B Gradient Derivation

B.1 Gradient of ApproxNDCG

We show how to derive the gradient for ApproxNDCG.

According to the chain rule, we obtain

$$\Delta\theta = \frac{\partial \widehat{\mathrm{NDCG}}}{\partial\theta} = N_n^{-1} \sum_x \frac{\partial \frac{2^{r(x)}-1}{\log_2(1+\hat{\pi}(x))}}{\partial\hat{\pi}(x)} \frac{\partial\hat{\pi}(x)}{\partial\theta}. \tag{41}$$

Further,

$$\begin{aligned} \tfrac{\partial\hat{\pi}(x)}{\partial\theta} &= -\alpha \textstyle\sum_{y\neq x} \tfrac{\exp(\alpha s_{xy})}{(1+\exp(\alpha s_{xy}))^2} \tfrac{\partial s_{xy}}{\partial\theta} \\ &= -\alpha \textstyle\sum_{y\neq x} \tfrac{\exp(\alpha s_{xy})}{(1+\exp(\alpha s_{xy}))^2} \left(\tfrac{\partial f(x;\theta)}{\partial\theta} - \tfrac{\partial f(y;\theta)}{\partial\theta}\right) \end{aligned} \tag{42}$$

$$\frac{\partial \frac{2^{r(x)}-1}{\log_2(1+\hat{\pi}(x))}}{\partial\hat{\pi}(x)} = -\frac{2^{r(x)}-1}{(\log_2(1+\hat{\pi}(x)))^2} \frac{1}{(1+\hat{\pi}(x))\ln 2} \tag{43}$$

Substituting Eq. (42) and (43) into (41), we get the gradient for ApproxNDCG.

Note that $\frac{\partial f(x;\theta)}{\partial\theta}$ in Eq. (42) depends on the specific form of the ranking function f. For example, for linear function, we have $\frac{\partial f(x;\theta)}{\partial\theta} = x$.

B.2 Gradient of ApproxAP

We next show how to derive the gradient for ApproxAP.

According to the chain rule, we obtain

$$\frac{\partial\widehat{\mathrm{AP}}}{\partial\theta} = \frac{-1}{|D_+|} \sum_y \frac{r(y)}{\hat{\pi}^2(y)} \frac{\partial\hat{\pi}(y)}{\partial\theta} + \frac{1}{|D_+|} \sum_y \sum_{x\neq y} r(y)r(x) \frac{\partial J(\theta)}{\partial\theta}, \tag{44}$$

where

$$J(\theta) = \frac{1}{\hat{\pi}(y)} \frac{\exp(\beta(\hat{\pi}(y)-\hat{\pi}(x)))}{1+\exp(\beta(\hat{\pi}(y)-\hat{\pi}(x)))}.$$

Again by the chain rule, we have

$$\frac{\partial J(\theta)}{\partial\theta} = \frac{\partial J(\theta)}{\partial\hat{\pi}(y)} \frac{\partial\hat{\pi}(y)}{\partial\theta} + \frac{\partial J(\theta)}{\partial\hat{\pi}(x)} \frac{\partial\hat{\pi}(x)}{\partial\theta} \tag{45}$$

Now we consider $\frac{\partial J(\theta)}{\partial\hat{\pi}(x)}$ and $\frac{\partial J(\theta)}{\partial\hat{\pi}(y)}$.

$$\frac{\partial J(\theta)}{\partial\hat{\pi}(x)} = \frac{-1}{\hat{\pi}(y)} \frac{\beta\exp(\beta(\hat{\pi}(x)-\hat{\pi}(y)))}{(1+\exp(\beta(\hat{\pi}(x)-\hat{\pi}(y))))^2}. \tag{46}$$

$$\begin{aligned} \frac{\partial J(\theta)}{\partial\hat{\pi}(y)} &= \frac{-1}{\hat{\pi}^2(y)} \frac{1}{1+\exp(\beta(\hat{\pi}(x)-\hat{\pi}(y)))} \\ &+ \frac{1}{\hat{\pi}(y)} \frac{\beta\exp(\beta(\hat{\pi}(x)-\hat{\pi}(y)))}{(1+\exp(\beta(\hat{\pi}(x)-\hat{\pi}(y))))^2}. \end{aligned} \tag{47}$$

Substituting Eq. (42), (46) and (47) into (45), and then substituting Eq.(45) into (44), we get the gradient for ApproxAP.