Position-Aware ListMLE: A Sequential Learning Process for Ranking

Yanyan Lan[1] **Yadong Zhu**[2] **Jiafeng Guo**[1] **Shuzi Niu**[2] **Xueqi Cheng**[1]
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, P. R. China
[1]{lanyanyan,guojiafeng,cxq}@ict.ac.cn, [2]{zhuyadong,niushuzi}@software.ict.ac.cn

Abstract

ListMLE is a state-of-the-art listwise learning-to-rank algorithm, which has been shown to work very well in application. It defines the probability distribution based on Plackett-Luce Model in a top-down style to take into account the position information. However, both empirical contradiction and theoretical results indicate that ListMLE cannot well capture the position importance, which is a key factor in ranking. To amend the problem, this paper proposes a new listwise ranking method, called position-aware ListMLE (p-ListMLE for short). It views the ranking problem as a sequential learning process, with each step learning a subset of parameters which maximize the corresponding stepwise probability distribution. To solve this *sequential multi-objective optimization* problem, we propose to use *linear scalarization* strategy to transform it into a single-objective optimization problem, which is efficient for computation. Our theoretical study shows that p-ListMLE is better than ListMLE in statistical consistency with respect to typical ranking evaluation measure NDCG. Furthermore, our experiments on benchmark datasets demonstrate that the proposed method can significantly improve the performance of ListMLE and outperform state-of-the-art listwise learning-to-rank algorithms as well.

1 INTRODUCTION

Ranking is an important problem in various applications, such as information retrieval, meta search and collaborative filtering. In recent years, machine learning technologies have been widely applied for ranking, and a new research branch named learning to rank has emerged. A learning-to-rank process can be described as follows. In training, a number of sets (queries) of objects (documents) are given and within each set the objects are labeled by assessors, mainly based on multi-level ratings. The target of learning is to create a model that provides a ranking over the objects that best respects the observed labels. In testing, given a new set of objects, the trained model is applied to generate a ranking list of the objects. To evaluate the performance of a ranking system, many position-aware evaluation measures such as NDCG [9], MAP [2], ERR [5] are used to reflect users' bias on different positions. That is, users often care more about the results on top positions in a ranking [3, 16, 21].

In literature, pointwise algorithms such as McRank [14] were first proposed to solve the ranking problem, which transformed ranking into (ordinal) regression or classification on individual documents. The idea is natural but it is comprehensively criticized for using different objectives from ranking. Therefore, pairwise algorithms such as RankSVM [10], RankBoost [7] and RankNet [1] were then proposed to view a pair of items as the object, and transform ranking into the pairwise classification problem. However, the pairwise approach highly ignores the position information over different pairs, which is quite important for ranking as mentioned above. To overcome the weakness of pairwise ranking algorithms, listwise ranking algorithms such as ListMLE [21], ListNet [4], RankCosine [17] and AdaRank [22] were proposed, which view the whole ranking list as the object. For example, ListMLE utilized the likelihood loss of the probability distribution based on Plackett-Luce model for optimization. According to previous studies [4, 15, 17, 21], the listwise approach can outperform the other two approaches on benchmark datasets.

Seemingly listwise approaches can well solve the ranking problem by directly modeling the ranking lists and thus taking into account the position information. However, both empirical contradiction and theoretical results indicate that listwise approaches cannot well capture the position importance, which is a key factor in ranking [13, 6]. In this paper, we take the typical listwise method ListMLE as an example to illustrate this problem. Empirically, given two ranking functions f_1 and f_2, the error of f_1 occurs in the top positions and that of f_2 occurs in the bottom positions. We find that ListMLE will prefer f_1 to f_2, leading to lower performance under the IR evaluation measure such as NDCG. Theoretically, it has been proven in [21] that ListMLE is consistent[1] with permutation level 0-1 loss, a loss without considering the importance of different positions.

We analyze the underlying reason behind these above results. We find that the probability in ListMLE is defined in a top-down style which seems to reflect the position importance in ranking. However, due to the chain rule of probability, the decomposition of probability in ListMLE is not unique, indicating that different positions are actually equally important in such definition. To amend this problem, we propose a new listwise ranking approach, namely position-aware ListMLE (p-ListMLE for short), which views ranking as a sequential learning process. Specifically, at step 1, it aims to maximize the top 1 probability distribution of Plackett-Luce model. At step i, it aims to maximize the i-th conditional probability distribution of Plackett-Luce model given the top i–1 items. To solve the sequential learning problem, we propose to transform it into a single-objective optimization problem, which is equivalent to minimizing a new surrogate loss.

Theoretically, we study the statistical consistency issue of p-ListMLE. Following the technique used in [11], we can prove that with RDPS as the assumption, p-ListMLE will be consistent with Weighted Pairwise Disagreement Loss (WPDL for short), which is equivalent to a certain NDCG. Further considering the previous result that ListMLE is consistent with permutation level 0-1 loss, we can see that p-ListMLE is better than ListMLE theoretically. We further conduct extensive experiments on benchmark datasets LETOR4.0, and the empirical results demonstrate that the proposed p-ListMLE can significantly outperform the original ListMLE as well as other state-of-the-art listwise learning-to-rank algorithms.

The contribution of this paper lies in the following aspects:

(1) We provide a novel view of ranking as a sequential learning process, with each step to learn a subset of parameters which maximize the corresponding stepwise probability distribution;

(2) We propose a new ranking algorithm to incorporate positions into the learning process;

(3) We provide theoretical analysis on the consistency of the proposed ranking algorithm.

The remainder of the paper are organized as follows. In section 2, we provide some backgrounds on ListMLE, including the framework of listwise learning to rank, the algorithm of ListMLE, and the theoretical results on ListMLE. Section 3 describes the motivation of this paper and section 4 presents our main results, including the novel sequential view of ranking and the loss function of the new algorithm. Section 5 and 6 presents our theoretical and experimental results, respectively. Section 7 concludes the paper.

2 BACKGROUNDS

In this section, we give some backgrounds on ListMLE, which is a famous listwise ranking algorithm. Listwise learning to rank addresses the ranking problem in the following way. In learning, it takes ranking lists of objects as instances and trains a ranking function through the minimization of a listwise loss function defined on predicted list and the ground-truth list. Following [21], we give the mathematical description of listwise learning-to-rank framework as follows.

2.1 Listwise Learning to Rank

Let $\mathbf{x} = \{x_1, \cdots, x_n\} \in X$ be a set of objects to be ranked, and $\mathbf{y} = \{y_1, \cdots, y_n\} \in Y$ be the ground-truth permutation of these objects, where y_i stands for the position of x_i and $\mathbf{y}^{-1}(i)$ stands for the index of items in the i-th position of $\mathbf{y}$. We assume that $(\mathbf{x}, \mathbf{y})$ are sampled according to a fixed but unknown joint probability distribution P_{XY}. Let $f : X \to \mathbb{R}^n$ be a ranking function, where $\mathbb{R}^n$ denotes a n-dimensional real-valued vector space. The task of listwise learning to rank is to learn a ranking function that can minimize the *expected risk* $R_0(h)$, defined as:

$$R_0(h) = \int_{X \times Y} L_0(f; \mathbf{x}, \mathbf{y})\, dP_{XY}(\mathbf{x}, \mathbf{y}),$$

where L_0 is a true loss of listwise learning to rank. For example, Xia et al. [21] utilized permutation level 0-1 loss as the true loss, which takes the following form.

$$L_{0-1}(f; \mathbf{x}, \mathbf{y}) = I_{\{\pi_f(\mathbf{x}) \neq \mathbf{y}\}}, \qquad (1)$$

where $I_{\{\cdot\}}$ is an indicator function, with $I_A = 1$, if A is true, and $I_A = 0$, otherwise. π_f stands for the output permutation induced by sorting the objects in descending order of scores produced by f, that is,

$$\pi_f(\mathbf{x}) = sort(f(x_1), \cdots, f(x_n)).$$

Since P_{XY} is unknown, the optimal ranking function which minimizes the expected loss cannot be easily obtained. In practice, we are usually given independently and identically distributed samples $S = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N \sim P_{XY}$, where $\mathbf{x}_i = (x_1^{(i)}, \cdots, x_{n_i}^{(i)})$ and $\mathbf{y}_i = (y_1^{(i)}, \cdots, y_{n_i}^{(i)})$.

[1] Please note that Xia et al. [19] prove that a modification to ListMLE is consistent with top-k 0-1 loss, however, top-k 0-1 loss take the top k positions as equal, therefore cannot well capture the position importance.

Therefore, we instead try to obtain a ranking function that minimize the *empirical risk*.

$$\hat{R}(h) = \frac{1}{N}\sum_{i=1}^{N} L_0(h; \mathbf{x}_i, \mathbf{y}_i).$$

However, the true loss is usually nonconvex, which poses a challenge to the optimization of the empirical risk. As is done in the literature of machine learning, people usually use surrogate losses as an approximation of the true loss, and turn to minimize the corresponding *surrogate empirical risk* instead.

$$R_\phi(f, \mathbf{x}, \mathbf{y}) = \frac{1}{N}\sum_{i=1}^{N} L_\phi(f; \mathbf{x}_i, \mathbf{y}_i).$$

2.2 ListMLE

ListMLE is such a listwise ranking algorithms which utilize a likelihood loss as the surrogate loss, defined as follows.

$$L(f; \mathbf{x}, \mathbf{y}) = -\log P(\mathbf{y}|\mathbf{x}; f), \tag{2}$$

where,

$$P(\mathbf{y}|\mathbf{x}; f) = \prod_{i=1}^{n} \frac{\exp\left(f(x_{\mathbf{y}^{-1}(i)})\right)}{\sum_{k=i}^{n}\exp(f(x_{\mathbf{y}^{-1}(k)}))}. \tag{3}$$

The above probability is defined according to Plackett-Luce model. That is to say, the probability of a permutation is first decomposed to the product of a stepwise conditional probability, with the i-th conditional probability standing for the probability that the document is ranked at the i-th position given the top $i-1$ objects are ranked correctly. The precise form are given in the following equations.

$$\begin{aligned} & P(\mathbf{y}|\mathbf{x}; f) \\ &= P(\mathbf{y}^{-1}(1), \mathbf{y}^{-1}(2), \cdots, \mathbf{y}^{-1}(n)|\mathbf{x}; f) \\ &= P(\mathbf{y}^{-1}(1)|\mathbf{x}; f)\prod_{i=2}^{n} P(\mathbf{y}^{-1}(i)|\mathbf{x}, \mathbf{y}^{-1}(1), \cdots \mathbf{y}^{-1}(i-1); f) \end{aligned} \tag{4}$$

where,

$$P(\mathbf{y}^{-1}(1)|\mathbf{x}; f) = \frac{\exp\left(f(x_{\mathbf{y}^{-1}(1)})\right)}{\sum_{k=1}^{n}\exp(f(x_{\mathbf{y}^{-1}(k)}))}, \tag{5}$$

$$\begin{aligned} & P(\mathbf{y}^{-1}(i)|\mathbf{x}, \mathbf{y}^{-1}(1), \mathbf{y}^{-1}(2), \cdots, \mathbf{y}^{-1}(i-1); f) \\ &= \frac{\exp\left(f(x_{\mathbf{y}^{-1}(i)})\right)}{\sum_{k=j}^{n}\exp(f(x_{\mathbf{y}^{-1}(k)}))}, \forall i = 2, \cdots, n \end{aligned} \tag{6}$$

2.3 Consistency

Previous theoretical analyses on ListMLE were mainly focused on generalization and consistency. For generalization analysis, Lan et al. [12] has derived the generalization bounds of listwise ranking methods, including ListMLE [21], ListNet [4] and RankCosine [17]. As to the statistical consistency analysis, the loss function in ListMLE has been proven to be consistent with permutation level 0-1 loss [20], where consistency is defined as follows.

Definition 1. *We say a surrogate loss L_ϕ is statistically consistent with respect to the true loss L_0, if $\forall\epsilon_1 > 0, \exists\epsilon_2 > 0$, such that for any ranking function $f \in \mathcal{F}$, $R_\phi(f) \leq \inf_{h\in\mathcal{F}} R_\phi(h) + \epsilon_2$ implies $R_0(f) \leq \inf_{h\in\mathcal{F}} R_0(h) + \epsilon_1$.*

Statistical consistency is a desired property for a good surrogate loss, which measures whether the expected true risk of the ranking function obtained by minimizing a surrogate loss converges to the expected true risk of the optimal ranking in the large sample limit. Therefore, the consistency of ListMLE with respect to permutation level 0-1 loss means that the the surrogate loss of ListMLE is a good surrogate of permutation level 0-1 loss theoretically. However, 0-1 loss is not a 'good' loss for the ranking problem, since it largely ignores the impact of positions, which is crucial in ranking. Therefore, in this paper, we propose to study how to improve ListMLE to make it consistent with a better loss for ranking.

3 Motivations

The motivation of this work comes from both empirical contradiction and theoretical results in ListMLE, indicating that position importance, which is a key factor in ranking, is actually ignored in this listwise approach.

3.1 Empirical Contradiction

Firstly, we take a case study on some toy data to show that position importance is actually not considered in ListMLE.

We are given a set of documents $\mathbf{x} = \{x_1, \cdots, x_5\}$ and their ground-truth labels $\mathbf{z} = (z_1, \cdots, z_5)$ in terms of 5-level ratings, where $z_i = i$. That is to say, the best ranking list is $\mathbf{y} = (1, 2, 3, 4, 5)$. Suppose we are given two ranking functions f_1 and f_2 with corresponding scores $\mathbf{y}_1 = (\ln 4, \ln 5, \ln 3, \ln 2, \ln 1)$ and $\mathbf{y}_2 = (\ln 5, \ln 4, \ln 1, \ln 2, \ln 3)$, where $\mathbf{y}_i(j) = f_i(x_j)$. As we can see, the first two documents are mistakenly ranked by $\mathbf{f}_1$, while the last three documents are mistakenly ranked by $\mathbf{f}_2$. The corresponding likelihood losses of these two ranking functions in ListMLE are listed as follows.

$$\begin{aligned}
&L(f_1, \mathbf{x}, \mathbf{y})\\
&= -\log\left(\frac{4}{4+5+3+2+1}\cdot\frac{5}{5+3+2+1}\cdot\frac{3}{3+2+1}\cdot\frac{2}{2+1}\cdot\frac{1}{1}\right),\\
&L(f_2, \mathbf{x}, \mathbf{y})\\
&= -\log\left(\frac{5}{5+4+3+2+1}\cdot\frac{4}{4+3+2+1}\cdot\frac{1}{1+2+3}\cdot\frac{2}{2+3}\cdot\frac{3}{3}\right).
\end{aligned}$$

Through comparison, we can see that f_1 is better than f_2 in terms of likelihood loss of ListMLE, as indicated by the fact that $L(f_1, \mathbf{x}, \mathbf{y}) < L(f_2, \mathbf{x}, \mathbf{y})$.

However, from the view of IR evaluation measure NDCG, we can see that $NDCG(f_1, \mathbf{x}, \mathbf{y}) < NDCG(f_2, \mathbf{x}, \mathbf{y})$, showing that f_2 should be preferred to f_1.

$$\begin{aligned}
&NDCG@5(f_1, \mathbf{x}, \mathbf{y})\\
&= \frac{1}{N_5}\left((2^5-1)\log\frac{1}{3}+(2^4-1)\log\frac{1}{2}+(2^3-1)\log\frac{1}{4}\right)\\
&\quad+\frac{1}{N_5}\left((2^2-1)\log\frac{1}{5}+(2^1-1)\log\frac{1}{6}\right),\\
&NDCG@5(f_2, \mathbf{x}, \mathbf{y})\\
&= \frac{1}{N_5}\left((2^5-1)\log\frac{1}{2}+(2^4-1)\log\frac{1}{3}+(2^3-1)\log\frac{1}{6}\right)\\
&\quad+\frac{1}{N_5}\left((2^2-1)\log\frac{1}{5}+(2^1-1)\log\frac{1}{4}\right),
\end{aligned}$$

As we know, IR evaluation measures such as NDCG reflect the fact that users are more concerned on results in top positions in a ranking. Therefore, the mistakes in top positions will be more severe than that in low positions. The empirical contradiction between ListMLE and IR evaluation measures thus indicates that the loss of ListMLE cannot well capture the position importance.

3.2 Theoretical Result

In [21], it was proven that the loss functions of ListMLE [21] are consistent with permutation level 0-1 loss. However, permutation level 0-1 loss actually does not take position importance into account. Therefore, theoretical consistency between ListMLE and permutation level 0-1 loss also demonstrates that the position importance cannot be well captured by the loss of ListMLE.

4 Position-Aware ListMLE

The above results indicates that ListMLE ingnores the position importance, which is a key factor for ranking. However, the probability in ListMLE is defined in a top-down style which seems to reflect the position importance in ranking. This contradiction makes us revisit the algorithm of ListMLE. As we can see, the probability on permutation in ListMLE can be decomposed as follows.

$$\begin{aligned}
&P(\mathbf{y}|\mathbf{x}; f) \qquad (7)\\
&= P(\mathbf{y}^{-1}(1), \mathbf{y}^{-1}(2), \cdots, \mathbf{y}^{-1}(n)|\mathbf{x}; f)\\
&= P(\mathbf{y}^{-1}(1)|\mathbf{x}; f)\prod_{i=2}^{n} P(\mathbf{y}^{-1}(i)|\mathbf{x}, \mathbf{y}^{-1}(1), \cdots, \mathbf{y}^{-1}(i-1); f)
\end{aligned}$$

However, the *chain rule of probability* described as follows tells us that this is not the unique decomposition.

$$\begin{aligned}
&P(A_1, \cdots, A_n)\\
&= P(A_{i_1})P(A_{i_2}|A_{i_1})\cdots P(A_{i_n}|A_{i_1}, \cdots, A_{i_{n-1}}),
\end{aligned}$$

where $(i_1, \cdots, i_n)$ is any permutation of $(1, \cdots, n)$.

As a consequence, the probability is equal to any decomposition described as follows.

$$\begin{aligned}
&P(\mathbf{y}|\mathbf{x}; f) \qquad (8)\\
&= P(\mathbf{y}^{-1}(1), \mathbf{y}^{-1}(2), \cdots, \mathbf{y}^{-1}(n)|\mathbf{x}; f)\\
&= P(\mathbf{y}^{-1}(i_1)|\mathbf{x}; f)\prod_{j=2}^{n} P(\mathbf{y}^{-1}(i_j)|\mathbf{x}, \mathbf{y}^{-1}(i_1), \cdots, \mathbf{y}^{-1}(i_{j-1}); f)
\end{aligned}$$

where $(i_1, \cdots, i_n)$ is any permutation of $(1, \cdots, n)$. In this way, different positions are actually equally important under the probability definition of ListMLE. In other words, the loss of ListMLE cannot reflect the position importance in a top-down style. Therefore, we propose a new listwise ranking approach, namely p-ListMLE, to capture the position importance in a 'true' top-down style.

4.1 Ranking As a Sequential Process

To reflect the position importance in a top-down style, i.e. higher position is more important, we propose a new sequential learning process for ranking as follows.

Step 1: Maximizing the probability that the top 1 object is selected with mathematical description as follows.

$$\max_{f\in\mathcal{F}} P(\mathbf{y}^{-1}(1)|\mathbf{x}; f);$$

Step i: For $i = 2, \cdots, n$, we denote the subset of ranking functions that reach the maximum in Step $i-1$ as S_{i-1}. The task of step i is to maximize the probability that the object with position i in ground-truth permutation is selected given the top $i-1$ objects ranked correctly. The mathematical formulation is described as follows.

$$\max_{f\in S_{i-1}} P(\mathbf{y}^{-1}(i)|\mathbf{x}, \mathbf{y}^{-1}(1), \cdots, \mathbf{y}^{-1}(i-1); f);$$

Step n+1: The learning process ends, and the ranking function f is randomly selected from S_{n-1} as the output ranking function.

4.2 Loss Function

In order to solve the above *sequential multi-objective optimization* problem, we propose to use *linear scalarization* strategy [8] to transform it into a single-objective optimization problem, and emphasize the early steps to reflect the position importance.

$$
\begin{aligned}
&\min_{f\in\mathcal{F}} \Phi(f),\\
&\Phi(f) = -\sum_{i=2}^{n}\alpha(i)\log P(\mathbf{y}^{-1}(i)|\mathbf{x},\mathbf{y}^{-1}(1),\cdots,\mathbf{y}^{-1}(i-1);f)\\
&\qquad - \alpha(1)\log P(\mathbf{y}^{-1}(1)|\mathbf{x};f),
\end{aligned}
$$

where $\alpha(\cdot)$ is a decreasing function, i.e. $\alpha(i) > \alpha(i+1)$.

Incorporating the probability based on Plackett-Luce model as described in Eq. (5) and Eq. (6) into the above optimization problem, we obtain a new algorithm which minimizes the following likelihood loss function for p-ListMLE.

$$
\begin{aligned}
&L_p(f;\mathbf{x},\mathbf{y}) \qquad (9)\\
&= \sum_{i=1}^{n}\alpha(i)(-f(x_{\mathbf{y}^{-1}(i)}) + \log(\sum_{j=i}^{n}\exp(f(x_{\mathbf{y}^{-1}(j)})))).
\end{aligned}
$$

4.3 Case Revisit

Here, we revisit the case used in section 3.1 to show that after introducing position factor α into ListMLE, the empirical contradiction will disappear. We first compute the likelihood losses of p-ListMLE with respect to f_1 and f_2, which are listed as follows.

$$
\begin{aligned}
&L_p(f_1,\mathbf{x},\mathbf{y})\\
&= -\alpha(1)\log((\frac{4}{4+5+3+2+1}) - \alpha(2)\log(\frac{5}{5+3+2+1})\\
&\quad -\alpha(3)\log(\frac{3}{3+2+1}) - \alpha(4)\log(\frac{2}{2+1}),
\end{aligned}
$$

$$
\begin{aligned}
&L_p(f_2,\mathbf{x},\mathbf{y})\\
&= -\alpha(1)\log(\frac{5}{5+4+3+2+1}) - \alpha(2)\log(\frac{4}{4+3+2+1})\\
&\quad -\alpha(3)\log(\frac{1}{1+2+3}) - \alpha(4)\log(\frac{2}{2+3}).
\end{aligned}
$$

In this way, the following equality holds:

$$
\begin{aligned}
&L_p(f_1,\mathbf{x},\mathbf{y}) - L_p(f_2,\mathbf{x},\mathbf{y})\\
&= (\alpha(1)-\alpha(2)-\alpha(4))\ln 5 - (\alpha(1)-\alpha(2))\ln 4\\
&\quad - (\alpha(3)-\alpha(4))\ln 3 + \alpha(2)(\ln 11 - \ln 10)\\
&> (\alpha(1)-\alpha(2)-\alpha(4))\ln 5 - (\alpha(1)-\alpha(2)\\
&\quad + \alpha(3)-\alpha(4))\ln 4,
\end{aligned}
$$

Therefore, as long as α satisfies the following condition in Eq. (10), we will have $L_p(f_1,\mathbf{x},\mathbf{y}) > L_p(f_2,\mathbf{x},\mathbf{y})$.

$$
\frac{\alpha(1)-\alpha(2)-\alpha(4)}{\alpha(1)-\alpha(2)+\alpha(3)-\alpha(4)} > \frac{\ln 4}{\ln 5}. \qquad (10)
$$

That is, f_2 is preferred to f_1 in terms of p-ListMLE, which is consistent with $NDCG$ as shown in Section 3.1. This condition is easy to be satisfied as long as $\alpha(1)$ is far lager than the other $\alpha(i), i = 2,\cdots,5$.

5 Statistical Consistency of p-ListMLE

In this section, we study the statistical consistency of p-ListMLE. In [21], it is proved that ListMLE is consistent with permutation level 0-1 loss. Since position factor $\alpha(\cdot)$ is introduced in p-ListMLE, we consider weighted pairwise loss (WPDL) defined in [11] as the true loss.

$$
L_{\text{WPDL}}(f;\mathbf{x},\mathbf{y}) = \sum_{i,j:r_i>r_j} D(r_i,r_j) I_{\{f(x_i)-f(x_j)\le 0\}}, \qquad (11)
$$

where $r_i = n - y_i$, $D(r_i,r_j) = \alpha(r_j) - \alpha(r_i) > 0$.

Firstly, we introduce the definition of a rank-differentiable probability space (RDPS for short), with which we can prove that p-ListMLE is consistent with WPDL.Hereafter, we will also refer to data from RDPS as having a rank-differentiable property.

5.1 A Rank-Differentiable Probability Space

Before introducing the definition of RDPS, we give two definitions, *an equivalence class of ratings* and *dual ratings*. Intuitively, we say two ratings are equivalent if they induce the same ranking or preference relationships. Meanwhile, we say two ratings are the dual ratings with respect to a pair of objects, if the two ratings only exchange the ratings of the two objects while keeping the ratings of other objects unchanged. The formal definitions are given as follows.

Definition 2. *A ratings* $\mathbf{r}$ *is called equivalent to* $\tilde{\mathbf{r}}$*, denoted as* $\mathbf{r} \sim \tilde{\mathbf{r}}$*, if* $\mathcal{P}(\mathbf{r}) = \mathcal{P}(\tilde{\mathbf{r}})$*. Where* $\mathcal{P}(\mathbf{r}) = \{(i,j) : r_i > r_j.\}$ *and* $\mathcal{P}(\tilde{\mathbf{r}}) = \{(i,j) : \tilde{r}_i > \tilde{r}_j.\}$ *stand for the preference relationships induced by* $\mathbf{r}$ *and* $\tilde{\mathbf{r}}$*, respectively.*

Therefore, an equivalence class of the ratings $\mathbf{r}$*, denoted as* $[\mathbf{r}]$*, is defined as the set of ratings which are equivalent to* $\mathbf{r}$*. That is,* $[\mathbf{r}] = \{\tilde{\mathbf{r}} \in \mathcal{R} : \tilde{\mathbf{r}} \sim \mathbf{r}.\}$*.*

Definition 3. *Let* $R(i,j) = \{\mathbf{r} \in \mathcal{R} : r_i > r_j.\}$*,* $\mathbf{r}'$ *is called the dual ratings of* $\mathbf{r} \in R(i,j)$ *with respect to* (i,j) *if* $r'_j = r_i, r'_i = r_j, r'_k = r_k, \forall k \neq i,j$*.*

Now we give the definition of RDPS. An intuitive explanation on this definition is that there exists a unique equivalence class of ratings that for each induced pairwise preference relationship, the probability will be able to separate the two dual ratings with respect to that pair.

Definition 4. *Let* $R(i,j) = \{\mathbf{r} \in \mathcal{R} : r_i > r_j.\}$*, a probability space is called rank-differentiable with* (i,j)*, if for any* $\mathbf{r} \in R(i,j), P(\mathbf{r}|\mathbf{x}) \geq P(\mathbf{r}'|\mathbf{x})$*, and there exists at least one ratings* $\mathbf{r} \in R(i,j)$*, s.t.* $P(\mathbf{r}|\mathbf{x}) > P(\mathbf{r}'|\mathbf{x})$*, where* $\mathbf{r}'$ *is the dual ratings of* $\mathbf{r}$*.*

Definition 5. *A probability space is called rank-differentiable, if there exists an equivalence class* $[\mathbf{r}*]$*, s.t.* $\mathcal{P}(\mathbf{r}^*) = \{(i,j) :$ *the probability space is rank-differentiable with*$(i,j).\}$*, where* $\mathcal{P}(\mathbf{r}^*) = \{(i,j) : r^*_i > r^*_j.\}$*. We will also call this probability space a RDPS or rank-differentiable with* $[\mathbf{r}^*]$*.*

5.2 Consistency with WPDL

Following the proof technique in [11], we prove that p-ListMLE is consistent with WPDL, as shown in the following theorem.

Theorem 1. *We assume that the probability space is rank-differentiable, then the surrogate loss function in p-ListMLE is consistent with WPDL.*

The proof of the theorem is similar to Theorem 5 in [11], and is based on the following theorem.

Theorem 2. *We assume that the probability space is rank-differentiable with an equivalence class* $[\mathbf{r}^*]$*. let* f *be a function such that* $R_p(f|\mathbf{x}) = \inf_{h\in\mathcal{F}} R_p(h|\mathbf{x})$*, then for any object pair* $(x_i, x_j), r^*_i > r^*_j$*, we have* $f(x_i) > f(x_j)$*.*

Proof. (1) We assume that $f(x_i) < f(x_j)$, and define f' as the function such that $f'(x_i) = f(x_j), f'(x_j) = f(x_i), f'(x_k) = f(x_k), \forall k \neq i,j$. We can then get the following equation,

$$\begin{aligned}
&R_p(f'|\mathbf{x}) - R_p(f|\mathbf{x}) \\
&= \sum_{\substack{\mathbf{r},\mathbf{r}',\\ \mathbf{r}\in R(i,j)}} \sum_{k:r_j<r_k<r_i} \alpha(r_k)[P(\mathbf{r}|\mathbf{x}) – P(\mathbf{r}'|\mathbf{x})] \\
&\quad \times \log\left(\frac{\sum_{l=y_k,l\neq y_j}^{n} \exp(f(x_{\mathbf{y}^{-1}(l)})) + \exp(f(x_i))}{\sum_{l=y_k,l\neq y_j}^{n} \exp(f(x_{\mathbf{y}^{-1}(l)})) + \exp(f(x_j))}\right) \\
&+ \sum_{\substack{\mathbf{r},\mathbf{r}',\\ \mathbf{r}\in R(i,j)}} [\alpha(r_i) - \alpha(r_j)][\log(f(x_i)) - \log(f(x_j))] \\
&\qquad \times [P(\mathbf{r}|\mathbf{x}) – P(\mathbf{r}'|\mathbf{x})] \\
&+ \sum_{\substack{\mathbf{r},\mathbf{r}',\\ \mathbf{r}\in R(i,j)}} \alpha(r_j)[P(\mathbf{r}|\mathbf{x}) – P(\mathbf{r}'|\mathbf{x})] \\
&\quad \times \log\left(\frac{\sum_{l=y_j+1}^{n} \exp(f(x_{\mathbf{y}^{-1}(l)})) + \exp(f(x_i))}{\sum_{l=y_j+1}^{n} \exp(f(x_{\mathbf{y}^{-1}(l)})) + \exp(f(x_j))}\right),
\end{aligned}$$

According to the conditions of RDPS and the requirements of $\alpha(\cdot)$, we can obtain

$$R_p(f'|\mathbf{x}) < R_p(f|\mathbf{x}).$$

This is a contradiction with $R_p(f|\mathbf{x}) = \inf_{h\in\mathcal{F}} R_p(h|\mathbf{x})$. Therefore, we have proven that $f(x_i) \leq f(x_j)$.

(2) Now we assume that $f(x_i) = f(x_j) = f_0$. From the assumption $R_p(f|\mathbf{x}) = \inf_{h\in\mathcal{F}} R_p(h|\mathbf{x})$, we can get

$$\left.\frac{\partial R_\Phi(f|\mathbf{x})}{\partial f(x_i)}\right|_{f_0} = 0, \left.\frac{\partial R_\Phi(f|\mathbf{x})}{\partial f(x_j)}\right|_{f_0} = 0.$$

Accordingly, we can obtain two equations as follows:

$$\sum_{\substack{\mathbf{r},\mathbf{r}',\\ \mathbf{r}\in R(i,j)}} A_1 P(\mathbf{r}|\mathbf{x}) + A_2 P(\mathbf{r}'|\mathbf{x}) = 0, \tag{12}$$

$$\sum_{\substack{\mathbf{r},\mathbf{r}',\\ \mathbf{r}\in R(i,j)}} B_1 P(\mathbf{r}|\mathbf{x}) + B_2 P(\mathbf{r}'|\mathbf{x}) = 0, \tag{13}$$

where,

$$\begin{aligned}
&A_1 = B_2 \\
&= \sum_{k:r_j<r_i<r_k} \alpha(r_k) \frac{\exp(f_0)}{\sum_{l=y_k}^{n} \exp(f(x_{\mathbf{y}^{-1}(k)}))} \\
&\quad + \alpha(r_i)(-1 + \frac{\exp(f_0)}{\sum_{l=y_i}^{n} \exp(f(x_{\mathbf{y}^{-1}(k)}))}),
\end{aligned}$$

$$
\begin{aligned}
&A_2 = B_1 \\
=& \sum_{k:r_j<r_i<r_k} \alpha(r_k)\frac{\exp(f_0)}{\sum_{l=y_k}^{n}\exp(f(x_{\mathbf{y}^{-1}(k)}))} \\
&+ \sum_{k:r_i<r_k<r_j} \alpha(r_k)\frac{\exp(f_0)}{\sum_{l=y_k}^{n}\exp(f(x_{\mathbf{y}^{-1}(k)}))} \\
&+ \alpha(r_i)\frac{\exp(f_0)}{\sum_{l=y_i}^{n}\exp(f(x_{\mathbf{y}^{-1}(l)}))} \\
&+ \alpha(r_j)(-1 + \frac{\exp(f_0)}{\sum_{l=y_j}^{n}\exp(f(x_{\mathbf{y}^{-1}(k)}))})
\end{aligned}
$$

Based on the requirements of RDPS and $\alpha(\cdot)$, we can obtain that,

$$
\begin{aligned}
&\sum_{\substack{\mathbf{r},\mathbf{r}',\\ \mathbf{r}\in R(i,j)}} (A_1 - B_1)P(\mathbf{r}|\mathbf{x}) + (A_2 - B_2)P(\mathbf{r}'|\mathbf{x}) \\
=& \sum_{\substack{\mathbf{r},\mathbf{r}',\\ \mathbf{r}\in R(i,j)}} (A_1 - A_2)[P(\mathbf{r}|\mathbf{x}) - P(\mathbf{r}'|\mathbf{x})] \\
\leq& \sum_{\substack{\mathbf{r},\mathbf{r}',\\ \mathbf{r}\in R(i,j)}} (\alpha(r_j) - \alpha(r_i))[P(\mathbf{r}|\mathbf{x}) - P(\mathbf{r}'|\mathbf{x})] < 0.
\end{aligned}
$$

This is a contradiction with Eq. (12). Therefore, we actually have proven that $f(x_i) > f(x_j)$. □

5.3 Discussion

In [21], ListMLE is proved to be consistent with permutation level 0-1 loss. However, consistency with permutation level 0-1 loss does not mean consistency with NDCG. For example, it has been proven in [18] that losses in RankCosine [17] and ListNet [4] are not consistent with NDCG. Since that we have proven that p-ListMLE is consistent with WPDL, it is natural to study the relationship between WPDL and NDCG. Here we show that WPDL with a certain weight, referred to as *difference-weight pairwise disagreement loss* (DWPDL for short), is equivalent to NDCG with a certain discount function, referred to as *sharp-NDCG*. In this way, p-ListMLE is better than ListMLE theoretically.

The formal definition of NDCG is as follows.

$$
NDCG@n(f;\mathbf{x},\mathbf{y}) = \frac{1}{N_n}\sum_{i=1}^{n} Gain(r(y_i))Disc(\pi_f(i)),
$$

where $Gain$ is the gain function which gives larger scores to objects with larger labels and $Disc$ is the discount function which gives larger scores to objects ranked higher in π_f. $r(y_i)$ is a function to mapping position y_i into relevance score. For consistency with above formulation, we use r_i to denote $r(y_i)$, and define $r_i = n - y_i$, $\pi_f(i)$ is the position of document x_i in permutation π_f and N_n is a normalization factor.

We define WDPDL, and sharp-NDCG as follows, respectively.

$$
\begin{aligned}
&l_w(f;\mathbf{x},\mathbf{y}) \\
=& \sum_{i,j,r_i>r_j} (Gain(r_i) - Gain(r_j))1_{\{f(x_i)\leq f(x_j)\}}, \\
&sharp\text{–}NDCG@n(f;\mathbf{x},\mathbf{y}) = \frac{\sum_{i=1}^{n}(n-\pi(i))Gain(r_i)}{N_n'},
\end{aligned}
$$

where N_n' is a normalization factor.

The following theorem shows the relationship between D-WPDL and sharp-NDCG.

Theorem 3. *We denote* $\mathcal{F}_0 = \{f \in \mathcal{F} : \forall\mathbf{x}, f(x_i) \neq f(x_j), \forall i, j.\}$*, then* $\forall(\mathbf{x},\mathbf{y}), f, g \in \mathcal{F}_0$*, we have*

$$
\begin{gathered}
l_w(f;\mathbf{x},\mathbf{y}) < l_w(g;\mathbf{x},\mathbf{y}) \\
\Leftrightarrow sharp\text{–}NDCG@n(\pi_f,\mathbf{x},\mathbf{y}) > sharp\text{–}NDCG@n(\pi_g,\mathbf{x},\mathbf{y}).
\end{gathered}
$$

Proof. According to the definition of sharp-NDCG, it has the following form:

$$
sharp-NDCG@n(\pi,\mathbf{y}) = \sum_{i=1}^{n} Gain(r_i)(n - \pi(i)).
$$

Rewrite $n - \pi(i)$ and $n - \pi_{\mathbf{y}}(i)$ as $\sum_{j=1}^{m} 1_{\{\pi(i)-\pi(j)<0\}}$ and $\sum_{j=1}^{m} 1_{\{\pi_{\mathbf{y}}(i)-\pi_{\mathbf{y}}(j)<0\}}$, respectively, then the following equation holds,

$$
\begin{aligned}
&\sum_{i=1}^{n}(n-\pi_{\mathbf{y}})Gain(r_i) - sharp\text{–}NDCG@n(\pi;\mathbf{x},\mathbf{y}) \\
=& \sum_{i=1}^{n}\sum_{j\neq i} Gain(r(y_i))(1_{\{\pi_{\mathbf{y}}(i)-\pi_{\mathbf{y}}(j)<0\}} - 1_{\{\pi(i)-\pi(j)<0\}}),
\end{aligned}
$$

Thus, for each pair $(i,j), r_i > r_j$, the term becomes:

$$
\begin{aligned}
&Gain(r_i)1_{\{\pi(i)-\pi(j)>0\}} - Gain(r(y_j))1_{\{\pi(j)-\pi(i)<0\}} \\
=& Gain(r_i)1_{\{\pi(i)-\pi(j)>0\}} - Gain(j)1_{\{\pi(i)-\pi(j)>0\}} \\
=& (Gain(r_i) - Gain(r(y_j)))1_{\{\pi(i)-\pi(j)>0\}},
\end{aligned}
$$

Further considering the relationship between π and f, the following equation holds:

$$
\begin{aligned}
&\sum_{i=1}^{m}(n-\pi_{\mathbf{y}})Gain(r_i) - sharp-NDCG@m(\pi;\mathbf{x},\mathbf{y}) \\
&= l_w(f;\mathbf{x},\mathbf{y}).
\end{aligned}
$$

Since $\sum_{i=1}^{m}(n-\pi_{\mathbf{y}})Gain(r_i)$ is just dependent on $\mathbf{y}$, the result in the theorem has been proved. □

Note that in applications such as information retrieval, people usually use the following specific discount and gain functions.

$$
Gain(i) = 2^{r(y_i)} - 1,\ Disc(\pi(i)) = \frac{1}{\log_2(1+\pi(i))}.
$$

According to the original paper of NDCG [9], however, this is not the only choice. Actually, the only difference between sharp-NDCG and the above NDCG is that the discount function in sharp-NDCG is sharper.

In summary, we have obtain that:

(1) p-ListMLE is statistically consistent with WPDL, where $D(r_i, r_j) = \alpha(r_j) - \alpha(r_i)$;

(2) WPDL with $D(r_i, r_j) = Gain(r_i) - Gain(r_j)$ is equivalent to sharp-NDCG;

(3) sharp-NDCG is intrinsic similar with NDCG.

Therefore, we can expect to obtain better performance w.r.t NDCG if we define $\alpha(r_i) = Gain(r_i) = 2^{n-y_i} - 1$.

6 Experiments

In this section, we conduct experiments on benchmark data sets LETOR 4.0[2] to show that p-ListMLE can achieve better performances compared to ListMLE as well as other state-of-the-art listwise learning-to-rank algorithms.

As the setting in this paper is listwise ranking, we choose two datasets in LETOR, i.e. MQ2007-list and MQ2008-list, in which the ground-truth is a full order ranking list. In MQ2007-list, there are about 1700 queries and 700 documents per query on average. In MQ2008-list, there are about 800 queries and 1000 documents per query on average. Both query sets are from Million Query track of TREC 2007 and TREC 2008.

We implement both ListMLE and p-ListMLE by Stochastic Gradient Descent (SGD). In p-ListMLE, we set $\alpha(i)$ as $2^{n-i} - 1$, as guided in the above section. The stopping criteria is chosen from $\{0.1^i\}_{i=1}^6$ to control when to stop, and the learning rates are selected from $\{0.1^i\}_{i=1}^5$ with the maximal number of iterations 500. Evaluation measure NDCG is adopted to evaluate the test performances of both algorithms, where the multi-level ground-truth label is taken as $l(x_i) = n - y_i$, where y_i is the rank of item i in the listwise ground-truth. For empirical comparison, we also include two other state-of-the-art listwise learning-to-rank algorithms such as ListNet and RankCosine, and the implementation is conducted strictly under the standard setting as shown in [4] and [17]. The experimental results are shown in Figure 1.

From the results, we can see that p-ListMLE significantly outperform ListMLE with NDCG as evaluation measure on both datasets. Taking NDCG@10 as an example, the improvement of p-ListMLE over ListMLE is 0.95% and 0.54% on MQ2007-list and MQ2008-list, respectively. Moreover, when comparing p-ListMLE with the other two baseline methods, we can see that it can also outperform traditional listwise ranking methods significantly. We also take NDCG@10 as an example, the improvement of p-ListMLE over ListNet is 0.85% and 0.22% on MQ2007-list and MQ2008-list, respectively. While, the improvement of p-ListMLE over RankCosine is 1.48% and 0.32% on on MQ2007-list and MQ2008-list, respectively.

7 Conclusion

In this paper, we address the problem of ListMLE that the position importance is highly ignored, which is however very important for ranking. We propose a new listwise ranking algorithm, namely position-aware ListMLE (p-ListMLE) to amend the problem. In p-ListMLE, ranking is viewed as a sequential learning process, with each step learning a subset of parameters which minimize the corresponding stepwise probability distribution. We prove that p-ListMLE is consistent with WPDL, which is equivalent to a certain form of NDCG. In addition, our experimental results on benchmark datasets show that p-ListMLE can significantly outperform ListMLE. Therefore, we have demonstrated both theoretically and empirically that p-ListMLE is better than ListMLE.

For the future work, we plan to further investigate the problem of how to set the position factor $\alpha(\cdot)$ in practice, or how to guide the settings from other theoretical aspects.

Acknowledgments

This research work was funded by the 973 Program of China under Grants No. 2012CB316303 and No. 2014CB340401, the 863 Program of China under Grants No. 2012AA011003, National Natural Science Foundation of China under Grant No. 61232010 and No. 61203298.

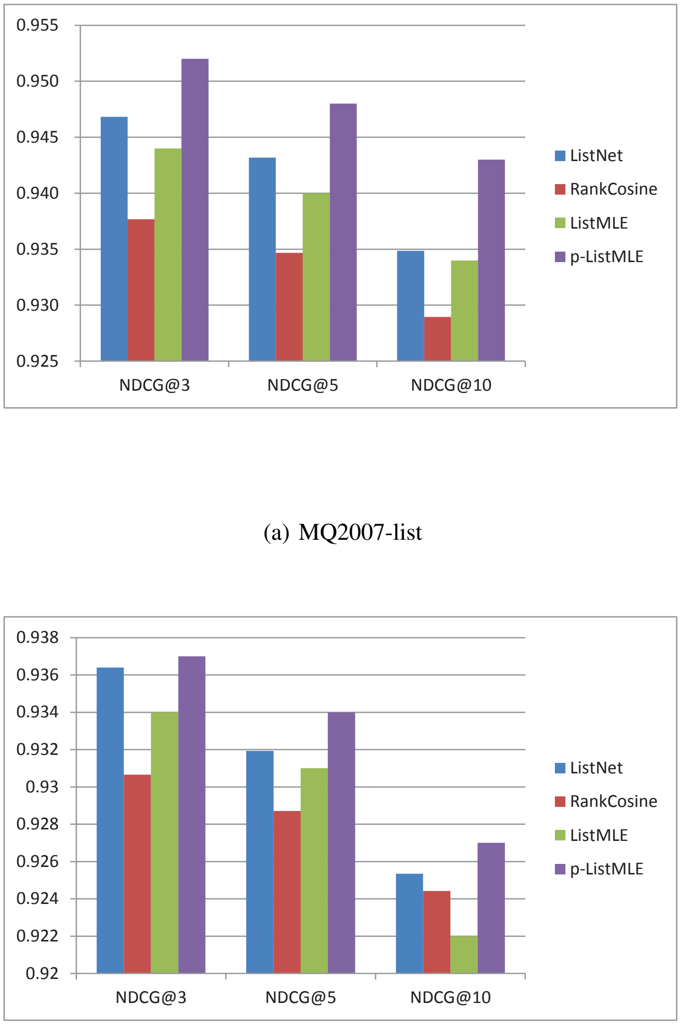
References

[1] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of the 22th International Conference on Machine Learning (ICML 2005)*, pages 89–96, 2005.

[2] C. Burkley and E. M. Voorhees. *Retrieval System Evaluation*. MIT Press, 2005.

[3] Y. Cao, J. Xu, T.-Y. Liu, H. Li, Y. Huang, and H.-W. Hon. Adapting ranking svm to document retrieval. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 186–193, New York, NY, USA, 2006. ACM.

[2] http://research.microsoft.com/en-us/um/beijing/projects/letor/.



(a) MQ2007-list

(b) MQ2008-list

Figure 1: Performance Comparison Between p-ListMLE and ListMLE

[4] Z. Cao, T. Qin, T. Y. Liu, M. F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th International Conference on Machine Learning (ICML 2007)*, pages 129–136, 2007.

[5] O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM ’09, pages 621–630, New York, NY, USA, 2009. ACM.

[6] W. Chen, T. yan Liu, Y. Lan, Z. Ma, and H. Li. Ranking measures and loss functions in learning to rank. In *Twenty-Fourth Annual Conference on Neural Information Processing Systems*, pages 315–323, 2009.

[7] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003.

[8] C.-L. Hwang and A. S. M. Masud. *Multiple Objective Decision Making, Methods and Applications*. springer-Verlag, 1979.

[9] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Informations and Systems*, 20(4):422–446, 2002.

[10] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2002)*, pages 133–142, 2002.

[11] Y. Lan, J. Guo, X. Cheng, and T.-Y. Liu. Statistical consistency of ranking methods in a rank-differentiable probability space. In P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1241–1249, 2012.

[12] Y. Lan, T.-Y. Liu, Z. Ma, and H. Li. Generalization analysis of listwise learning-to-rank algorithms. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 577–584, New York, NY, USA, 2009. ACM.

[13] Y. Lan, S. Niu, J. Guo, and X. Cheng. Is top-k sufficient for ranking? In *Proceedings of the 22nd ACM international conference on Conference on information &#38; knowledge management*, CIKM '13, pages 1261–1270, New York, NY, USA, 2013. ACM.

[14] P. Li, C. Burges, and Q. Wu. Mcrank: Learning to rank using multiple classification and gradient boosting. In *Twenty-Second Annual Conference on Neural Information Processing Systems*, 2007.

[15] T.-Y. Liu. Learning to rank for information retrieval. *Foundation and Trends on Information Retrieval*, 3:225–331, 2009.

[16] S. Niu, J. Guo, Y. Lan, and X. Cheng. Top-k learning to rank: labeling, ranking and evaluation. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '12, pages 751–760, New York, NY, USA, 2012. ACM.

[17] 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*, 2007.

[18] P. Ravikumar, A. Tewari, and E. Yang. On ndcg consistency of listwise ranking methods. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, volume 15 of *JMLR Workshop and Conference Proceedings*, pages 618–626, 2011.

[19] F. Xia, T.-Y. Liu, and H. Li. Statistical consistency of top-k ranking. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 2098–2106. 2009.

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

[21] F. Xia, T. Y. Liu, J. Wang, W. S. Zhang, and H. Li. Listwise approach to learning to rank - theory and algorithm. In *Proceedings of the 25th International Conference on Machine Learning (ICML 2008)*, 2008.

[22] J. Xu and H. Li. Adarank: A boosting algorithm for information retrieval. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 391–398, New York, NY, USA, 2007. ACM.