# Directly Optimize Diversity Evaluation Measures: A New Approach to Search Result Diversification

JUN XU, LONG XIA, YANYAN LAN, JIAFENG GUO, and XUEQI CHENG,
CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology,
Chinese Academy of Sciences

The queries issued to search engines are often ambiguous or multifaceted, which requires search engines to return diverse results that can fulfill as many different information needs as possible; this is called *search result diversification*. Recently, the relational learning to rank model, which designs a learnable ranking function following the criterion of maximal marginal relevance, has shown effectiveness in search result diversification [Zhu et al. 2014]. The goodness of a diverse ranking model is usually evaluated with diversity evaluation measures such as $\alpha$-NDCG [Clarke et al. 2008], ERR-IA [Chapelle et al. 2009], and D#-NDCG [Sakai and Song 2011]. Ideally the learning algorithm would train a ranking model that could directly optimize the diversity evaluation measures with respect to the training data. Existing relational learning to rank algorithms, however, only train the ranking models by optimizing loss functions that loosely relate to the evaluation measures. To deal with the problem, we propose a general framework for learning relational ranking models via *directly optimizing any diversity evaluation measure*. In learning, the loss function upper-bounding the basic loss function defined on a diverse ranking measure is minimized. We can derive new diverse ranking algorithms under the framework, and several diverse ranking algorithms are created based on different upper bounds over the basic loss function. We conducted comparisons between the proposed algorithms with conventional diverse ranking methods using the TREC benchmark datasets. Experimental results show that the algorithms derived under the diverse learning to rank framework always significantly outperform the state-of-the-art baselines.



CCS Concepts: • **Information systems → Learning to rank**; **Information retrieval diversity;**

Additional Key Words and Phrases: Search result diversification, relational learning to rank, diversity evaluation measure

**ACM Reference Format:**
Jun Xu, Long Xia, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. 2017. Directly optimize diversity evaluation measures: A new approach to search result diversification. ACM Trans. Intell. Syst. Technol. 8, 3, Article 41 (January 2017), 26 pages.
DOI: http://dx.doi.org/10.1145/2983921

## 1. INTRODUCTION

It has been widely observed that users' information needs, described by keyword-based queries, are often ambiguous or multifaceted. It is important for commercial search engines to provide search results that balance query-document relevance and

This work was funded by the 973 Program of China under Grant No. 2014CB340401 and 2013CB329606, the 863 Program of China under Grant No. 2014AA015204 and 2015AA020104, the National Natural Science Foundation of China (NSFC) under Grant No. 61232010, 61425016, 61472401 and 61203298, and the Youth Innovation Promotion Association CAS under Grant No. 20144310 and 2016102.
Authors' addresses: J. Xu, L. Xia, Y. Lan, J. Guo, and X. Cheng, Institute of Computing Technology, Chinese Academy of Sciences, No. 6 Kexueyuan South Road, Zhongguancun, Haidian District, Beijing, China 100190; emails: junxu@ict.ac.cn, xialong@software.ict.ac.cn, {lanyanyan, guojiafeng, cxq}@ict.ac.cn.

DOI: http://dx.doi.org/10.1145/2983921

document novelty, called *search result diversification* [Zhai et al. 2003; Agrawal et al. 2009]. One of the key problems in search result diversification is ranking: specifically, how to develop a ranking model that can sort documents based on their relevance to the given query as well as by the novelty of the information in the documents.

Methods for search result diversification can be categorized into heuristic approaches and learning approaches. The heuristic approaches construct diverse rankings with handcrafted ranking rules. As a representative method in the category, Carbonell and Goldstein proposed the Maximal Marginal Relevance (MMR) criterion for guiding the construction of ranking models [Carbonell and Goldstein 1998]. In MMR, constructing a diverse ranking is formulated as a process of sequential document selection. At each iteration, the document with the highest marginal relevance is selected. The marginal relevance can be defined as, for example, a linear combination of the query-document relevance and the maximum distance of the document to the selected document set. A number of approaches have been proposed [Rafiei et al. 2010; Santos et al. 2010; Dang and Croft 2012; Raman et al. 2012] on the basis of the criterion and promising results have been achieved. User studies also shows that user browsing behavior matches very well with the MMR criterion: Usually, users browse the web search results in a top-down manner and perceive diverse information from each individual document based on what they have obtained in the preceding results [Clarke et al. 2008]. Therefore, in a certain sense, we can say that MMR has been widely accepted as a criterion for guiding the construction of diverse ranking models.

Machine learning approaches have also been proposed for the task of search result diversification [Radlinski et al. 2008; Li et al. 2009; Mihalkova and Mooney 2009; Zhu et al. 2014], especially those methods that can directly optimize evaluation measures on training data [Yue and Joachims 2008; Liang et al. 2014]. Yue and Joachims propose SVM-DIV, which formulates the task as a problem of structured output prediction [Yue and Joachims 2008]. In the model, the measure of subtopic diversity is directly optimized under the structural SVM framework. Liang et al. propose to conduct personalized search result diversification via directly optimizing the measure of $\alpha$-NDCG, also under the structural SVM framework [Liang et al. 2014]. All these methods try to resolve the mismatch between the objective function used in training and the final evaluation measure used in testing. Experimental results also showed that directly optimizing the diversity evaluation measures can indeed improve diverse ranking performances [Yue and Joachims 2008; Liang et al. 2014].

One problem with the direct optimization approaches is that it is hard, if not impossible, to define a ranking model that can meet the MMR criterion under the direct optimization framework. Recently, Zhu et al. proposed the Relational Learning to Rank (R-LTR) model [Zhu et al. 2014] in which the ranking function is designed following the criterion of MMR. In training, R-LTR maximizes the likelihood of the "positive" rankings derived from the training data, which is loosely related to the diversity evaluation measures.

Thus, there is an open question regarding machine learning approaches to search result diversification. Is there a general theory that can guide the development of new diverse learning to rank algorithms, one that can utilize the MMR model for ranking as well as directly optimize diversity evaluation measure in training?

In this article, we conduct a study on directly optimizing diversity evaluation measures in diverse learning to rank and answered the posed question. Specifically, we develop a new diverse learning to rank framework that adopts the R-LTR as its ranking model. In training, the model parameters are estimated by minimizing a loss function upper bounding the basic loss function defined on the diversity evaluation measures. New algorithms can be easily derived and studied under the framework. As examples, we created several algorithms by optimizing different upper bounds.

The framework offers several advantages: (i) it makes use of the R-LTR model in ranking, which has been shown to be effective for the task of search result diversification; (ii) it has the ability to easily derive different algorithms that can directly optimize any diversity evaluation measure in training by minimizing different upper bounds of the basic loss function; and (iii), it has the possibility to analyze new diverse ranking algorithms under a unified framework.

To evaluate the effectiveness of the framework and the new derived diverse ranking algorithms, we conducted extensive experiments on three public TREC benchmark datasets. The experimental results show that the new derived diverse ranking algorithms under the diverse ranking framework significantly outperformed the state-of-the-art approaches including MMR, SVM-DIV, and R-LTR. We analyzed the results and showed that the new derived algorithms attained good balances between relevance and novelty in ranking and directly optimizing evaluation measures in training. We also showed that by directly optimizing a measure in training, the derived algorithms can indeed enhance the ranking performances in terms of the measure used in training. From the experimental results, we also observed that there exists no significant difference among the performances of the algorithms derived under the framework.

The rest of the article is organized as follows. After a summary of related work in Section 2, we describe R-LTR in Section 3. In Section 4, we describe the proposed general framework for directly optimizing diversity evaluation measures. The example algorithms derived under the framework are shown in Section 5. In Section 6, we give a summary of the upper bounds and analyze the advantages of our approaches. Experimental results and discussions are given in Section 7. Section 8 concludes this article and gives future work.

## 2. RELATED WORK

Methods of search result diversification can be categorized into heuristic approaches and learning approaches.

### 2.1. Heuristic Approaches

It is a common practice to use heuristic rules to construct a diverse ranking list in search. Usually, the rules are created based on the observation that, in diverse ranking, a document's novelty depends not only on the document itself but also on the documents ranked in previous positions. Carbonell and Goldstein propose the MMR criterion to guide the design of diverse ranking models [Carbonell and Goldstein 1998]. The criterion is implemented through a process of iteratively selecting documents from the candidate document set. At each iteration, the document with the highest marginal relevance score is selected, where the score is a linear combination of the query-document relevance and the maximum distance of the document to the documents in the current result set. The marginal relevance score is then updated in the next iteration as the number of documents in the result set increases by one. More methods have been developed under the criterion. PM-2 [Dang and Croft 2012] treats the problem of finding a diverse search result as finding a proportional representation for the document ranking. xQuAD [Santos et al. 2010] directly models different aspects underlying the original query in the form of subqueries and estimates the relevance of the retrieved documents to each identified subquery. Dou et al. [2011] proposes to explicitly diversify search results based on multiple dimensions of subtopics. Hu et al. [2015] proposed a diversification framework that leverages the hierarchical intents of queries and selects those documents that maximize diversity in the hierarchical structure [Radlinski and Dumais 2006; Carterette and Chandar 2009; Gollapudi and Sharma 2009; Guo and Sanner 2010; He et al. 2012; Dang and Croft 2013; Wu et al. 2016].

Heuristic approaches rely on the utility functions that can only use a limited number of ranking signals. Also, the parameter tuning cost is high, especially in complex search settings. In this article, we focus on machine learning approaches to constructing diverse ranking models that can meet the MMR criterion.

### 2.2. Learning Approaches

Methods of machine learning have been applied to search result diversification. In these approaches, rich features can be utilized and the parameters are automatically estimated from the training data. Some promising results have been obtained. For example, Zhu et al. [2014] proposed the R-LTR model, in which diverse ranking is constructed with a process of sequential document selection. The training of R-LTR amounts to optimizing the likelihood of ground truth rankings. Radlinski et al. proposed online learning algorithms that directly learn a diverse ranking of documents based on users' clicking behavior [Radlinski et al. 2008]; for more such works, please refer to Li et al. [2009], Mihalkova and Mooney [2009], and Yue and Joachims [2009]. All these methods, however, formulate the learning problem as optimizing loss functions that are loosely related to diversity evaluation measures.

Recently, methods that can directly optimize evaluation measures have been proposed and applied to search result diversification. Yue and Joachims formulate the task of constructing a diverse ranking as a problem of predicting diverse subsets [Yue and Joachims 2008]. The structural SVM framework is adopted to perform the training. Liang et al. propose to conduct personalized search result diversification, also under the structural SVM framework [Liang et al. 2014]. In the model, the loss function is defined based on the diversity evaluation measure of $\alpha$-NDCG. Thus, the algorithm can be considered as directly optimizing $\alpha$-NDCG in training. One issue with these methods is that it is hard to learn a MMR model under the structural SVM framework.

In this article, we propose a framework that can learn a MMR model at the same time it can minimize loss functions upper bounding the basic loss function defined on diversity evaluation measures.

## 3. RELATIONAL LEARNING TO RANK

Zhu et al. proposed R-LTR for search result diversification [Zhu et al. 2014]. In R-LTR, the ranking of documents is designed as a process of selecting documents sequentially, and thus it meets the criterion of MMR. Specifically, suppose that we are given a query $q$, which is associated with a set of retrieved documents $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$, where each document $\mathbf{x}_i$ is represented as a $D$-dimensional relevance feature vector. Let $R = \mathcal{R}^{M\times M\times K}$ denotes a 3-way tensor representing relationship among the $M$ documents, where $R_{ijk}$ stands for the $k$-th relationship feature of document $\mathbf{x}_i$ and document $\mathbf{x}_j$.

### 3.1. Document Level Ranking Model $f_S$

The MMR model creates a diverse ranking over $X$ with a process of sequential document selection. At each step, the document with the highest marginal relevance is selected and added to the tail of the list [Zhu et al. 2014]. Specifically, let $S \subseteq X$ be the set of documents that have been selected for query $q$ at one of the document selection steps. Given $S$, the marginal relevance score of each document $\mathbf{x}_i \in X\backslash S$ at the current step is defined as a linear combination of the query-document relevance and novelty of the document to the documents in $S$:

$$f_S(\mathbf{x}_i, R_i) = \omega_r^T \mathbf{x}_i + \omega_d^T h_S(R_i), \tag{1}$$

where $\mathbf{x}_i$ denotes the relevance feature vector of the document, $R_i \in \mathcal{R}^{M\times K}$ is the matrix representation of the relationship between document $i$ and the other documents (note that $R_{ij} \in \mathcal{R}^K$ denotes the relationship feature vector of document pair $(i, j)$), and $\omega_r$

**ALGORITHM 1:** Ranking with Relational Learning to Rank Model

**Input:** documents $X$, document relation $R$, and ranking model parameters $\omega_r$ and $\omega_d$
**Output:** ranking $\mathbf{y}$

1: $S_0 \leftarrow \phi$
2: **for** $r = 1, \ldots, M$ **do**
3: $\quad y(r) \leftarrow \arg\max_{j:\mathbf{x}_j \in X \setminus S_{r-1}} f_{S_{r-1}}(\mathbf{x}_j, R_j)$
4: $\quad S_r \leftarrow S_{r-1} \cup \{\mathbf{x}_{y(r)}\}$
5: **end for**
6: **return y**

and $\omega_d$ are the weights for the relevance features and novelty features, respectively. The first term in Equation (1) represents the relevance of document $i$ to the query, and the second term represents the novelty of documents $i$ with respect to the documents in $S$. Following the practice in Zhu et al. [2014], the relational function $h_S(R_i)$ is defined as the minimal distance:

$$h_S(R_i) = \left( \min_{\mathbf{x}_j \in S} R_{ij1}, \ldots, \min_{\mathbf{x}_j \in S} R_{ijK} \right).$$

According to the MMR criterion, a sequential document selection process can be used to create a diverse ranking, as shown in Algorithm 1. Specifically, given a query $q$, the retrieved documents $X$, and document relationship $R$, the algorithm initializes $S_0$ as an empty set. It then iteratively selects the documents from the candidate set. At iteration $r$ ($r = 1, \ldots, M$), the document with the MMR score $f_{S_{r-1}}$ is selected and ranked at position $r$. At the same time, the selected document is inserted into $S_{r-1}$.

### 3.2. Query Level Ranking Model *F*

The query level ranking model $F(X, R, \mathbf{y})$ is the query level ranking function. $F$ takes the document set $X$, document relationship $R$, and ranking over the documents $\mathbf{y}$ as inputs. The output of $F$ is the confidence score of the ranking $\mathbf{y}$. The predicted $\hat{\mathbf{y}}^{(n)}$ can be considered as the ranking that maximizes $F$:

$$\hat{\mathbf{y}}^{(n)} = \arg\max_{\mathbf{y} \in \mathcal{Y}^{(n)}} F(X^{(n)}, R^{(n)}, \mathbf{y}), \tag{2}$$

where $\mathcal{Y}^{(n)}$ is the set of all possible rankings over $X^{(n)}$. Here, $F$ is defined as the logarithm of the probability of generating the ranking list $\mathbf{y}$ with a process of iteratively selecting the top-ranked documents from the remaining documents and using the marginal relevance function $f_S$ in Equation (1) as the selection criterion:

$$\begin{aligned} F(X, R, \mathbf{y}) &= \log \Pr(\mathbf{y}|X, R) \\ &= \log \Pr(\mathbf{x}_{y(1)} \ldots \mathbf{x}_{y(M)}|X, R) \\ &= \log \prod_{r=1}^{M-1} \Pr(\mathbf{x}_{y(r)}|X, S_{r-1}, R) \\ &= \sum_{r=1}^{M-1} \log \frac{\exp\{f_{S_{r-1}}(\mathbf{x}_i, R_{y(r)})\}}{\sum_{k=r}^{M} \exp\{f_{S_{r-1}}(\mathbf{x}_i, R_{y(k)})\}} \end{aligned} \tag{3}$$

where $y(r)$ denotes the index of the document ranked at the $r$-th position in $\mathbf{y}$, $S_{r-1} = \{\mathbf{x}_{y(k)}\}_{k=1}^{r-1}$ is the documents ranked at the top $r-1$ positions in $\mathbf{y}$, $f_{S_{r-1}}(\mathbf{x}_i, R_i)$ is the marginal relevance score of document $\mathbf{x}_i$ with respect to the selected documents in $S_{r-1}$, and $S_0 = \phi$ is an empty set.

With the definition of query level ranking model $F$, it is obvious that the MMR process of Algorithm 1 actually greedily searches the solution for optimizing the problem of Equation (2).

In learning, the R-LTR algorithm optimizes the log likelihood of ground truth rankings derived from the training data:

$$\max_{\omega_r,\omega_d} \sum_{m=1}^{M} F(X_m, R_m, \mathbf{y}_m^*),$$

where $\mathbf{y}_m^*$ is the ground truth ranking for the $m$-th query.

## 4. A GENERAL FRAMEWORK FOR DIRECTLY OPTIMIZING DIVERSITY EVALUATION MEASURES

In this section, we describe the general framework for learning the diverse ranking model via directly optimizing diversity evaluation measures. The framework directly adopts the R-LTR model described in Section 3 which meets the criterion of MMR.

As for the loss functions, the framework first defines the basic loss function defined over diversity evaluation measures. Different loss functions upper bounding the basic loss function are then defined and optimized, achieving different diverse learning to rank algorithms.

### 4.1. The Basic Loss Function

Suppose we are given $N$ labeled training queries $\{(X^{(n)}, R^{(n)}, J^{(n)})\}_{n=1}^{N}$, where $J^{(n)}$ denotes the human labels on the documents in the form of a binary matrix. $J^{(n)}(i, s) = 1$ if document $\mathbf{x}_i^{(n)}$ contains the $s$-th subtopic of $q_n$ and 0 otherwise.[1] The learning process, thus, amounts to minimizing the loss over all of the training queries:

$$\min_{\omega_r,\omega_d} \sum_{n=1}^{N} L\big(\hat{\mathbf{y}}^{(n)}, J^{(n)}\big), \tag{4}$$

where $\hat{\mathbf{y}}^{(n)}$ is the ranking constructed by the diverse learning to rank model (Algorithm 1) for documents $X^{(n)}$, and $L(\hat{\mathbf{y}}^{(n)}, J^{(n)})$ is the function for judging the "loss" of the predicted ranking $\hat{\mathbf{y}}^{(n)}$ compared with the human labels $J^{(n)}$.

An ideal learning algorithm for search result diversification would train a ranking model that could directly optimize the diversity evaluation measures with respect to the training data. That is, the loss function of Equation (4) can be written as

$$\min_{\omega_r,\omega_d} \sum_{n=1}^{N} \big(1 - E(X^{(n)}, \hat{\mathbf{y}}^{(n)}, J^{(n)})\big), \tag{5}$$

where $E$ is an evaluation measure for search result diversification such as $\alpha$-NDCG or D#-NDCG or the like. $\hat{\mathbf{y}}^{(n)}$ is the permutation selected for applying to $X^{(n)}$ by ranking model $F$. We refer to Equation (5) as the "basic loss function" and those methods that minimize the basic loss function as the "direct optimization approach" to search result diversification.

It is obvious that the basic loss function is hard to optimize because $E$ is not a smooth and convex function. We resort to optimizing the upper bound of the loss function under the framework of structured output prediction. According to Theorem 2 in Xu et al.

[1]Some datasets also use graded judgments. In this article, we assume that all labels are binary.

[2008], the loss function defined in Equation (5) can be upper bounded by the function defined over the ranking pairs:

$$\sum_{n=1}^{N} \max_{\substack{\mathbf{y}^+ \in \mathcal{Y}^{+(n)}; \\ \mathbf{y}^- \in \mathcal{Y}^{-(n)}}} \left(E(X^{(n)}, \mathbf{y}^+, J^{(n)}) - E(X^{(n)}, \mathbf{y}^-, J^{(n)})\right) \cdot [\![F(\mathbf{y}^+, X^{(n)}, R^{(n)}) \leq F(\mathbf{y}^-, X^{(n)}, R^{(n)})]\!] = \sum_{n=1}^{N} \max_{\substack{\mathbf{y}^+ \in \mathcal{Y}^{+(n)}; \\ \mathbf{y}^- \in \mathcal{Y}^{-(n)}}} \Delta E(\mathbf{y}^+, \mathbf{y}^-) \cdot [\![\Delta F(\mathbf{y}^+, \mathbf{y}^-) \leq 0]\!], \tag{6}$$

where $\mathcal{Y}^{+(n)}$ is the set of all possible *positive* rankings for the $n$-th query, $\mathcal{Y}^{-(n)}$ is the set of all possible *negative* rankings for the $n$-the query (note that *positive* rankings and *negative* rankings are relative to each other, which means that the score of *positive* rankings is higher than of *negative* rankings), and $[\![\cdot]\!]$ is 1 if the condition is satisfied, otherwise zero. Here, $\Delta E(\mathbf{y}^+, \mathbf{y}^-) = E(X^{(n)}, \mathbf{y}^+, J^{(n)}) - E(X^{(n)}, \mathbf{y}^-, J^{(n)})$ and $\Delta F(\mathbf{y}^+, \mathbf{y}^-) = F(\mathbf{y}^+, X^{(n)}, R^{(n)}) - F(\mathbf{y}^-, X^{(n)}, R^{(n)})$ are the differences between the positive ranking $\mathbf{y}^+$ and negative ranking $\mathbf{y}^-$ in terms of the evaluation measure $E$ and ranking model $F$, respectively.

Practically, in order to leverage existing optimization technologies like Perceptron and stochastic gradient descent, bound optimization has been widely used. We can consider upper bounds over Equation (6). Different diverse learning to rank algorithms can be derived by defining different upper bounds and adopting different optimization techniques.

### 4.2. Upper Bounds on the Basic Loss Function

We show that the loss function in Equation (6) can be upper bounded by the following loss functions.

(1) We can get rid of the max by replacing it with the sum function. This is because $\sum_i x_i \geq \max_i x_i$ if $x_i \geq 0$ holds for all $i$. Thus, Equation (6) can be upper bounded by

$$\sum_{n=1}^{N} \sum_{\substack{\mathbf{y}^+ \in \mathcal{Y}^{+(n)}; \\ \mathbf{y}^- \in \mathcal{Y}^{-(n)}}} \Delta E(\mathbf{y}^+, \mathbf{y}^-) \cdot [\![\Delta F(\mathbf{y}^+, \mathbf{y}^-) \leq 0]\!]. \tag{7}$$

(2) Moving the term $\Delta E(\mathbf{y}^+, \mathbf{y}^-)$ into $[\![\cdot]\!]$ as margin, we get an upper bound on Equation (7):

$$\sum_{n=1}^{N} \sum_{\substack{\mathbf{y}^+ \in \mathcal{Y}^{+(n)}; \\ \mathbf{y}^- \in \mathcal{Y}^{-(n)}}} [\![\Delta F(\mathbf{y}^+, \mathbf{y}^-) \leq \Delta E(\mathbf{y}^+, \mathbf{y}^-))]\!]. \tag{8}$$

It can be shown that Equation (8) upper bounds the basic loss function in Equation (7) if $E \in [0, 1]$.

The preceding loss functions are still not continuous and differentiable because they contain the 0-1 function $[\![\cdot]\!]$, which is not continuous and differentiable. We can consider using continuous, differentiable, and even convex upper bounds, which are also upper bounds on the basic loss function in Equation (5).

(3) Replacing the 0-1 function $[\![\cdot]\!]$ with its upper bounds, such as the logistic function, exponential function, and hinge function, we can get three different upper bounds on Equation (7), as shown in Table I.

Table I. Upper Bounds on the 0-1 Loss Function $[\![\cdot]\!]$

| Name | Upper bound |
|---|---|
| Logistic function | $\sum_{n=1}^{N}\sum_{\mathbf{y}^+\in\mathcal{Y}^{+(n)};\mathbf{y}^-\in\mathcal{Y}^{-(n)}}\Delta E(\mathbf{y}^+,\mathbf{y}^-)\cdot\log\left(1+e^{-\Delta F(\mathbf{y}^+,\mathbf{y}^-)}\right)$ |
| Exponential function | $\sum_{n=1}^{N}\sum_{\mathbf{y}^+\in\mathcal{Y}^{+(n)};\mathbf{y}^-\in\mathcal{Y}^{-(n)}}\Delta E(\mathbf{y}^+,\mathbf{y}^-)\cdot e^{-\Delta F(\mathbf{y}^+,\mathbf{y}^-)}$ |
| Hinge function | $\sum_{n=1}^{N}\sum_{\mathbf{y}^+\in\mathcal{Y}^{+(n)};\mathbf{y}^-\in\mathcal{Y}^{-(n)}}\Delta E(\mathbf{y}^+,\mathbf{y}^-)\cdot\left[1-\Delta F(\mathbf{y}^+,\mathbf{y}^-)\right]_+$ |

Note that the relaxations in Equations (1), (2), and (3) can be applied separately or simultaneously, leading to different upper bounds on the basic loss function.

## 5. DERIVED ALGORITHMS

In this section, without loss of generality, we show several diverse ranking algorithms that optimize different upper bounds over the basic loss function. All these upper bounds are those shown in Section 4.2.

### 5.1. Perceptron Algorithm with Measures as Margin

Optimizing the upper bound shown in Equation (7) with Perceptron [Collins 2002; Li et al. 2002], we can get the algorithm called Perceptron Algorithm with Measures as Margin (PAMM), as shown in Algorithm 2.

PAMM takes a training set $\{(X^{(n)}, R^{(n)}, J^{(n)})\}_{n=1}^{N}$ as input and takes the diversity evaluation measure $E$, learning rate $\eta$, number of positive rankings per query $\tau^+$, and number of negative rankings per query $\tau^-$ as parameters. For each query $q_n$, PAMM first generates $\tau^+$ positive rankings $PR^{(n)}$ and $\tau^-$ negative rankings $NR^{(n)}$ (Lines 2 and 3). $PR^{(n)}$ and $NR^{(n)}$ play as the random samples of $\mathcal{Y}^{+(n)}$ and $\mathcal{Y}^{-(n)}$, respectively. PAMM then optimizes the model parameters $\omega_r$ and $\omega_d$ iteratively in a stochastic manner over the ranking pairs: At each round, for each pair between a positive ranking and a negative ranking $(\mathbf{y}^+, \mathbf{y}^-)$, the gap of these two rankings in terms of the query-level ranking model $\Delta F = F(X, R, \mathbf{y}^+) - F(X, R, \mathbf{y}^-)$ is calculated based on current parameters $\omega_r$ and $\omega_d$ (Line 9). If $\Delta F$ is smaller than the margin in terms of evaluation measure $\Delta E = E(X, \mathbf{y}^+, J) - E(X, \mathbf{y}^-, J)$ (Line 10), the model parameters will be updated so that $\Delta F$ will be enlarged (Lines 11 and 12). The iteration continues until convergence. Finally, PAMM outputs the optimized model parameters $(\omega_r, \omega_d)$.

Next, we explain the key steps of PAMM in detail.

*5.1.1. Generating Positive and Negative Rankings.* In PAMM, it is hard to directly conduct the optimization over the sets of positive rankings $\mathcal{Y}^{+(n)}$ and negative rankings $\mathcal{Y}^{-(n)}$ because, in total, these two sets have $M!$ rankings if the candidate set contains $M$ documents. Thus, PAMM samples the rankings to reduce the training time.

For each training query, PAMM first samples a set of positive rankings. Algorithm 3 illustrates the procedure. Similar to the online ranking algorithm shown in Algorithm 1, the positive rankings are generated through a sequential document selection process, and the selection criteria is the diversity evaluation measure $E$. After generating the first positive ranking $\mathbf{y}^{(1)}$, the algorithm constructs other positive rankings based on $\mathbf{y}^{(1)}$ by randomly swapping the positions of two documents whose subtopic coverage is identical.

For each training query, PAMM also samples a set of negative rankings. Algorithm 4 shows the procedure. The algorithm simply generates random rankings iteratively. If the generated ranking is not a positive ranking and satisfies the user predefined

**ALGORITHM 2:** The PAMM Algorithm

**Input:** Training data $\{(X^{(n)}, R^{(n)}, J^{(n)})\}_{n=1}^{N}$, learning rate $\eta$, diversity evaluation measure $E$, number of positive rankings per query $\tau^+$, number of negative rankings per query $\tau^-$.
**Output:** Model parameters $(\omega_r, \omega_d)$

**for** $n = 1$ **to** $N$ **do**
  $PR^{(n)} \leftarrow$PositiveRankings($X^{(n)}$, $J^{(n)}$, $E$, $\tau^+$) {Algorithm 3}
  $NR^{(n)} \leftarrow$NegativeRankings($X^{(n)}$, $J^{(n)}$, $E$, $\tau^-$) {Algorithm 4}
**end for**
initialize $\{\omega_r, \omega_d\} \leftarrow$ random values in $[0, 1]$
**repeat**
  **for** $n = 1$ **to** $N$ **do**
    **for all** $\{\mathbf{y}^+, \mathbf{y}^-\} \in PR^{(n)} \times NR^{(n)}$ **do**
      $\Delta F \leftarrow F(X^{(n)}, R^{(n)}, \mathbf{y}^+) - F(X^{(n)}, R^{(n)}, \mathbf{y}^-)$
      $\{F(X, R, \mathbf{y})$ is defined in Equation (3)$\}$
      **if** $\Delta F \leq E(X^{(n)}, \mathbf{y}^+, J^{(n)}) - E(X^{(n)}, \mathbf{y}^-, J^{(n)})$ **then**
        calculate $\nabla\omega_r^{(n)}$ and $\nabla\omega_d^{(n)}$ {Equation (9) and Equation (11)}
        $(\omega_r, \omega_d) \leftarrow (\omega_r, \omega_d) + \eta \times (\nabla\omega_r^{(n)}, \nabla\omega_d^{(n)})$
      **end if**
    **end for**
  **end for**
**until** convergence
**return** $(\omega_r, \omega_d)$

**ALGORITHM 3:** PositiveRankings

**Input:** Documents $X$, diversity labels $J$, evaluation measure $E$, and the number of positive rankings $\tau^+$
**Output:** Positive rankings $PR$

**for** $r = 1$ **to** $|X|$ **do**
  $y^{(1)}(r) \leftarrow \arg\max_{j:\mathbf{x}_j \in X \setminus S_{r-1}} E\left(S_{r-1} \cup \{\mathbf{x}_j\}, \left(y^{(1)}(1), \ldots, y^{(1)}(r-1), j\right), J\right)$
  $S_r \leftarrow S_{r-1} \cup \{\mathbf{x}_{y^{(1)}(r)}\}$
**end for**
$PR \leftarrow \{\mathbf{y}^{(1)}\}$
**while** $|PR| < \tau^+$ **do**
  $\mathbf{y} \leftarrow \mathbf{y}^{(1)}$
  $(k, l) \leftarrow$ randomly choose two documents whose human labels are identical, i.e., $J(y(k)) = J(y^{(1)}(l))$
  $y(k) \leftrightarrow y(l)$ {swap documents at rank $k$ and $l$}
  **if** $\mathbf{y} \notin PR$ **then**
    $PR \leftarrow PR \cup \{\mathbf{y}\}$
  **end if**
**end while**
**return** $PR$

constraints (e.g., $\alpha$-NDCG@20 $\leq 0.8$), the ranking will be added into the ranking set $NR$.

Note that, in some extreme cases, Algorithm 3 and Algorithm 4 cannot create enough rankings. In our implementations, the algorithms are forced to return after running enough iterations.

**ALGORITHM 4:** NegativeRankings

**Input:** Documents $X$, diversity labels $J$, evaluation measure $E$, and number of negative rankings $\tau^-$
**Output:** $NR$

```
NR = φ
while |NR| < τ⁻ do
    y ← random shuffle (1, ..., |X|)
    if y ∉ NR and E(X, y, J) is as expected then
        NR ← NR ∪ {y}
    end if
end while
return NR
```

*5.1.2. Updating $\omega_r$ and $\omega_d$ for PAMM.* Given a ranking pair $(\mathbf{y}^+, \mathbf{y}^-) \in PR^{(n)} \times NR^{(n)}$, PAMM updates $\omega_r$ and $\omega_d$ as

$$\omega_r \leftarrow \omega_r + \eta \times \nabla\omega_r,$$
$$\omega_d \leftarrow \omega_d + \eta \times \nabla\omega_d,$$

if $F(X, R, \mathbf{y}^+) - F(X, R, \mathbf{y}^-) \leq E(X, \mathbf{y}^+, J) - E(X, \mathbf{y}^-, J)$. The goal of the update is to enlarge the margin between $\mathbf{y}^+$ and $\mathbf{y}^-$ in terms of query level model: $\Delta F = F(X, R, \mathbf{y}^+) - F(X, R, \mathbf{y}^-)$. $\nabla\omega_r$ can be calculated as the gradient:

$$\nabla\omega_r = \frac{\partial F(X, R, \mathbf{y}^+)}{\partial \omega_r} - \frac{\partial F(X, R, \mathbf{y}^-)}{\partial \omega_r}, \tag{9}$$

where

$$\begin{aligned}\frac{\partial F(X, R, \mathbf{y})}{\partial \omega_r} &= \frac{\partial \sum_{j=1}^{|X|-1} \log \Pr(\mathbf{x}_{y(j)} | X \backslash S_{j-1}, R)}{\partial \omega_r} \\ &= \sum_{j=1}^{|X|-1} \left\{ \mathbf{x}_{y(j)} - \frac{\sum_{k=j}^{|X|} \mathbf{x}_{y(k)} e^{f_{S_{j-1}}(\mathbf{x}_{y(k)}, R_{y(k)})}}{\sum_{k=j}^{|X|} e^{f_{S_{j-1}}(\mathbf{x}_{y(k)}, R_{y(k)})}} \right\}. \end{aligned} \tag{10}$$

Similarly, $\nabla\omega_d$ can be calculated as

$$\nabla\omega_d = \frac{\partial F(X, R, \mathbf{y}^+)}{\partial \omega_d} - \frac{\partial F(X, R, \mathbf{y}^-)}{\partial \omega_d}, \tag{11}$$

where

$$\frac{\partial F(X, R, \mathbf{y})}{\partial \omega_d} = \sum_{j=1}^{|X|-1} \left\{ h_{S_{j-1}}(R_{y(j)}) - \frac{\sum_{k=j}^{|X|} h_{S_{j-1}}(R_{y(k)}) e^{f_{S_{j-1}}(\mathbf{x}_{y(k)}, R_{y(k)})}}{\sum_{k=j}^{|X|} e^{f_{S_{j-1}}(\mathbf{x}_{y(k)}, R_{y(k)})}} \right\}. \tag{12}$$

Intuitively, the gradients $\nabla\omega_r$ and $\nabla\omega_d$ are calculated so that Line 12 of Algorithm 2 will increase $F(X, R, \mathbf{y}^+)$ and decrease $F(X, R, \mathbf{y}^-)$.

### 5.2. Algorithms that Optimize the Logistic and Exponential Upper Bounds

Optimizing the logistic and exponential upper bounds in Table I with stochastic gradient descent, we can get different algorithms that directly optimize the diversity evaluation measures, referred to as the Stochastic Gradient Descent with Measure as Margin and Logistic (SGDMM-Log) function as upper bound and Stochastic Gradient Descent with Measure as Margin and Exponential (SGDMM-Exp) function as upper bound, respectively. The procedures of SGDMM-Log and SGDMM-Exp are shown in

**ALGORITHM 5:** The SGDMM-Log/SGDMM-Exp Algorithms

**Input:** Training data $\{(X^{(n)}, R^{(n)}, J^{(n)})\}_{n=1}^{N}$, learning rate $\eta$, diversity evaluation measure $E$, number of positive rankings per query $\tau^+$, number of negative rankings per query $\tau^-$.
**Output:** Model parameters $(\omega_r, \omega_d)$

1: **for** $n = 1$ **to** $N$ **do**
2: $PR^{(n)} \leftarrow$PositiveRankings($X^{(n)}$, $J^{(n)}$, $E$, $\tau^+$) {Algorithm 3}
3: $NR^{(n)} \leftarrow$NegativeRankings($X^{(n)}$, $J^{(n)}$, $E$, $\tau^-$) {Algorithm 4}
4: **end for**
5: initialize $\{\omega_r, \omega_d\} \leftarrow$ random values in $[0, 1]$
6: **repeat**
7: **for** $n = 1$ **to** $N$ **do**
8: **for all** $\{\mathbf{y}^+, \mathbf{y}^-\} \in PR^{(n)} \times NR^{(n)}$ **do**
9: calculate $\nabla\omega_r^{(n)}$ and $\nabla\omega_d^{(n)}$ {For SGDMM-Log, calculating with Equation (13) and Equation (14); for SGDMM-Exp, calculating with Equation (15) and Equation (16)}
10: $(\omega_r, \omega_d) \leftarrow (\omega_r, \omega_d) - \eta \times (\nabla\omega_r^{(n)}, \nabla\omega_d^{(n)})$
11: **end for**
12: **end for**
13: **until** convergence
14: **return** $(\omega_r, \omega_d)$

Algorithm 5. Here, we show both in the same Algorithm 5 because the only difference is the method for calculating the gradients $\omega_r$ and $\omega_d$ (Line 9).

The procedures shown in Algorithm 5 are almost identical to the PAMM algorithm in Algorithm 2, except that (i) in PAMM, the gradients are calculated only for those training pairs that satisfy $\Delta F(\mathbf{y}^+, \mathbf{y}^-) \leq \Delta E(\mathbf{y}^+, \mathbf{y}^-)$ (Line 10). In SGDMM-Log and SGDMM-Exp, however, the gradients are calculated for all ranking pairs; and (ii) PAMM, SGDMM-Log, and SGDMM-Exp make use of different methods for calculating the gradients (Line 11 in Algorithm 2 and Line 9 in Algorithm 5).

*5.2.1. Updating $\omega_r$ and $\omega_d$ for SGDMM-Log.* According to the logistic loss in Table I, given a ranking pair $(\mathbf{y}^+, \mathbf{y}^-) \in PR^{(n)} \times NR^{(n)}$, SGDMM-Log updates $\omega_r$ and $\omega_d$ as

$$\omega_r \leftarrow \omega_r + \eta \times \nabla\omega_r,$$
$$\omega_d \leftarrow \omega_d + \eta \times \nabla\omega_d,$$

where

$$\nabla\omega_r = \Delta E(\mathbf{y}^+, \mathbf{y}^-) \cdot \frac{e^{-\Delta F(\mathbf{y}^+, \mathbf{y}^-)}}{1 + e^{-\Delta F(\mathbf{y}^+, \mathbf{y}^-)}} \cdot \left( \frac{\partial F(X, R, \mathbf{y}^-)}{\partial \omega_r} - \frac{\partial F(X, R, \mathbf{y}^+)}{\partial \omega_r} \right), \tag{13}$$

where $\frac{\partial F(X,R,\mathbf{y})}{\partial \omega_r}$ is calculated as in Equation (10).

Similarly, $\nabla\omega_d$ can be calculated as

$$\nabla\omega_d = \Delta E(\mathbf{y}^+, \mathbf{y}^-) \cdot \frac{e^{-\Delta F(\mathbf{y}^+, \mathbf{y}^-)}}{1 + e^{-\Delta F(\mathbf{y}^+, \mathbf{y}^-)}} \cdot \left( \frac{\partial F(X, R, \mathbf{y}^-)}{\partial \omega_d} - \frac{\partial F(X, R, \mathbf{y}^+)}{\partial \omega_d} \right), \tag{14}$$

where $\frac{\partial F(X,R,\mathbf{y})}{\partial \omega_d}$ is defined in Equation (12).

*5.2.2. Updating $\omega_r$ and $\omega_d$ for SGDMM-Exp.* Similarly, according to the exponential loss in Table I, given a ranking pair $(\mathbf{y}^+, \mathbf{y}^-) \in PR^{(n)} \times NR^{(n)}$, SGDMM-Exp updates $\omega_r$ and $\omega_d$ as

$$\omega_r \leftarrow \omega_r + \eta \times \nabla\omega_r,$$
$$\omega_d \leftarrow \omega_d + \eta \times \nabla\omega_d,$$

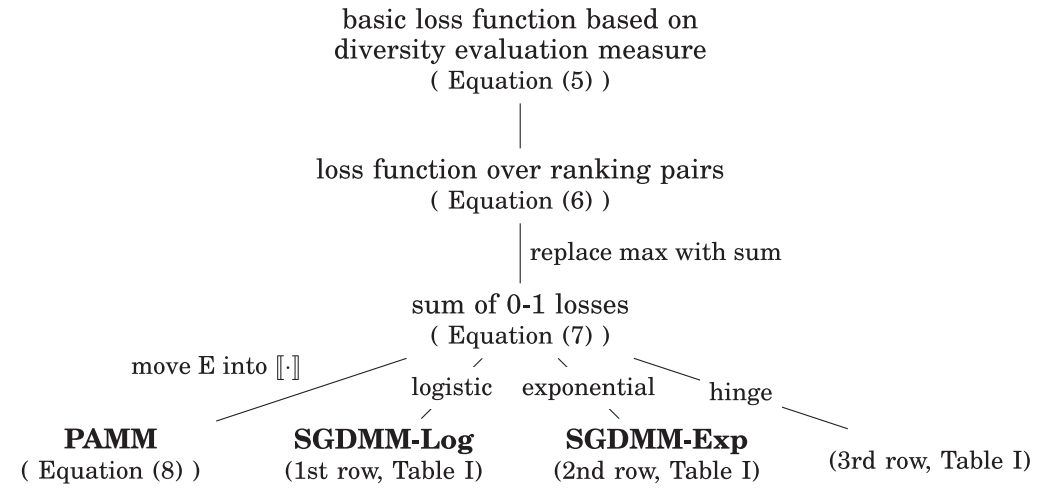


Fig. 1. Summary on upper bounds.

where

$$\nabla\omega_r = \Delta E(\mathbf{y}^+, \mathbf{y}^-) \cdot e^{-\Delta F(\mathbf{y}^+, \mathbf{y}^-)} \cdot \left( \frac{\partial F(X, R, \mathbf{y}^-)}{\partial \omega_r} - \frac{\partial F(X, R, \mathbf{y}^+)}{\partial \omega_r} \right), \tag{15}$$

where $\frac{\partial F(X,R,\mathbf{y})}{\partial \omega_r}$ is defined in Equation (10).

Similarly, $\nabla\omega_d$ can be calculated as

$$\nabla\omega_d = \Delta E(\mathbf{y}^+, \mathbf{y}^-) \cdot e^{-\Delta F(\mathbf{y}^+, \mathbf{y}^-)} \cdot \left( \frac{\partial F(X, R, \mathbf{y}^-)}{\partial \omega_d} - \frac{\partial F(X, R, \mathbf{y}^+)}{\partial \omega_d} \right), \tag{16}$$

where $\frac{\partial F(X,R,\mathbf{y})}{\partial \omega_d}$ is defined in Equation (12).

## 6. ANALYSIS

### 6.1. Summary on Upper Bounds

Here, we give a summary of the upper bounds on the basic loss function and different optimization methods. Figure 1 shows the relationships. To maximize the diverse ranking accuracy in terms of a diversity evaluation measure, the basic loss function is defined as the sum of 1 minus the performance of each query (Equation (5)). The basic loss function can be upper bounded by a loss function over possible ranking pairs, as shown in Equation (6). Replacing the max operation with sum, Equation (6) can be further upper bounded by the sum of 0-1 losses, as shown in Equation (7). Based on Equation (7), different upper bounds can derived, each corresponds to one algorithm. For example, moving $E$ into $[\![\cdot]\!]$, we get the loss function for PAMM (Equation (8)). Replacing $[\![\cdot]\!]$ with its upper bounds of logistic function and exponential function, we get the loss function of SGDMM-Log and SGDMM-Exp, respectively.

### 6.2. Advantages

The proposed diverse ranking framework in this paper is simple yet powerful for search result diversification. It provides a general method for directly optimizing the diversity evaluation measures in learning diversity ranking models. New algorithms can be

easily derived under the framework. Compared with the existing learning methods such as R-LTR [Zhu et al. 2014], SVM-DIV [Yue and Joachims 2008], and structural SVM [Liang et al. 2014], the algorithms derived from the framework have several advantages.

First, these algorithms employ a more reasonable ranking model. The relational ranking model follows the MMR criterion and thus can be mimic the process of sequential document selection, which naturally meets search user behaviors. In contrast, existing approaches to search result diversification, such as structural SVM approaches [Liang et al. 2014], calculate all ranking scores within a single step, that of in relevance ranking. The marginal relevance of each document cannot be taken into consideration at ranking time.

Second, these algorithms have the ability to incorporate any diversity evaluation measure in training, which makes the algorithm focus on the specified measure when updating the model parameters. In contrast, existing algorithms such as R-LTR only minimizes a loss function that is loosely related to diversity evaluation measures, and SVM-DIV is trained to optimize the subtopic coverage.

Third, these algorithms utilize the pairs between the positive rankings and the negative rankings in training, which makes it possible to leverage more information in training. Specifically, the algorithms enlarge the margins between the positive rankings and negative rankings when updating the parameters. In contrast, R-LTR only uses the information in the positive rankings, and the training is aimed at maximizing the likelihood.

### 6.3. Comparison of the Derived Algorithms

PAMM optimizes a modified 0-1 loss in which the difference of the two rankings is used as the margin. The penalty to a ranking pair will be zero if the model correctly predicts the ranking pair with some high confidence; otherwise, 1. Since the penalty to one pair is upper bounded by 1, PAMM is not sensitive to data noise. However, the 0-1 loss function is non-smooth and non-convex. Thus, PAMM makes use of the Perceptron updating rules in training. At each iteration, PAMM updates the model parameters only if the ranking pairs are not correctly predicted or the confidence is not large enough. After a few training iterations, most ranking pairs will be correctly predicted. This property makes PAMM perform faster than SGDMM-Log and SGDMM-Exp.

SGDMM-Log optimizes a logistic loss function with stochastic gradient descent. The logistic loss function is smooth and convex. It does not assign zero penalty to any ranking pairs. The ranking model that correctly predicts a ranking pair with high confidence is penalized less. This leads SGDMM-Log to be more sensitive to outliers in the data than PAMM.

SGDMM-Exp optimizes an exponential loss function with stochastic gradient descent. The exponential loss function is also smooth and convex. Compared with the 0-1 loss and the logistic loss, the exponential loss penalizes incorrect prediction of a ranking pair very severely, whereas it penalizes almost nothing when the prediction is correct. This is a very desirable property in most cases. However, this property also means that SGDMM-Exp has the disadvantage of being very sensitive to noisy data since the learning process will spend more effort on the outliers and tend to be vulnerable to noisy data.

## 7. EXPERIMENTS

We conducted experiments to test the performances of the example algorithms (PAMM, SGDMM-Log, and SGDMM-Exp) derived under the framework of directly optimizing diversity evaluation measures.

Table II. Statistics on WT2009, WT2010, and WT2011

| Dataset | #queries | #labeled docs | #subtopics per query |
|---|---|---|---|
| WT2009 | 50 | 5149 | 3∼8 |
| WT2010 | 48 | 6554 | 3∼7 |
| WT2011 | 50 | 5000 | 2∼6 |

### 7.1. Experimental Settings

*7.1.1. Datasets and Evaluation Measures.* Three TREC benchmark datasets for diversity tasks are used: TREC 2009 Web Track (WT2009), TREC 2010 Web Track (WT2010), and TREC 2011 Web Track (WT2011). Each dataset consists of queries, corresponding retrieved documents, and human-judged labels. Each query includes several subtopics identified by TREC assessors. The document relevance labels were made at the subtopic level, and the labels are binary.[2] Statistics on the datasets are given in Table II.

All experiments were carried out on the ClueWeb09 Category B data collection,[3] which comprises 50 million English web documents. Porter stemming, tokenization, and stop-words removal (using the INQUERY list) were applied to the documents as preprocessing. We conducted 5-fold cross-validation experiments on the three datasets. For each dataset, we randomly split the queries into five even subsets. At each fold, three subsets were used for training, one was used for validation, and one was used for testing. The results reported were the average over the five trials.

The current official evaluation metrics of the diversity task, ERR-IA [Chapelle et al. 2009], $\alpha$-NDCG [Clarke et al. 2008], and NRBP [Clarke et al. 2009] were adopted in our experiments. They measure the diversity of a result list by explicitly rewarding novelty and penalizing redundancy observed at every rank. The associated parameters $\alpha$ and $\beta$ are all set to 0.5, which is consistent with the default settings in the official TREC evaluation program. D#-NDCG [Sakai and Song 2011], which encourages high intent recall in a search output within the D-measure framework [Sakai et al. 2010], is also used in our experiments. The parameter $\gamma$ in D#-NDCG is set to 0.5. In all measures, the scores are computed over the top-$k$ search results ($k = 20$).

*7.1.2. Baselines.* We compared PAMM, SGDMM-Log, and SGDMM-Exp with several types of baselines. The baselines include the conventional relevance ranking models in which document novelty is not taken into consideration:

**Query likelihood (QL)** [Manning et al. 2008]: Language models for information retrieval.

**ListMLE** [Liu 2009; Li 2014]: A representative learning-to-rank model for information retrieval.

We also compared PAMM, SGDMM-Log, and SGDMM-Exp with three heuristic approaches to search result diversification in the experiments:

**MMR** [Carbonell and Goldstein 1998]: A heuristic approach to search result diversification in which the document ranking is constructed via iteratively selecting the document with the MMR.

**xQuAD** [Santos et al. 2010]: A representative heuristic approach to search result diversification.

**PM-2** [Dang and Croft 2012]: Another widely used heuristic approach to search result diversification.

Note that these three baselines require a prior relevance function to implement their diversification steps. In our experiments, ListMLE was chosen as the relevance function.

[2]WT2011 has graded judgments. In this article, we treat them as binary.
[3]http://boston.lti.cs.cmu.edu/data/clueweb09.

Table III. Relevance Features Used in the Experiments

| Name | Description | # Features |
|---|---|---|
| TF-IDF | The tf-idf model | 5 |
| BM25 | BM25 with default parameters | 5 |
| LMIR | LMIR with Dirichlet smoothing | 5 |
| MRF [Metzler and Croft 2005] | MRF with ordered/unordered phrase | 10 |
| PageRank | PageRank score | 1 |
| #inlinks | number of inlinks | 1 |
| #outlinks | number of outlinks | 1 |

The first four lines are query-document matching features, each applied to the fields of body, anchor, title, URL, and the whole documents. The latter three lines are document quality features. Zhu et al. [2014].

Learning approaches to search result diversification are also used as baselines in the experiments:

**SVM-DIV** [Yue and Joachims 2008]: A representative learning approach to search result diversification. It utilizes structural SVMs to optimize the subtopic coverage. SVM-DIV does not consider relevance. For fair performance comparison, in the baseline, we first apply ListMLE to capture relevance and then apply SVM-DIV to re-rank the top-K retrieved documents.

**Structural SVM** [Tsochantaridis et al. 2005]: Structural SVM can be configured to directly optimize diversity evaluation measures, as shown in Liang et al. [2014]. In this article, we used structural SVM to optimize $\alpha$-NDCG@20 and ERR-IA@20, denoted as StructSVM($\alpha$-NDCG) and StructSVM(ERR-IA), respectively.

**R-LTR** [Zhu et al. 2014]: A state-of-the-art learning approach to search result diversification. The ranking function is a linear combination of the relevance score and novelty score between the current document and those previously selected. Following the practice in Zhu et al. [2014], in our experiments, we used the results of R-LTR$_{min}$ which defines the relation function $h_S(R)$ as the minimal distance.

*7.1.3. Parameters Settings.* PAMM, SGDMM-Log, and SGDMM-Exp have some parameters. The learning rate parameter $\eta$ was tuned based on the validation set during each experiment. In all of the experiments in this subsection, we set the number of positive rankings per query $\tau^+ = 5$ and number of negative rankings per query $\tau^- = 20$. As for the parameter $E$, $\alpha$-NDCG@20, ERR-IA@20, and D#-NDCG@20 were utilized, denoted as PAMM($\alpha$-NDCG), PAMM(ERR-IA), PAMM(D#-NDCG), SGDMM-Log($\alpha$-NDCG), SGDMM-Log(ERR-IA), SGDMM-Log(D#-NDCG), SGDMM-Exp($\alpha$-NDCG), SGDMM-Exp(ERR-IA), and SGDMM-Exp(D#-NDCG), respectively.

*7.1.4. Features.* We adopted the features used in the work of R-LTR [Zhu et al. 2014]. There are two types of features: relevance features, which capture the relevance information of a query with respect to a document, and novelty features, which represent the relation information among documents. Tables III and IV list the relevance and novelty features used in the experiments, respectively.

### 7.2. Experimental Results

The experimental results on WT2009, WT2010, and WT2011 are reported in Tables V, VI, and VII, respectively. Boldface indicates the highest score in terms of the corresponding evaluation measure. From the results, we can see that all of our methods (PAMM($\alpha$-NDCG), PAMM(ERR-IA), PAMM(D#-NDCG), SGDMM-Log($\alpha$-NDCG), SGDMM-Log(ERR-IA), SGDMM-Log(D#-NDCG), SGDMM-Exp($\alpha$-NDCG), SGDMM-Exp(ERR-IA), and SGDMM-Exp(D#-NDCG)) outperform all of the baselines on all three datasets in terms of all evaluation measures. We conducted significant testing

Table IV. The Seven Novelty Features Used in the Experiments

| Name | Description |
|---|---|
| Subtopic Diversity | Euclidean distance based on PLSA [Hofmann 1999] |
| Text Diversity | Cosine-based distance on term vectors |
| Title Diversity | Text diversity on title |
| Anchor Text Diversity | Text diversity on anchor |
| ODP-Based Diversity | ODP[4] taxonomy-based distance |
| Link-Based Diversity | Link similarity of document pair |
| URL-Based Diversity | URL similarity of document pair |

Each feature is extracted over two documents. Zhu et al. [2014].

Table V. Performance Comparison of All Methods on WT2009

| Method | ERR-IA | $\alpha$-NDCG | D#-NDCG | NRBP |
|---|---|---|---|---|
| QL | 0.1637 | 0.2691 | 0.1445 | 0.1382 |
| ListMLE | 0.1913 | 0.3074 | 0.1571 | 0.1681 |
| MMR | 0.2022 | 0.3083 | 0.1863 | 0.1715 |
| xQuAD | 0.2316 | 0.3437 | 0.1899 | 0.1956 |
| PM-2 | 0.2294 | 0.3369 | 0.1858 | 0.1788 |
| SVM-DIV | 0.2408 | 0.3526 | 0.2121 | 0.2073 |
| StructSVM(ERR-IA) | 0.2613 | 0.3732 | 0.2134 | 0.2066 |
| StructSVM($\alpha$-NDCG) | 0.2602 | 0.3771 | 0.2127 | 0.2130 |
| StructSVM(D#-NDCG) | 0.2555 | 0.3756 | 0.2249 | 0.2091 |
| R-LTR | 0.2714 | 0.3964 | 0.2296 | 0.2339 |
| PAMM(ERR-IA) | **0.2945*** | 0.4229* | 0.2430* | 0.2363 |
| PAMM($\alpha$-NDCG) | 0.2842* | **0.4271*** | 0.2421* | 0.2411* |
| PAMM(D#-NDCG) | 0.2844* | 0.4241* | **0.2502*** | 0.2476* |
| SGDMM-Log(ERR-IA) | 0.2888* | 0.4152* | 0.2433* | **0.2531*** |
| SGDMM-Log($\alpha$-NDCG) | 0.2877* | 0.4221* | 0.2431* | 0.2340 |
| SGDMM-Log(D#-NDCG) | 0.2851* | 0.4096 | 0.2497* | 0.2409* |
| SGDMM-Exp(ERR-IA) | 0.2914* | 0.4007 | 0.2411 | 0.2373 |
| SGDMM-Exp($\alpha$-NDCG) | 0.2876* | 0.4142* | 0.2407 | 0.2400 |
| SGDMM-Exp(D#-NDCG) | 0.2880* | 0.4130* | 0.2489* | 0.2451* |

Boldface indicates the highest score and '*' indicates that the improvement over R-LTR is statistically significant (p-value $< 0.05$).

(t-test) on the improvements of our methods over the baseline of R-LTR (the best baseline) in terms of $\alpha$-NDCG@20, ERR-IA@20, D#-NDCG, and NRBP. The asterisks in Tables V, VI, and VII indicate that the corresponding improvements are statistically significant (p-value $< 0.05$). The results show that the algorithms derived from the framework of directly optimizing diversity evaluation measures are effective for the task of search result diversification.

From the results, we can see that all of the direct optimization methods derived under the framework perform better than the baselines. These methods themselves perform equally well on all three datasets. There exist no significant differences among them.

We also observed that, on all three datasets, the derived algorithms trained with $\alpha$-NDCG@20 performed best in terms of $\alpha$-NDCG@20, the algorithms trained with ERR-IA@20 performed best in terms of ERR-IA@20, and the algorithms trained with D#-NDCG performed best in terms of D#-NDCG. The results indicate that the algorithms derived under the framework can enhance diverse ranking performances in terms of a measure by using the measure in training. More detailed analysis of the phenomenon is reported in Section 7.3.2 and Appendix A.

[4]http://www.dmoz.org

Table VI. Performance Comparison of All Methods on WT2010

| Method | ERR-IA | $\alpha$-NDCG | D#-NDCG | NRBP |
|---|---|---|---|---|
| QL | 0.1980 | 0.3024 | 0.1678 | 0.1549 |
| ListMLE | 0.2436 | 0.3755 | 0.2221 | 0.1949 |
| MMR | 0.2735 | 0.4036 | 0.2345 | 0.2252 |
| xQuAD | 0.3278 | 0.4445 | 0.2679 | 0.2872 |
| PM-2 | 0.3296 | 0.4478 | 0.2673 | 0.2901 |
| SVM-DIV | 0.3331 | 0.4593 | 0.3164 | 0.2934 |
| StructSVM(ERR-IA) | 0.3557 | 0.4724 | 0.3379 | 0.2933 |
| StructSVM($\alpha$-NDCG) | 0.3521 | 0.4764 | 0.3393 | 0.2991 |
| StructSVM(D#-NDCG) | 0.3497 | 0.4722 | 0.3451 | 0.2947 |
| R-LTR | 0.3647 | 0.4924 | 0.3506 | 0.3293 |
| PAMM(ERR-IA) | **0.3876*** | 0.5119* | 0.3726* | 0.3333 |
| PAMM($\alpha$-NDCG) | 0.3802* | **0.5249*** | 0.3712* | **0.3431*** |
| PAMM(D#-NDCG) | 0.3811* | 0.5220* | 0.3793* | 0.3417* |
| SGDMM-Log(ERR-IA) | 0.3855* | 0.5100* | 0.3713* | 0.3411* |
| SGDMM-Log($\alpha$-NDCG) | 0.3825* | 0.5221* | 0.3716* | 0.3400* |
| SGDMM-Log(D#-NDCG) | 0.3856* | 0.5166* | 0.3758* | 0.3397* |
| SGDMM-Exp(ERR-IA) | 0.3862* | 0.5135* | 0.3741* | 0.3384 |
| SGDMM-Exp($\alpha$-NDCG) | 0.3826* | 0.5219* | 0.3700* | 0.3424* |
| SGDMM-Exp(D#-NDCG) | 0.3821* | 0.5149* | **0.3799*** | 0.3403* |

Boldface indicates the highest score and '*' indicates that the improvement over R-LTR is statistically significant (p-value $< 0.05$).

Table VII. Performance Comparison of All Methods on WT2011

| Method | ERR-IA | $\alpha$-NDCG | D#-NDCG | NRBP |
|---|---|---|---|---|
| QL | 0.3520 | 0.4531 | 0.3177 | 0.3123 |
| ListMLE | 0.4172 | 0.5169 | 0.3947 | 0.3887 |
| MMR | 0.4284 | 0.5302 | 0.4034 | 0.3913 |
| xQuAD | 0.4753 | 0.5645 | 0.4395 | 0.4274 |
| PM-2 | 0.4873 | 0.5786 | 0.4473 | 0.4318 |
| SVM-DIV | 0.4898 | 0.5910 | 0.4702 | 0.4475 |
| StructSVM(ERR-IA) | 0.5137 | 0.6134 | 0.4973 | 0.4574 |
| StructSVM($\alpha$-NDCG) | 0.5127 | 0.6179 | 0.4930 | 0.4630 |
| StructSVM(D#-NDCG) | 0.5121 | 0.6122 | 0.4988 | 0.4620 |
| R-LTR | 0.5389 | 0.6297 | 0.5082 | 0.4982 |
| PAMM(ERR-IA) | 0.5483* | 0.6373* | 0.5221* | 0.4981 |
| PAMM($\alpha$-NDCG) | 0.5417 | 0.6433* | 0.5211* | 0.5012 |
| PAMM(D#-NDCG) | 0.5466* | 0.6368 | **0.5260*** | 0.5000 |
| SGDMM-Log(ERR-IA) | **0.5499*** | 0.6359 | 0.5141 | 0.4999 |
| SGDMM-Log($\alpha$-NDCG) | 0.5423 | 0.6403* | 0.5208* | 0.4986 |
| SGDMM-Log(D#-NDCG) | 0.5476* | 0.6396* | 0.5233* | 0.5001 |
| SGDMM-Exp(ERR-IA) | 0.5477* | 0.6371* | 0.5218* | 0.4975 |
| SGDMM-Exp($\alpha$-NDCG) | 0.5416 | **0.6436*** | 0.5231* | 0.5008 |
| SGDMM-Exp(D#-NDCG) | 0.5468* | 0.6365 | 0.5220* | **0.5018** |

Boldface indicates the highest score and '*' indicates that the improvement over R-LTR is statistically significant (p-value $< 0.05$).

### 7.3. Discussions

We conducted experiments to show the reasons that the algorithms derived from the framework of directly optimizing diversity evaluation measures outperform the baselines, using the PAMM algorithm and results of the WT2009 dataset as examples. The experimental results of SGDMM-Log and SGDMM-Exp on the WT2009 dataset are shown in the Appendix.

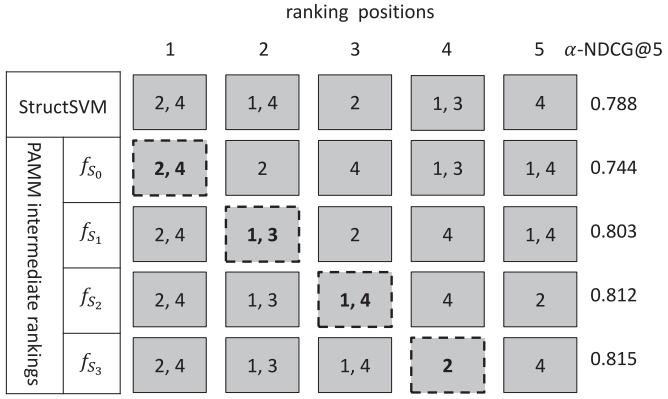


Fig. 2. Example rankings from WT2009. Each shaded block represents a document, and the number(s) in the block represent the subtopic(s) covered by the document.

*7.3.1. Effect of Maximizing Marginal Relevance.* We found that PAMM makes a good trade-off between the query-document relevance and document novelty via maximizing marginal relevance. Here, we use the result with regard to query number 24 ("diversity," which contains four subtopics) to illustrate why our method is superior to the baseline method of Structural SVM trained with $\alpha$-NDCG@20 (denoted as StructSVM($\alpha$-NDCG)). Note that structural SVM cannot leverage the marginal relevance in its ranking model. Figure 2 shows the top five ranked documents by StructSVM($\alpha$-NDCG), as well as four intermediate rankings generated by PAMM($\alpha$-NDCG) (denoted as $f_{S_0}$, $f_{S_1}$, $f_{S_2}$, and $f_{S_3}$). The ranking denoted as $f_{S_r}$ is generated by first sequentially selecting the documents for ranking positions of $1, 2, \ldots, r-1$ with models $f_{S_0}, f_{S_1}, \ldots, f_{S_{r-2}}$, respectively, then ranking the remaining documents with $f_{S_{r-1}}$. For example, the intermediate ranking denoted $f_{S_2}$ is generated by selecting one document with $f_{S_0}$ and setting it to rank 1, then selecting one document with $f_{S_1}$ and set it to rank 2, and finally ranking the remaining documents with $f_{S_2}$ and putting them at the tail of the list. Each of the shaded blocks indicates a document, and the number(s) in the block indicate the subtopic(s) assigned to the document by the human annotators. The performances in terms of $\alpha$-NDCG@5 are also shown in the last column. Here, we used $\alpha$-NDCG@5 because only the top five documents are shown.

The results in Figure 2 indicate the effectiveness of the MMR criterion. We can see that the $\alpha$-NDCG@5 increases steadily with increasing rounds of document selection iterations. In the first iteration, $f_{S_0}$ selects the most relevant document and puts it in the first position without considering the document novelty. Thus, the $\alpha$-NDCG@5 of the ranking generated by $f_{S_0}$ is lower than that of StructSVM($\alpha$-NDCG). In the second iteration, the ranking function $f_{S_1}$ selects the document associated with subtopics 1 and 3 and ranks it to the second position, according to the MMR criterion. From the viewpoint of diverse ranking, this is obviously a better choice than StructSVM($\alpha$-NDCG) made, which selects the document with subtopics 1 and 4. (Note that both Structural SVM and PAMM select the document with subtopics 2 and 4 for the first position.) In the following steps, $f_{S_2}$ and $f_{S_3}$ select documents for ranking positions of 3 and 4, also following MMR criterion. As a result, $f_{S_1}$, $f_{S_2}$, and $f_{S_3}$ outperform StructSVM($\alpha$-NDCG).

*7.3.2. Ability to Improve the Evaluation Measures.* We conducted experiments to see whether PAMM has the ability to improve the diverse ranking quality in terms of a measure by using the measure in training. Specifically, we trained models using

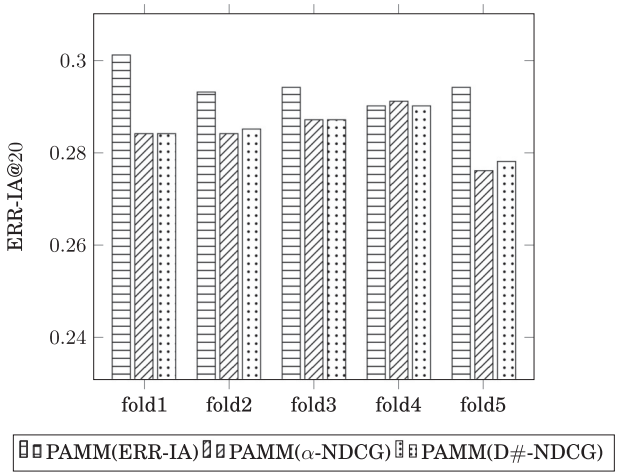


Fig. 3. Performance in terms of ERR-IA@20 when model is trained with ERR-IA@20, $\alpha$-NDCG@20, or D#-NDCG@20.

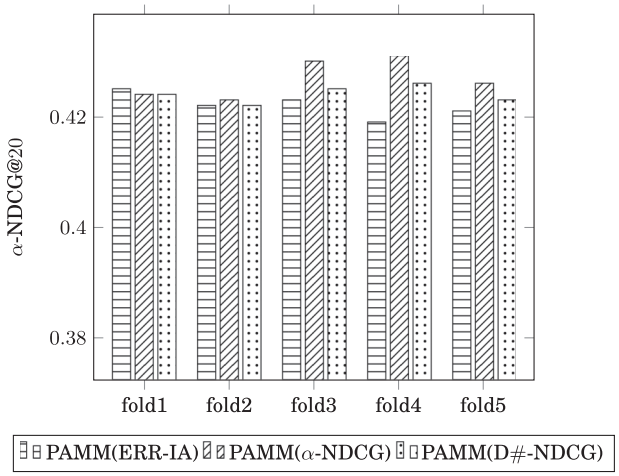


Fig. 4. Performance in terms of $\alpha$-NDCG@20 when model is trained with ERR-IA@20, $\alpha$-NDCG@20, or D#-NDCG@20.

ERR-IA@20, $\alpha$-NDCG@20, and D#-NDCG@20 and evaluated their accuracies on the test dataset in terms of ERR-IA@20, $\alpha$-NDCG@20, and D#-NDCG@20. The experiments were conducted for each fold of the cross-validation, and performances on each fold are reported. Figures 3, 4, and 5 show the results in terms of $\alpha$-NDCG@20, ERR-IA@20, and D#-NDCG@20, respectively. From Figure 3, we can see that on all five folds (except fold four), PAMM(ERR-IA) trained with ERR-IA@20 performs better in terms of ERR-IA@20. Similarly, from Figure 4, we can see that on all five folds (except fold four), PAMM($\alpha$-NDCG) trained with $\alpha$-NDCG@20 performs better in terms of $\alpha$-NDCG@20. From Figure 5, we can see that on all five folds, PAMM(D#-NDCG) trained with D#-NDCG@20 performs better in terms of D#-NDCG@20. Similar results have also been observed in the experiments on other datasets (see the results in Tables V, VI, and VII) and in the experiments for the other derived algorithms of SGDMM-Log and SGDMM-Exp (see the results in Appendix A). All results indicate that the algorithms derived

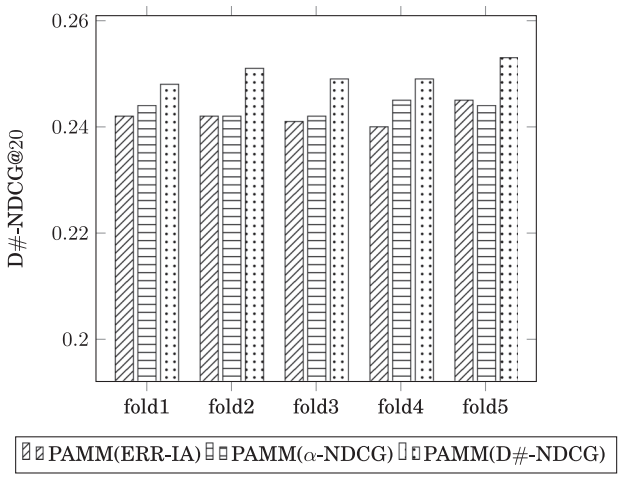


Fig. 5. Performance in terms of D#-NDCG@20 when model is trained with ERR-IA@20, $\alpha$-NDCG@20, or D#-NDCG@20.

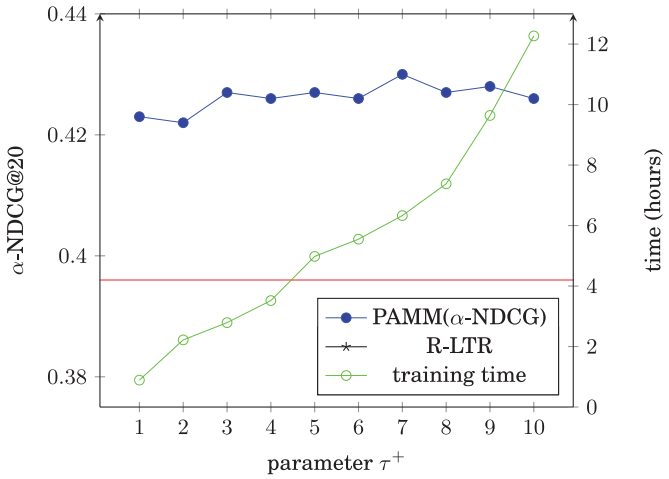


Fig. 6. Ranking accuracies and training time with respect to $\tau^+$.

under the proposed framework can indeed enhance diverse ranking quality in terms of a measure by using the measure in training.

*7.3.3. Effects of Positive and Negative Rankings.* We examined the effects of the number of positive rankings generated per query (parameter $\tau^+$). Specifically, we compared the performances of PAMM($\alpha$-NDCG) with respect to different $\tau^+$ values. Figure 6 shows the performance curve in terms of $\alpha$-NDCG@20. The performance of R-LTR baseline is also shown for reference. From the result, we can see that the curve does not change much with different $\tau^+$ values, which indicates the robustness of PAMM. Figure 6 also shows training time (in hours) with respect to different $\tau^+$ values. The training time increased dramatically with large $\tau^+$ because more ranking pairs are generated for training. In our experiments, $\tau^+$ was set to 5.

We further examined the effect of the number of negative rankings per query (parameter $\tau^-$). Specifically, we compared the performances of PAMM($\alpha$-NDCG) with respect

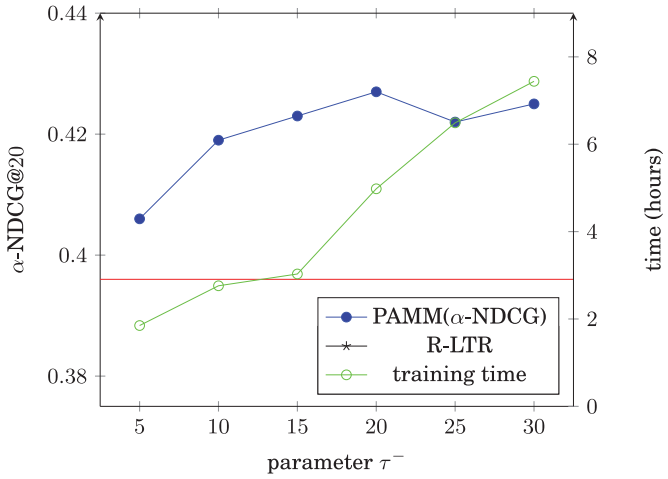


Fig. 7. Ranking accuracies and training time with respect to $\tau^-$.

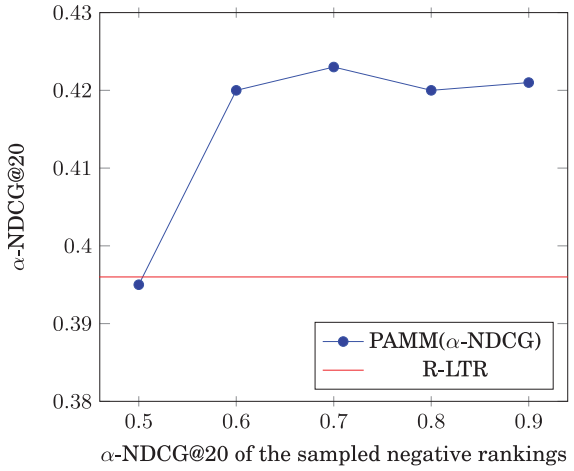


Fig. 8. Ranking accuracies with respect to different $\alpha$-NDCG@20 values of the negative rankings.

to different $\tau^-$, and the results are shown in Figure 7. From the results, we can see that the performance of PAMM increased steadily with increasing $\tau^-$ values until $\tau^- = 20$, which indicates that PAMM can achieve better ranking performance with more information from the negative rankings. As the cost, the training time increased dramatically because more training instances are involved. In our experiments, $\tau^-$ was set to 20.

We also conducted experiments to show the effect of sampling the negative rankings with different $\alpha$-NDCG values. Specifically, in each experiment, we configured Algorithm 4 to choose the negative rankings whose $\alpha$-NDCG@20 values are 0.5, 0.6, 0.7, 0.8, and 0.9, respectively. Figure 8 shows the performances of PAMM($\alpha$-NDCG) with respect to different $\alpha$-NDCG@20 values of the sampled negative rankings. From the results, we can see that PAMM performs best when the $\alpha$-NDCG@20 of the sampled negative rankings ranges from 0.6 to 0.9. The results also indicate that PAMM (and

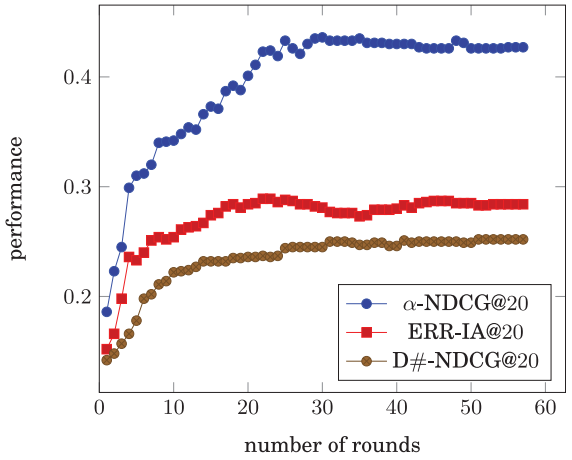


Fig. 9. Learning curve of PAMM($\alpha$-NDCG).

the algorithms derived under the proposed framework) is robust and not very sensitive to different methods of sampling the negative rankings.

*7.3.4. Improving the User Experience.* We conducted A/B testing to check whether the derived algorithms can actually improve user experience. Specifically, for each of the 50 queries from WT2009, the rankings generated by the R-LTR model (ranking A) and the rankings generated by the PAMM($\alpha$-NDCG) (ranking B) were shown to 19 annotators simultaneously. Each annotator compared these two rankings and selected a label of "win," "draw," or "loss" based on user experience. Note that the names of the two ranking models are anonymized before being sent to the annotators. Table VIII shows the results, and each line correspond to a query. The query string, the $\alpha$-NDCG@20 of R-LTR, the $\alpha$-NDCG@20 of PAMM($\alpha$-NDCG), the number of annotators who think PAMM won, the number of annotators who think PAMM and R-LTR are equally good, and the number of annotators who think PAMM lost are shown in the table. From the results, we can see that the user annotations match well with $\alpha$-NDCG@20, which indicates that $\alpha$-NDCG@20 is an effective evaluation measure for simulating user satisfaction judgments. Among the 950 judgements (50 queries and 19 judgments per query), 524 voted for a PAMM win, 173 voted for a PAMM loss, and 253 voted for a draw. The results showed that PAMM (and the algorithms derived from the framework) can actually improve user experience.

*7.3.5. Convergence and Training Time.* We conducted experiments to show whether PAMM can converge in terms of the diversity evaluation measures. Specifically, we showed the learning curve of PAMM($\alpha$-NDCG) in terms of $\alpha$-NDCG@20, ERR-IA@20, and D#-NDCG during the training phase. At each training iteration, the model parameters are outputted and evaluated on the test data. Figure 9 shows the performance curves with respect to the number of training iterations. From the results, we can see that the ranking accuracy of PAMM($\alpha$-NDCG) steadily improves in terms of $\alpha$-NDCG@20, ERR-IA@20, and D#-NDCG as the training goes on. PAMM converges and returns after running about 60 iterations. We also observed that, in all of our experiments, PAMM usually converges and returns after running 50$\sim$100 iterations. A similar phenomenon was also observed from the learning curve of SGDMM-Log and SGDMM-Exp (see Appendix C). The results indicates that the algorithms derived under the proposed framework converge fast and conduct training efficiently.

Table VIII. A/B Testing for R-LTR and PAMM($\alpha$-NDCG) on WT2009 Queries

| ID: Query | $\alpha$-NDCG@20 of R-LTR | $\alpha$-NDCG@20 of PAMM | PAMM win | draw | PAMM loss |
|---|---|---|---|---|---|
| 1: obama family tree | 0.866 | 0.920 | 13 | 1 | 5 |
| 2: french lick resort and casino | 0.322 | 0.378 | 16 | 1 | 2 |
| 3: getting organized | 0.694 | 0.669 | 1 | 13 | 5 |
| 4: toilet | 0.445 | 0.478 | 15 | 0 | 4 |
| 5: mitchell college | 0.295 | 0.338 | 15 | 1 | 3 |
| 6: kcs | 0.094 | 0.109 | 2 | 13 | 4 |
| 7: air travel information | 0.163 | 0.172 | 14 | 3 | 2 |
| 8: appraisals | 0.358 | 0.357 | 0 | 14 | 5 |
| 9: used car parts | 0.383 | 0.405 | 14 | 4 | 1 |
| 10: cheap internet | 0.220 | 0.276 | 14 | 3 | 2 |
| 11: gmat prep classes | 0.503 | 0.518 | 12 | 6 | 1 |
| 12: djs | 0.567 | 0.617 | 18 | 1 | 0 |
| 13: map | 0.471 | 0.496 | 16 | 3 | 0 |
| 14: dinosaurs | 0.759 | 0.755 | 2 | 10 | 7 |
| 15: espn sports | 0.349 | 0.356 | 1 | 14 | 4 |
| 16: arizona game and fish | 0.381 | 0.426 | 13 | 3 | 3 |
| 17: poker tournaments | 0.368 | 0.374 | 13 | 2 | 4 |
| 18: wedding budget calculator | 0.541 | 0.514 | 1 | 1 | 17 |
| 19: the current | 0.000 | 0.000 | 0 | 19 | 0 |
| 20: defender | 0.101 | 0.171 | 19 | 0 | 0 |
| 21: volvo | 0.457 | 0.483 | 10 | 3 | 6 |
| 22: rick warren | 0.135 | 0.208 | 17 | 1 | 1 |
| 23: yahoo | 0.000 | 0.000 | 1 | 18 | 0 |
| 24: diversity | 0.830 | 0.875 | 12 | 5 | 2 |
| 25: euclid | 0.479 | 0.539 | 13 | 3 | 3 |
| 26: lower heart rate | 0.530 | 0.554 | 5 | 9 | 5 |
| 27: starbucks | 0.314 | 0.346 | 10 | 6 | 3 |
| 28: inuyasha | 0.327 | 0.379 | 12 | 4 | 3 |
| 29: ps 2 games | 0.286 | 0.318 | 14 | 2 | 3 |
| 30: diabetes education | 0.144 | 0.181 | 10 | 8 | 1 |
| 31: atari | 0.817 | 0.812 | 1 | 9 | 9 |
| 32: website design hosting | 0.676 | 0.744 | 12 | 6 | 1 |
| 33: elliptical trainer | 0.489 | 0.496 | 10 | 5 | 4 |
| 34: cell phones | 0.181 | 0.225 | 11 | 7 | 1 |
| 35: hoboken | 0.209 | 0.236 | 12 | 4 | 3 |
| 36: gps | 0.218 | 0.195 | 3 | 4 | 12 |
| 37: pampered chef | 0.475 | 0.541 | 15 | 3 | 1 |
| 38: dogs for adoption | 0.681 | 0.662 | 1 | 1 | 17 |
| 39: disneyland hotel | 0.393 | 0.426 | 15 | 3 | 1 |
| 40: michworks | 0.361 | 0.391 | 16 | 2 | 1 |
| 41: orange county convention center | 0.268 | 0.373 | 13 | 5 | 1 |
| 42: the music man | 0.428 | 0.455 | 15 | 3 | 1 |
| 43: the secret garden | 0.220 | 0.296 | 17 | 1 | 1 |
| 44: map of the united states | 0.156 | 0.213 | 16 | 1 | 2 |
| 45: solar panels | 0.284 | 0.329 | 14 | 3 | 2 |
| 46: alexian brothers hospital | 0.332 | 0.371 | 15 | 3 | 1 |
| 47: indexed annuity | 0.779 | 0.832 | 9 | 9 | 1 |
| 48: wilson antenna | 0.680 | 0.755 | 13 | 4 | 2 |
| 49: flame designs | 0.237 | 0.264 | 9 | 4 | 6 |
| 50: dog heat | 0.556 | 0.527 | 4 | 5 | 10 |
| total | | | 524 | 253 | 173 |

Table IX. Comparison of the Training Time for Different Algorithms Derived from the Framework

| Method | training time (hour) |
|---|---|
| PAMM($\alpha$-NDCG) | ∼5h |
| SGDMM-Log($\alpha$-NDCG) | ∼6.5h |
| SGDMM-Exp($\alpha$-NDCG) | ∼6.5h |

We compared the training time of PAMM($\alpha$-NDCG), SGDMM-Log($\alpha$-NDCG), and SGDMM-Exp($\alpha$-NDCG) on the first fold of the WT2009 dataset. All experiments were conducted on a server with 24GB memory and two Intel Xeon E5410 2.33 GHz Quad-Core processors. From the results reported in Table IX,[5] we can see that PAMM($\alpha$-NDCG) used less training time than SGDMM-Exp($\alpha$-NDCG) and SGDMM-Exp($\alpha$-NDCG) to converge. This is because (i) we empirically found that PAMM usually converges at about 60 iterations while SGDMM-Exp needs about 70∼80 iterations to converge; and (ii) at each iteration, PAMM updates the model parameter only if the condition $F(X^{(n)}, R^{(n)}, \mathbf{y}^+) - F(X^{(n)}, R^{(n)}, \mathbf{y}^-) \leq E(X^{(n)}, \mathbf{y}^+, J^{(n)}) - E(X^{(n)}, \mathbf{y}^-, J^{(n)})$ (Line 10 of Algorithm 2) is satisfied. SGDMM-Exp and SGDMM-Log, however, need to update the parameter for all pairs $(\mathbf{y}^+, \mathbf{y}^-)$. The results show that PAMM is more efficient than other derived algorithms.

## 8. CONCLUSION

In this article, we proposed a novel framework that can directly optimize diversity evaluation measures for learning ranking models for search result diversification. The framework makes use of the MMR model for constructing the diverse rankings. In training, the diversity evaluation measure on training queries is directly optimized. New diversity ranking algorithms can be easily derived under the framework by optimizing the loss functions upper bounding the basic loss defined over the diversity evaluation measure. The algorithms derived under the framework offer several advantages: they employ a ranking model that meets the MMR criterion, they have the ability to directly optimize any diversity evaluation measure, and they have the ability to utilize both positive and negative rankings in training. Experimental results based on three benchmark datasets show that the algorithms derived under the framework significantly outperformed the state-of-the-art baseline methods including SVM-DIV, structural SVM, and R-LTR.

## REFERENCES

Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. 2009. Diversifying search results. In *Proceedings of the 2nd ACM International Conference on Web Search and Data Mining (WSDM'09)*. ACM, New York, 5–14. DOI: http://dx.doi.org/10.1145/1498759.1498766

Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98)*. ACM, New York, 335–336. DOI: http://dx.doi.org/10.1145/290941.291025

Ben Carterette and Praveen Chandar. 2009. Probabilistic models of ranking novel documents for faceted topic retrieval. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM'09)*. ACM, New York, 1287–1296. DOI: http://dx.doi.org/10.1145/1645953.1646116

Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. 2009. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM'09)*. ACM, New York, 621–630. DOI: http://dx.doi.org/10.1145/1645953.1646033

Charles L. A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. 2008. Novelty and diversity in information retrieval evaluation. In *Proceedings of*

[5]The wavy line before the number means the time is approximated.

*the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'08)*. ACM, New York, 659–666. DOI:http://dx.doi.org/10.1145/1390334.1390446

Charles L. Clarke, Maheedhar Kolla, and Olga Vechtomova. 2009. An effectiveness measure for ambiguous and underspecified queries. In *Proceedings of the 2nd International Conference on Theory of Information Retrieval: Advances in Information Retrieval Theory (ICTIR'09)*. Springer-Verlag, Berlin, 188–199.

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10 (EMNLP'02)*. Association for Computational Linguistics, Stroudsburg, PA, 1–8. DOI:http://dx.doi.org/10.3115/1118693.1118694

Van Dang and Bruce W. Croft. 2013. Term level search result diversification. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'13)*. ACM, New York, 603–612. DOI:http://dx.doi.org/10.1145/2484028.2484095

Van Dang and W. Bruce Croft. 2012. Diversity by proportionality: An election-based approach to search result diversification. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'12)*. ACM, New York, 65–74. DOI:http://dx.doi.org/10.1145/2348283.2348296

Zhicheng Dou, Sha Hu, Kun Chen, Ruihua Song, and Ji-Rong Wen. 2011. Multi-dimensional search result diversification. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*. ACM, 475–484.

Sreenivas Gollapudi and Aneesh Sharma. 2009. An axiomatic approach for result diversification. In *Proceedings of the 18th International Conference on World Wide Web (WWW'09)*. ACM, New York, 381–390. DOI:http://dx.doi.org/10.1145/1526709.1526761

Shengbo Guo and Scott Sanner. 2010. Probabilistic latent maximal marginal relevance. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'10)*. ACM, New York, 833–834. DOI:http://dx.doi.org/10.1145/1835449.1835639

Jiyin He, Vera Hollink, and Arjen de Vries. 2012. Combining implicit and explicit topic representations for result diversification. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'12)*. ACM, New York, 851–860. DOI:http://dx.doi.org/10.1145/2348283.2348397

Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99)*. ACM, New York, 50–57. DOI:http://dx.doi.org/10.1145/312624.312649

Sha Hu, Zhicheng Dou, Xiaojie Wang, Tetsuya Sakai, and Ji-Rong Wen. 2015. Search result diversification based on hierarchical intents. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM'15)*. ACM, New York, 63–72. DOI:http://dx.doi.org/10.1145/2806416.2806455

Hang Li. 2014. *Learning to Rank for Information Retrieval and Natural Language Processing; 2nd ed.* Morgan & Claypool Publ., San Rafael, CA.

Liangda Li, Ke Zhou, Gui-Rong Xue, Hongyuan Zha, and Yong Yu. 2009. Enhancing diversity, coverage and balance for summarization through structure learning. In *Proceedings of the 18th International Conference on World Wide Web (WWW'09)*. ACM, New York, 71–80. DOI:http://dx.doi.org/10.1145/1526709.1526720

Yaoyong Li, Hugo Zaragoza, Ralf Herbrich, John Shawe-Taylor, and Jaz S. Kandola. 2002. The perceptron algorithm with uneven margins. In *Proceedings of the 19th International Conference on Machine Learning (ICML'02)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, 379–386. http://dl.acm.org/citation.cfm?id=645531.655993

Shangsong Liang, Zhaochun Ren, and Maarten de Rijke. 2014. Personalized search result diversification via structured learning. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'14)*. ACM, New York, 751–760. DOI:http://dx.doi.org/10.1145/2623330.2623650

Tie-Yan Liu. 2009. Learning to rank for information retrieval. *Foundation and Trends in Information Retrieval* 3, 3 (March 2009), 225–331. DOI:http://dx.doi.org/10.1561/1500000016

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York.

Donald Metzler and W. Bruce Croft. 2005. A Markov random field model for term dependencies. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'05)*. ACM, New York, 472–479. DOI:http://dx.doi.org/10.1145/1076034.1076115

Lilyana Mihalkova and Raymond Mooney. 2009. Learning to disambiguate search queries from short sessions. In *Machine Learning and Knowledge Discovery in Databases*, Wray Buntine, Marko Grobelnik,

Dunja Mladeni, and John Shawe-Taylor (Eds.). Lecture Notes in Computer Science, Vol. 5782. Springer Berlin, 111–127. DOI: http://dx.doi.org/10.1007/978-3-642-04174-7_8

Filip Radlinski and Susan Dumais. 2006. Improving personalized web search using result diversification. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'06)*. ACM, New York, 691–692. DOI: http://dx.doi.org/10.1145/1148170.1148320

Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. 2008. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th International Conference on Machine Learning (ICML'08)*. ACM, New York, 784–791. DOI: http://dx.doi.org/10.1145/1390156.1390255

Davood Rafiei, Krishna Bharat, and Anand Shukla. 2010. Diversifying web search results. In *Proceedings of the 19th International Conference on World Wide Web (WWW'10)*. ACM, New York, 781–790. DOI: http://dx.doi.org/10.1145/1772690.1772770

Karthik Raman, Pannaga Shivaswamy, and Thorsten Joachims. 2012. Online learning to diversify from implicit feedback. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'12)*. ACM, New York, 705–713. DOI: http://dx.doi.org/10.1145/2339530.2339642

Tetsuya Sakai, Nick Craswell, Ruihua Song, Stephen Robertson, Zhicheng Dou, and Chin-Yew Lin. 2010. Simple evaluation metrics for diversified search results. In *EVIA@NTCIR*. Citeseer, 42–50.

Tetsuya Sakai and Ruihua Song. 2011. Evaluating diversified search results using per-intent graded relevance. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'11)*. ACM, New York, 1043–1052. DOI: http://dx.doi.org/10.1145/2009916.2010055

Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. 2010. Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th International Conference on World Wide Web (WWW'10)*. ACM, New York, 881–890. DOI: http://dx.doi.org/10.1145/1772690.1772780

Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. 2005. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research* 6 (Dec. 2005), 1453–1484. http://dl.acm.org/citation.cfm?id=1046920.1088722

Le Wu, Qi Liu, Enhong Chen, Nicholas Jing Yuan, Guangming Guo, and Xing Xie. 2016. Relevance meets coverage: A unified framework to generate diversified recommendations. *ACM Transactions on Intelligent Systems Technology* 7, 3, Article 39 (Feb. 2016), 30 pages. DOI: http://dx.doi.org/10.1145/2700496

Jun Xu, Tie-Yan Liu, Min Lu, Hang Li, and Wei-Ying Ma. 2008. Directly optimizing evaluation measures in learning to rank. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'08)*. ACM, New York, 107–114. DOI: http://dx.doi.org/10.1145/1390334.1390355

Yisong Yue and Thorsten Joachims. 2008. Predicting diverse subsets using structural SVMs. In *Proceedings of the 25th International Conference on Machine Learning (ICML'08)*. ACM, New York, 1224–1231. DOI: http://dx.doi.org/10.1145/1390156.1390310

Yisong Yue and Thorsten Joachims. 2009. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML'09)*. ACM, New York, 1201–1208. DOI: http://dx.doi.org/10.1145/1553374.1553527

Cheng Xiang Zhai, William W. Cohen, and John Lafferty. 2003. Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval (SIGIR'03)*. ACM, New York, 10–17. DOI: http://dx.doi.org/10.1145/860435.860440

Yadong Zhu, Yanyan Lan, Jiafeng Guo, Xueqi Cheng, and Shuzi Niu. 2014. Learning for search result diversification. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR'14)*. ACM, New York, 293–302. DOI: http://dx.doi.org/10.1145/2600428.2609634

Received March 2016; revised June 2016; accepted August 2016