Incorporating Robustness into Web Ranking Evaluation

Xin Li Fan Li Shihao Ji Zhaohui Zheng Yi Chang Anlei Dong
Yahoo! Labs, 701 First Avenue, Sunnyvale, CA 94089
{xinli, fli, shihao, zhaohui, yichang, anlei}@yahoo-inc.com

ABSTRACT

In many Web search engines, a ranking function is selected for deployment mainly by comparing the relevance measurements over candidates. Due to the dynamical nature of the Web, the ranking features and the query and URL distribution on which the ranking functions are built, may change dramatically over time. The actual relevance of the function may degrade, and thus the previous function selection conclusions become invalid. In this work we suggest to select Web ranking functions according to both their relevance and robustness to the changes that may lead to relevance degradation over time. We argue that the ranking robustness can be effectively measured by taking into account the ranking score distribution across search results. We then propose two alternatives to the NDCG metric that both incorporate ranking robustness into ranking function evaluation and selection. A machine learning approach is developed to learn the parameters that control the metric sensitivity to score turbulence, from human-judged preference data.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Measurement, Reliability

Keywords

Web ranking, ranking robustness, NDCG

1. INTRODUCTION

In Web search, a ranking function usually ranks the search result pages for a query, by first assigning a score that measures the relevance of each page to the query, and then ranking pages in the descendent order of the score. Many popular learning-to-rank algorithms, such as RankSVM [5], RankNet [3], GBrank [7], and SoftRank [6], belong to this category. For function selection purpose, metrics such as Mean Average Precision (MAP) and NDCG [4] have been developed and are commonly used to measure the search result relevance. The underlying assumption for this strategy is that the relevance of the selected function will be consistently better over time after deployment, compared with other candidates.

However, this assumption usually does not hold for Web search. As the Web size expands, Web content updates, Web link structure evolves, and user search behaviors change, the rank features that the function was trained may get updated, and the query and search result page distribution it is applied to may drift too. This brings practical issues to the ranking quality in this dynamical Web search environment. On one hand, a deployed ranking function may not be so robust in the sense that its relevance may degrade with these changes. On the other hand, the ranking function itself, can not always be retrained so timely to capture the changes. Consequently, the relevance comparison previously done for function selection may not be valid any longer.

In this paper we claim that, selecting web ranking functions purely based on relevance is inadequate, and ranking robustness should also be measured and taken into account in ranking function evaluation. Rather than functions with only higher static relevance measurement, those whose relevance measurement are more robust to potential changes, should be considered for deployment. One observation we made is: the impact from potential change in rank features and the distribution of user queries and search result pages on a ranking function, is mostly reflected to the ranking scores assigned by the function. Consider two score lists with the same ranking order: (1) $\{2.1, 2.0, 1.0\}$ and (2) $\{2.1, 1.5, 1.0\}$. The human-judged relevance grades of the three pages are {"Excellent", "Bad", "Bad"}. Although the two lists have identical NDCG values, apparently (2) is better than (1) in terms of ranking robustness with regard to potential turbulence in scores. As rank features update, the first two pages in (1) can be easily swaped, and the corresponding NDCG measurement can drop dramatically. This indicates the inadequacy of NDCG as the only criteria in function selection when score turbulence can happen.

Therefore in this work, we propose two alternatives to NDCG, that measure both relevance and ranking robustness for ranking function selection. We design the first robustness metric, namely rNDCG, to add in the probability that neighboring pairs of search results in the original ranking order may switch positions when ranking score turbulence happens. The second metric is SoftNDCG, first proposed in [6]. SoftNDCG calculates the expected NDCG of a ranked list according to a probability distribution that a result page

CIKM'09, November 2–6, 2009, Hong Kong, China.

takes each rank, which is also defined over the ranking scores across pages. Both of the proposed metrics are sensitive to the possible ranking score turbulence as we will demonstrate in this paper. In addition to these two manually defined metrics, we also develop a machine learning approach to automatically learn a robustness metric that maximizes its consistency with user preference data. For doing that, we first parameterize the SoftNDCG metric with parameters that control the sensitiveness of the measurement to ranking scores. Then we collect training data by providing randomly permutated search results to human editors and collect their preference over the ranked lists. A maximum likelihood optimization is then applied for the purpose of parameter learning.

How to evaluate the proposed metrics with respect to their measurements of robustness is a key issue in this study. In the empirical study, we compare different metrics with regard to their reliability in function selection, according to three criterions. A metric that can better measure the ranking robustness should be 1) more stable (i.e. having less variance) to the noise that causes ranking score turbulence; 2) more reliable in their function selection across different data samplings, better measuring the robustness to the possible drifting in the distribution of queries and result pages; and 3) more stable in their function selection when the evaluation data is updated over time. Our experimental results indeed demonstrate the superiority of the proposed metrics over the standard NDCG metric, based on these criterions.

There are some related works that attempt to address the ranking robustness problem from different angles. Some works [1, 2] address the problem from the perspective of spamming, and attempt to improve the ranking relevance in a noisy environment by actively adding noise to training data. [9] defines a ranking robustness metric to predict query search performance, which applies a noise model to perturbable documents and then measure the search result similarity before and after perturbation. The calculation of designed metric highly relies on a specific retrieval function. SoftRank [6] defines SoftNDCG as some optimization loss function and applies it directly to function training. We first apply it directly as a ranking evaluation metric in function selection. From the perspective of learning ranking metrics, [8] has conducted similar experiments in automatically learning the gain value and rank discounting factors for NDCG. The learned metric is a static relevance measurement, and does not measure robustness to changes.

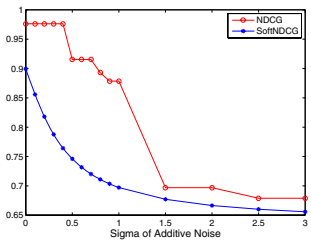

Figure 1: Evolutions of NDCG and SoftNDCG for one query as the intensity of additive noise increases.

2. WEB RANKING ROBUSTNESS

In the real Web search environment, changes are happening all the time, and these changes may cause the quality of a ranking function to degrade over time. For example: 1) Page content update — the web pages are not static and may get updated frequently; 2) Web scale expansion — more pages, more link structures and more users, may lead to drifting of the feature distribution; 3) User search behavior change – evolution in search query distribution may cause the data distribution used for evaluation to drift. The basic assumption for ranking robustness is that, the less relevance degradation a ranking function may suffer, the higher robustness the function has. Therefore we focus on measuring the expected relevance of a ranking function over possible changes, and predict its potential relevance over time.

Can the future relevance of a ranking function be predicted with existing data ? Our answer is yes, at least partially. We notice that the dynamical changes are reflected on the value and distribution of the ranking scores assigned by a function. If we take into account the ranking scores in the relevance evaluation, the ranking robustness can be at least partially measured. Let's use a real Web query and its search results as an example to demonstrate the relationship between the ranking scores and the potential relevance change. The relevance scores for Web pages in a ranked list and the relevance label for each page (Perfect, Excellent, Good, Fair and Bad) are known and used to compute the NDCG value. We incrementally add larger Gaussian white noise to the scores. Figure 1 presents the evolution of the NDCG values as the intensity of the additive noise increases. As seen, the NDCG curve is unstable with respect to the additive noise: at some points NDCG is insensitive to the additive noise, while at some points a small increase in the noise results in a dramatic change in the NDCG value. This is because the NDCG computation only relies on the order of a ranked list, but is regardless to the ranking scores. As one of the advocated metrics for robust ranking function selection, the SoftNDCG curve (to be discussed next) is smooth in response to the additive noise.

3. RANKING ROBUSTNESS METRICS

In order to select ranking function according to both relevance and robustness, two metrics, namely rNDCG and SoftNDCG, are proposed in this section as alternatives to NDCG. Both of the metrics go beyond NDCG which only considers the absolute ranks, and take into account the probability that a search result page belongs to a rank position. The probability, although calculated differently, is defined over the normalized relevance scores of the pages.

3.1 rNDCG

The first metric rNDCG (namely robust NDCG) considers the potential ranking order changes in the neighboring positions of a ranked list. Consider a ranked list of N Web pages in the decreasing order of their relevance score: $L = \{< g_1, s_1 >, < g_2, s_2 >, \ldots, < g_N, s_N >\}$. $< g_r, s_r >$ $(r = 1, 2, \ldots, N)$ represents the human-judged grade and the relevance score of the document at rank r. Without loss of generality, we assume $s_1 > s_2 > \cdots > s_N$. When we compute the relevance gain from each ranking position r, three neighboring result pages at rank $r-1$, r and $r+1$ in the original order, may contribute their relevance gain with probabilities p_{r-1}, $1 - p_{r-1} - p_r$, and p_r, respectively. The rNDCG of the ranked list is then defined as follows:

$$\mathcal{G}(L) = G_{\max}^{-1} \sum_{r=1}^{N} D(r)[p_{r-1}g(r-1) + p_r g(r+1) + (1-p_{r-1}-p_r)g(r)] \quad (1)$$

where p_r is the probability that the page at rank r may take rank $r+1$, due to score turbulence. $D(r)$ is the same rank-discounting function as in NDCG that is often chosen in the form of $D(r) = 1/\log(2+r)$, and $G_{\max}$ is the maximum possible value of $\sum_{r=1}^{N} D(r)[p_{r-1}\, g(r-1) + p_r g(r+1) + (1 - p_{r-1} - p_r) g(r)]$ achieved when pages are optimally ordered. The probability p_r is defined as $p_r = \frac{1}{2+e^{\frac{s(r+1)-s(r)}{\sigma}}}$ if $r = 1, 2, \ldots, N-1$; 0 otherwise. As noted, the difference between the relevance scores of two neighboring pages decides the probability: the closer the two relevance scores are, the more likely the two pages may switch ranks in the future. Here σ is a positive normalizer that controls sensitive of the rank probability to the score difference.

3.2 SoftNDCG

The second metric that considers the ranking score distribution across documents is SoftNDCG, introduced in [6]. SoftNDCG calculates the expected NDCG with regard to a distribution that defines the probability of a result page taking each specific rank. Consider the same ranked list of Web result pages: $L = \{< g_1, s_1 >, < g_2, s_2 >, \ldots, < g_N, s_N >\}$:

$$\mathcal{G}(L) = G_{\max}^{-1} \sum_{j=1}^{N} g(j) \sum_{r=0}^{N-1} D(r) p_j(r), \tag{2}$$

where $p_j(r)$ is the probability of document j ranked at position r, given a score list $\{s_1, s_2, \cdots, s_N\}$. In SoftNDCG, a ranking form of a Binomial distribution is used to approximate the distribution of ranks. In order to estimate the rank distribution of document j, it first directly estimates π_{ij}, the probability that document i out-ranks documents j, for every other document $i \neq j$. It then computes the probability $p_j(r)$ that document j has rank r based on them.

In the computation of SoftNDCG, one parameter σ is required to be pre-determined. This parameter controls how to measure the closeness among ranking scores. As $\sigma \to \infty$, the rank distribution $p_j(r)$ becomes a uniform distribution, i.e., no score differences are differentiated, while as $\sigma \to 0$, SoftNDCG degenerates to the normal NDCG.

Our approach to determining σ is to learn from preference pair, e.g., $L_1 \succ L_2$, that signifies the ranking generated by score list L_1 is preferred to that generated by L_2. We model this pairwise preference in a probability model:

$$p(\mathcal{G}(L_1) > \mathcal{G}(L_2)) = \frac{1}{1 + \exp(\mathcal{G}(L_2) - \mathcal{G}(L_1))}. \tag{3}$$

By adjusting parameter σ, we attempt to maximize the likelihood to favor users' and/or human editors' preferences.

Assume we have preference lists $\mathcal{L} = \{L_1^i \succ L_2^i\}_{i=1}^{M}$. An optimal σ^* can be sought by maximizing the following log-likelihood function:

$$\mathcal{F}(\sigma) = \frac{1}{M} \sum_{i=1}^{M} \log p(\mathcal{G}(L_1^i) > \mathcal{G}(L_2^i)). \tag{4}$$

In this study, a simple gradient descent procedure is applied to searching for the optimal σ^*, with the gradient computed as follows:

$$\frac{\partial \mathcal{F}(\sigma)}{\partial \sigma} = \frac{1}{M} \sum_{i=1}^{M} \frac{\frac{\partial \mathcal{G}(L_1^i)}{\partial \sigma} - \frac{\partial \mathcal{G}(L_2^i)}{\partial \sigma}}{1 + \exp(\mathcal{G}(L_1^i) - \mathcal{G}(L_2^i))} \tag{5}$$

We executed the experiment of learning σ using $M = 2183$ Web queries. The queries are randomly sampled from the query log of a commercial search engine, and the top five search results are also scraped from the same search engine. For each query, we randomly permutate the search results to get two different ranked lists $L_1^{(i)}$ and $L_2^{(i)}$ for each query. Then we ask human editors to give their preference over the two lists: $L_1^{(i)} \succ L_2^{(i)}$ or $L_1^{(i)} \prec L_2^{(i)}$, for $i = 1, 2, \cdots, M$. The optimal σ^* that maximizes the log-likelihood as in Equation 4 converges to around 0.2.

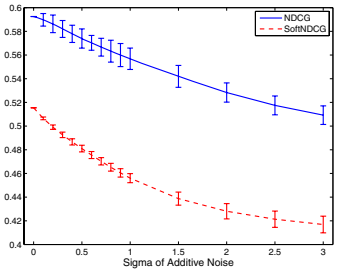

Figure 2: Evolutions of NDCG and SoftNDCG as the intensity of additive noise increases. Results are averaged over ten runs, with mean and error-bars reported.

4. EMPIRICAL EVALUATION

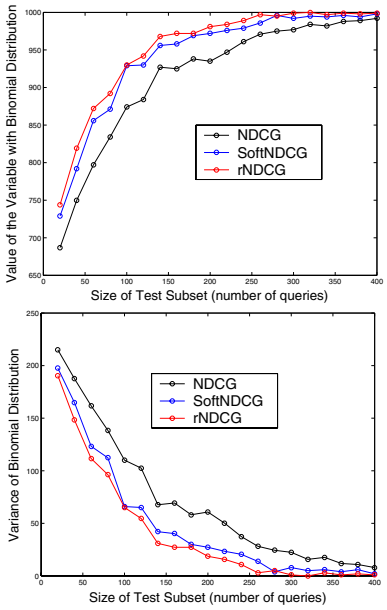

Figure 3: The value and the variance of the Binomial distributions, with different test sizes.

In this section, we compare the three metrics: NDCG, rNDCG, SoftNDCG[1] in three experiments: 1) Study how the relevance of a ranking function evaluated in those metrics degrade with artificial noise added to the ranking scores. 2) Study how function selection results evaluated by different metrics are affected by the size of the test data. This experiment simulates the situation when the distribution of evaluation data could change over time. 3) Study how function selection results are affected over time. For this purpose, we train two ranking functions with data from an earlier time,

[1]All metrics are calculated only for top 5 rank positions.

and then evaluate the variance of function comparison results with data sampled data from different time points. The ranking functions for all these experiments are trained with GBrank [7] a state-of-art ranking algorithm. The features used for training can be roughly divided into four categories: link-based features, content-based features, click-based features and others. The calculation of the rank probability in rNDCG and SoftNDCG both exploit function-dependent ranking scores. We normalize raw scores into their percentile positions in a large score sampling for each function, making them comparable across functions.

4.1 Stability to Ranking Score Turbulence

The data set used for evaluating the three metrics contains $4,000$ queries, sampled from a large commercial web search engine. All the query and result page pairs have human labels, with five relevance levels (Perfect, Excellent, Good, Fair and Bad). We compute NDCG and SoftNDCG over the ranking scores given by a ranking function, but under additive noise from a Gaussian noise model, while the NDCG and the SoftNDCG values here are averaged over all the $4,000$ queries. To demonstrate the instability of NDCG, we implement the experiment ten times with the additive noise generated by ten different random seeds. The results reported in Figure 2, show that the SoftNDCG measurement has smaller variances. It indicates that compared with the standard NDCG, SoftNDCG is more stable with regard to potential score turbulence, because it has already considered the impact of score turbulence in the measurement.

4.2 Reliability over Different Data Samplings

This experiment is to study whether rNDCG and SoftNDCG are more reliable than NDCG in ranking function selection, with respect to different data samplings. For this purpose, We first train two ranking functions (named as f_1 and f_2) using GBRank [7] with different learning parameters and feature sets. For SoftNDCG, we learned its parameter using the methods described in Section 3.2. The σ in rDCG is simply set to 0.5. We randomly sample $1,000$ subsets (each contains K queries, we also study the impact of different K) from our test data (which is different from our holdout set), and calculate the average NDCG, rNDCG and SoftNDCG over each subset. After that, we construct a Binomial distribution for each metric, using the following rule: For each subset, a binary random variable is assigned 1 if the metric shows a higher and equal value (tied values are very rare) for f_1 than f_2 averaged over the queries in the subset, and 0 otherwise. This way, we have a Binomial distribution for each metric over the $1,000$ subsets.

Figure 3 plots the value and variance curve for the evolutions of the three Binomial distributions as the subset size K increases. The Binomial distribution with $K = 4,000$ is treated our ground truth result, where all three metrics agree that f_1 is better than f_2. As seen in the graph, the value curves of three Binomial distributions converge to the ground truth as the subset size increases, while the variances converge to zero. However, as test sets become smaller, rNDCG and SoftNDCG outperform NDCG in the sense they are more consistent with ground truth results, and with much smaller variances.

4.3 Reliability over Data Samplings over Time

In this experiment, we sample latest news queries and Web pages (including the rank features) from nine individual days over three months after training, 300 queries each day. The two ranking functions f_1' and f_2' were also trained at an earlier time. As we did in the Section 4.2, for each test set, we randomly sample $1,000$ subsets, with 50 queries each, and construct a similar Binomial distribution for each metric. Figure 4.3 plots the value and variance curves of three Binomial distributions across different dates. As seen, the NDCG metric cannot distinguish f_1' and f_2' very well. In contrast, based on rNDCG and SoftNDCG, it is clear that f_1' outperforms f_2' constantly across time.

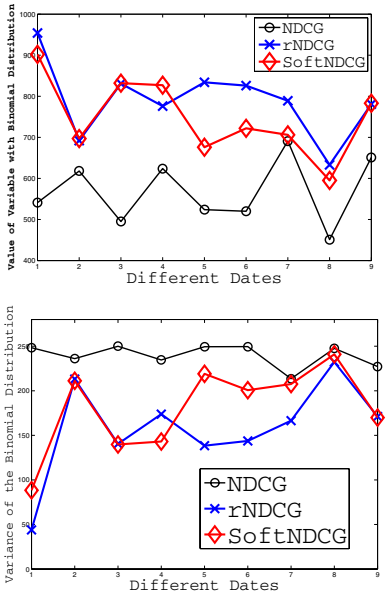

Figure 4: The value and variance of Binomial distributions over test sets sampled from multiple time points.

5. CONCLUSION

In this work, we argue that not only relevance, but also ranking robustness should be considered in Web ranking function selection. We further model the ranking robustness as the expected relevance of a ranking function with regard to potential ranking score turbulence.

6. REFERENCES

[1] R. Bhattacharjee and A. Goel. Algorithms and incentives for robust ranking. In *SODA*, 2007.

[2] J. Bian, Y. Liu, E. Agichtein, and H. Zha. A few bad votes too many?: towards robust ranking in social media. In *AIRWEB*, 2008.

[3] C. J. C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. N. Hullender. Learning to rank using gradient descent. In *ICML*, 2005.

[4] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems*, 20(4), 2002.

[5] T. Joachims. Optimizing search engines using clickthrough data. In *KDD*, 2002.

[6] M. Taylor, J. Guiver, S. Robertson, and T. Minka. Softrank: optimizing non-smooth rank metrics. In *WSDM*, 2008.

[7] Z. Zheng, K. Chen, G. Sun, and H. Zha. A regression framework for learning ranking functions using relative relevance judgments. In *SIGIR*, 2007.

[8] K. Zhou, H. Zha, G.-R. Xue, and Y. Yu. Learning the gain values and discount factors of dcg. In *SIGIR Workshop*, 2008.

[9] Y. Zhou and B. Croft. Ranking robustness: a novel framework to predict query performance. In *CIKM*, 2006.