Explore Click Models for Search Ranking

Dong Wang[1,2,*], Weizhu Chen[2], Gang Wang[2], Yuchen Zhang[1], Botao Hu[1]

[1]Institute for Theoretical Computer Science Tsinghua University Beijing, China, 100084
[2]Microsoft Research Asia No. 49 Zhichun Road Haidian District Beijing, China, 100080

dongw89@gmail.com, {wzchen, gawa}@microsoft.com {zhangyuc, botao.a.hu}@gmail.com

ABSTRACT

Recent advances in *click model* have positioned it as an effective approach to estimate document relevance based on user behavior in web search. Yet, few works have been conducted to explore the use of click model to help web search ranking. In this paper, we focus on learning a ranking function by taking the results from a click model into account. Thus, besides the editorial relevance data arising from the explicit manually labeled search result by experts, we also have the estimated relevance data that is automatically inferred from click models based on user search behavior. We carry out extensive experiments on large-scale commercial datasets and demonstrate the effectiveness of the proposed methods.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval; I.2.6 [**ARTIFICIAL INTELLIGENCE**]: Learning

General Terms

Algorithms, Design, Experimentation, Theory

Keywords

Click Model, Search Ranking, Log Mining

1. INTRODUCTION

Since click-through logs encode user preferences on search results, utilizing a user's click-through behavior on search results to automatically estimate document relevance has attracted more and more research attention recently. This task is challenging due to the well-known positional bias problem [4]. A number of studies [3, 5, 6, 8] have attempted to address this problem so as to infer unbiased relevance. Most of these works attempt to model user behaviors on search results and accurately predict future user activities. These kinds of methods are also called *click models*. For example, [5] proposed a User Browsing Model (UBM) by extending the examination hypothesis. [6] proposed a Click Chain Model (CCM) and [3] proposed a Dynamic Bayesian Model (DBN) by analyzing user behaviors in a chain-style network. These click models have been considered as one of the most effective approaches to interpret user clicks and infer search relevance, and recent advances in click models have moved forward aggressively.

Yet, few works have been conducted to explore the estimated relevance to learn a ranking function. Existing works on learning to rank [1, 2] mostly rely on editorial relevance data. However, collecting editorial relevance data is very expensive because it is indispensable to cover a diverse set of queries in the context of web search. In contrast to the scarcity of editorial relevance data, terabytes of click-through logs are generated every day and user preferences are encoded inside the data. They can be collected at a very low cost and used by click models to automatically infer the document relevance. Thus, it would be very desirable if we can replace the editorial relevance data by estimated relevance data when learning a ranking function.

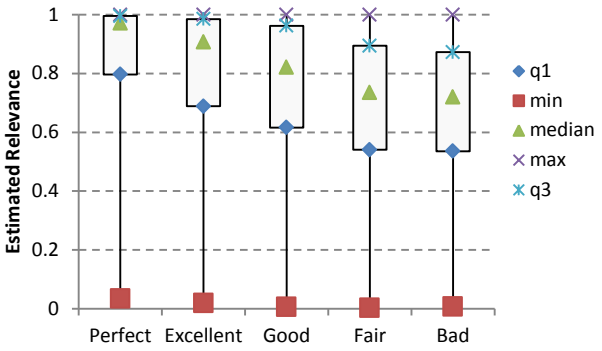


Figure 1. The correlation between estimated relevance and human's label

In our study, we observe that there are strong correlations between editorial relevance and estimated relevance. Figure 1 shows the box plot between editorial relevance and estimated relevance. The x-axis is human labeling, which has five grades. Perfect means the most relevant document and Bad means the most irrelevant document. The y-axis indicates the estimated relevance computed in the General Click Model (GCM) [8].

In this paper, we propose three methods to better explore the estimated relevance in learning a ranking function. Since there are bunches of works investigating learning a relevance function from click-through logs, this paper does not argue that the proposed method can outperform each of them. Instead, this paper aims to give a study on how to leverage the estimated relevance inferred from a click model to learn a good ranking function.

CIKM'10, October 26-30, 2010, Toronto, Ontario, Canada.

*This work was done when the first authors were visiting Microsoft Research Asia.

The remainder of the paper is organized as follows. Firstly, we describe the background in Section 2. Then we propose several ranking models in Section 3 and conduct several experiments in Section 4. Finally, we conclude the paper in Section 5.

2. PROBLEM BACKGROUND

Here we introduce some background from two categories: one is click model and the other is search ranking.

2.1 Click Model

Click models were proposed to model users' search behaviors, and compute the estimated relevance for each document query pair. In this paper, we assume that click log stores a lot of search sessions. In a session, a user submits a query q to the search engine, and gets a set of documents $D_q = \{d_i\}_{i=1}^n$. The user might examine the search results, clicks on some search results relevant to his query and then finish the session.

For a document d_i corresponding to a query q, click model can automatically infer an estimated relevance r_i based on the user click behavior. This relevance value indicates the degree of correlation between a document d_i and a query q. For example, the estimated relevance in CCM and UBM can be represented as:

$$r_j = P\big(C_j = 1 \big| E_j = 1\big) \tag{1}$$

Here $C_j = 1$ indicates that the user clicks on document d_j and $E_j = 1$ indicates that the user examines the document d_j.

Recently, GCM proposed a more general representation of the estimated relevance and demonstrates that most of the previous works, including DBN, CCM and UBM, can be reduced to GCM as special cases of the general representation. In GCM, the authors assume that a user chooses to click a document d_i in search results after examining it according to a distribution. This distribution is represented as a random variable $R_i \sim N(\mu_i, \sigma_i^2)$. Then the click event will happen if $R_i > 0$ and the estimated relevance is defined as $r_i = P(R_i > 0)$.

2.2 Search Ranking

In the learning to rank area, there are n documents for a query q and the ith document is d_i. The objective of learning to rank is to train a ranking function:

$$f: R^m \to R \tag{3}$$

Here the input of the ranking function is the feature vector R^m of d_i corresponds to query q. The output of the function is a score s indicating the predicted relevance of a document to a query.

To train this ranking function, we provide each query document pair a label l_i. This label is an editorial relevance value within five grades and indicates the relevance degree between d_i and q. Then a ranking algorithm is adopted to minimize a given cost function. For example, RankNet [2], a pairwise ranking model, defines the probability that d_i should rank higher than d_j with probability as:

$$P_{ij} = \frac{e^{o_{ij}}}{1+e^{o_{ij}}} \tag{4}$$

The cost function here is defined as the cross entropy cost:

$$C = \sum C_{ij} = \sum\big(-\overline{P_{ij}} o_{ij} + log(1 + e^{o_{ij}})\big) \tag{5a}$$

Here $o_{ij} = s_i - s_j$. For the target probability, $\overline{P_{ij}} = 1$ if d_i should rank higher than d_j in the training data, $\overline{P_{ij}} = 0$ otherwise. The derivative of RankNet cost is:

$$\frac{\delta C}{\delta o_{ij}} = -\overline{P_{ij}} + \frac{e^{o_{ij}}}{1+e^{o_{ij}}} \tag{6a}$$

We use NDCG [7] to measure the performance of ranking algorithm. The NDCG is often truncated at a rank position p as:

$$NDCG@p = \frac{1}{IDCG@p} \sum_{i=1}^{p} \frac{2^{l_i}-1}{log_2(1+i)} \tag{7}$$

Here $IDCG@p$ is chosen such that the perfect ranking would result in $NDCG@p = 1$.

3. ESTIMATED RELEVANCE RANKING

In this section, we first introduce the basic pairwise ranking model and then design three methods to exploit the estimated relevance when learning a ranking function. Our proposed ranking models are based on Neural Network and we can combine it with LambdaRank for editorial dataset.

3.1 VP: Value-based Pairwise Rank

We firstly outline the approach leveraging estimated relevance in the experimental Section of the DBN [3]. To the best of our knowledge, it is the only work which incorporates the estimated relevance inferred from click model to train a ranking function. This work assumes that a preference pair that d_i should rank higher than d_j is generated if $r_i > r_j$. After all the pairs are generated, a ranking model may be adopted to learn a ranking function to minimize the pair-wise error. In this paper, we use the GCM to calculate the estimated relevance of document d_i as:

$$r_i = P(R_i > 0) \tag{8}$$

Referring to the definition of RankNet in Section 2.2, the probability of d_i more relevant than d_j is defined in equation (4). The cost function to optimize is the cross entropy as (5a). The target probability $\overline{P_{ij}} = 1$ if $r_i > r_j$ and the target probability is $\overline{P_{ij}} = 0$ otherwise. We adopt RankNet as training model and optimize cross entropy loss in training data by the gradient descent algorithm.

3.2 DP: Distribution-based Pairwise Rank

In above ranking model, the target probability of each preference pair is $\overline{P_{ij}} = 1$ if

$$dif = r_i - r_j > 0 \tag{9}$$

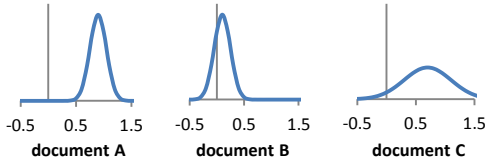
However, this approach neglects the value of dif. This value might encode the magnitude of each pair. Therefore, we propose Distribution-based Pairwise Rank aiming to compute a more reasonable target probability of each preference.

Thus, for a pair of (d_i, d_j), we have the estimated relevance distribution defined as R_i and R_j, and the target probability is:

$$\overline{P_{ij}}^* = P\big(R_i > R_j\big) \tag{10}$$

Then, we define the new cross entropy cost function as:

$$C^* = \sum C_{ij}^* = \sum\big(-\overline{P_{ij}}^* o_{ij} + log(1 + e^{o_{ij}})\big) \tag{5b}$$



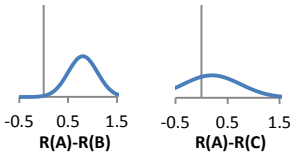


Figure 2. The estiamted relevance distribution and probability of each preference pair.

Here we present an example in Figure 2 to illustrate the basic idea. Given three documents A, B and C, we use $R(A), R(B)$ and $R(C)$ to indicate the estimated relevance distribution. We show their distribution difference in lower part of Figure 2. We can see that the probability $P(R_A - R_B > 0)$ is larger than $P(R_A - R_C > 0)$. This means that we have high confidence to believe that d_A is superior than d_B, while the low confidence for saying d_A is superior than d_C.

3.3 VL: Value-based λ Rank

The traditional pairwise ranking algorithm defines a smooth cost function to approximate the target evaluation measure. However, despite its merits, the pairwise ranking algorithm unnecessarily neglects the position effects in the rank list, while the evaluation measure NDCG is strongly related with the position in this list.

In this section, we propose Value-based Lambda Rank to optimize search rank with estimated relevance data. In this ranking method, we change the cumulative gains as (11a):

$$gain_i = 2^{r_i} - 1 \quad (11a)$$

Therefore, the evaluation metrics g_q for each query with estimated relevance is defined as:

$$g_q = \frac{1}{IdealG_q} \sum_i \frac{2^{r_i}-1}{\log(1+i)} \quad (12a)$$

Here r_i indicates the estimated relevance of document ranked at position i, and $IdealG_q$ is a normalization factor that normalizes g_q between 0 and 1. In order to maximize the score computed by g_q, we adopt λ-gradient, which is similar to the one used in LambdaRank equal to the RankNet cost scaled by the difference in g_q found by swapping two documents. For example, the λ-gradient for d_i and d_j rank at the position p_i and p_j can be defined as (13a), and the gradient of g_q can be defined as (14a):

$$\lambda_{ij} = \frac{1}{IdealG} \Delta g_q \frac{\delta C}{\delta o_{ij}} \quad (13a)$$

$$\Delta g_q = (2^{r_i} - 2^{r_j}) \left(\frac{1}{\log(1+p_i)} - \frac{1}{\log(1+p_j)} \right) \quad (14a)$$

3.4 DL: Distribution-based λ Rank

As we introduce in Section 3.2, characterizing the estimated relevance as a distribution is more advantageous than as a deterministic value. It is possible to derive more information. Simultaneously, designing a ranking algorithm by taking the position effect into consideration is important to optimize the NDCG. Thus, we take advantages of both the superiority in Section 3.2 and 3.3 to design a Distribution-based Lambda Rank.

Considering estimated relevance as a random variable, the cumulative gain formula in (11a) becomes:

$$gain_i^* = E(gain_i) = \int_{-\infty}^{\infty} (P(R_i = r)2^r - 1) dr \quad (11b)$$

Therefore, we refine our evaluation function for query q as:

$$g_q^* = \frac{1}{IdealG} \sum_i \frac{E(gain_{\phi(i)})}{log(1+i)} \quad (12b)$$

Similar to (13a) and (14a), suppose document d_i ranks at the position p_i, and document d_j is ranked at position p_j. The new lambda function is (15b), and the gradient is (16b):

$$\lambda_{ij}^* = S_{ij} \frac{1}{IdealG} \left| \Delta g_{ij}^* \frac{\delta C_{ij}^*}{\delta o_{ij}} \right| \quad (15b)$$

$$\Delta g_{ij}^* = \left(gain_i^* - gain_j^* \right) \left(\frac{1}{log(1+p_i)} - \frac{1}{log(1+p_j)} \right) \quad (16b)$$

Here $S_{ij} = 1$ if both $r_i > r_j$ and $E(gain_i) > E(gain_j)$, $S_{ij} = -1$ if both $r_i < r_j$ and $E(gain_i) < E(gain_j)$, and $S_{ij} = 0$ otherwise.

4. EXPERIMENT

The datasets we use for training and testing are extracted from two sources: an editorial relevance dataset labeled by human experts and an estimated relevance dataset inferred by GCM model. We carry out several experiments in order to answer following questions:

1. When it is only trained with the estimated relevance method, can our proposed method outperform the state-of-the-art method? Can it replace the editorial relevance data?
2. Suppose we have a small amount of editorial data, can we achieve the same NDCG score as we have a large amount of editorial data by incorporating estimated relevance data?
3. Can we combine both types of relevance dataset to achieve a better ranking function?

4.1 Experiments with Estimated Relevance

First and foremost, we conduct an experiment to show the results of ranking models with estimated relevance only. The experimental result is shown in Figure 4.

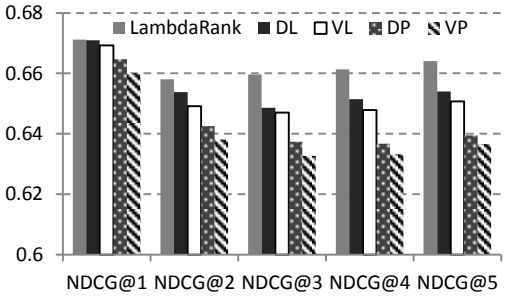


Figure 4. NDCG score of different ranking models on estimated relevance dataset

Comparing our three estimated relevance ranking models with the state-of-the-art model VP Rank, we find that our results are better in all positions and the improvements are consistent and significant. Among all our three proposed models, DL performs the best while VL is the worse in term of NDCG. This superiority is consistent in all positions. However, we find that LambdaRank trained on editorial relevance data still achieve the best NDCG value in most of positions.

4.2 Experiments with Partial Editorial Data

To answer the second question, we conduct an experiment with different size of the editorial data. We use the estimated relevance dataset as the basic training data and editorial judgment data as supplementary data in this experiment. The result is shown in Figure 5. The four lines illustrate the changes

of NDCG@5 value with the increase of editorial relevance data. Moreover, we add a black horizontal line indicating the NDCG@5 score of LambdaRank trained on 100% editorial judgment data only.

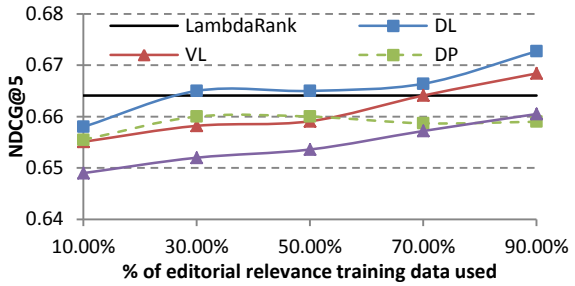


Figure 5. NDCG score of different ranking models on estimated relevance dataset and editorial relevance dataset

With the DL model, the percentage value to achieve the same NDCG@5 is 30%, while this value is about 70% for VL model. From this perspective, it shows that DL is much better in terms of the effectiveness of leveraging the data. We think this experiment shows that DL can bring huge benefit for commercial search engines for a lot of small market/language, there is always insufficient editorial relevance data due to the high cost.

4.3 Experiments with Combined Dataset

To answer the last question, we evaluate our click models on combined datasets. We define a parameter α to measure the ratio between estimated relevance data editorial judgment data. In general, suppose D_e is the number of editorial judgment training pairs and D_r is number of the estimated relevance training pairs. The whole training data with parameter α is defined as:

$$D = D_e + D_r' \tag{17}$$

Here $D_r' \in D_r$ and $|D_r'| = \alpha|D_e|$.

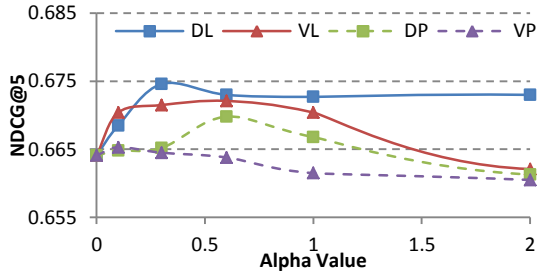


Figure 6. NDCG@5 score of different α

From the result in Figure 6, we can see that the DP Rank and VL Rank could improve NDCG@5 with a particular ratio, and our DL Rank could improve NDCG@5 for 1 percent and the result on combinational data is consistently better than LambdaRank based on editorial relevance dataset only. Moreover, to verify this improvement in other position, we draw the best NDCG results in Figure 7, which doubly verify that as compared with the LambdaRank trained with 100% percent of editorial relevance data, introducing the estimated relevance data can improve the NDCG@1, and this improvement is consistent.

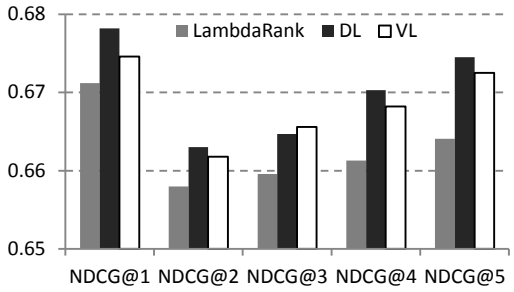


Figure 7. Best NDCG@1 among LambdaRank, DL and VL

5. CONCLUSION

In this paper, we focus on the approaches to incorporate the estimated relevance generated by click model to learn a ranking function. To achieve this objective, we propose three methods and compare them with a state-of-the-art method. Our Distribution-based λ Rank model which regards estimated relevance as a distribution and uses lambda gradient to learn the ranking function perform significantly the best in all our experiments. Secondly, we combine two types of relevance data, which is applied to demonstrate that with about 30% of the editorial relevance data and estimated relevance data, we can achieve the same accuracy as that trained on 100% editorial relevance data. Finally, we learn a better ranking function of two types of dataset. The result show that with the introducing of estimated relevance data, the accuracy can be improved about 1 point in terms of both NDCG@1 and NDCG@5.

6. REFERENCES

[1] Burges C.J.C., Ragno R., and Le Q.V. Learning to rank with non-smooth cost function. In *Proceedings of NIPS*, 2006.

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

[3] Chapelle O. and Zhang Y. A Dynamic Bayesian network click model for web search ranking. In *Proceedings of WWW2009*, 2009.

[4] Craswell N., Zoeter O., Taylor M., and Ramsey B. An experimental comparison of click position-bias models. In *Proceedings of WSDM2008*, 2008.

[5] Dupret G. and Piwowarski B. User browsing model to predict search engine click data from past observations. In *Proceedings of SIGIR2008*, 2008.

[6] Guo F., Liu C., Kannan A., Minka T., Taylor M., Wang Y., and Faloutsos C.. Click chain model in web search. In *Proceedings of WWW2009*, 2009.

[7] Jarvelin, K., and Kekalainen, J. (2000). IR evaluation methods for retrieving highly relevant documents. In *Proceedings of SIGIR '00,* 41-48.

[8] Zhu Z.A., Chen W., Minka T., Zhu C., and Chen Z. A Novel Click Model and Its Applications to Online Advertising. In *Proceedings of WSDM2010*, 2010.