Application of Ensemble Models in Web Ranking

Homa Baradaran Hashemi, Nasser Yazdani, Azadeh Shakery, and Mahdi Pakdaman Naeini
School of Electrical and Computer Engineering
College of Engineering, University of Tehran
Tehran, Iran
Email: H.B.Hashemi@ece.ut.ac.ir, Yazdani@ut.ac.ir, Shakery@ut.ac.ir, and M.Pakdaman@ece.ut.ac.ir

***Abstract*—One of the most important parts of search engines is the ranking unit. Many different classical ranking algorithms based on content (such as TF-IDF and BM25) and connectivity (such as HITS and PageRank) have been used in web search engines to find pages in response to a user query. Although these algorithms have been developed to improve retrieval results, none of them can take advantage of power of contents as well as useful link structures. Thus, it remains a challenging research question how to effectively combine these available information to maximize search accuracy. In this study, we investigate the application of different ensemble models in ranking algorithms. Some of them are simple such as Sum, Product and Borda rule, and the others are more complicated methods. We present three complex ensemble approaches. The first one is OWA operator to merge the results of various ranking algorithms. In the second approach, a state-of-the-art method, simulated click-through data, is used to learn how to combine many content and connectivity features of web pages. Moreover, we present a modified version of SVM classifier customized for ranking problems as the third complex fusion approach. The proposed methods are evaluated using the LETOR and dotIR benchmark data sets. The experimental results show that in most of the cases ensemble methods give better results and the improvements are very encouraging. These results also show that the OWA and SVM fusion methods are promising respect to other ensemble models.**

***Keywords*-Web Ranking; Document Feature Combination; Support Vector Machine; Ordered Weighted Averaging; Simulated Click-through Data;**

I. Introduction and Related Work

The fast growth of the World Wide Web and the need of acquiring accurate information have attracted much attention in research on web search engines. Since users usually prefer to look at top retrieved results, ranking the web pages to end up with top appealing pages is one of the main issues in search engines. Ranking algorithms generally classified into two main groups, content based methods such as TF-IDF [1] and BM25 [2] and connectivity based methods such as PageRank [3] and HITS [4]. Each of these methods only considers one type of information and none of them fully take advantage of all the available information. But intuitively, all information can be potentially exploited; for example, both content and connectivity may be useful.

More recently, there has been a lot of interest in combining content with the link structure information. Several ranking methods that combine these features have been developed to improve retrieval results [5], [6], [7], [8], [9], [10], [11], [12], [13]. Some researchers have used machine learning techniques to automatically construct a ranking function from training data, such that the function can sort documents according to their degree of relevance to the query. They have used different supervised methods in this direction, namely neural networks [6], genetic programming [12] and support vector machines [7][13]. In addition, some other solutions take advantage of useful information in query logs such as user clicks to maximize search accuracy [8][14] [10][11][15]. Generally, these methods use click-through data for training, for example Joachim [8] has proposed a learning algorithm based on support vector machine using click-through data and Zareh et al. [10] have developed an adaptive ranking method called "A3CRank" by using user clicks to merge results of ranking algorithms.

In this work, we apply different ensemble methods to combine content and link information, which can fully take advantage of content and connectivity information in a principled way. Among these ensemble methods, some are simple such as Sum, Product and Borda rule, and the others are more complicated. We present three complex combination approaches. As the first complicated ensemble method, we use Ordered Weighted Aggregate (OWA) fusion operator [16] to learn how to combine results of various ranking algorithms.

In the second approach, we use simulated click-through data to aggregate content and connectivity features of web pages. In order to find an appropriate weight to each feature, we follow the general procedure proposed in [11]. Zareh et. el, in [11] present a method to find the best combination of various features of web pages using simulated user clicks. In their work, they combine two features at each stage, one of them is result of the best combination of previous features and the other is the next feature. The result of two combinations are interleaved and the best combination is found by the clicks of simulated user.

In the third approach, we present a modified version of SVM classifier proper for ranking problems. Its basic idea is to use value of classification instead of the class sign.

In our experiments, we evaluate several ensemble methods, specially the proposed complicated methods with two recently constructed benchmark collections. As the basic

assessment, we use LETOR benchmark from Microsoft Research [17] which is derived from the existing English test collections. Moreover, we evaluate our solutions by a newly constructed web test collection on Iran web, dotIR benchmark [18], which is released by Iran Telecommunication Research Center (ITRC) [19].

The experimental results show that using LETOR collection, almost all ensemble methods improve search accuracy respect to the basic features. Specially, OWA and SVM ensemble methods give the most promising results. On the other hand, our work, which is, to the best of our knowledge, the first work on dotIR collection shows special characteristics of the collection. The experimental results show that the accuracy of complicated methods are as effective as current BM25 feature. Furthermore, we provide some analysis on the collection and discuss the different behavior of the two collections based on the applied ensemble methods.

The rest of the paper is organized as follows. We explain the details of used ensemble methods in section 2. We present the experiment results and analysis in section 3 and finally bring the conclusions and future work of our study in section 4.

II. Ensemble models

In this section, we will briefly introduce several ensemble methods applied in this work. In order to examine the performance of these ensemble methods, we apply them on the extracted features from Letor [17] and dotIR [18] datasets.

A. Sum Rule

In this method, we add values of different methods together. Since these values may be in different ranges, we also use the normalize sum, in which, the values are first normalized and then accumulated. Considering each of the basic features as the output of different learning model, using Sum rule and averaging out the final result, we can decrease the variance error of learner such as in the bagging ensemble method [20].

B. Product Rule

In this method, in order to obtain the fused output, we multiply different values together. Since these values may be in different ranges, we normalize them at first and then multiply the normalize values. Intuitively, by scaling the output of each basic approach and considering it as the probability measure we obtain the posterior probability using the multiplication of each probability value assuming the independence constraint between the information sources. Actually, in Sum and Product models, the final ranking is calculated based on combination of basic information resources. In this sense, we can say that these methods use low level fusion approaches in order to combine the basic information resources.

C. Borda Rule

The Borda rule is a high level fusion approach. In this method to calculate combination of ranking from different information resources, we add ranking vectors of these features in the same order. Then the final ranking will be calculated from the sorted vector of obtained results.

D. Ordered Weighted Averaging

OWA operator is one of the advanced fusion operators which maps a vector of size *n* different values to a single fused value using a normalized weight vector. This fusion operator actually has the capability of spanning the whole averaging operator's domains [16]. In OWA method, if $A = [a_1...a_n]$ is our page vector and $W = [w_1, w_2, ...w_n]$ is the OWA weight vector in which $\sum_{j=1}^{n} w_j = 1$, the result of aggregation is $f(a_1, ..., a_n) = \sum_{j=1}^{n} w_j b_j$, where $< b_1, .., b_n >$ is the sorted permutation of elements of vector A. There are many different OWA operator based on the learning algorithms on the weight vector W. We use the Exponential OWA operator to find the weights of each vector as the following.

$w_1 = \lambda$
$w_2 = \lambda(1-\lambda)$
$w_3 = \lambda(1-\lambda)^2$
...
$w_{n-1} = \lambda(1-\lambda)^{n-2}$
$w_n = (1-\lambda)^{n-1}$

Where parameter λ belongs to the unit interval $0 \leq \lambda \leq 1$. Experimentally, we have found that $\lambda = 0.3$ is suitable for the aggregation of the results same as [10].

E. Support Vector Machine

In this part, we establish a brief background on the theory of SVM and its modification customized for ranking purpose. Given a linearly separable set of points $D = \{(x_i, c_i)|X_i \in R^n, c_i \in \{-1, 1\}\}_{i=1}^{N}$, the optimal separating hyper plane, the hyper plane with the largest margin, can be obtained by solving the following optimization problem:

$$Minimizing : \frac{1}{2} w.w \tag{1}$$

Subject to:

$$c_i(w..x_i + b) \geq 1, i = 1, 2, ..., N \tag{2}$$

If the set D is not linear separable then the above optimization problem has no solution. In this case, we use the idea of soft margin method for the classification purpose. This method introduces slack variables, ξi, which measures the degree of misclassification of the data point x_i

$$c_i(w..x_i + b) \geq 1 - \xi_i, i = 1, 2, ..., N \tag{3}$$

The objective function is then increased by a function which penalizes non-zero ξi, and the optimization becomes a trade off between a large margin, and a small error penalty.

If the penalty function is linear, then the optimal solution can be obtained by solving the following optimization problem:

$$Minimizing : -\frac{1}{2}w.w + C\sum_{i=1}^{N}\xi_i \quad (4)$$

Subject to:

$$c_i(wx_i + b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, 2, ..., N \quad (5)$$

The Idea of SVMs can be generalized simply to the non-linear discriminative classifier by mapping the input vector into a high dimensional feature space using the trick of kernel functions which is an inner product in the new space. Typical kernel functions are polynomial kernels, radial basis kernels and wavelet kernels [21].

In the fusion of different basic ranking methods using the Support Vector Machines, in order to obtain a ranked list in the output of SVM fusion model, we used the distance of a data point from the discriminating hyperplane as the measure for relevancy of a page to a specific query. It means that a page would be relevant to a query if it will lay on the positive side of maximum margin hyperplane in the feature space. Moreover, intuitively the more distant from the discriminative hyperplane the more relevant the result will be.

F. Simulated Click-through Data

In this section, we describe the general idea of simulated click-through data ranking ensemble algorithm which is used as one of the complex ranking ensemble methods in our experiments. The main idea is to use the simulated clicks of the users on the retrieved results to find the best combination of basic ranking results. As the user clicks on a page are based on its relevance, we need a function for relevance. In the basic simulated click-through data model proposed by Radlinski and Joachims [22] the contents and relevance of documents are simulated, while we use real documents and actual relevance judgments from LETOR and dotIR datasets.

In ensemble model of ranking algorithms we usually use a linear combination of different basic ranking model's output as the final result of ranking. The general linear ranking combination formula is given in equation 6 where $w_f(i)$ denotes the normalized weight of the result i in feature f and m is the number of features. The results are sorted in decreasing order by $w(i)$ and shown to the user. The normalized vector $\overrightarrow{C}$ gives the coefficients of features in which c_f is the coefficient of feature f [11]:

$$w(i) = \sum_{f=1}^{m} c_f * w_f(i), \sum_{f=1}^{m} c_f = 1 \quad (6)$$

Generally, finding the optimum vector $\overrightarrow{C}$ is an NP-hard problem. There are some solutions in the literature to find the suboptimal vector using a feasible time algorithm. To overcome this problem in the simulated click-through data model, instead of choosing all features together, we choose a feature and combine it with results of the previous combination using an iterative greedy algorithm. In this method, to compare the performance of two different combination coefficient at each iteration, the combination functions will be evaluated by interleaving the results of the current combination with the best combination found so far and presenting the interleaved list to the user. Then the user clicks will be used to find the better combination. For implementing this method we used the algorithm presented by [11]. Actually, the solution that we will find would be a suboptimal solution to the problem, however the running time of algorithm would be linear.

III. Experiments and Results

In this section, we report our experiments on applying several ensemble methods and analysis of their accuracy. In our experiments, we use two benchmark data collections, Letor [17] and dotIR [18] collections, which are constructed for research on information retrieval and are publicly available. They also contain some extracted features that make them to be useful for evaluation of ranking algorithms.

A. The Datasets

1) LETOR data collection: As our basic data set, we use a benchmark collection called LETOR [23] released by Microsoft Research Asia. Since its release, it is widely used in information retrieval research community for evaluation of learning to rank algorithms. It is constructed based on the existing datasets and query sets, namely, the "Gov" and OHSUMED corpora. We use 50 TREC-2003 queries on the "Gov" corpus. In LETOR, for each query-document pair, 64 various features are extracted including classical content based methods such as BM25 [2] and language model [24], connectivity based methods such as PageRank [3] and HITS [4], combined features proposed recently such as probabilistic relevance propagation [5], and low-level features such as TF and IDF.

An important advantage of using this collection is that it is created carefully for the purpose of evaluating learning to rank algorithms with a significant number of extracted features available for quantitatively comparing different methods. Since we use several ensemble methods to combine basic features of documents, LETOR seems to be a suitable dataset for our experiments. The LETOR package contains queries, relevance judgments, the extracted features, and some tools to compute the accuracy of the newly proposed ranking algorithms.

2) dotIR data collection: More recently, a benchmark collection called dotIR [18] is released by Iran Telecommunication Research Center (ITRC). The dataset contains the contents of the web pages, queries, and human judgments on the retrieved documents with respect to the queries. There are 997,462 web pages, 50 queries and around 18,000

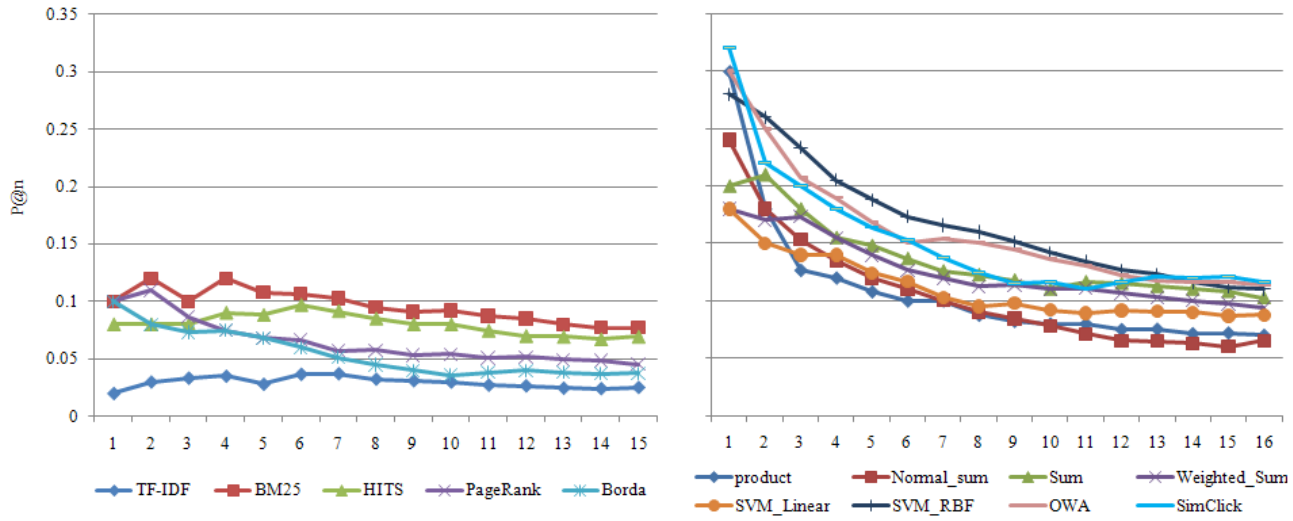

Figure 1. P@n on Letor dataset

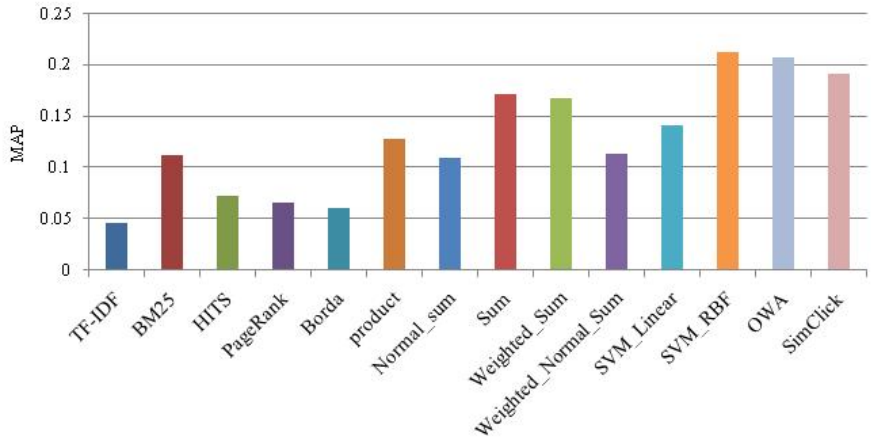

Figure 2. Mean Average Precision on Letor dataset

labeled (relevant or irrelevant) documents for each pair of query and document. Same as LETOR, 56 various features are extracted from dotIR collection. So, it can be another promising evaluation dataset for our presented ensemble methods.

B. Result Analysis

The main research question we want to answer is whether applying our proposed ensemble methods based on content and connectivity based methods would improve the performance. The problem of ranking of web pages has been studied extensively. Most existing studies of ranking algorithms focus on machine learning techniques, but there is not a comprehensive study to compare different algorithms on standard datasets.

To answer this question, we combine four different basic features with several ensemble methods and compare their performance on two benchmark collections. As the four basic features, we consider two classical content-based feature: TF-IDF [1] and BM25 [2], and two basic connectivity-based features: PageRank [3] and HITS [4]. Although this selection may seems to be a small portion from the big set of features, we were made to choose them. Because the other features are either too complicated and extracted for comparing the newly proposed methods or too simple for using as a combination feature. We use precision at different cut-off points ($P@n$) and mean average precision (MAP) for comparison.

1) Experiments on LETOR Data Collection: Figures 1 and 2 compare the performance of different ensemble methods with performance of the basic features on the LETOR collection. Figure 1 has been split in two charts in order to be more readable. As can be seen, the performances of proposed ensemble methods (right chart) are much better than the performance of basic features (left chart).

Also, Figure 1 shows that the simulated click through

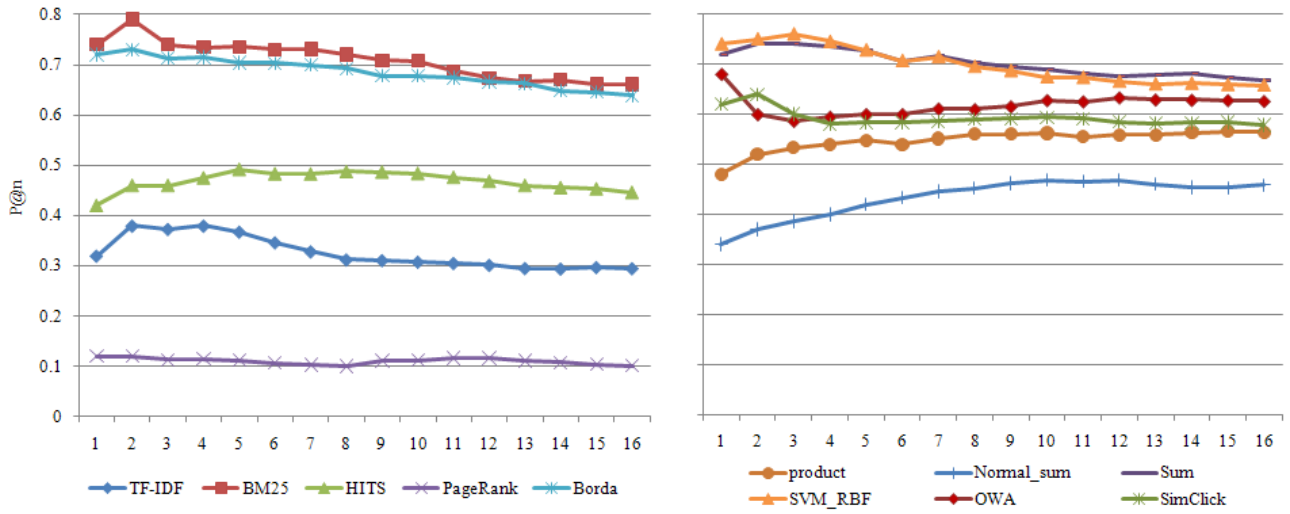

Figure 3. P@n on dotIR dataset

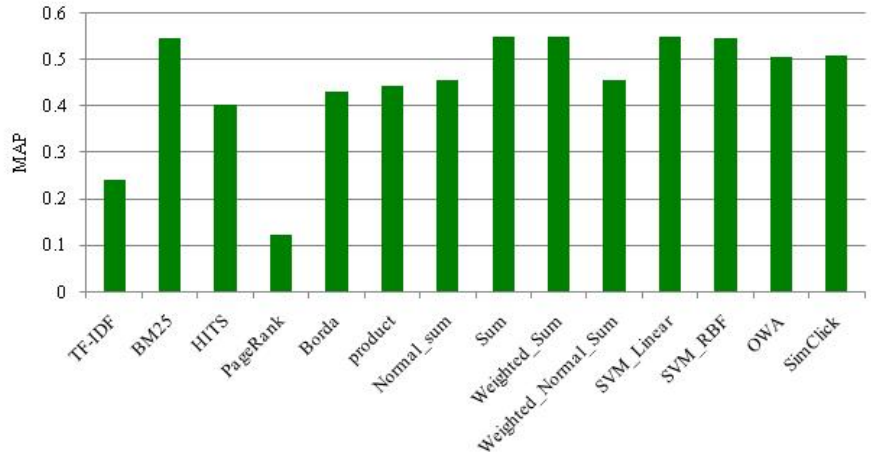

Figure 4. Mean Average Precision on dotIR dataset

data, SVM based fusion and OWA methods are the three best fusion approaches which have the most improvement in the final results. Furthermore, in figure 2, we compare the mean average precision of the different methods on the TREC-2003 of Letor benchmark. Although the mean average precision of the SVM based fusion is better than other fusion methods, as one can see in Figures 1 the OWA and simulated click through data fusion methods works slightly better than SVM for the high number of retrieved pages. However, the SVM based fusion and OWA fusion are more promising respect to other fusion methods.

2) Experiments on dotIR Data Collection: The extracted features of dotIR collection are the same as Letor. So we just apply our methods on this new dataset. Figures 3 and 4 compare the performance of different ensemble methods with the performance of the basic ranking methods as done in the previous experiment. As can be seen, these results have different behavior than the LETOR results. Since the accuracy of BM25 feature is very high, we can conclude that the content-based methods seems promising on this dataset. On the other hand, the performance of PageRank feature is relatively low. This may be as the result of the out-links removal. Since this collection is built by sampling the .ir domain, there are many broken links that should be omitted and they have direct influence on the PageRank accuracy.

In the right side of figure 3, we compare the results of ensemble methods. Some methods, such as weighted sum, have obtained almost the same result as the other methods. Thus, we have omitted them from the figure, in order to increase readability. As the figure shows, SVM and Sum methods give better results than the other ensemble methods. However, none of them is as accurate as BM25 in all the cases. In addition, mean average precision of BM25 is strangely high (it is almost 0.53). This means

that more research needed to be done on dotIR collection since its behavior is not same as the other standard IR test collections'.

IV. Conclusion and Future Work

In this work, we used different ensemble models such as Sum, Product, Borda rules and OWA, SVM and simulated click-through data methods to combine basic ranking features. Overall, we see that the ensemble methods are reasonable and all these specific fusion algorithms can help improve search results. The LETOR and dotIR benchmark datasets have been used for evaluation. Moreover, the results show that, more research needed to be done on the newly built dotIR collection. These models could also be applied to other collected real data sets such as Yahoo! learning to rank competition but that remains as future work.

References

[1] R. Baeza-Yates, B. Ribeiro-Neto *et al.*, *Modern information retrieval*. Addison-Wesley Reading, MA, 1999.

[2] S. Robertson, "Overview of the okapi projects," *Journal of Documentation*, vol. 53, no. 1, pp. 3–7, 1997.

[3] S. Brin, L. Page, R. Motwami, and T. Winograd, "The PageRank citation ranking: bringing order to the web," in *Proceedings of ASIS*, vol. 98, 1998, pp. 161–172.

[4] J. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM (JACM)*, vol. 46, no. 5, pp. 604–632, 1999.

[5] A. Shakery and C. Zhai, "A probabilistic relevance propagation model for hypertext retrieval," in *Proceedings of the 15th ACM international conference on Information and knowledge management*, 2006, pp. 550–558.

[6] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, "Learning to rank using gradient descent," in *Proceedings of the 22nd international conference on Machine learning*, 2005, pp. 89–96.

[7] Y. Cao, J. Xu, T. Liu, H. Li, Y. Huang, and H. Hon, "Adapting ranking SVM to document retrieval," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 2006, pp. 186–193.

[8] T. Joachims, "Optimizing search engines using clickthrough data," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002, p. 132142.

[9] Y. Freund, R. Iyer, R. Schapire, and Y. Singer, "An efficient boosting algorithm for combining preferences," *The Journal of Machine Learning Research*, vol. 4, pp. 933–969, 2003.

[10] A. Zareh Bidoki, P. Ghodsnia, N. Yazdani, and F. Oroumchian, "A3CRank: An adaptive ranking method based on connectivity, content and click-through data," *Information Processing & Management*, vol. 46, pp. 159–169, 2010.

[11] A. Bidoki and J. Thom, "Combination of Documents Features Based on Simulated Click-through Data," in *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, 2009, pp. 538–545.

[12] J. Yeh, J. Lin, H. Ke, and W. Yang, "Learning to rank for information retrieval using genetic programming," in *SIGIR 2007 workshop: Learning to rank for information retrieval*, vol. 27, 2007.

[13] R. Herbrich, T. Graepel, and K. Obermayer, "Large margin rank boundaries for ordinal regression," *Advances in Large Margin Classifiers*, pp. 115–132, 2000.

[14] G. Xue, H. Zeng, Z. Chen, Y. Yu, W. Ma, W. Xi, and W. Fan, "Optimizing web search using web click-through data," in *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, 2004, pp. 118–126.

[15] T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay, "Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search," *ACM Transactions on Information Systems (TOIS)*, vol. 25, no. 2, p. 7, 2007.

[16] R. Yager, "On ordered weighted averaging aggregation operators in multicriteria decisionmaking," *IEEE transactions on Systems, Man and Cybernetics*, vol. 18, no. 1, pp. 183–190, 1988.

[17] T. Qin, T. Liu, J. Xu, and H. Li, "LETOR: A benchmark collection for research on learning to rank for information retrieval," *Information Retrieval*, pp. 1–29, 2010.

[18] E. Darrudi, H. Baradaran Hashemi, A. Aleahmad, A. Habibian, A. Zareh Bidoki, A. Shakery, and M. Rahgozar, "A Standard Web test collection for .ir domain," *submitted to Iranian Journal of Electrical and Computer Engineering (IJECE)*, Fall 2009.

[19] dotir benchmark collection. [Online]. Available: http://ece.ut.ac.ir/dbrg/webir/

[20] P. Domingos, "A unified bias-variance decomposition and its applications," in *machine learning international conference*, 2000, pp. 231–238.

[21] C. Bishop, *Pattern recognition and machine learning*, 2006.

[22] F. Radlinski and T. Joachims, "Evaluating the Robustness of Learning from Implicit Feedback," *Learning in Web Search (LWS 2005)*, p. 42.

[23] T. Liu, J. Xu, T. Qin, W. Xiong, and H. Li, "Letor: Benchmark dataset for research on learning to rank for information retrieval," in *Proceedings of SIGIR 2007 Workshop on Learning to Rank for Information Retrieval*, 2007, pp. 3–10.

[24] C. Zhai and J. Lafferty, "A study of smoothing methods for language models applied to information retrieval," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 2, pp. 179–214, 2004.