# An Ensemble Approach to Learning to Rank

Dong Li, Yang Wang, Weijian Ni
*College of Information Technology Science*
*Nankai University*
*Tianjin 300071, China*
*{lidong.nk, wangyang022, niweijian}@gmail.com*

Yalou Huang, Maoqiang Xie
*College of Software*
*Nankai University*
*Tianjin 300071, China*
*{huangyl, xiemq}@nankai.edu.cn*

## Abstract

*In recent years, 'learning to rank' is a focused approach for information retrieval, which can learn the ranking order given by experts and construct a uniform model to rank for new query. But in practice user queries vary in large diversity, it makes a single learned ranker not representative. Therefore, we propose an ensemble approach to 'learning to rank, in which a lower generalization error can be gotten by generating a set of rankers and leveraging these rankers for the final prediction. Moreover, two strategies of creating multiple base rankers are proposed to make the ensemble more effective for information retrieval. The experiment results on two real world datasets indicate that the proposed approach can outperform the original 'learning to rank' methods significantly.*

## 1. Introduction

Ranking is the key problem in many information retrieval (IR) applications, such as document retrieval, web search and expert search. To solve the ranking problem, 'Learning to rank' is considered as a promising approach in recent years. By using machine learning methods, learning to rank can train ranking model with the order information given by experts, which makes the ranking results more accurate than unsupervised ranking methods.

Many methods of learning to rank have been proposed and applied in IR applications. Representatively, Herbrich et al. [1] and Joachims [2] took a SVM approach to learn the ranking function and Ranking SVM (RSVM) algorithm is proposed. And Burgus et al. [3] use the neural network method for ranking in RankNet, Freund et al. [4] adopted the boosting approach in RankBoost.

However, when the method of learning to rank are applied in real world, the prediction performance will be affected significantly by the defective training data. For example, in document retrieval, the training data consists of a set of queries, their corresponding retrieved documents, and relevance levels of them given by experts. Because the labeling task is subjective and expensive, the number of human labeled document instances varies largely from query to query. And since the type of query in practice varies a lot, the ranking characters of queries might be much diverse. Therefore, the single model learned from defective training data will be biased toward queries with more document instances, and the biased model will take biased ranking result for the diverse queries.

To these problems, we propose an ensemble approach to learning to rank. The ensemble method trains a set of base rankers, instead of an isolated one. All base rankers have same accuracy level, and also have their individual decision to prediction. Combing these base rankers, the ensemble can get a lower generalization error than base rankers by the leverage of rankers. [5]

Because the performance of ensemble is decided by the accuracy and diversity of base predictors [6], the key problem of constructing an effective ensemble for IR becomes finding a set of accurate rankers with high diversity. In our proposed approach, two strategies, bootstrapping and clustering, are introduced to create the base rankers with high accuracy and diversity. Moreover, the defects in training data is considered and utilized sufficiently in these strategies.

For validating the performance of the proposed ensemble approach, experiments on two real-world datasets: OHSUMED [7] and .Gov [8] are conducted. Experimental results indicate that the ensemble ranking can outperform the original 'learning to rank' method used as base learning algorithm significantly. Comparing and analyzing the results between two strategies, we also validate that the accuracy and diversity is the key of constructing the effective ensemble of ranking.

The rest of this paper is organized as follows: the summary of learning to rank and the problems they met in IR are given in Section 2. Focused on the problems, the ensemble ranking methods are proposed in details in Section 3. For validation, experimental results and analysis on two real-world datasets are given in Section 4. At last, Section 5 concludes this paper.

This work is supported by National Science Foundation of China under the grant 60673009, Tianjin Science and Technology Research Foundation under the grant 05YFGZGX24000 and Microsoft Research Asia Foundation.

978-0-7695-3305-6/08 $25.00 © 2008 IEEE
DOI 10.1109/FSKD.2008.188

## 2. Learning to Rank

### 2.1. Model

'Learning to rank' as a new and popular topic in machine learning have been applied for information retrieval in recent years. Using machine learning techniques, learning to rank can learn a ranking function with the training data labeled by human experts, and give the ranking prediction to new queries. To demonstrate, we describe the model for learning to rank in document retrieval as follows.

In training data, a set of queries $Q = \{q^{(1)}, q^{(2)}, \ldots q^{(m)}\}$ is given, where $n$ is denoted as the number of queries. As the labeled data, experts retrieve a list of document $d^{(i)} = \{d_1^{(i)}, d_2^{(i)}, \ldots d_{n^{(i)}}^{(i)}\}$ for each query $q^{(i)}$ and label the relevance level $y^{(i)} = \{y_1^{(i)}, y_2^{(i)}, \ldots y_{n^{(i)}}^{(i)}\}$ for each query-document pair $(q^{(i)}, d_j^{(i)})$, where $n^{(i)}$ is denoted as the number of retrieved documents for query $q^{(i)}$. So the training set can be represented as $S = \{(q^{(i)}, d^{(i)}, y^{(i)})\}_{i=1}^{m}$.

For each query-document pair, the feature vector $\vec{x}_j^{(i)} = \Psi(q^{(i)}, d_j^{(i)}) \in X$ is created. And the supervised ranking aim to find a ranking function $f : X \to \Re$. The ranking function can calculate the relevance score for each query and its corresponding document, and give the correct rank list by these scores. This model can also be applied in other IR application by using the retrieved instance replace the documents.

### 2.2. Prior works

When ranking problem is described as a machine learning problem, proposing and minimizing the ranking loss function becomes the key to learning to rank. There are several popular approaches to constructing the ranking loss function which are considered on different instance level. One is building the loss function on document instance level. SVOR [9] is proposed to minimize the rank loss by aggregating the error on each document instance. Another approach is pair-wise loss function, which create the pair instance between two documents with different relevance level, and denote correct rank pair as positive (+1) instance while incorrect as negative (-1). So the ranking problem is transformed into a binary classification problem in RSVM [1], RankBoost [2], RankNet [3]. And a recent approach is list-wise, which define rank loss with the difference between predicting document list and labeled list for each query, in AdaRank [10], ListNet [11].

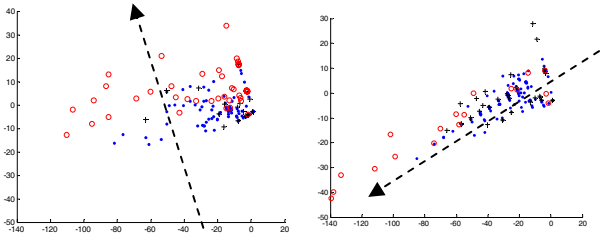
### 2.3. Problems

In real-world application, there are some defects in the training data, which will decrease the prediction performance of learning to rank significantly. One is the unbalance of training set. In fact, it is difficult for human experts to label the same size of documents for each query, because the label task is subjective and expensive. So in training data, the number of human labeled document instances varies largely from query to query. For example, in a real-world document retrieval dataset, OHSUMED, some queries have two or three times more instance than others. The distribution of the labeled is shown in Table 1.

**Table 1. The distribution of labeled instance number on OHSUMED**

| Labeled instance number | Query Number |
|---|---|
| >240 | 11 |
| >200 | 9 |
| >160 | 15 |
| >120 | 47 |
| >80 | 13 |
| <=80 | 10 |

Another problem is the diversity of queries. The type of query in real world is so various that different query may have diverse ranking character. We also use OHSUMED dataset to demonstrate the problem. Figure 1 shows two groups of queries dealt with PCA analysis and described by two principal coordinates. The instances with three rank levels from high to low (definitely relevant, partially relevant, and irrelevant) are denoted as the circle, plus and dot. The arrows denote the ranking characters for queries, through which the instances can be ranked most closely to the labeled rank level. In the figure, the directions of two arrows are quite different. It indicates that the ranking characters of these queries may have much diversity.



(a) Query No 5, 11 ,12 (b) Query No 10,41

**Figure 1. PCA analysis on OHSUMED**

Learning with these defective data, the performance of learning to rank methods will be affected significantly. Firstly, in most methods, especially in the approaches of document instance level and pair-wise level, the ranking

error in training data is strongly relative with the number of labeled instances. The trained model will be biased since some queries might have several times more instances than others. Secondly, because the queries in training set might have diverse ranking characters, the uniform model can not cover the overall queries in training set. Therefore, the trained biased ranker performs well only for a part of queries, but worse for the others. To solve these problems, we propose ensemble approach to learning to rank for IR.

## 3. Ensemble Approach to Learning to Rank

### 3.1. Ensemble of Ranking

In machine learning research, ensemble learning is a popular research direction. The ensemble firstly generates a set of base predictors which have same accuracy and individual decision for prediction, and then, combines their outputs in some way (typically by weighed or unweighted voting). If the base predictors have an error rate of better than random guessing and make errors uncorrelatedly, the ensemble of these base predictors will be more accurate than any of individual members [5].

The ensemble of ranking can be described as follow. Firstly, a set of base training sets are created by drawn instances from the original training set. Then, the base rankers are trained with these base training sets by using conventional algorithm of learning to rank. Finally, the results of base rankers are combined by aggregating function for the final prediction.

Using ensemble approach to learning to rank have three advantages. First is that a lower generalization error of ranking can be gained by effective ensemble. Second is that dejects of training data can be considered and utilized in the process of constructing base rankers. Thirdly, the conventional algorithms of learning to rank can be applied and improved in ensemble without much adapted.

In following, we analyze the generalization error of the ensemble of ranking, and propose the strategies of creating good base rankers to make ensemble more effective for information retrieval.

### 3.2. The Generalization Error Analysis

It is proved that the generalization error of ensemble can be decreased in regression and classification problems [6,12]. In ranking problem, the decreasing is also existed, which can be testified as follows.

Assume there are N base rankers in ensemble. Each ranker $\alpha$ is denoted as a ranking function $h_\alpha(x)$ from the instance feature vector $R^N$ to ranking score $R$. Using average aggregating function to construct the ensemble, the result of ensemble can be calculated as

$$\bar{h}(x) = \frac{1}{N}\sum_{\alpha} h_\alpha(x) \tag{1}$$

For the ranking instance $x$ with its target ranking level $y$, the ranking quadratic errors of base rankers and ensemble are

$$e_\alpha(x) = (y - h_\alpha(x))^2 \tag{2}$$

$$e(x) = (y - \bar{h}(x))^2 \tag{3}$$

Decomposed (3) and adding (2) for all base rankers, the error of ensemble is yielded as

$$e(x) = \frac{1}{N}\sum_{\alpha} e_\alpha(x) - \frac{1}{N}\sum_{\alpha}(h_\alpha(x) - \bar{h}(x))^2 \tag{4}$$

Denote the average accuracy $\bar{e}(x)$ and the average diversity of base rankers $\bar{d}(x)$ as

$$\bar{e}(x) = \frac{1}{N}\sum_{\alpha} e_\alpha(x) \tag{5}$$

$$\bar{d}(x) = \frac{1}{N}\sum_{\alpha}(h_\alpha(x) - \bar{h}(x))^2 \tag{6}$$

So (4) becomes $e(x) = \bar{e}(x) - \bar{d}(x)$.

Assume the predicted instances $x$ are drawn randomly from the distribution $p(x)$. All these formulas can be averaged over the distribution. Denote $E$, $\bar{E}$ and $\bar{D}$ as the average version for $e$, $\bar{e}$ and $\bar{d}$. The generalization error of ensemble ranking becomes

$$E = \bar{E} - \bar{D} \tag{7}$$

According to (7), the generalization error of ensemble $E$ will be lower the average error of base rankers $\bar{E}$, because the diversity of base rankers $\bar{D}$ are always positive. Moreover, the lower accuracy $\bar{E}$ and the higher diversity $\bar{D}$ will make the generalization error of ensemble lower. Therefore, the key of constructing an effective ensemble of ranking becomes how to find highly accurate and highly diverse base rankers.

### 3.3. Strategies of Creating Base Rankers

To create a set of accurate and diverse base rankers, creating proper base training sets is the most common approach in ensemble learning. We propose two strategies of creating base training sets, bootstrapping and clustering. They will consider and utilize the defects in training data to increase the accuracy and diversity of base rankers.

**3.2.1. Bootstrapping** (random with replacement) is a classic method to manipulate the training set in ensemble learning [12]. In our approach, we adapt the original bootstrapping to weaken the effect of unbalanced training data in order to get high accurate base rankers. The original re-sampling process is split into two steps: firstly draw a query from the training set, and then draw an instance from the selected query random with replacement. So after $n$ drawing processes, $n$ instances will be drawn into the base training set and all drawn instance will have the same probability in query level. Therefore, learned with these base training sets, the base rankers can get certain accuracy since the unbias of training data, and also get diversity with the random re-sampling.

**3.2.2. Clustering** strategy utilizes the diversity of queries. We cluster the queries by their ranking character to create diverse base training sets. The queries in same cluster will have similar ranking characters, and queries in different cluster are diverse.

In implement, we use ranking character of queries as the measure of clustering. Because the linear function $f(x) = \langle \boldsymbol{\omega}, \mathbf{x} \rangle$ is adopted by most learning to rank algorithm as the ranking function, we define the $\boldsymbol{\omega}^{(i)}$ as the ranking character of query $q^{(i)}$. $\boldsymbol{\omega}^{(i)}$ is the coefficient vector in ranking function $f^{(i)}(x^{(i)}) = \langle \boldsymbol{\omega}^{(i)}, \mathbf{x}^{(i)} \rangle$ which is learned by query $q^{(i)}$. Then, the cosine similarity between the ranking characters is defined as the distance measure of clustering.

$$D(q^{(i)}, q^{(j)}) = \cos(\boldsymbol{\omega}^{(i)}, \boldsymbol{\omega}^{(j)}) = \frac{\langle \boldsymbol{\omega}^{(i)}, \boldsymbol{\omega}^{(j)} \rangle}{\|\boldsymbol{\omega}^{(i)}\| \cdot \|\boldsymbol{\omega}^{(j)}\|} \quad (11)$$

Using common clustering method such as K-means algorithm, the queries in original training set can be partitioned into several groups. Learned with these groups as base training sets, the base rankers will have high diversity because of the diversity of queries.

# 4. Experiment

## 4.1. Experimental Settings

In our experiment, we use two benchmark datasets for information retrieval: OHSUMED and .Gov. And these two datasets is gained from Letor [13] Dataset released by Microsoft Research Asia.

We used two evaluation measures for evaluating the result of ranking methods: Mean Average Precision (MAP) [14] and Normalized Discounted Cumulative Gain (NDCG) [15]. MAP can evaluate the ranking algorithm's average overall accuracy, and NDCG is proposed to evaluate the accuracy of Top N ranking results. Both MAP and NDCG are the important evaluation measures in information retrieval.

In our experiments, Ranking SVM is selected as the baseline and the algorithm of training base rankers, because it is the representative methods of supervised ranking. For our ensemble ranking methods, we apply two ensemble strategies, bootstrapping and clustering respectively. We denote them as B-RSVM and C-RSVM. In bootstrapping, 20 base rankers are learned with base training sets, and each base training set contains 63.2% of original training set. In clustering, the training set is partitioned into 8 groups.

All the experiments conduct 5-cross validation. The datasets are divided into five parts. For each cross, three parts are chosen as training set, one as validation set to tune the parameter and the other one as the test set.

## 4.2. Experimental Results

We take the comparison with the original ranking algorithm RSVM and the ensemble ranking based on it. Figure 2 shows the ranking accuracies of RSVM and ensemble ranking methods on the OHSUMED data set in terms of both NDCG and MAP. Table 2 shows the relative improvements of ensemble ranking methods over the original RSVM. We can see the two ensemble methods both outperform the single model algorithm. Between the two ensemble strategies, B-RSVM is better on MAP, since C-RSVM is better on NDCG.

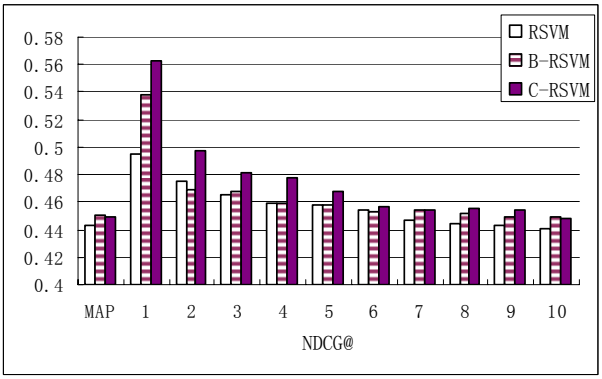



**Figure 2. Results for OHSUMED data**

**Table 2. Relative improvements of ensemble ranking methods on OHSUMED data**

| | MAP | NDCG@1 | NDCG@10 |
|---|---|---|---|
| B-RSVM | 1.76% | 8.58% | 1.94% |
| C-RSVM | 1.42% | 13.66% | 1.66% |

Figure 3 and Table 3 show the experimental results on the .Gov dataset. Again, ensemble methods improve upon

RSVM in ranking accuracy in terms of all measures. Particularly, B-RSVM and C-RSVM achieves about 20% and 12% relative improvements over RSVM in terms of MAP, which are higher than OHSUMED. Between the two ensemble strategies, B-RSVM is better than C-RSVM on both MAP and NDCG.

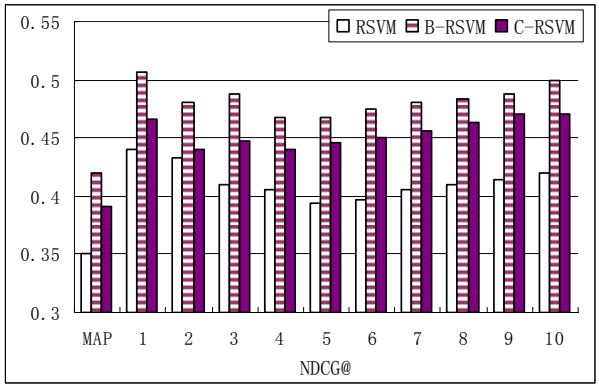



**Figure 3. Results for .Gov data**

**Table 3. Relative improvements of ensemble ranking methods on .Gov data**

| | MAP | NDCG@1 | NDCG@10 |
|---|---|---|---|
| B-RSVM | 19.81% | 15.15% | 18.76% |
| C-RSVM | 11.52% | 6.06% | 12.10% |

### 4.3. Comparison with Two Strategies

In Table 4, we show the average, standard deviation of base rankers, result of ensemble and relative improvement than base rankers on MAP. On the ensemble result on MAP, bootstrapping is better than clustering, because the average MAP of base rankers in bootstrapping, which represents the average accuracy of base rankers is much higher than clustering. On the other hand, on the relative improvement than base rankers, clustering is better than bootstrapping, since the representation of diversity, the standard deviation MAP of base rankers in clustering is much higher than clustering. There, it validates that both accuracy and diversity of base rankers are important to ensemble of ranking.

**Table 4. MAP analysis between two strategies**

| | E(MAP) | σ(MAP) | MAP | Relative impro. |
|---|---|---|---|---|
| | | OHSUMED | | |
| Bootstrapping | 0.43136 | 0.00586 | 0.45069 | 4.48% |
| Clustering | 0.39602 | 0.02501 | 0.44917 | 13.43% |
| | | .Gov | | |
| Bootstrapping | 0.35802 | 0.02225 | 0.41988 | 17.28% |
| Clustering | 0.26980 | 0.10197 | 0.39085 | 44.87% |

## 5. Conclusion

In this paper, we have proposed ensemble approach to learning to rank, which is more effective for information retrieval. We prove that ensemble can reduce the ranking generalization error, and the accuracy and diversity of base rankers is the key to the ensemble performance. To get more effective ensemble of ranking, we proposed two new strategies to create base rankers. One can obtain the accurate rankers by weakening the effect of unbalanced training data, and the other can take diverse rankers by utilizing the query diversity. Experimental results on real-world datasets indicate that the proposed approach outperforms original methods significantly.

## References

[1] R. Herbrich, T. Graepel, and K. Obermayer. "Large Margin Rank Boundaries for Ordinal Regression". Advances in Large Margin Classifiers, pages 115-132, 2000

[2] T. Joachims, "Optimizing Search Engines Using Clickthrough Data", Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD), ACM, 2002.

[3] C. Burges, T. Shaked, and et al. "Learning to rank using gradient descent". Proceedings of ICML 2005, Germany, 2005

[4] Y. Freund, R. D. Iyer, R. E. Schapire, Y. Singer. "An Efficient Boosting Algorithm for Combining Preferences". Journal of Machine Learning Research 4. pages 933-969, 2003

[5] T. G. Dietterich, "Ensemble Methods in Machine Learning". First International Workshop on Multiple Classifier Systems, 2000

[6] A. Krogh, J. Vedelsby. "Neural Network Ensembles, Cross Validation, and Active Learning". Proceedings of NIPS 1994, Denver, Colorado, USA, 1994

[7] W. R. Hersh, C. Buckley, and et al. "OHSUMED: An interactive retrieval evaluation and new large test collection for research". Proceedings of SIGIR 1994, Dublin, Ireland, 1994.

[8] N. Craswell and D. Hawking. "Overview of the TREC-2004 Web Track", In TREC, 2004.

[9] C. Wei, S. S. Keerthi. "New approaches to support vector ordinal regression". ICML 2005, Bonn, Germany, 2005

[10] J. Xu, H. Li. "AdaRank: a boosting algorithm for information retrieval". Proceedings of SIGIR 2007, Amsterdam, The Netherlands, 2007

[11] Z. Cao, T. Qin, and et al. "Learning to rank: from pairwise approach to listwise approach". Proceedings of ICML 2007. Oregon, USA, 2007

[12] L. Breiman, "Bagging predictors". Machine Learning, pages 123–140, 24, 1996

[13] T.Y. Liu, T. Qin, and et al., "LETOR: Benchmark dataset for research on learning to rank for information retrieval", Proceedings of LR4IR 2007, in conjunction with SIGIR 2007, The Netherlands, 2007

[14] R. A. Baeza-Yates, B. Ribeiro-Neto. Modern Information Retrieval, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1999

[15] K. Jarvelin and J. Kekalainen. "Cumulated Gain-based Evaluation of IR Techniques". ACM Transactions on Information Systems, Vol 20(4), pp.422~446, 2002.