Hierarchical Feature Selection for Ranking

Guichun Hua, Min Zhang, Yiqun Liu, Shaoping Ma, Liyun Ru
State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
huaguichun@gmail.com, {z-m, msp, yiqunliu}@tsinghua.edu.cn, lyru@vip.sohu.com

ABSTRACT

Ranking is an essential part of information retrieval(IR) tasks such as Web search. Nowadays there are hundreds of features for ranking. So learning to rank(LTR), an interdisciplinary field of IR and machine learning(ML), has attracted increasing attention. Those features used in the IR are not always independent from each other, hence the feature selection, an important issue in ML, should be paid attention to for LTR. However, the state-of-the-art LTR approaches merely analyze the connection among the features from the aspects of feature selection. In this paper, we propose a hierarchical feature selection strategy containing 2 phases for ranking and learn ranking functions. The experimental results show that ranking functions based on the selected feature subset significantly outperform the ones based on all features.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Retrieval models

General Terms: Algorithms, Experimentation

Keywords: Learning to Rank, Feature Selection

1. INTRODUCTION

Web search engines are often referred to when people are requiring some information from Internet, and ranking is an essential part in the structure of search engines. Nowadays, hundreds of features for ranking have been proposed e.g. content-based features such as $TFIDF$, $BM25$; link-based features such as $PageRank$, $HITS$; user behavior features based on click-through data. It is a hot research field to construct more efficient ranking functions based on these features, so LTR, an interdisciplinary field of IR and ML, has gained increasing attention for a few recent years.

The conventional ML research shows that the features and the composition of the features affect the performance of learning methods, and the construction methods of ranking functions for IR show that the features are not independent from each other. For example, the features of TF(Term Frequency) and IDF(Inverse Document Frequency) are elements to construct the feature $BM25$. However, the state-of-the-art LTR approaches merely analyze the connection among the features from the aspects of feature selection except [6, 2] to the best of our knowledge. [6] applies the boosted regression trees to select the proper feature subset. [2] considers the feature importance and similarity between two features, and proposes an efficient greedy feature selection method. However, they are both flat feature selection methods which may be biased, and they could not decide which number of features selected is proper.

The main contributions of this paper are that : (1) we propose a hierarchical feature selection strategy containing 2 phases to make the selected features not biased. (2) design a quality measure to decide the proper number of selected features. We use Ranking SVM(RankSVM) [3, 4] and ListNet [1] to verity the strategy because they are powerful and commonly used approaches in LTR [7, 5]. The experimental results show that our feature selection methods do significantly improve the performance of the ranking functions.

2. FEATURE SELECTION STRATEGY

The process of the hierarchical feature selection strategy contains 2 phases: (1) the similarity between any two features is measured, and the similar features are aggregated into groups; (2) the representative feature in each group is selected through either delegation method. By this way, the selected features are not biased to a group of features which are more representative than the ones in other group.

2.1 Cluster-based feature similarity analysis

The Kendall's τ is chosen as the feature similarity measure and the similarity between features f_i and f_j: $sim(f_i, f_j)$ is calculated as follows:

$$\tau_q(f_i, f_j) = \frac{\#\big\{(d_s, d_t) \in D_q | d_s \prec_{f_i} d_t \ and \ d_s \prec_{f_j} d_t\big\}}{\#\big\{(d_s, d_t) \in D_q\big\}}$$

where $d_s \prec_{f_i} d_t$ denotes d_t ranks higher than d_s based on f_i for document pair (d_s, d_t) in the set D_q w.r.t. a query q, and $\#\{.\}$ denotes the number of elements in the set $\{.\}$.

Features are clustered according to their similarities. We define a measure based on the intra-cluster similarities to estimate the quality of clustering results. But the intra-cluster similarity would be maximum if the cluster have only one element. Therefore the $Penalty$ is defined to reduce such effect, and it is the average of all similarities as: $\frac{2}{N*(N-1)}\big(\sum_{f_v, f_u \in F} sim(f_v, f_u)\big)$ where N is the number of features in feature set F. The quality measure is as follows:

$$Quality_n = \sum_{i=1}^{n} \begin{cases} Penalty & N_i = 1 \\ \frac{2}{N_i*(N_i-1)}\big(\sum_{f_v, f_u \in F_i} sim(f_v, f_u)\big) & N_i > 1 \end{cases}$$

where n is the number of clusters. The clustering method we choose is K-Means. By this way, the number of clusters can be decided with the quality value.

WWW 2010, April 26–30, 2010, Raleigh, North Carolina, USA.
ACM 978-1-60558-799-8/10/04.

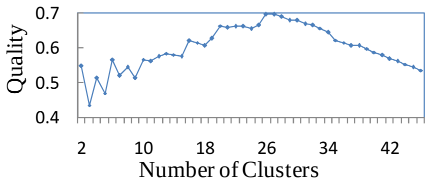

Figure 1: Quality of all clustering results

2.2 Delegation methods

We proposed two delegation methods to choose features from each cluster.

First is the Delegation Method Based on Evaluation Measure (BEM). In each cluster, every feature is used solely for ranking on training. Both normal and inverse value of the feature are applied respectively. The ranking results are measured with most commonly used criteria in IR such as MAP and $NDCG@n$. The feature (or the inverse of the feature) leading to the best performance is selected from the cluster.

Second is Delegation Method Implied by LTR Method ($ILTR$). Most of the LTR algorithms generate the final ranking functions as linear mode:$\sum_{i=1}^{N} \omega_i * f_i$, where ω_i is the weight of the feature f_i. The feature leading to the highest weight is selected from the cluster.

3. EXPERIMENT

3.1 Experiment Settings

The experiment dataset is LETOR which is broadly used in LTR research. The LETOR4.0 is released in July 2009 with 25,205,179 web pages and two query sets from Million Query Track of TREC2007(1692 queries) and TREC2008(784 queries) marked as MQ2007(MQ7) and MQ2008(MQ8) respectively in the following. 5-fold cross validation has been made in the experiments: three for training, one for validation and one for test. And experimental results are analyzed in terms of NDCG@n.

3.2 Experimental Results and Analysis

The quality measure of each clustering result shows in Fig 1, then the number of clusters is decided as 26.

The ranking function name is shown as *Dataset_LTR Algorithm_Feature Selection Algorithm*. *Dataset* is MQ7 or MQ8. *LTR Algorithm* is RankSVM(RS) or ListNet(LN). *Feature Selection Algorithm* is MAP denoting BEM with MAP, rs denoting $ILTR$ with RankSVM, *ln* denoting $ILTR$ with ListNet, or *All* denoting method based on all features(the baseline of our work that do not use feature selection).

Comparative results of feature selection on MQ7 are shown in Fig 2(a). Features selected in MQ7 are applied directly in MQ8, whose performance is shown in Fig 2(b). The paired T-Tests are conducted on the improvements of NDCG@n(p-value<0.05 means significant improvements; p-value<0.01 means very significant improvements).

The Fig 2 shows that: (1) the feature selection using BEM with MAP consistently achieves significant performance for ranking: *MQ7_RS_MAP*, *MQ7_LN_MAP*, *MQ8_RS_MAP* and *MQ8_LN_MAP* outperform the baselines with p-value= 0.0249, 0.0008, 0.0002 and 0.0004 independently. (2) in MQ7, the best performance is obtained by *MQ7_RS_rs* with p-value= 0.0006 v.s. the baseline *MQ7_RS_All*. In MQ8, all ranking functions outperform the baseline ones, and *MQ8_LN_MAP* obtains the best performance with p-value= 0.0004 v.s. the baseline *MQ8_LN_All*. (3) the feature selection using $ILTR$ with RankSVM does improve the performance for ranking, while the one using $ILTR$ with ListNet gains poor performance in MQ7.

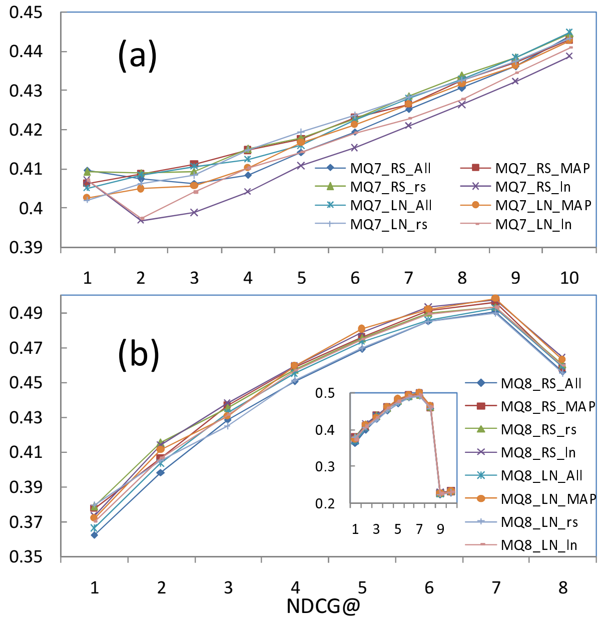

Figure 2: Comparison Results with NDCG@1∼10 on MQ7 and MQ8

4. CONCLUSIONS

In this paper, we propose a hierarchical feature selection strategy containing 2 phases and design a quality measure with which the number of clusters can be decided. The experimental results show that our methods could achieve significant improvement for ranking.

5. ACKNOWLEDGMENTS

This work is supported by Natural Science Foundation (60736044, 60903107) and Research Fund for the Doctoral Program of Higher Education of China (20090002120005).

6. REFERENCES

[1] Z. Cao and et.al. Learning to rank: from pairwise approach to listwise approach. In *ICML 2007*, pages 129–136.

[2] X. Geng and et.al. Feature selection for ranking. In *SIGIR 2007*, pages 407–414.

[3] R. Herbrich and et.al. Large margin rank boundaries for ordinal regression. In *Advances in Large Margin Classifiers*, pages 115–132, 2000.

[4] T. Joachims. Optimizing search engines using clickthrough data. In *KDD 2002*, pages 133–142.

[5] T.-Y. Liu. Learning to rank for information retrieval. In *Foundation and Trends on Information Retrieval*, pages 641–647, 2009.

[6] F. Pan and et.al. Feature selection for ranking using boosted trees. In *CIKM 2009*, pages 2025–2028.

[7] M. Zhang and et.al. Is learning to rank effective for web search. In *SIGIR 2009 workshop: Learning to Rank for Information Retrieval*, pages 641–647.