

On Domain Similarity and Effectiveness of Adapting-to-Rank

Keke Chen
Department of Computer Science and
Engineering
Wright State University, Dayton, OH 45435
keke.chen@wright.edu

Jing Bai Srihari Reddy Belle Tseng
Yahoo! Labs
2811 Mission College Blvd., Santa Clara, CA
94089
{jingbai,sriharir,belle}@yahoo-inc.com

ABSTRACT

Adapting to rank address the the problem of insufficient domain-specific labeled training data in learning to rank. However, the initial study shows that adaptation is not always effective. In this paper, we investigate the relationship between the domain similarity and the effectiveness of domain adaptation with the help of two domain similarity measure: relevance correlation and sample distribution correlation.

Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Retrieval Models

General Terms

Algorithms

Keywords

Learning to Rank, Model Adaptation, Similarity Measure

1. INTRODUCTION

Learning to rank has been a promising method for continuously and efficiently improving the quality of relevance for information retrieval systems [6]. However, learning to rank usually requires sufficient amount of labeled training data of good quality, which is highly expensive. Search in multiple domains and continuously maintaining multi-domain ranking functions make this problem urgent to handle. Domain adaptation, or adapting to learn, has been known to be one of the effective methods to address the emerging multi-domain learning-to-rank problem. The basic idea is to share the data or functions between different domains, and thus for those domains that have little data we can still obtain good-quality ranking functions.

Although domain adaptation has been studied recently in other areas like speech recognition and language modeling, their problem setting and modeling methods have been very different from learning to rank in search. Especially, many questions for domain

adaptation were not addressed and satisfactorily answered. 1) It has been observed that if the overall performance downgrade is small, it is often appealing to apply *one* unified ranking function to a set of domains. However, for what kind of domains can we apply this one-function approach? 2) The simple weighted data combination approach, i.e., combining the appropriately weighted training datasets from different domains to train a function for the target domain, has been shown effective in some areas [4], while not always effective in learning to rank [1]. It is thus important to understand when and why adaptation works.

In this paper, we propose to analyze the domain similarity to understand the effectiveness of the domain adaptation methods. Two novel methods - relevance correlation analysis and sample distribution similarity analysis, are developed to analyze the domain similarity. We first look at the domain similarity from the marco-level, i.e., similar domains should have highly correlated relevance performance on a set of relevance measures. Then, we study the sample-level similarity by developing a method based on regression-tree modeling [2] to visually analyze the similarity of sample distributions. Experiments are performed to study the relationship between domain similarity and the effectiveness of model adaptation with two popular adapting-to-rank methods: data combination and Trada tree adaptation [1]. The results show that if the domain similarity is very high or very low, the two adaptation methods do not bring additional benefits, and adaptation methods works well when the domain similarity is medium.

2. DOMAIN SIMILARITY ANALYSIS FOR RANKING ADAPTATION

We argue that domain similarity is an important factor that affects the result of domain adaptation. This can be formally derived from the Bayesian analysis of the multi-domain training. Let a labeled training example be $\{(\mathbf{x}_i, y_i)\}$, where \mathbf{x}_i is the feature vector derived from the match of the i -th (query, document) pair, and y_i is the target value judged by the relevance expert. Let s and t denote the source and target domains, respectively. We ignore the process of getting the following result

$$p(y_t|\mathbf{x}_t, \mathbf{x}_s, y_s) = \frac{p(\mathbf{x}_t|\mathbf{x}_s, y_t)p(y_t|y_s)}{p(\mathbf{x}_t|\mathbf{x}_s)} \quad (1)$$

Therefore, the key is to understand the similarity (or the dependency, if the data from both domains are not normalized) between the domains. To conveniently understand this similarity we design two methods: relevance correlation and multidimensional sample similarity analysis.

2.1 Relevance Correlation

The goal of similarity analysis is to find those domains that have

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

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

Copyright 2009 ACM 978-1-60558-512-3/09/11 ...\$10.00.

high correlation in terms of relevance so that we can share training resources between similar domains in adaptation. Ideally, if we improve the relevance of one domain in a group of similar domains, other domains can benefit from this improvement as well. Therefore, a natural way to define the domain similarity is to use *relevance correlation* as the similarity measure. We propose to use the following “landmark” based relevance correlation analysis. (1) We first train a set of ranking functions based respectively on the data from a set of domains ($M_j, j = 1, \dots, k$) as the “*relevance landmarks*”. (2) For each domain $D_i, i = 1, \dots, l$, under investigation we use the available domain-specific data to test the landmark ranking functions with *some relevance measure*. With the relevance test results, we obtain a “relevance vector” of k elements for each D_i , $R_i = \{r_{i1}, r_{i2}, \dots, r_{ik}\}$. (3) By evaluating the correlation (i.e., Pearson correlation) between any pair of relevance vectors, we obtain a similarity matrix.

There are two important factors in calculating relevance correlation: the relevance measure and the selection of relevance landmarks. We consider a few relevance metrics that have been popularly used in literature [3, 5], including Normalized Discounted Cumulative Gain (NDCG) (or the non-normalized version: DCG) and Mean Average Precision (MAP). NDCG is a metric designed for evaluating the quality of ranked list if the grades for items in the list are known. Suppose there are k documents used for testing the query q . Each query-document pair (q, d_i) in the test set is labeled with a grade l_i . A particular ranking will sort the list of the k documents in certain order. Let $i = 1, \dots, k$ be this order.

$$NDCG_k = Z_k \sum_{i=1}^k \frac{2^{l_i} - 1}{\log(i + 1)}$$

where Z_k is normalization factor so that the perfect ranking will give $NDCG_k = 1$. We often use the average NDCG of the queries in the test set as the final quality measure. Note that different queries in the test set might have different normalization factor Z_k , depending on the grade distribution for the particular query. Therefore, DCG that removes Z_k normalization from NDCG has a different distribution from NDCG.

MAP is defined on the precision at position k , P_k , often used for the two-level grading scheme $\{irrelevant, relevant\}$. Let $rel(i)$ be a boolean function indicate whether the document i is relevant or not. P_k is defined by $P_k = \frac{\sum_{i=1}^k rel(i)}{k}$, and average precision at position k is defined by

$$AP_k = \frac{\sum_{i=1}^k P_i \cdot rel(i)}{\sum_{i=1}^k rel(i)}$$

MAP is the average AP over all queries in the test set. We found that NDCG has very strong correlation with MAP (> 0.9) in experiments. Therefore, we can either use NDCG or MAP in relevance correlation analysis.

2.2 Similarity Analysis on Sample Distributions

Relevance correlation checks the similarity at the macro-level, which gives a rough idea on how the domains are related. To further understand the difference between any pair of domains, we need to analyze the samples. In this section, we propose a pairwise domain similarity analysis method based on regression tree (and gradient boosting trees (GBT)) and sample datasets.

Analyzing multidimensional distribution, particularly high dimensional data, is well known as a difficult problem. However, datasets for the learning problem has the additional information: the labels. We use the labels to guide the joint feature-label distribution analysis in the proposed method. To start with, we will

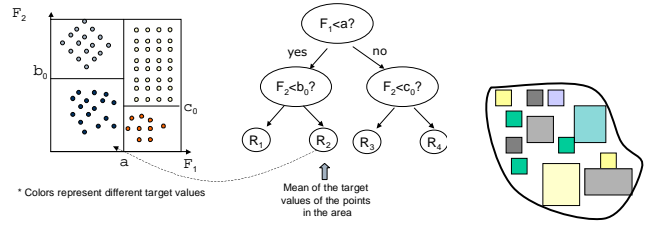


Figure 1: Left: a perfectly fitted tree. Right: the sample space partitioned by a regression tree.

briefly introduce the regression tree and GBT modeling method [2]. In Figure 1, we use small blocks to illustrate the local areas in the high-dimensional space. Different colors represent different target values for the blocks. A regression tree model tries to model the blocks with multi-dimensional bounding boxes defined by the tree and the approximate predicted value for the blocks. In regression tree, each leaf node tries to approximately model one of these blocks, and each internal node groups a few nearby blocks with close target values. GBT modeling refines the approximation by multiple boosting trees [2].

The basic idea of regression-tree based sample distribution analysis is described as follows. Suppose we have two domains: the source domain with large training data (as the reference domain), and the target domain with small training data. We train a single regression tree with a number of leaf nodes, say $p = 100$ (this number depends on the level of granularity we desire) with the source domain data. Each leaf node is defined by a set of disjoint conditions, e.g., “ $F_1 < v_{11}$ and $F_2 < v_{21}$ and $F_1 \geq v_{12}$ ”. We use this tree as the reference model, and map the data from the target domain onto the subspaces formed by the leaf nodes. This mapping is easily done by passing each feature vector of target domain through the tree. Suppose there are n_i records falling onto the leaf node i . Let $t_{i,j}$ be the grade for the j -th record and R_i be the response value of the node. After all records are mapped, the mean square error of prediction for the node i for the target domain can be calculated: $1/n_i \sum_{j=1}^{n_i} (t_{i,j} - R_i)^2$. We use a couple of statistics to describe the samples from one particular domain in each leaf node.

(1) The *normalized* number of samples falling onto each node, $n_i, i = 1 \dots p$. Since the total number of samples may not be the same for a pair of domains, we need to normalize this number. Assume the sample is uniformly drawn from the domain, i.e., with the increase of total population, the samples in each subspace will be increased proportionally. Let the rate r between two sample sets be $r = N_1/N_2$, where N_1 and N_2 are the total number of samples in the domain 1 and 2, respectively. The number of domain 2 samples at a node n_{2i} is normalized to $r \times n_{2i}$.

(2) The mean square error (MSE) on each node, $e_i, i = 1 \dots p$. Since the goal of this similarity analysis is to find whether two datasets are consistent in terms of regression modeling, and a leaf node models a part of the regression function, it is very meaningful to compare the leaf-level modeling error for both the source and target domains.

Based on the above definitions, we analyze several typical patterns in the sample comparison and MSE comparison graphs. First, the left subfigure in Figure 2 illustrates a well matched sample distributions. Whereas the right subfigure shows another situation, in which some nodes do not have samples from the target domain, which shows the significantly different parts between the two domains. The pattern at the right subfigure implies a good opportunity

that domain adaptation can succeed – the missing part can be possibly patched by the source domain data.

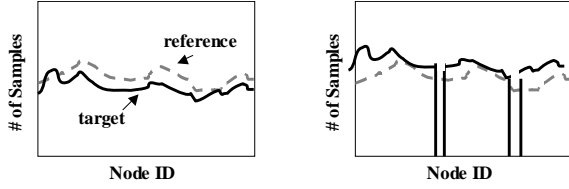


Figure 2: Comparing sample distribution patterns.

If the sample distributions are very similar except for some missing parts, we can turn to check the MSE distribution. Often, we see the situation illustrated by the left subfigure in Figure 3, where the MSEs of the target domain seem much higher than the average level of the reference model. There are two possibilities: either the target domain data have lots of noisy labels or its label distribution is very different from that of the source domain. The average MSE on the model trained on the small target domain data (the right subfigure) can help us understand which of the two causes is more likely. If the average MSEs are very close, it is more likely the target labels are noisy. In this case, a large dataset from the source domain may help reduce the effect of noise data.

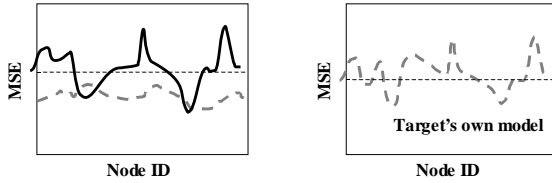


Figure 3: Comparing MSE distribution patterns.

3. EXPERIMENTS

Datasets. We will use the publicly available LETOR datasets [5] for experiments. LETOR datasets include the 2003/2004 TREC web track data. There are three search tasks in TREC Web track: topic distillation (TD), homepage finding (HP) and named page finding (NP). TREC evaluators provide two-grade judgments for these tasks, i.e., {relevant, irrelevant}. The recent LETOR datasets (version 3.0) include six TREC web track datasets, i.e., 2003/2004 TD/HP/NP data. Note that the same type of datasets may have different distributions from year to year due to the evolution of the Web. Because they share the same set of features, we are able to use them to simulate six different domains.

Domain Similarity Analysis Since no other public domain data is available as relevance landmarks, we use these six domains as “mutual landmarks” – when we analyze two domains, the remaining four domains serve as the relevance landmarks. Although the number of landmarks is small, we can still find some interesting information. For each landmark domain, we train a GBT-based ranking model with approximately optimal parameter setting in (100, 150) trees, (8, 10) terminal nodes per tree, 0.05 shrinkage rate, and (0.4, 0.5) sampling rate - The numbers in () are the possible choices. Then, these four models are used to test the two target domain datasets, which results in two 4-element relevance vectors. We use NDCG5 to generate the relevance vectors: three domains HP04, NP04 and TD04 are highly correlated to each other to form a group (Figure 4), while the remaining domains have low correlation with other domains. We also compare the three evaluation

metrics: NDCG, DCG, and MAP, on all testing results. Figure 5 shows that NDCG and MAP are highly correlated ($\rho = 0.9878$), while DCG seems less correlated with NDCG and MAP.

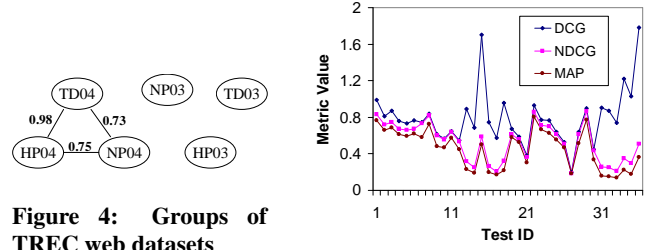


Figure 4: Groups of TREC web datasets

Figure 5: Correlation between metrics

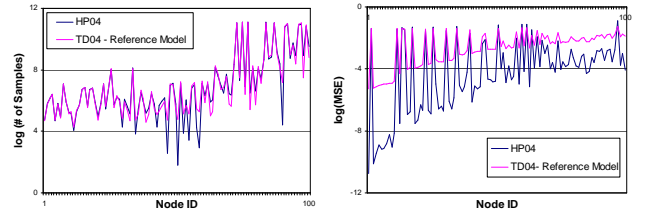


Figure 6: HP04 data distribution is well matched with TD04 data distribution (left). The MSE of HP04 data is generally smaller than TD04's (right).

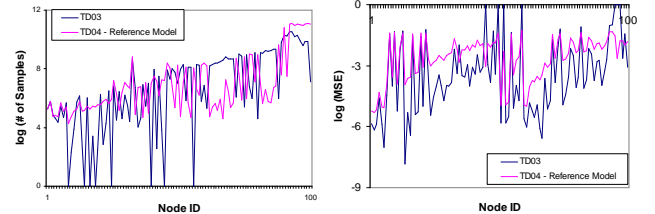


Figure 7: TD03 data distribution is very different from TD04 data distribution, and no sample at some nodes (left). TD03 data generate large MSEs in many nodes (right).

Next, we use tree-based sample distribution analysis to validate the relevance correlation result, focusing on the typical pairs of domains. We use TD04 data to train a reference model with GBT method (10 trees and each tree has 10 leaf nodes). Then, we map both TD04 and HP04 data to the TD04 model and see the difference between the statistics. The two domains have similar sizes of data, $\sim 75,000$ examples. The numbers in the figures are transformed with log function to make the presentation clearer. The left subfigure in Figure 6 shows that the sample distributions are very close for most nodes. The right subfigure shows that HP04 has even lower MSE on most nodes than TD04 data, which means HP04's data distribution has lower complexity than TD04's. We also look at TD03 and TD04, which have a low relevance correlation. Figure 7 shows that the two sets of data have very different sample distributions, and the label difference between the overlapped distribution can be large (right subgraph). Therefore, the result of sample-level analysis is consistent with relevance correlation analysis.

Effectiveness of Adaptation In this set of experiments, we will see how domain similarity is related to the effectiveness of adaptation. Three training algorithms are compared: 1) directly applying the function learned in the source domain to the target domain; 2) combining data by appropriately weighting the target domain data; 3) adapting the source domain function to the target domain with the Trada tree adaptation algorithm [1] that adjusts the gradient boosting tree structure with the target domain data. The reported numbers are based on the average of the five-fold testing results. The preliminary parameter probing experiments are performed to determine the acceptable range of parameter settings, and the validation sets are used to finally determine the best set of parameter. We also perform significance test (t -test) on some comparisons (p -value < 0.05 means statistically significant). With each correlation level, we also investigate the effect of the target training data size to the result. We vary the size of the target domain training data (5, 10, 20, 30, 40 queries, respectively) to study the effect of the size of target domain data.

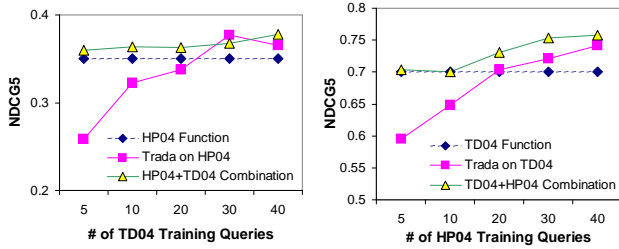


Figure 8: For TD04 and HP04, using either one of the datasets can generate reasonably good models for the other domain, while data combination is slightly better.

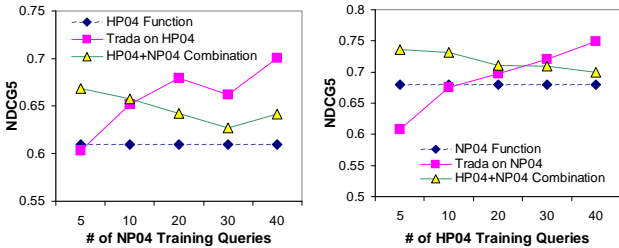


Figure 9: For NP04 and HP04, with the increase of data size, data combination becomes worse than Trada.

TD04 and HP04 are the high-correlation example. Figure 8 shows the result on two cases: TD04 as the target domain (left) and HP04 as the target domain (right). In the left subfigure, “HP04 function” means the function trained with only the HP04 data. “Trada on HP04” means using the HP04 function as the base model to perform Trada adaptation. “TD04+HP04 Data” means data combination method. Similar annotations are applied to other figures later. We find that simply applying the function learned in one domain to another will give sufficiently good performance if the domains have a high similarity. All small improvement brought by adaptation methods are not statistically significant. Data combination with target domain overweighting is slightly better than Trada. Figure 9 shows that both data combination and Trada adaptation help for less strongly correlated domains: NP04 and HP04. With extremely small data, data combination works better, while Trada

outperforms with more data. We also look at two low-correlated domains, TD03 and TD04 (Figure 10). Both Trada and data combination improve the relevance compared to the function from the source domain. However, they do not outperform functions trained with sufficient highly correlated domain data (the function ‘HP04’ on the left subfig).

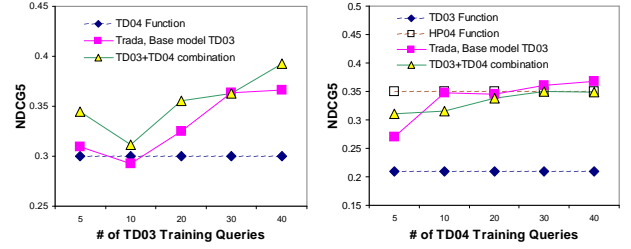


Figure 10: TD03 and TD04 are with low correlation. Domain adaptation helps in this case, while not better than a function from a highly correlated domain.

4. CONCLUSION

With the increasing requirements from different domains, domain adaptation has become an important method to address the problem of insufficient training data. In this paper, we study the similarity between search domains and its relationship with the effectiveness of different adaptation algorithms. Two similarity measures: relevance correlation and sample similarity are developed to study the domains similarity. Experimental results show that adaptation algorithms help more on medium domain similarity while it is less helpful on extremely high or low domain similarity.

5. REFERENCES

- [1] CHEN, K., LU, R., WONG, C., SUN, G., HECK, L., AND TSENG, B. Trada: Tree based ranking function adaptation. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)* (2008).
- [2] FRIEDMAN, J. H. Greedy function approximation: A gradient boosting machine. *Annals of Statistics* 29, 5 (2001), 1189–1232.
- [3] JARVELIN, K., AND KEKALAINEN, J. IR evaluation methods for retrieving highly relevant documents. In *Proceedings of ACM SIGIR Conference* (2000).
- [4] JIANG, J., AND ZHAI, C. Instance weighting for domain adaptation in NLP. In *Conference of the Association for Computational Linguistics (ACL)* (2007).
- [5] LIU, T.-Y., QIN, T., XU, J., XIONG, W., AND LI, H. LETOR: Benchmark dataset for research on learning to rank for information retrieval. In *Workshop of Learning to Rank for Information Retrieval, in conjunction with SIGIR* (2007).
- [6] ZHENG, Z., CHEN, K., SUN, G., AND ZHA, H. A regression framework for learning ranking functions using relative relevance judgments. In *SIGIR* (2007), pp. 287–294.