Knowledge and Data Engineering, IEEE Transactions on Publication Year: 2012 , Page(s): 745 - 758

Ranking Model Adaptation for Domain-Specific Search

Bo Geng, *Member, IEEE*, Linjun Yang, *Member, IEEE*, Chao Xu, Xian-Sheng Hua, *Member, IEEE*

Abstract—With the explosive emergence of vertical search domains, applying the broad-based ranking model directly to different domains is no longer desirable due to domain differences, while building a unique ranking model for each domain is both laborious for labeling data and time-consuming for training models. In this paper, we address these difficulties by proposing a regularization based algorithm called ranking adaptation SVM (RA-SVM), through which we can adapt an existing ranking model to a new domain, so that the amount of labeled data and the training cost is reduced while the performance is still guaranteed. Our algorithm only requires the prediction from the existing ranking models, rather than their internal representations or the data from auxiliary domains. In addition, we assume that documents similar in the domain-specific feature space should have consistent rankings, and add some constraints to control the margin and slack variables of RA-SVM adaptively. Finally, *ranking adaptability* measurement is proposed to quantitatively estimate if an existing ranking model can be adapted to a new domain. Experiments performed over Letor and two large scale datasets crawled from a commercial search engine demonstrate the applicabilities of the proposed ranking adaptation algorithms and the *ranking adaptability* measurement.

Index Terms—Information Retrieval, Support Vector Machines, Learning to Rank, Domain Adaptation.

1 Introduction

Learning to rank is a kind of learning based information retrieval techniques, specialized in learning a ranking model with some documents labeled with their relevancies to some queries, where the model is hopefully capable of ranking the documents returned to an arbitrary new query automatically. Based on various machine learning methods, e.g., Ranking SVM [12], [14], RankBoost [9], RankNet [4], ListNet [5], LambdaRank [3], etc., the learning to rank algorithms have already shown their promising performances in information retrieval, especially Web search.

However, as the emergence of domain-specific search engines, more attentions have moved from the broad-based search to specific verticals, for hunting information constraint to a certain domain. Different vertical search engines deal with different topicalities, document types or domain-specific features. For example, a medical search engine should clearly be specialized in terms of its topical focus, whereas a music, image or video search engine would concern only the documents in particular formats.

Since currently the broad-based and vertical search engines are mostly based on text search techniques, the ranking model learned for broad-based can be utilized directly to rank the documents for the verticals. For example, most of current image search engines only utilize the text information accompanying images as the ranking features, such as the term frequency (TF) of query word in image title, anchor text, alternative text, surrounding text, URL and so on. Therefore, Web images are actually treated as text-based documents that share similar ranking features as the document or Web page ranking, and text based ranking model can be applied here directly. However, the broad-based ranking model is built upon the data from multiple domains, and therefore cannot generalize well for a particular domain with special search intentions. In addition, the broad-based ranking model can only utilize the vertical domain's ranking features that are same to the broad-based domain's for ranking, while the domain-specific features, such as the content features of images, videos or music can not be utilized directly. Those features are generally important for the semantic representation of the documents and should be utilized to build a more robust ranking model for the particular vertical. Alternatively, each vertical can learn its own ranking model independently. However, it's laborious to label sufficient training samples and time-consuming to train different models for various verticals, since the number of verticals is large and increasing drastically.

Based on our experimental results, the ranking model of the broad-based search can provide a reasonable, though not as perfect as the specifically trained, ranking model for vertical search applications. Thereafter, we can make a trade-off between the direct using of the broad-based model and the independent learning of a completely new ranking model, for each specific vertical. That is, the broad-based ranking model can be adapted, with the help of several labeled samples and their domain-specific features, for ranking the documents in new domains. Because the existing broad-based ranking

Manuscript received Mar. 17, 2010.

B. Geng and C. Xu are with the Key Laboratory of Machine Perception (Ministry of Education), Peking University, Beijing 100871, China (email: bogeng@pku.edu.cn; xuchao@cis.pku.edu.cn).
L. Yang and X.-S. Hua are with Microsoft Research Asia, Beijing 100190, China (email: linjuny@microsoft.com; xshua@microsoft.com).

model provides a lot of common information in ranking documents, only few training samples are needed to be labeled in the new domain. From the probabilistic perspective, the broad-based ranking model provides a prior knowledge, so that only a small number of labeled samples are sufficient for the target domain ranking model to achieve the same confidence. Hence, to reduce the cost for new verticals, how to adapt the auxiliary ranking models to the new target domain and make full use of their domain-specific features, turns into a pivotal problem for building effective domain-specific ranking models.

Ranking adaptation is closely related to classifier adaptation, which has shown its effectiveness for many learning problems [2], [7], [8], [25], [18], [30], [32]. However, to the best of our knowledge, there are no prior works on the adaptation for the ranking problem. Besides the general difficulties faced by the classifier adaptation, such as covariate shift (or namely sample selection bias) [25], [32] and concept drifting [18], ranking adaptation is comparatively more challenging. Unlike classifier adaptation, which mainly deals with binary targets, ranking adaptation desires to adapt the model which is used to predict the rankings for a collection of documents. Though the documents are normally labeled with several relevance levels, which seems to be able to be handled by multi-class classification or regression, it is still difficult to directly use classifier adaption for ranking. The reason lies in two-fold: (1) in ranking, the mainly concerned is about the preference of two documents or the ranking of a collection of documents, which is difficult to be modeled by classification or regression; (2) the relevance levels between different domains are sometimes different and need to be aligned.

In this paper, we focus on the adaptation of ranking models, instead of utilizing the labeled data from auxiliary domains directly, which may be inaccessible due to privacy issue or data missing. Moreover, Model adaptation is more desirable than data adaptation, because the learning complexity is now only correlated to the size of the target domain training set, which should be much smaller than the size of auxiliary dataset. In this paper, we're going to investigate three problems of ranking model adaptation:

- whether we can adapt ranking models learned for the existing broad-based search or some verticals, to a new domain, so that the amount of labeled data in the target domain is reduced while the performance requirement is still guaranteed;
- how to adapt the ranking model effectively and efficiently;
- how to utilize domain-specific features to further boost the model adaptation.

The first problem is solved by the proposed *ranking adaptability* measure, which quantitatively estimates whether an existing ranking model can be adapted to the new domain, and predicts the potential performance for the adaptation. We address the second problem from the regularization framework and a ranking adaptation SVM algorithm is proposed. Our algorithm is a black-box ranking model adaptation, which needs only the predictions from the existing ranking model, rather than the internal representation of the model itself or the data from the auxiliary domains. With the black-box adaptation property, we achieved not only the flexibility but also the efficiency. To resolve the third problem, we assume that documents similar in their domain-specific feature space should have consistent rankings, e.g., images that are similar in their visual feature space should be ranked into similar positions and vice versa. We implement this idea by constraining the margin and slack variables of RA-SVM adaptively, so that similar documents is assigned with less ranking loss if they are ranked in a wrong order.

The rest of the paper is organized as follows. In Section 2, we formally present and analyze the proposed ranking adaptation algorithm. Section 3 explores the ranking adaptability. We discuss and formulate the ranking adaptation with the utilization of domain-specific feature in Section 4. The experimental results are shown and discussed in Section 5. Section 6 analyzes the efficiency problem of the proposed method. We remind some related works in Section 7. Section 8 concludes the paper.

2 RANKING ADAPTATION

We define the ranking adaptation problem formally as follows: for the target domain, a query set $Q = \{q_1, q_2, \ldots, q_M\}$ and a document set $D = \{d_1, d_2, \ldots, d_N\}$ are given. For each query $q_i \in Q$, a list of documents $d_i = \{d_{i1}, d_{i2}, \ldots, d_{i,n(q_i)}\}$ are returned and labeled with the relevance degrees $\mathbf{y}_i = \{y_{i1}, y_{i2}, \ldots, y_{i,n(q_i)}\}$ by human annotators. The relevance degree is usually a real value, i.e., $y_{ij} \in \mathbb{R}$, so that different returned documents can be compared for sorting an ordered list. For each query document pair $< q_i, d_{ij} >$, an s-dimensional query dependent feature vector $\phi(q_i, d_{ij}) \in \mathbb{R}^s$ is extracted, e.g., the term frequency of the query keyword q_i in the title, body, URL of the document d_{ij}. Some other hyperlink based static rank information is also considered, such as Pagerank [21], HITS [17] and so on. $n(q_i)$ denotes the number of returned documents for query q_i. The target of learning to rank is to estimate a ranking function $f \in \mathbb{R}^s \rightarrow \mathbb{R}$ so that the documents d can be ranked for a given query q according to the value of the prediction $f(\phi(q, d))$.

In the setting of the proposed ranking adaptation, both the number of queries m and the number of the returned documents $n(q_i)$ in the training set are assumed to be small. They are insufficient to learn an effective ranking model for the target domain. However, an auxiliary ranking model f^a, which is well trained in another domain over the labeled data Q^a and D^a, is available. It is assumed that the auxiliary ranking model f^a contains a lot of prior knowledge to rank documents, so it can be used to act as the base model to be adapted to the new domain. Few training samples can be sufficient to adapt the ranking model since the prior knowledge is available.

Before the introduction of our proposed ranking adaptation algorithm, it's important to review the formulation of Ranking Support Vector Machines (Ranking SVM), which is one of the most effective learning to rank algorithms, and is here employed as the basis of our proposed algorithm.

2.1 Ranking SVM

Similar to the conventional Support Vector Machines (SVM) for the classification problem [27], the motivation of Ranking SVM is to discover a one dimensional linear subspace, where the points can be ordered into the optimal ranking list under some criteria. Thus, the ranking function takes the form of the linear model $f(\phi(q,d)) = \mathbf{w}^T\phi(q,d)$, where the bias parameter is ignored, because the final ranking list sorted by the prediction f is invariant to the bias. The optimization problem for Ranking SVM is defined as follows:

$$\begin{aligned}
&\min_{f,\xi_{ijk}} \ \frac{1}{2}||f||^2 + C\sum\nolimits_{i,j,k}\xi_{ijk}\\
&\text{s.t.} \ \ f(\phi(q_i,d_{ij})) - f(\phi(q_i,d_{ik})) \geq 1-\xi_{ijk}\\
&\qquad \xi_{ijk} \geq 0,\\
&\text{for} \ \ \forall i \in \{1,2,\ldots,M\},\\
&\qquad \forall j \forall k \in \{1,2,\ldots,n(q_i)\} \text{ with } y_{ij} > y_{ik}, \qquad (1)
\end{aligned}$$

where C is the trade-off parameter for balancing the large-margin regularization $||f||^2$ and the loss term $\sum_{i,j,k}\xi_{ijk}$.

Because f is a linear model , we can derive that $f(\phi(q_i,d_{ij}))-f(\phi(q_i,d_{ik})) = f(\phi(q_i,d_{ij})-\phi(q_i,d_{ik}))$, with $\phi(q_i,d_{ij})-\phi(q_i,d_{ik})$ denoting the difference of the feature vectors between the document pair d_{ij} and d_{ik}. If we further introduce the binary label $\text{sign}(y_{ij}-y_{ik})$ for each pair of documents d_{ij} and d_{ik}, the above Ranking SVM problem can be viewed as a standard SVM for classifying document pairs into positive or negative, i.e., whether the document d_{ij} should be ranked above d_{ik} or not.

Since the number of labeled samples for the new domain is small, if we train the model using only the samples in the new domain, it will suffer from the insufficient training sample problem, which is ill-posed and the solution may be easily overfitting to the labeled samples with low generalization ability. Moreover, the current SVM solver requires super-quadratic computational cost for the training [22], as a consequence, it is quite time-consuming and nearly infeasible to train models using the training data from both the auxiliary domain and the target domain. This problem is more severe for the ranking SVM since the training are based on pairs and so the problem size is quadratic to the sample size.

In the following, we will develop an algorithm to adapt the auxiliary model using the few training samples labeled in the new domain. By model adaption, both the effectiveness of the result ranking model and the efficiency of the training process are achieved.

2.2 Ranking Adaptation SVM

It can be assumed that, if the auxiliary domain and the target domain are related, their respective ranking functions f^a and f should have similar shapes in the function space $\mathbb{R}^s \to \mathbb{R}$. Under such an assumption, f^a actually provides a prior knowledge for the distribution of f in its parameter space. The conventional regularization framework, such as L_p-norm regularization, manifold regularization [1] designed for SVM [27], regularized neural network [11] and so on, shows that the solution of an ill-posed problem can be approximated from variational principle, which contains both the data and the prior assumption [11]. Consequently, we can adapt the regularization framework which utilizes the f^a as the prior information, so that the ill-posed problem in the target domain, where only few query document pairs are labeled, can be solved elegantly. By modeling our assumption into the regularization term, the learning problem of Ranking Adaptation SVM (RA-SVM) can be formulated as:

$$\begin{aligned}
&\min_{f,\xi_{ijk}} \ \frac{1-\delta}{2}||f||^2 + \frac{\delta}{2}||f-f^a||^2 + C\sum\nolimits_{i,j,k}\xi_{ijk}\\
&\text{s.t.} \ \ f(\phi(q_i,d_{ij})) - f(\phi(q_i,d_{ik})) \geq 1-\xi_{ijk}\\
&\qquad \xi_{ijk} \geq 0,\\
&\text{for} \ \ \forall i \in \{1,2,\ldots,M\},\\
&\qquad \forall j \forall k \in \{1,2,\ldots,n(q_i)\} \text{ with } y_{ij} > y_{ik}. \qquad (2)
\end{aligned}$$

The objective function (2) consists of the adaptation regularization term $||f-f^a||^2$, which minimizes the distance between the target ranking function and the auxiliary one in the function space or the parameter space, to make them close; the large-margin regularization $||f||^2$; and the loss term $\sum_{i,j,k}\xi_{ijk}$. The parameter $\delta \in [0,1]$ is a trade-off term to balance the contributions of large-margin regularization $||f||^2$ which makes the learned model numerically stable, and adaptation regularization $||f-f^a||^2$ which makes the learned model similar to the auxiliary one. When $\delta = 0$, Problem (2) degrades to the conventional Ranking SVM (1), in other words, RA-SVM is equivalent to directly learning Ranking SVM over the target domain, without the adaptation of f^a. The parameter C is the same as in Ranking SVM, for balancing the contributions between the loss function and the regularization terms. It can be observed that when $C = 0$ and $\delta = 1$, Eq. (2) actually discards the labeled samples in the target domain, and directly output a ranking function with $f = f^a$. This is sometimes desirable, since if the labeled samples in the target domain are unavailable or unusable, f^a is believed to be better than random guess for ranking the documents in the target domain, as long as the auxiliary domain and the target domain are related.

2.3 Optimization Methods

To optimize Problem (2), we briefly denote $\mathbf{x}_{ijk} = \phi(q_i, d_{ij}) - \phi(q_i, d_{ik})$ and introduce the Lagrange multipliers to integrate the constraints of (2) into the objective function, which results in the primal problem:

$$\begin{aligned} L_P = & \frac{1-\delta}{2}||f||^2 + \frac{\delta}{2}||f - f^a||^2 + C\sum_{i,j,k}\xi_{ijk} \\ & - \sum_{i,j,k}\mu_{ijk}\xi_{ijk} - \sum_{i,j,k}\alpha_{ijk}(f(\mathbf{x}_{ijk}) - 1 + \xi_{ijk}). \quad (3) \end{aligned}$$

Taking the derivatives of L_P w.r.t. f, and setting it to zero, we can obtain the solution as:

$$f(\mathbf{x}) = \delta f^a(\mathbf{x}) + \sum\nolimits_{i,j,k}\alpha_{ijk}\mathbf{x}_{ijk}^T\mathbf{x}. \quad (4)$$

Denoting $\Delta f(\mathbf{x}) = \sum_{i,j,k}\alpha_{ijk}\mathbf{x}_{ijk}^T\mathbf{x}$, which can be viewed as the part of support vectors learned from the target domain, we can derive from (4) that the final ranking function f, which we would like to achieve for the target domain, is a linear combination between the auxiliary function f^a and the target part Δf, and the parameter δ controls the contribution of f^a.

In addition to (4), the optimal solution of problem (2) should satisfy the Karush-Kuhn-Tucker (KKT) conditions, which are composed of:

$$\begin{aligned} & \alpha_{ijk}(f(\mathbf{x}_{ijk}) - 1 + \xi_{ijk}) = 0 \\ & \alpha_{ijk} \geq 0 \\ & f(\mathbf{x}_{ijk}) - 1 + \xi_{ijk} \geq 0 \\ & \mu_{ijk}\xi_{ijk} = 0 \\ & \mu_{ijk} \geq 0 \\ & \xi_{ijk} \geq 0 \\ & C - \alpha_{ijk} - \mu_{ijk} = 0. \quad (5) \end{aligned}$$

Substituting (4) and (5) back into (3), we can derive the dual problem formulation as:

$$\begin{aligned} \max_{\alpha_{ijk}} \quad & -\frac{1}{2}\sum\nolimits_{i,j,k}\sum\nolimits_{l,m,n}\alpha_{ijk}\alpha_{lmn}\mathbf{x}_{ijk}^T\mathbf{x}_{lmn} \\ & + \sum\nolimits_{i,j,k}(1 - \delta f^a(\mathbf{x}_{ijk}))\alpha_{ijk} \\ \text{s.t.} \quad & 0 \leq \alpha_{ijk} \leq C, \\ \text{for} \quad & \forall i \in \{1, 2, \ldots, M\}, \\ & \forall j \forall k \in \{1, 2, \ldots, n(q_i)\} \text{ with } y_{ij} > y_{ik}. \quad (6) \end{aligned}$$

The above problem is a standard Quadratic Programming (QP) problem, and any standard QP solvers, e.g. SMO [22], can be utilized to solve it.

Notice that we can firstly train a ranking model in the target domain, and then linearly combine it with the auxiliary model, which shows the same solution as shown in (4). However, because of the scarcity of labeled data, purely training a ranking model in the target domain will lead the model overfitting to the training samples, and cannot effectively combine with auxiliary model for a satisfactory performance. RA-SVM differs in that it learns a joint ranking model by considering f^a during the learning phase, as shown in (6). The overfitting problem can be overcomed by utilizing the prior information from the auxiliary model.

2.4 Discussions

The proposed RA-SVM has several advantages, which makes our algorithm highly applicable and flexible when applied to the practical applications. We'll give more discussions of the characteristics of RA-SVM in the following.

- Model adaptation: the proposed RA-SVM does not need the labeled training samples from the auxiliary domain, but only its ranking model f^a . Such a method is more advantageous than data based adaptation, because the training data from auxiliary domain may be missing or unavailable, for the copyright protection or privacy issue, but the ranking model is comparatively easier to obtain and access.
- Black-box adaptation: The internal representation of the model f^a is not needed, but only the prediction of the auxiliary model to the training samples from the target domain $f^a(x)$ is used. It brings a lot of flexibilities in some situations where even the auxiliary model itself may be unavailable. Also, in some cases, we would like to use a more advanced algorithm for learning the ranking model for the new target domain, than the one used in the old auxiliary domain, or in other cases, the algorithm used in the old domain is even unknown to us. By the black-box adaptation property, we don't need to have any idea on the model used in the auxiliary domain, but only the model predictions are required.
- Reducing the labeling cost: by adapting the auxiliary ranking model to the target domain, only a small number of samples need to be labeled, while the insufficient training sample problem will be addressed by the regularization term $||f - f^a||^2$, which actually assigns a prior to the target ranking model. In Section 5, we'll experimentally demonstrate that the proposed RA-SVM model is quite robust and well-performed, even with only a small number of training samples labeled.
- Reducing the computational cost: It has been shown that our ranking adaptation algorithm can be transformed into a Quadratic Programming (QP) problem, with the learning complexity directly related to the number of labeled samples in the target domain. Platt [22] proposed the sequential minimal optimization (SMO) algorithm which can decompose a large QP problem into a series of subproblems and optimize them iteratively. The time complexity is around $O(n^{2.3})$ for general kernels [22]. In [15], cutting-plane method is adopted to solve SVM for the linear kernel, which further reduces the time complexity to $O(n)$. Here, n is the number of labeled document pairs in the target domain. According to the above discussion, the size of the labeled training set is greatly reduced. Thus, n is substantially

small, which in turn leads to the efficiency of our algorithm.

2.5 Adaptation from Multiple Domains

Our proposed RA-SVM can be extended to a more general setting, where ranking models learned from multiple domains are provided. Denoting the set of auxiliary ranking functions by $\mathcal{F} = \{f_1^a, f_2^a, \ldots, f_R^a\}$, the RA-SVM for the multiple domain adaptation setting can be formulated as:

$$\begin{aligned}\min_{f,\xi_{ijk}} \ & \frac{1-\delta}{2}||f||^2 + \frac{\delta}{2}\sum\nolimits_{r=1}^{R} \theta_r ||f - f_r^a||^2 + C\sum\nolimits_{i,j,k}\xi_{ijk} \\ \text{s.t.} \ & f(\phi(q_i, d_{ij})) - f(\phi(q_i, d_{ik})) \geq 1 - \xi_{ijk} \\ & \xi_{ijk} \geq 0, \\ \text{for} \ & \forall i \in \{1, 2, \ldots, M\}, \\ & \forall j \forall k \in \{1, 2, \ldots, n(q_i)\} \text{ with } y_{ij} > y_{ik}, \end{aligned} \tag{7}$$

where θ_r is the parameter that controls the contribution of ranking model f_r^a obtained from the rth auxiliary domain, and we can further constrain $\sum_{r=1}^{R} \theta_r = 1$ without any loss of generality. Similar to the analysis in the one domain adaptation setting, the solution for problem (7) is:

$$f(\mathbf{x}) = \delta \sum\nolimits_{r=1}^{R} \theta_r f_r^a(\mathbf{x}) + \sum\nolimits_{i,j,k} \alpha_{ijk} \mathbf{x}_{ijk}^T \mathbf{x}. \tag{8}$$

If we represent $f^a(\mathbf{x}) = \sum_{r=1}^{R} \theta_r f_r^a(\mathbf{x})$, the auxiliary ranking functions can be regarded as a single one, which lies in the convex hull of $\mathcal{F}$. Thus, similar to the discussion of (4), the final ranking model is a linear combination of two parts, i.e., the convex combination of ranking functions from auxiliary domains f^a, and the part from the target set $\Delta f = \sum_{i,j,k} \alpha_{ijk} \mathbf{x}_{ijk}^T \mathbf{x}$, with the parameter θ_r controlling the contribution of the auxiliary model f_r^a, while δ controlling all the contributions from $\mathcal{F}$ globally.

3 EXPLORE RANKING ADAPTABILITY

Though the ranking adaptation can mostly provide benefits for learning a new model, it can be argued that when the data from auxiliary and target domains share little common knowledge, the auxiliary ranking model can provide little help or even negative influence, to the ranking of the documents in the target domain. Consequently, it is imperative to develop a measure for quantitatively estimating the adaptability of the auxiliary model to the target domain. However, given a ranking model and a dataset collected for a particular target domain, it's nontrivial to measure their correlations directly, because neither the distribution of the ranking model nor that of the labeled samples in the target domain is trivial to be estimated. Thus, we present some analysis on the properties of the auxiliary model, based on which the definition of the proposed *ranking adaptability* is presented.

3.1 Auxiliary Model Analysis

We analyze the effects of auxiliary models through the loss constraint in the formulation of our RA-SVM. By substituting (4) into (2), we can obtain that:

$$\begin{aligned} & \delta f^a(\mathbf{x}_{ijk}) + \Delta f(\mathbf{x}_{ijk}) \geq 1 - \xi_{ijk} \\ & \text{with } y_{ij} > y_{ik}, \text{ and } \xi_{ijk} \geq 0, \end{aligned} \tag{9}$$

where, as defined before, $\mathbf{x}_{ijk} = \phi(q_i, d_{ij}) - \phi(q_i, d_{ik})$ and $\Delta f = \sum_{i,j,k} \alpha_{ijk} \mathbf{x}_{ijk}^T \mathbf{x}$. Thus, in order to minimize the ranking error ξ_{ijk} for the document pair d_{ij} and d_{ik}, we hope to get a large prediction value on the left-hand side of the first inequation in (9). For a given auxiliary ranking function f^a, a comparatively large $f^a(\mathbf{x}_{ijk})$ suggests that f^a can correctly judge the order for the document pair d_{ij} and d_{ik}, and vice versa. According to the constraints (9), if f^a is capable of predicting the order of the documents correctly, we can correspondingly lower the contribution of the part of the ranking function learned in the target domain, i.e., Δf. At an extreme case, if f^a is able to predict all pairs of documents correctly in the target domain, namely it can give perfect ranking lists for all the labeled queries, we may derive that f^a should be applied to the target domain directly with only small modifications, i.e., satisfying the "large margin" requirement in the target domain. On the other hand, if f^a cannot give a desirable ordering of the document pairs, we have to rely on Δf more to eliminate the side effects of f^a, so that the ranking error over labeled samples is reduced. Consequently, the performance of f^a over the labeled document pairs in the target domain can greatly boost the learning of RA-SVM for the ranking adaptation.

3.2 Ranking Adaptability

Based on the above analysis of f^a, we develop the *ranking adaptability* measurement by investigating the correlation between two ranking lists of a labeled query in the target domain, i.e., the one predicted by f^a and the ground-truth one labeled by human judges. Intuitively, if the two ranking lists have high positive correlation, the auxiliary ranking model f^a is coincided with the distribution of the corresponding labeled data, therefore we can believe that it possesses high ranking adaptability towards the target domain, and vice versa. This is because the labeled queries are actually randomly sampled from the target domain for the model adaptation, and can reflect the distribution of the data in the target domain.

Here, we adopt the well-known Kendall's τ [16] to calculate the correlation between the two ranking lists, and based on which, the proposed *ranking adaptability* is defined. For a given query q_i, we denote the rank list predicted by the ranking function f by $\mathbf{y}_i^* = \{y_{i1}^*, y_{i2}^*, \ldots, y_{i,n(q_i)}^*\}$, and define a pair of documents (d_{ij}, y_{ij}) and (d_{ik}, y_{ik}) by concordant if $(y_{ij} - y_{ik})(y_{ij}^* - y_{ik}^*) > 0$, and discordant if $(y_{ij} - y_{ik})(y_{ij}^* - y_{ik}^*) < 0$. Furthermore, we represent the number of concordant pairs as $N_i^c = \sum_{j=1}^{n(q_i)} \sum_{k=j+1}^{n(q_i)} \mathrm{sign}[(y^*_{ij} - y^*_{ik})(y_{ij} - y_{ik}) > 0]$ and the number of discordant pairs as $N_i^c = \sum_{j=1}^{n(q_i)} \sum_{k=j+1}^{n(q_i)} \mathrm{sign}[(y^*_{ij} - y^*_{ik})(y_{ij} - y_{ik}) < 0]$, where $\mathrm{sign}(x)$ is the sign function with $\mathrm{sign}(x) = 1$ if $x > 0$, $\mathrm{sign}(x) = -1$ if $x < 0$, and $\mathrm{sign}(x) = 0$ otherwise. Suppose q_i has neither tied prediction (i.e., for $\forall j \forall k y^*_{ij} \neq y^*_{ik}$) nor tied relevance (i.e., for $\forall j \forall k y_{ij} \neq y_{ik}$), then $N_i^c + N_i^d = n(q_i)(n(q_i) - 1)/2$. In such a situation where no tie exists, we can define the rank correlation for function f over the query q_i based on the Kendall's τ as:

$$\tau_i(f) = \frac{N_i^c - N_i^d}{n(q_i)(n(q_i) - 1)/2} \ . \tag{10}$$

However, ties are quite common for general applications, especially in the Web search scenario. When ties do exist, we can handled them by adding 0.5 to N_i^c and 0.5 to N_i^d if $y_{ij} = y_{ik}$, and ignore the pairs with $y^*_{ij} = y^*_{ik}$. Therefore, a more general definition for the correlation is:

$$\tau_i(f) = \frac{N_i^c - N_i^d}{N_i^c + N_i^d} \ . \tag{11}$$

Thus, it is obvious $\tau_i(f) \in [-1, 1]$, where $\tau_i(f) = 1$ corresponds to the positive correlation between $\mathbf{y}^*_i$ and $\mathbf{y}_i$, $\tau_i(f) = -1$ equals to the negative correlation, and $\tau_i(f) = 0$ means uncorrelated.

Based on (11), the proposed *ranking adaptability* of the auxiliary ranking model f^a for the target domain, is defined as the mean of the Kendall's τ correlation between the predicted rank list and the ground truth rank list, for all the labeled queries in the target domain, namely,

$$\mathcal{A}(f^a) = \frac{1}{M} \sum\nolimits_{i=1}^{M} \tau_i(f^a) \ . \tag{12}$$

The proposed *ranking adaptability* measures the correlation between the ranking lists sorted by auxiliary model prediction and the ground truth, which in turn gives us an indication of whether the auxiliary ranking model can be adapted to the target domain, and how much assistance it can provide. Based on the *ranking adaptability*, we can perform automatic model selection for determining which auxiliary models will be adapted. The effectiveness of the proposed *ranking adaptability* measurement will be demonstrated experimentally in Section 6.5.

4 RANKING ADAPTATION WITH DOMAIN-SPECIFIC FEATURE

Conventionally, data from different domains are also characterized by some domain-specific features, e.g., when we adopt the ranking model learned from the Web page search domain to the image search domain, the image content can provide additional information to facilitate the text based ranking model adaptation. In this section, we discuss how to utilize these domain-specific features, which are usually difficult to translate to textual representations directly, to further boost the performance of the proposed RA-SVM.

The basic idea of our method is to assume that documents with similar domain-specific features should be assigned with similar ranking predictions. We name the above assumption as the consistency assumption, which implies that a robust textual ranking function should perform relevance prediction that is consistent to the domain-specific features.

To implement the consistency assumption, we are inspired by the work [26] and recall that for RA-SVM in (2), the ranking loss is directly correlated to the slack variable, which stands for the ranking loss for pairwise documents, and is nonzero as long as the ranking function predicts a wrong order for the two documents. In addition, as a large margin machine, the ranking loss of RA-SVM is also correlated to the large margin specified to the learned ranker. Therefore, to incorporate the consistency constraint, we rescale the ranking loss based on two strategies, namely margin rescaling and slack rescaling. The rescaling degree is controlled by the similarity between the documents in the domain-specific feature space, so that similar documents bring about less ranking loss if they are ranked in a wrong order. We discuss the detailed formulations of margin rescaling and slack rescaling as follows.

4.1 Margin Rescaling

Margin rescaling denotes that we rescale the margin violation adaptively according to their similarities in the domain-specific feature space. Specifically, the Ranking Adaptation SVM with Margin Rescaling (RA-SVM-MR) can be defined as the following optimization problem:

$$\begin{aligned} \min_{f, \xi_{ijk}} \ & \frac{1-\delta}{2}||f||^2 + \frac{\delta}{2}||f - f^a||^2 + C \sum\nolimits_{i,j,k} \xi_{ijk} \\ \text{s.t.} \ \ & f(\phi(q_i, d_{ij})) - f(\phi(q_i, d_{ik})) \geq 1 - \xi_{ijk} - \sigma_{ijk} \\ & \xi_{ijk} \geq 0, \\ \text{for} \ \ & \forall i \in \{1, 2, \ldots, M\}, \\ & \forall j \forall k \in \{1, 2, \ldots, n(q_i)\} \text{ with } y_{ij} > y_{ik}. \end{aligned} \tag{13}$$

where $0 \leq \sigma_{ijk} \leq 1$ denotes the similarities between document d_{ij} and d_{ik} returned for query q_i in the domain-specific feature space. The above optimization problem differs from (2) in the first linear inequality constraint, which varies the margin adaptively. Compared to a pair of dissimilar documents, similar ones with larger σ_{ijk} will result in a smaller margin to satisfy the linear constraint, which produces less ranking loss in terms of a smaller slack variable ξ_{ijk} if the document pair d_{ij} and d_{ik} (namely d_{ijk}) is ranked in a wrong order by the function f. The dual problem of (13) is:

$$\begin{aligned} \max_{\alpha_{ijk}} \ & -\frac{1}{2} \sum\nolimits_{i,j,k} \sum\nolimits_{l,m,n} \alpha_{ijk} \alpha_{lmn} \mathbf{x}_{ijk}^T \mathbf{x}_{lmn} \\ & \quad + \sum\nolimits_{i,j,k} (1 - \delta f^a(\mathbf{x}_{ijk}) - \sigma_{ijk}) \alpha_{ijk} \\ \text{s.t.} \ \ & 0 \leq \alpha_{ijk} \leq C, \\ \text{for} \ \ & \forall i \in \{1, 2, \ldots, M\}, \\ & \forall j \forall k \in \{1, 2, \ldots, n(q_i)\} \text{ with } y_{ij} > y_{ik}, \end{aligned} \tag{14}$$

and the desired ranking function takes the same form as (2), as shown in (4).

4.2 Slack Rescaling

Compared to margin rescaling, slack rescaling is intended to rescale the slack variables according to their similarities in the domain specific feature space. We define the corresponding Ranking Adaptation SVM with Slack Rescaling (RA-SVM-SR) as the following optimization problem:

$$\begin{aligned}
&\min_{f,\xi_{ijk}} \ \frac{1-\delta}{2}||f||^2 + \frac{\delta}{2}||f - f^a||^2 + C\sum\nolimits_{i,j,k}\xi_{ijk} \\
&\text{s.t.} \ \ f(\phi(q_i,d_{ij})) - f(\phi(q_i,d_{ik})) \geq 1 - \frac{\xi_{ijk}}{1-\sigma_{ijk}} \\
&\qquad \xi_{ijk} \geq 0, \\
&\text{for} \ \ \forall i \in \{1,2,\dots,M\}, \\
&\qquad \forall j \forall k \in \{1,2,\dots,n(q_i)\} \text{ with } y_{ij} > y_{ik}. \qquad (15)
\end{aligned}$$

Different from margin rescaling, slack rescaling varies the amplitude of slack variables adaptively. If a pair of documents are dissimilar in the domain-specific feature space, by dividing $1-\sigma_{ijk}$, the slack variables that control the ranking loss of the two documents are correspondingly amplified in order to satisfy the first linear equality, and vice versa. The dual problem of (15) is:

$$\begin{aligned}
&\max_{\alpha_{ijk}} \ -\frac{1}{2}\sum\nolimits_{i,j,k}\sum\nolimits_{l,m,n}\alpha_{ijk}\alpha_{lmn}\mathbf{x}_{ijk}^T\mathbf{x}_{lmn} \\
&\qquad + \sum\nolimits_{i,j,k}(1-\delta f^a(\mathbf{x}_{ijk}))\alpha_{ijk} \\
&\text{s.t.} \ \ 0 \leq \alpha_{ijk} \leq (1-\sigma_{ijk})C, \\
&\text{for} \ \ \forall i \in \{1,2,\dots,M\}, \\
&\qquad \forall j \forall k \in \{1,2,\dots,n(q_i)\} \text{ with } y_{ij} > y_{ik}, \qquad (16)
\end{aligned}$$

and the solution of the ranking function, as for RA-SVM-MR, is same to (2), as shown in (4). It can be observed from the dual format of (16) that, slack rescaling is equivalent to rescaling the trade-off parameters C for each pairwise documents, based on their similarities.

The optimizations of RA-SVM-MR (14) and RA-SVM-SR (16) have the exactly same time complexity as for the RA-SVM (6), i.e., $O(n^{2.3})$ by using SMO algorithm and $O(n)$ by means of cutting plane algorithm for the linear kernel. Therefore, although domain-specific features are incorporated for the model adaptation, we didn't bring about any additional efficiency problems.

5 EXPERIMENTS

In this section, we perform several experiments under two different settings, to demonstrate the effectiveness of the proposed RA-SVM based algorithms and the *ranking adaptability* measurement.

5.1 Datasets and Evaluation Measure

We firstly conduct the experiments over the Letor benchmark dataset [20], and adapt the ranking model learned from TD2003 dataset to the ranking of TD2004 dataset. Letor TD2003 and TD2004 datasets are gathered from the topic distillation task of TREC 2003 and TREC 2004, with 50 queries for TD2003 and 75 ones for TD2004. The documents are collected by crawling from the .gov domain. For each query, about 1000 associated documents are returned, and labeled with a binary judgment, i.e., relevant or irrelevant. The features of TD2003 and TD2004 include the low-level features such as term frequency, inverse document frequency, and document length, as well as high-level features such as BM25, LMIR, PageRank, and HITS, for totally 44 dimensional features. However, Letor is a comparatively small dataset, and each document is only labeled with a binary relevance degree, which cannot reflect the practical Web search scenarios with multiple relevance degrees. Also, there are no domain-specific features for the target domain data, where we cannot demonstrate the effectiveness of the proposed ranking adaptation with domain-specific feature algorithms.

Therefore, to give a more thorough analysis of our proposed RA-SVM based methods and to demonstrate the effectiveness of domain specific features, we collect more large scale datasets from a commercial internet search engine. Two datasets are separately gathered from different domains, i.e. the Web page search and the image search engines. There are totally 2625 queries for the Web page search domain, and 1491 queries for image. At most 50 documents for each query are crawled and labeled, and therefore we obtain 122815 query-document pairs for Web page search and 71246 query-image pairs, resulting in 46.79 documents returned for each Web page search query and 47.78 images for each image query on average. We take the visual features of images as domain-specific features for the image search domain, and try to utilize these features to boost the performance of adaptation from Web page search to image search. Note that the dataset of image search is a subset of the one used in our conference version [10]. This is because we have to crawl the images from the Web to extract their visual features as domain-specific features, in order to test the performance of RA-SVM-MR and RA-SVM-SR. However, the URLs of some images are currently invalid and we cannot download the corresponding images. Therefore, we have to select a subset of image queries from the original dataset, where each query has at least 30 images successfully downloaded.

Query-dependent textual features are extracted for all query-document pairs based on different document sources, e.g., the anchor text, the URL, the document title and the body. Some other features are also incorporated, such as the static rank, the junk page and so on. Totally 354 dimensional textual features are extracted. Each query-document pair is labeled with a relevance

TABLE 1
Ranking Adaptation Dataset Information.

Dataset	#Query	#Query-Document	Relevance Degree	Feature Dimension
TD2003	50	49171	2	44
TD2004	75	74170	2	44
Web Page Search	2625	122815	5	354
Image Search	1491	71246	3	354

degree by the human judges. For the visual features of each image, we extract several visual features that are widely used in computer vision, i.e., Attention Guided Color Signature, Color Spatialet, Wavelet, SIFT, Multi-Layer Rotation Invariant EOH (MRI-EOH), Histogram of Gradient (HoG), and Facial Feature. The distances computed on each feature are linearly combined as the ultimate distance between the images [6]. We transform the distance between documents d_{ij} and d_{ik} into their similarities by exponential function, i.e. $\sigma_{ijk} = \exp^{-\beta r_{ijk}}$, where r_{ijk} is the distance computed based on visual features and $\beta > 0$ is a parameter to control the transformation scale.

The range of the relevance degree for Web page search is from 0 (i.e. "bad") to 4 (i.e. "perfect match") with totally five degrees, while for image, they are labeled 0 (i.e. "irrelevant"), 1 (i.e. "relevant") and 2 (i.e. "highly relevant") with three degrees. The documents labeled as "detrimental" are removed from both datasets. The detailed information of each dataset, including the number of queries, query document pairs, relevance degrees, and feature dimensions are shown in Table 1.

The performance evaluations of the ranking results are based on two measures, namely, mean average precision (MAP) and normalized discounted cumulative gain at different rank truncation levels (NDCG@n) [13], for a comprehensive analysis of the performances of different algorithms. MAP, one of the most frequently used measure to evaluate the average performance of a ranking algorithm, denotes the mean of the average precision (AP), where the AP computes the area under precision/recall curve with non-interpolated manner and prefers relevant samples with higher rank. Since AP is evaluated only for binary judgement, we define relevance level 0 as non-relevant and all the other relevance degrees as relevant for all the datasets. To measure the ranking performance for multiple degree relevance, NDCG is proposed as a cumulative, multilevel measure of ranking quality, which is usually truncated at a particular rank level [13]. For a given query q_i, the NDCG is calculated as:

$$\mathcal{N}_i = N_i \sum_{j=1}^{L} \frac{2^{r(j)} - 1}{\log(1+j)} \ , \tag{17}$$

where $r(j)$ is the relevance degree of the jth document, N_i is the normalization coefficient to make the perfect order list with $\mathcal{N}_i = 1$, and L is the ranking truncation level at which NDCG is computed. In this paper, we evaluate NDCG@n by setting the truncation level n as at most 20.

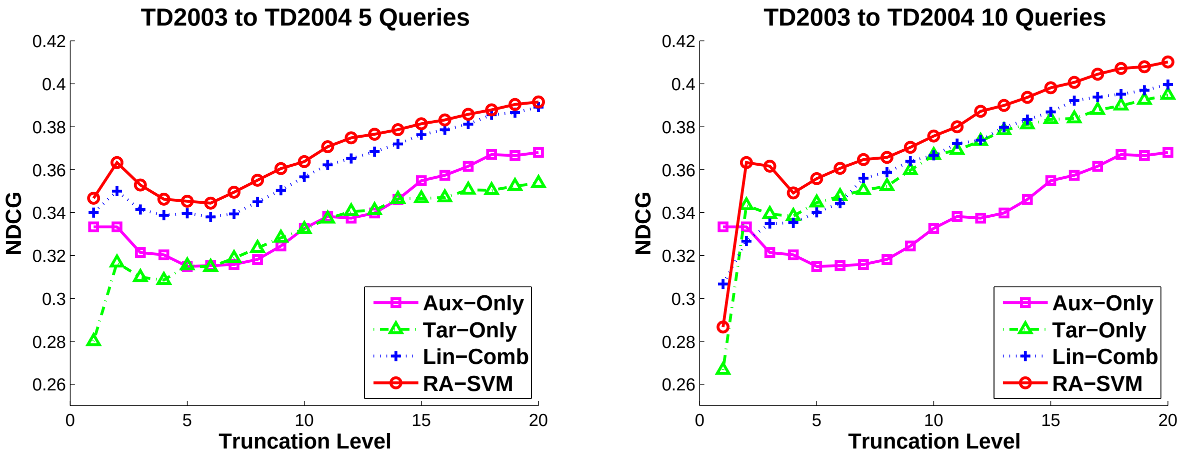
TABLE 2
Ranking Adaptation Experiment Settings.

Auxiliary Domain	Train	Validate	Test
TD2003	30	-	20
Web Page Search	500	-	2125
Target Domain	Adapt Pool	Validate	Test
TD2004	30	5	30
Image search	500	10	981

5.2 Experiment Settings

We build the auxiliary ranking model by training Ranking SVM with different parameters over some labeled queries randomly sampled from the auxiliary domain, namely Letor TD2003 dataset and Web page search dataset, and then select the models that are best performed over the remained data in the auxiliary domain as the auxiliary models for adaptation. In the adaptation target domain, where the performance of different algorithms are reported, we randomly select several queries as the pool of the labeled data as candidate data for the adaptation, several queries as the validation set to determine the parameters of different algorithms, and the remaining queries as the test set for the performance evaluation. We vary the size of adaptation set gradually by selecting different number of queries from the pool of the labeled data, so that we can see the influence of different numbers of labeled samples to the performance of the adapted ranking model. For each size of adaptation set, we generate five different adaptation sets by randomly sampling from the labeled adaptation data pool created before. We apply each algorithm over each generated set separately, resulting into five different ranking models. The final performance reported in this paper is the average results of the five ranking models, validated over the identical validation set and evaluated over the identical test set. The details of two experiment settings are summarized in Table 2.

In order to illustrate the effectiveness of our proposed RA-SVM, RA-SVM-MR and RA-SVM-SR, we compare their performance to the results of several baseline methods, i.e., (1) the Ranking SVM models learned purely from the adaptation sets of the target domain without adaptation (Tar-Only); (2) the results of applying the auxiliary ranking model directly to the test set (Aux-Only);



(a)

(b)

Fig. 1. The NDCG of TD2003 to TD2004 adaptation, with (a) 5 and (b) 10 labeled queries in TD2004 respectively.

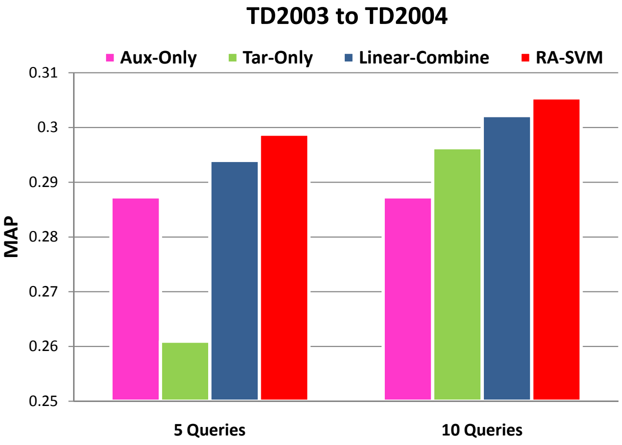


Fig. 2. The MAP of TD2003 to TD2004 adaptation results, with 5 and 10 labeled queries in TD2004 respectively.

(3) the performance of linearly combining the Aux-Only model and the Tar-Only model (Lin-Comb); (4) the performance of slack rescaling by the relative relevance of the pairwise documents (RA-SVM-GR). The intention of comparing with Lin-Comb is to show that RA-SVM based methods can utilize the auxiliary model more effectively to learn a robust target ranking model, than directly combining the two independent models. We also report the performance of RA-SVM-GR, to demonstrate that the rescaling of domain-specific features are more meaningful compared to rescaling from the groundtruth relevance difference.

5.3 Adapt from TD2003 to TD2004

The NDCG and MAP of utilizing 5 and 10 labeled queries, which are randomly selected from adaptation data pool, are shown in Fig. 1 and Fig. 2 respectively. For 5 labeled queries, the Aux-Only model performs better than the Tar-Only one, whose performance is suffered from the insufficient labeled queries in the target domain. Meanwhile, it can be observed that the Lin-Comb and RA-SVM outperform the above two methods significantly, since both the Aux-Only model and target domain's data information are utilized. In addition, RA-SVM shows the best performance over all, especially for the first several truncation levels. For 10 labeled adaptation queries, as the number of labeled queries increased, the Tar-Only comes to outperform Aux-Only model. However, Lin-Comb almost performs equally to the Tar-Only results. We argue that directly combining Aux-Only and Tar-Only cannot effectively utilize the information of both auxiliary model and target domain, because the two models are trained independently and combined intuitively. For Lin-Comb, the Tar-Only model may have overfitted to limited queries in the adaptation set, while the Aux-Only model cannot discover the domain-specific knowledge, and their combination is consequently limited for inducing a robust ranking model. Finally, RA-SVM, by leveraging the auxiliary model to build the target domain's model jointly in one step, leads to the best performance.

5.4 Adapt from Web Page Search to Image Search

To further demonstrate the effectiveness of the proposed RA-SVM algorithm, we perform several experiments by adapting the ranking model trained from Web page search domain to the image search domain. The performances with 5, 10, 20, 30, 40 and 50 labeled queries are shown in Fig. 3 and Fig. 4 respectively. It can be observed that, at each adaptation size, RA-SVM consistently outperforms the baseline methods significantly at all truncation levels, while RA-SVM-MR and RA-SVM-SR further improve the performance. In addition, we can derive that for the 5, 10 and 20 queries settings, the performance of Aux-Only model is much better than Tar-only one, because of the insufficient labeled sample problem. On the contrary, for the 40 and 50 queries settings, Tar-only model performs better than Aux-Only

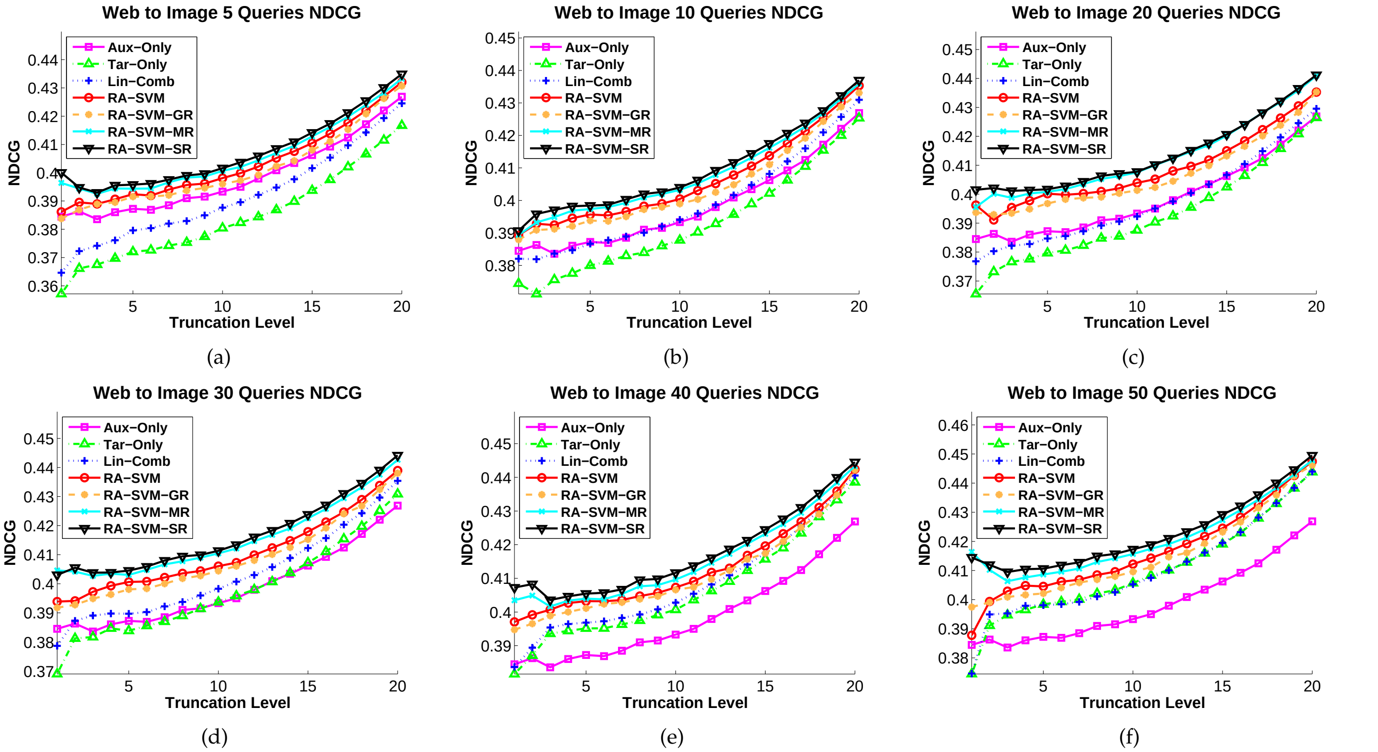


Fig. 3. The NDCG results of Web page search to image search adaptation, with (a) 5, (b) 10, (c) 20, (d) 30, (e) 40 and (f) 50 labeled queries of image search dataset utilized respectively.

one, due to the larger size of training set and the limited performance of the auxiliary model caused by the domain differences. For Lin-Comb, as we discussed for the TD2003 to TD2004 adaptation, simply combine two ranking models, cannot stably obtain desirable results. RA-SVM, by leveraging both the auxiliary model and the few labeled data in the target domain jointly, shows the best performance for both measurements. Furthermore, with the help of domain-specific features to adaptively control the loss of different pairwise documents, RA-SVM-MR and RA-SVM-SR can further improve the adaptation performance significantly, especially for the top several truncation levels. It can be observed that RA-SVM-SR is slightly more robust than RA-SVM-MR, especially for larger adaptation sets. We also find that RA-SVM-GR performs no better, or even worse than RA-SVM. This is because that conventional pairwise based learning to rank algorithms (e.g., Ranking SVM, RankBoost, RankNet) implicitly takes the relative relevance in to consideration, by creating pairwise documents that encouraging highly relevant documents to be ranked higher than both moderately relevant and irrelevant documents, while controlling the moderately relevant documents to be ranked higher than irrelevant ones but lower than highly relevant ones. RA-SVM-GR brings negative effects for the reweighting of each pair of documents.

In order to show the relationship between the performance and the size of adaptation set, we vary the adaptation set from 1 to 50 queries gradually and test the performance of each algorithm. The results are shown in Fig. 5. We observe that for a small number of labeled queries in the target set, the ranking model of Tar-Only cannot give satisfying results, due to the insufficient training sample problem. However, by adapting the auxiliary model, the performance of RA-SVM steadily outperforms all the three baseline methods. On the other hand, the auxiliary model itself can only give a poor performance over the test set of the target domain, due to the domain differences between Web pages and image documents, as mentioned before. However, with the help of only several labeled query document pairs in the target domain, the ranking performance can be substantially improved. In addition, the domain-specific features in the target domain can improve the performance of RA-SVM a lot, even for a small number of labeled queries. We observe that the performance of RA-SVM-SR and RA-SVM-MR over 10 labeled queries setting is comparable to the performance of Lin-Comb in the 50 labeled queries setting. Finally, we observe that the performance improvement does not degrade much as the increment of the size of adaptation set, which proves that our algorithms are quite effective for maximally utilizing the auxiliary ranking model, as well as the domain-specific features, even for a comparatively large number of labeled queries available in the target domain.

In order to prove that the performance improvements of our RA-SVM based methods are significant than the baseline methods, we conduct the statistical t-test

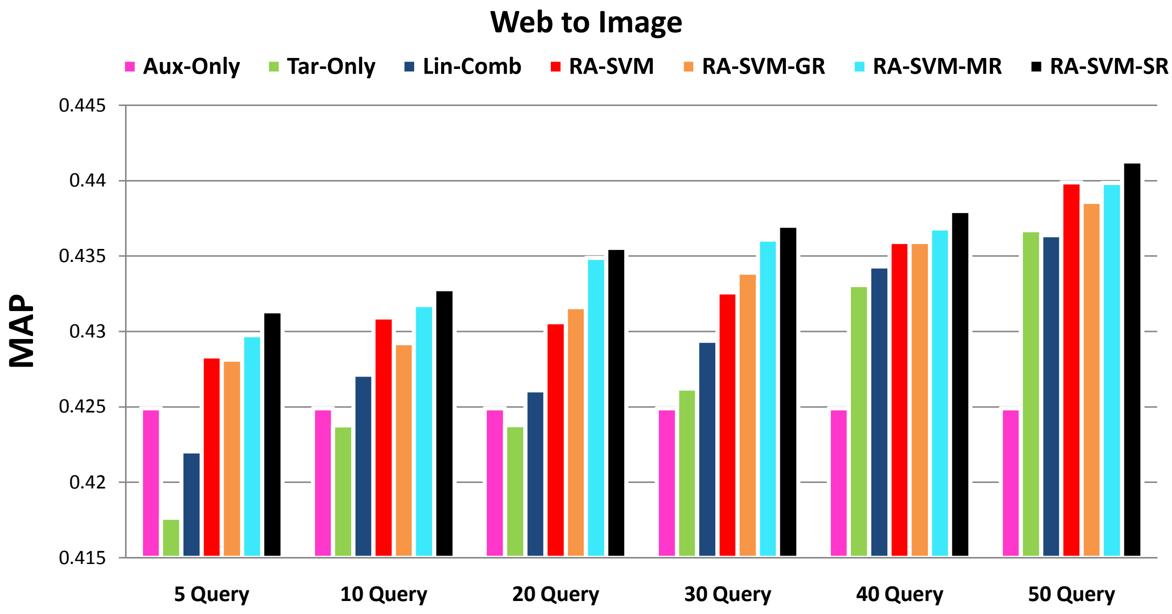
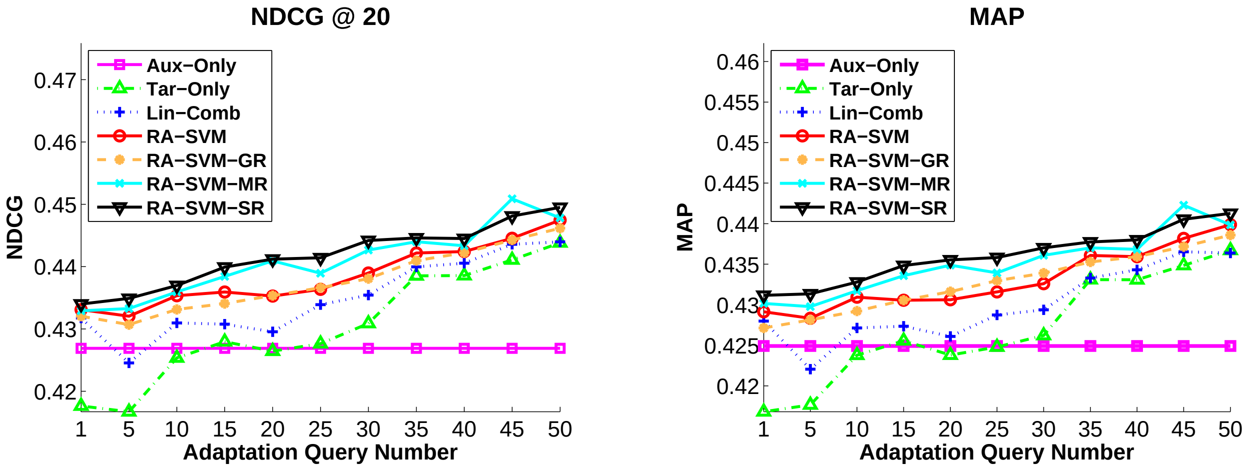


Fig. 4. The MAP of Web page search to image search adaptation results, with 5, 10, 20, 30, 40, and 50 labeled queries of image search dataset utilized respectively.



(a) (b)

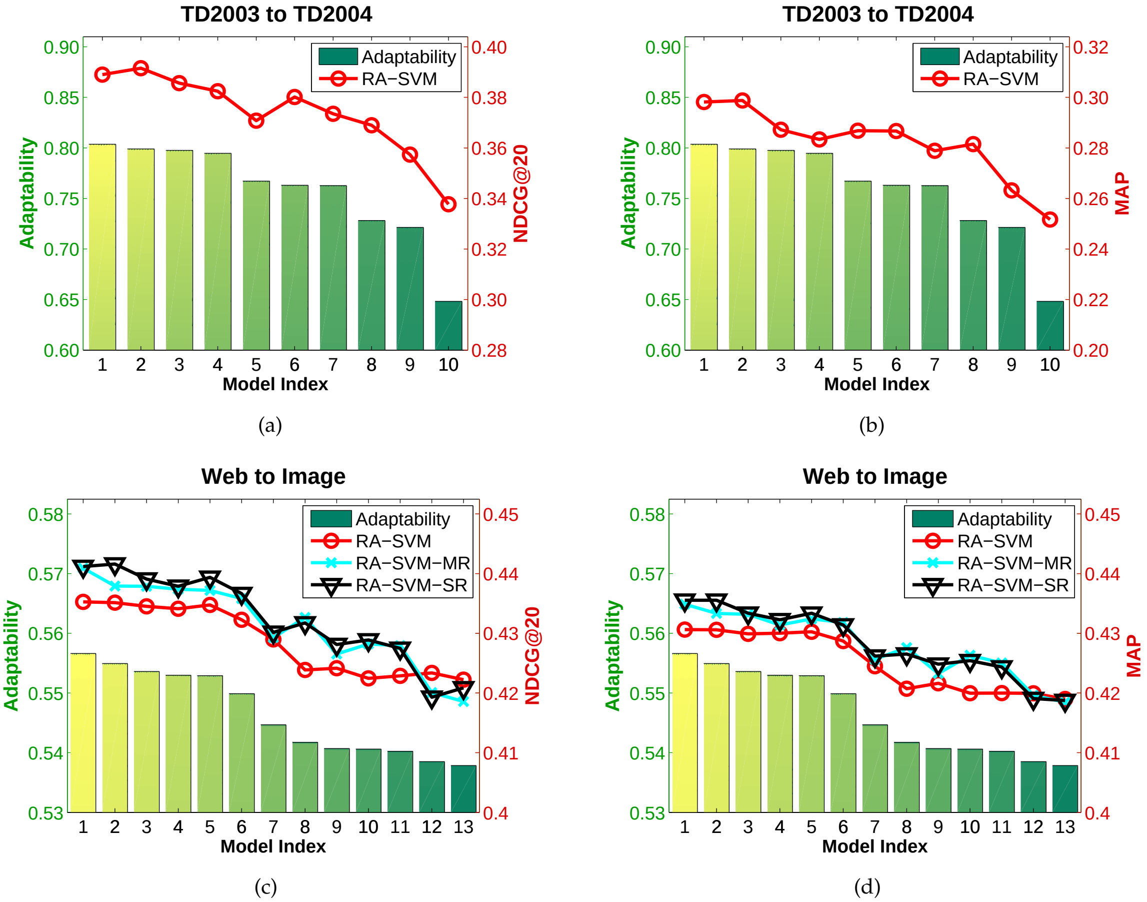
Fig. 5. (a) NDCG@20 and (b) MAP vs. the size of adaptation set.

TABLE 3
The p value of the significance test for NDCG@20.

Query	10 Query	20 Query	30 Query	40 Query	50 Query
Lin-Comb vs RA-SVM	2.08e-3	6.76e-4	1.18e-3	1.13e-2	6.71e-3
RA-SVM vs RA-SVM-MR	1.12e-2	5.23e-4	4.17e-3	1.51e-2	1.57e-1
RA-SVM vs RA-SVM-SR	9.12e-3	2.83e-4	9.75e-4	6.08e-3	2.39e-3

TABLE 4
The p value of the significance test for MAP.

Query	10 Query	20 Query	30 Query	40 Query	50 Query
Lin-Comb vs RA-SVM	2.49e-3	1.89e-4	9.36e-4	1.44e-2	2.44e-3
RA-SVM vs RA-SVM-MR	1.10e-2	9.14e-4	1.13e-3	1.80e-2	2.87e-1
RA-SVM vs RA-SVM-SR	5.36e-3	3.60e-4	1.24e-4	1.15e-2	1.33e-3



(a)

(b)

(c)

(d)

Fig. 6. The adaptability vs. performance (NDCG@20 and MAP) for different auxiliary models. Each column corresponds to an auxiliary ranking model, and the height of the vertical bar denotes the predicted ranking adaptability. Each line corresponds to the performance of a specific ranking adaptation method using different auxiliary models. (a) TD2003 to TD2004 NDCG@20; (b) TD2003 to TD2004 MAP; (c) Web page to image NDCG@20; (d) Web page to image MAP.

between the results of the compared methods and report the p values of the significance test. Due to the space limitation, we only present the results of the NDCG@20 and MAP over 10, 20, 30, 40, 50 adaptation queries, and consider three settings, i.e., Lin-Comb vs RA-SVM, RA-SVM vs RA-SVM-MR, and RA-SVM vs RA-SVM-SR. The results are shown in Table 3 and Table 4. We can derive that except for RA-SVM vs RA-SVM-MR over 50 adaptation queries, all the other improvements are significant. As we analyzed before, RA-SVM significantly outperforms Lin-Comb, while RA-SVM-SR is comparatively more stable for utilizing the domain-specific features to boost the ranking model adaptation.

5.5 Ranking Adaptability

In this subsection, we perform several experiments to prove the effectiveness of the proposed *ranking adaptability*, and the applicability for auxiliary model selection.

Firstly, ten ranking models are learned over the training set of the auxiliary domain, i.e., the TD2003 and the Web page search domain respectively, with the same training set used for the experiments in section 6.2 and Table 2. We still adopt Ranking SVM to learn the ranking models as the candidate auxiliary models. The ten models are learned by varying the parameter C of Ranking SVM. Then, we apply each model respectively to the target domain for adaptation experiments, using our RA-SVM, RA-SVM-MR and RA-SVM-SR. Finally, according to (12), the *ranking adaptabilities* of all the models over the adaptation sets from image search domain are calculated. The performances and the *ranking adaptabilities* to be reported are averaged over the five random splits of adaptation sets. To be concise, we only show the results on the adaptation set composed of five labeled queries for TD2004 dataset and twenty labeled queries for image search dataset, while the results of other sizes of adaptation sets are similar.

The evaluation measures of NDCG@20 and MAP are plotted together with the *ranking adaptabilities* in Fig. 6. We can conclude that, for both TD2003 to TD2004 and Web page search to image search, the performances of the adapted ranking models are approximately coincided with the proposed *ranking adaptability*, i.e., the

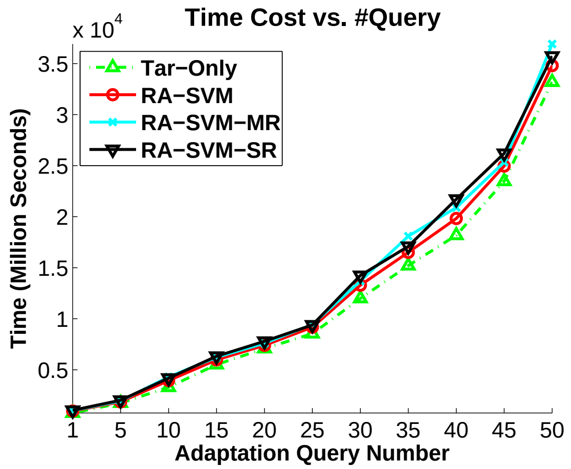


Fig. 7. The time cost of each method under different sizes of adaptation sets.

ranking models with high adaptability will achieve a better performance in the target domain if being adapted, and vice versa. As discussed before, such a property can be utilized for automatic model selection of the auxiliary ranking models for adaptation, given some labeled queries in the target domain.

6 EFFICIENCY ANALYSIS

To analyze the efficiency of the proposed RA-SVM based methods, we compare the learning time of different methods by varying the adaptation query number in the Web page search to image search setting. Because the Aux-Only will not spend time learning a new ranking model, and Lin-Comb needs the Tar-Only to be trained beforehand and then linearly combines it with Aux-Only, we only compare the time cost of Tar-Only, RA-SVM, RA-SVM-MR and RA-SVM-SR. The time reported for each method is the summation of the five random splits. All the experiments are done under the same hardware setting, i.e., the Intel Xeon E5440 core with 8GB memory.

The results are shown in Fig. 7, and we can observe that for small number of adaptation query number, the time costs of different algorithms are very similar. For large adaptation sets, even though Tar-Only is slightly better than RA-SVM based methods, the variance of different methods is not significant. We can conclude that the proposed RA-SVM is quite efficient compared with direct training a model in the target domain. Also, the results of RA-SVM-MR and RA-SVM-SR show that the incorporation of domain-specific features doesn't brings further learning complexity. These conclusions are consistent with our theoretical analysis mentioned in the previous sections.

7 RELATED WORK

We present some works that closely relate to the concept of ranking model adaptation here. To create a ranking model that can rank the documents according to their relevance to a given query, various types of models have been proposed, some of which have even been successfully applied to Web search engines. Classical BM25 [24] and Language Models for Information Retrieval (LMIR) [19], [23] work quite stable for the broad-based search with few parameters needing adjusted. However, with the development of statistical learning methods, and more labeled data with complicated features being available, sophisticated ranking models become more desirable for achieving better ranking performance. Recently, a dozen of learning to rank algorithms based on machine learning techniques have been proposed. Some of them transform the ranking problem into a pairwise classification problem, which takes a pair of documents as a sample, with the binary label taken as the sign of the relevance difference between the two documents, e.g., Ranking SVM [12], [14], RankBoost [9], RankNet [4] and etc. Some other methods including ListNet [5], SVM^{Map} [31], AdaRank [28], PermuRank [29], LambdaRank [3] and etc., focus on the structure of ranking list and the direct optimization of the objective evaluation measures such as Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG). In this paper, instead of designing a new learning algorithm, we focus on the adaptation of ranking models across different domains based on the existing learning to rank algorithms.

A lot of domain adaptation methods have also been proposed to adapt auxiliary data or classifiers to a new domain. Daume and Marcu proposed a statistical formulation in terms of a mixture model to address the domain distribution differences between training and testing sets [8]. A boosting framework was also presented for the similar problem [7]. For natural language processing, Blitzer and et al. [2] introduced a structural correspondence learning method which can mine the correspondences of features from different domains. For multimedia application, Yang and et al. [30] proposed Adaptive SVM algorithm for the cross-domain video concept detection problem. However, these works are mainly designed for classification problems, while we focused on the domain adaptation problem for ranking in this paper.

8 CONCLUSION

As various vertical search engines emerge and the amount of verticals increases dramatically, a global ranking model, which is trained over a dataset sourced from multiple domains, cannot give a sound performance for each specific domain with special topicalities, document formats and domain-specific features. Building one model for each vertical domain is both laborious for labeling the data and time-consuming for learning the model. In this paper, we propose the ranking model adaptation, to adapt the well learned models from the broad-based search or any other auxiliary domains to a new target domain. By model adaptation, only a small number of samples need to be labeled, and the computational cost for the training process is greatly reduced.

Based on the regularization framework, the Ranking Adaptation SVM (RA-SVM) algorithm is proposed, which performs adaptation in a black-box way, i.e., only the relevance predication of the auxiliary ranking models is needed for the adaptation. Based on RA-SVM, two variations called RA-SVM margin rescaling (RA-SVM-MR) and RA-SVM slack rescaling (RA-SVM-SR) are proposed to utilize the domain specific features to further facilitate the adaptation, by assuming that similar documents should have consistent rankings, and constraining the margin and loss of RA-SVM adaptively according to their similarities in the domain-specific feature space. Furthermore, we propose *ranking adaptability*, to quantitatively measure whether an auxiliary model can be adapted to a specific target domain and how much assistance it can provide.

We performed several experiments over Letor benchmark datasets and two large scale datasets obtained from a commercial internet search engine, and adapted the ranking models learned from TD2003 to TD2004 dataset, as well as from Web page search to image search domain. Based on the results, we can derive the following conclusions:

- The proposed RA-SVM can better utilize both the auxiliary models and target domain labeled queries to learn a more robust ranking model for the target domain data.
- The utilization of domain-specific features can steadily further boost the model adaptation, and RA-SVM-SR is comparatively more robust than RA-SVM-MR.
- The adaptability measurement is consistent to the utility of the auxiliary model, and it can be deemed as an effective criterion for the auxiliary model selection.
- The proposed RA-SVM is as efficient as directly learning a model in a target domain, while the incorporation of domain-specific features doesn't brings much learning complexity for algorithms RA-SVM-SR and RA-SVM-MR.

Acknowledgments

This work is supported by 973 Program (2011CB302400), NSFC (60975014) and NSFB (4102024).

References

[1] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7(Nov):2399–2434, 2006.

[2] J. Blitzer, R. Mcdonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *EMNLP '06: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 120–128. Association for Computational Linguistics, July 2006.

[3] C. J. C. Burges, R. Ragno, and Q. V. Le. Learning to rank with nonsmooth cost functions. In *NIPS '06: Advances in Neural Information Processing Systems*, pages 193–200. MIT Press, Cambridge, MA, 2006.

[4] C. J. C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *ICML '05: Proceedings of the 22th International Conference on Machine Learning*, 2005.

[5] Z. Cao and T. yan Liu. Learning to rank: From pairwise approach to listwise approach. In *ICML '07: Proceedings of the 24th International Conference on Machine Learning*, pages 129–136, 2007.

[6] J. Cui, F. Wen, and X. Tang. Real time google and live image search re-ranking. In *ACM Multimedia*, pages 729–732, 2008.

[7] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu. Boosting for transfer learning. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 193–200, 2007.

[8] H. Daume, III and D. Marcu. Domain adaptation for statistical classifiers. *Journal of Artficial Intelligence Research*, 26:101–126, 2006.

[9] Y. Freund, R. Iyer, R. E. Schapire, Y. Singer, and G. Dietterich. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003.

[10] B. Geng, L. Yang, C. Xu, and X.-S. Hua. Ranking model adaptation for domain-specific search. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*, pages 197–206, 2009.

[11] F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7:219–269, 1995.

[12] R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. In *Advances in Large Margin Classifiers*, 2000.

[13] K. Järvelin and J. Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 41–48, 2000.

[14] T. Joachims. Optimizing search engines using clickthrough data. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, 2002.

[15] T. Joachims. Training linear svms in linear time. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 217–226, 2006.

[16] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, June 1938.

[17] J. M. Kleinberg, S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. The web as a graph: Measurements, models and methods. In *Proceedings of the International Conference on Combinatorics and Computing, no. 1627 in Lecture Notes in Computer Science*, pages 1–18, 1999.

[18] R. Klinkenberg and T. Joachims. Detecting concept drift with support vector machines. In *ICML '00: In Proceedings of the Seventeenth International Conference on Machine Learning*, pages 487–494. Morgan Kaufmann, 2000.

[19] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 111–119, 2001.

[20] T.-Y. Liu, T. Qin, J. Xu, W. Xiong, and H. Li. Benchmark dataset for research on learning to rank for information retrieval. In *SIGIR '07: Proceedings of SIGIR 2007 Workshop on Learning to Rank for Information Retrieval (LR4IR 2007)*, 2007.

[21] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. In *Technical report, Stanford University*, 1998.

[22] J. C. Platt. Fast training of support vector machines using sequential minimal optimization. *Advances in kernel methods: support vector learning*, pages 185–208, 1999.

[23] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281, 1998.

[24] S. Robertson and D. A. Hull. The trec-9 filtering track final report. In *Proceedings of the 9th Text Retrieval Conference*, pages 25–40, 2000.

[25] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90:227–244(18), 2000.

[26] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output vari-

ables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.

[27] V. N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

[28] J. Xu and H. Li. Adarank: a boosting algorithm for information retrieval. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 391–398, 2007.

evaluation measures in learning to rank. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 107–114, 2008.

[30] J. Yang, R. Yan, and A. G. Hauptmann. Cross-domain video concept detection using adaptive svms. In MULTIMEDIA '07: *Proceedings of the 15th international conference on Multimedia*, pages 188–197, 2007.

[31] Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 271–278, 2007.

[32] B. Zadrozny. Learning and evaluating classifiers under sample selection bias. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 114, 2004.

Bo Geng received the B.Sci. degree from the Fudan University in 2007. Currently, he is a Ph.D candidate with the Key Laboratory of Machine Perception (Ministry of Education) in the Peking University. Previously, he was a research intern with the Internet Media group in the Microsoft Research Asia, and a research assistant with the Department of Computing in the Hong Kong Polytechnic University. His research interests lie primarily in machine learning, multimedia search, information retrieval and computer vision. He is a student member of IEEE.

Linjun Yang received the B.S. and M.S. degrees from East China Normal University and Fudan University, Shanghai, China, in 2001 and 2006, respectively. Since 2001, he has been with Microsoft Research Asia, Beijing, China, where he is currently an Associate Researcher in the Media Computing Group. His current interests are in the broad areas of multimedia information retrieval, with focus on multimedia search ranking and large-scale Web multimedia mining. He has authored or coauthored more than 30 publications in these areas and has more than 10 filed patents or pending applications. He is a member of ACM and IEEE.

Chao Xu received the B.E. degree from Tsinghua University in 1988, the M.S. degree from University of Science and Technology of China in 1991 and the Ph.D degree from Institute of Electronics, Chinese Academy of Sciences in 1997. Between 1991 and 1994 he was employed as an assistant professor by University of Science and Technology of China. Since 1997 Dr. Xu has been with School of EECS at Peking University where he is currently a Professor. His research interests are in image and video coding, processing and understanding. He has authored or co-authored more than 80 publications and 5 patents in these fields.

Xian-Sheng Hua received the B.S. and Ph.D. degrees from Peking University, Beijing, China, in 1996 and 2001, respectively, both in applied mathematics. Since 2001, he has been with Microsoft Research Asia, Beijing, where he is currently a Lead Researcher with the media

are in the areas of video content analysis, multimedia search, management, authoring, sharing, mining, advertising and mobile multimedia computing. He has authored or co-authored more than 180 publications in these areas and has more than 50 filed patents or pending applications.

He is now an adjunct professor of University of Science and Technology of China, and serves as an Associate Editor of IEEE Transactions on Multimedia, Associate Editor of ACM Transactions on Intelligent Systems and Technology, Editorial Board Member of Advances in Multimedia and Multimedia Tools and Applications, and editor of Scholarpedia (Multimedia Category). He also has successfully served or is serving as vice program chair (VCIP 2005), workshop organizers (workshops of ICME 2009/2010, ICDM 2009 and ACM Multimedia 2010), demonstration chairs, tutorial chairs, special session chairs, senior TPC members (ACM Multimedia and ACM KDD) and PC members of a large number of international conferences.

Dr. Hua won the Best Paper Award and Best Demonstration Award in ACM Multimedia 2007, Best Poster Award in 2008 IEEE International Workshop on Multimedia Signal Processing, Best Student Paper Award in ACM Conference on Information and Knowledge Management 2009, and Best Paper Award in International Conference on MultiMedia Modeling 2010. He also won 2008 MIT Technology Review TR35 Young Innovator Award for his outstanding contributions to video search, and named as one of the "Business Elites of People under 40 to Watch" by Global Entrepreneur. Dr. Hua has invented and shipped more than six technologies into Microsoft mainstream products. He is a member of IEEE, a member of VSPC TC and MAS TC in IEEE CAS society, the chair of the Interest Group on Visual Analysis and Content Management in Multimedia Communication TC of IEEE Communications Society, and a senior member of ACM.