JMLR: Workshop and Conference Proceedings 36:214–228, 2014 BIGMINE 2014

Scalable Heterogeneous Transfer Ranking

Mohammad Taha Bahadori MOHAMMAB@USC.EDU
University of Southern California

Yi Chang YICHANG@YAHOO-INC.COM
Yahoo Labs

Bo Long BO.LONG@GMAIL.COM
LinkedIn Inc.

Yan Liu YANLIU.CS@USC.EDU
University of Southern California

Editors: Wei Fan, Albert Bifet, Qiang Yang and Philip Yu

Abstract

In this paper, we propose to study the problem of *heterogeneous transfer ranking*, a transfer learning problem with heterogeneous features in order to utilize the rich large-scale labeled data in popular languages to help the ranking task in less popular languages. We develop a large-margin algorithm, namely LM-HTR, to solve the problem by mapping the input features in both the source domain and target domain into a shared latent space and simultaneously minimizing the feature reconstruction loss and prediction loss. We analyze the theoretical bound of the prediction loss and develop fast algorithms via stochastic gradient descent so that our model can be scalable to large-scale applications. Experiment results on two application datasets demonstrate the advantages of our algorithms over other state-of-the-art methods.

1. Introduction

In many applications of big data, we sometimes are confronted with a dilemma of small data. That is, even though we have collected a large amount of observatory or experimental data in a domain, the reality is that for specific tasks of interest, the actual amount of data we are able to utilize may be very limited. For example, in information retrieval task, many labeled examples are available for search query in English and Chinese while very few examples are available for Thai or other less spoken languages. Similar in health care, many labeled examples are available for personalized treatment for pneumonia, but only one or two examples are available for TaySachs disease, a rare genetic disease. Transfer learning, the process of leveraging the information from other domains (i.e., source domain) to train a better model for the target domain, is a natural solution for the small data dilemma (Thrun and Pratt, 1998; Pan and Yang, 2010).

Transfer learning for classification settings has been demonstrated useful in many applications, such as natural language processing (Daume, 2007; Blitzer et al., 2007), image classification (Raina et al., 2007), intrusion detection (He et al., 2009) and so on. However, learning to rank has many applications (Joachims, 2002; Burges et al., 2005; Xu and Li, 2007; Cortes et al., 2007; Zheng et al., 2007; Guiver and Snelson, 2008; Cao et al., 2007), while

transfer ranking is less studied. For example, (Chen et al., 2008) proposes the TransRank algorithm that selects k-best queries from source domain as training examples and utilizes feature augmentations to train a new classifier via rank SVM, (Bai et al., 2010) adapted ranking models that are trained with multi-grade labeled training data to the target domain using the domain-specific pair-wise preference data, and (Gao et al., 2010) estimated the importance of examples in the source domain to the target domain and transformed the importance into pairwise weight of document pairs for ranking algorithms.

Most existing work on transfer ranking have been focused on the learning scenario where the source domain and the target domain share the same feature space. In practical applications, we are usually confronted with even more challenging problems. For example, in vertical search, we can obtain labeled ranking results for popular languages (e.g. English and Spanish), whereas we are interested in building a search ranking algorithm for other regions in different languages (e.g. Vietnamese and Thai) with very few or even no labeled examples. This learning problem, where the source domain and the target domain have heterogeneous feature space, is also known as heterogeneous transfer learning (Yang et al., 2009). In this paper, we will study this problem under the ranking applications and refer it to as *heterogeneous transfer ranking*. It is a more challenging task because the ranking model usually requires a significant larger number of labeled examples due to its model complexity while the heterogeneous feature space makes it difficult to transfer the information from the source domain to the target domain effectively. In addition, the target domain usually comes with very few or even no labeled examples, which exacerbate the aforementioned issue.

In this paper, we propose a scalable large-margin based model for heterogeneous transfer ranking (LM-HTR), which assumes a shared prediction function in a latent space and learns the *domain-specific* mapping functions and the prediction function by minimizing the reconstruction error and prediction error in one unified function. Different from most transfer learning algorithms, which have the same mapping functions across domains, our model relaxes this assumption by introducing a domain-specific mapping function and guide the search of the mapping functions by minimizing the ranking loss. As a result, they are more flexible and do not significantly rely on the assumption of strong similarities between the source and target domains. In particular, we provide theoretical analysis on the generalization bound of the ranking loss in the target domain. Since scalability is one of the most important features for practical applications, we develop a fast optimization algorithm based on stochastic gradient descent to solve the resulting optimization problem. We demonstrate the effectiveness of LM-HTR on both synthetic datasets and application datasets, even for those applications where the similarities between the source domain and target domain are relatively low.

The rest of the paper is organized as follows: after reviewing related work, we describe the proposed model LM-HTR followed by discussions on the generalization bound and the scalability of LM-HTR. In the experiment results we verify the superior performance of LM-HTR. Finally, we summarize the paper and provide hints on future work.

2. Related Work

Transfer learning has been extensively studied in the literature (Thrun and Pratt, 1998; Blitzer et al., 2007; Daume, 2007; Raina et al., 2007; Pan and Yang, 2010; He et al., 2009). Unlike the classification task, there have been very few algorithms developed for the transfer ranking task. For example, (Duh and Kirchhoff, 2008) proposed a method that aims to find patterns in the documents labeled for each query in the target domain, project the data in source domain into another space with the inferred patterns and learn the ranking function in the projected space. This approach is vulnerable to overfitting when there are very few documents labeled for each query in the target domain. (Bai et al., 2010) adapted a small amount of ranked target data to the decision tree inferred from the source domain. (Gao et al., 2010) describes a weighting method to give higher weights to the examples in the source domain that more similar to those in the target domain. This weighting approach may suffer from higher estimator variance, and relies heavily on the assumption that there exist highly similar examples in the source and target domain.

Our proposed model has several advantages over existing work: First, it is *heterogeneous* transfer learning in that it does not have the assumption that the source domain and the target domain have to share the same feature space. In other words, the dimensions of the features could be different or even the feature space could be different. This is extremely useful for cross-lingual applications. Several recent work have explored heterogeneous transfer learning for classification task (Yang et al., 2009; Zhu et al., 2011; Duan et al., 2012), but none of them studied the ranking problem. Second, it is *transductive* transfer learning in that it does not require any labeled example in the target domain. Practically this is very important since in many applications it could be even difficult to identify the labeling experts in the first place. Several promising methods have been developed for transductive transfer learning (Arnold et al., 2007; Quanz and Huan, 2009; Bahadori et al., 2011), but most of them are either ineffective or extremely slow. Third, our model learns the *domain-specific* mapping functions, which are more flexible and do not significantly rely on the assumption of strong similarities between the source and target domains. Theoretical analysis and empirical performance demonstrate these advantages.

3. Methodology

In this section, we first formally define the problem, then discuss an example to motivate why domain-specific mapping functions are preferred for heterogeneous transfer learning, and describe in detail our proposed model, and finally we present fast optimization algorithms so that our model can be scalable to large-scale datasets.

3.1. Problem Definition

Learning to rank has been extensively studied and several different approaches have been proposed to formulate the problem (Joachims, 2002; Burges et al., 2005; Xu and Li, 2007; Cortes et al., 2007; Zheng et al., 2007; Guiver and Snelson, 2008; Cao et al., 2007). Throughout this section, we use pairwise preference as an example to describe our model. Suppose we have a document collection as well as a set of queries, all of which are in language A (e.g. English). For each query j, we are given the ranking labels for a set of docu-

ments $\mathcal{S}_j^S$, $j = 1, \ldots, J^S$. For each query-document pair, we can construct a feature vector $\mathbf{z}_i \in \mathcal{X}_S, i = 1, \ldots, m$ consisting of features defined over the query, the document, and the query-document matching scores, and in total we can have m such pairs. The label $y_{ii'} \in \{1, -1\}$ specifies the relative position of query-document pair $\mathbf{z}_i$ with respect to another query-document pair $\mathbf{z}_{i'}$ in the ranking list (below (-1) or above (+1)) for all the query-document pairs corresponding to the same query. Given another set of queries and documents in language B (e.g. Thai), we can also have the query-document feature set $\{\mathbf{x}_i \in \mathcal{X}_T | i = 1, \ldots, n\}$. Our goal is to predict the pairwise preference label for the new set in language B by utilizing the labeled examples in language A. Different from existing work in transfer ranking, we do not require that the feature space of the source domain $\mathcal{X}_S$ is the same as that of the target domain $\mathcal{X}_T$. Practically, they could be totally different. In addition, we do not require to have any labeled examples in the target domain. Therefore our task is "heterogeneous" and "transductive" in nature.

3.2. Large-Margin Heterogeneous Transfer Ranking (LM-HTR)

We propose a large-margin approach that automatically learns the domain-specific mapping functions and ranking function in one unified framework in order to solve the heterogeneous transfer learning problem. Here we use the large-margin approach because it has been demonstrated effective in many existing work on learning to rank (Joachims, 2002; Burges et al., 2005). Notice that our model is general enough and other type of ranking algorithms can be easily applicable. Before delving into our model, we first review transductive Support Vector Machines (TSVM), which attempt to solve the *transductive* labeling problem defined as following (Vapnik, 1995):

$$\min_{\mathbf{w},\mathbf{b},\{y_i^*\}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i + C^*\sum_{i=m+1}^{m+n}\xi_i^* \tag{1}$$

subject to:

$$\forall i \in \{1, \cdots, m\} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0,$$
$$\forall i \in \{m+1, \cdots, m+n\} \quad y_i^*(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i^*, \quad \xi_i^* \geq 0,$$

where $\mathbf{w}$ is the classifier and b is the bias component. $(\mathbf{x}_i, y_i), i = 1, \ldots, m$ are the samples with their labels and $\mathbf{x}_i, i = m+1, \ldots, m+n$ comprise the unlabeled samples. $y_i^*, i = m+1, \ldots, m+n$ are the labels learned for the unlabeled samples; C, C^*, ξ_i and ξ_i^* are the costs and hinge losses for the labeled and unlabeled samples, respectively.

The formulation of the Large-Margin Heterogeneous Transfer learning Ranking (LM-HTR) is similar except that we need two additional terms for learning domain-specific mapping functions. More specifically, we have:

$$\min_{\substack{\mathbf{w},b,\mathbf{\Phi},\tilde{\mathbf{\Phi}},\\ \{\mathbf{a}_i\},\{\mathbf{e}_i\},\{y_{ii'}^*\}}} \frac{1}{2}\|\mathbf{w}\|^2 + C_1\sum_{i=1}^{m}\eta_i + C_2\sum_{i=1}^{n}\zeta_i + C_3\sum_{\substack{i,i'=1\\ i\neq i'}}^{m}\xi_{ii'} + C_4\sum_{\substack{i,i'=1\\ i\neq i'}}^{n}\xi_{ii'}^* \tag{2}$$

subject to:

$$\begin{aligned}
&\forall j\in\{1,\dots,J^S\},\ \forall i,i'\in\mathcal{S}_j^S && y_{ii'}\mathbf{w}^\top(\mathbf{a}_i-\mathbf{a}_{i'})\geq 1-\xi_{ii'},\quad \xi_{ii'}\geq 0,\\
&\forall j\in\{1,\dots,J^T\},\ \forall i,i'\in\mathcal{S}_j^T && y^*_{ii'}\mathbf{w}^\top(\mathbf{e}_i-\mathbf{e}_{i'})\geq 1-\xi^*_{ii'},\quad \xi^*_{ii'}\geq 0\\
&\forall i\in\{1,\cdots,m\} && \|\mathbf{z}_i-\mathbf{\Phi}\mathbf{a}_i\|_2^2+\beta\|\mathbf{\Phi}\|_F^2\leq\eta_i,\\
&\forall i\in\{1,\cdots,n\} && \left\|\mathbf{x}_i-\tilde{\mathbf{\Phi}}\mathbf{e}_i\right\|_2^2+\beta\left\|\tilde{\mathbf{\Phi}}\right\|_F^2\leq\zeta_i
\end{aligned}$$

The optimization problem in Eq. (2) jointly minimizes five loss terms: (i) a L_2 regularization term for restricting the complexity of the classifier in the latent space, (ii) two reconstruction loss terms η_i and ζ_i for both source and target samples and (iii) hinge losses $\xi_{ii'}$ and $\xi^*_{ii'}$ for large-margin ranking of pairs in the source and target domains. In this optimization problem, there are four types of variables that need to be optimized. $\mathbf{\Phi}\in\mathbb{R}^{d_S\times r}$ and $\tilde{\mathbf{\Phi}}\in\mathbb{R}^{d_T\times r}$ are basis vectors for the r dimensional hidden spaces underlying the source and target domain, respectively. $\{\mathbf{a}_i\}$ and $\{\mathbf{e}_i\}$ are the representations of the source and target samples in the latent space, respectively. The vector $\mathbf{w}$ is the ranking function in the latent spaces which points in the direction of the preferred documents. The binary values $y_{ii'}$ are the ±1 pairwise preference information in the source domain and $y^*_{ii'}$ are the predicted pairwise preference between two samples $\mathbf{e}_i$ and $\mathbf{e}_{i'}$ in the latent domain. β is the regularization parameter, ξ_{ij} and ξ^*_{ij} are hinge loss variables, and C_1 - C_4 are the cost parameters that control the reconstruction error and ranking error in the source and target domains.

In order to solve the optimization problem in eq (2), an iterative searching approach as discussed in (Bradley and Bagnell, 2009) can be applied. That is, in each iteration, we fix a group of variables and solve the resulting simpler subproblems:

1. Fixing $\{\mathbf{a}_i\},\{\mathbf{e}_i\}$, we have two independent subproblems. One is the following PCA-type problem,

$$\min_{\mathbf{\Phi}}\left\{\sum_{i=1}^m\|\mathbf{z}_i-\mathbf{\Phi}\mathbf{a}_i\|_2^2+\beta\|\mathbf{\Phi}\|_F^2\right\}\quad\text{and}\tag{3}$$

$$\min_{\tilde{\mathbf{\Phi}}}\left\{\sum_{i=1}^n\left\|\mathbf{x}_i-\tilde{\mathbf{\Phi}}\mathbf{e}_i\right\|_2^2+\beta\left\|\tilde{\mathbf{\Phi}}\right\|_F^2\right\},\tag{4}$$

and the other is TSVM-type problem as follows:

$$\min_{\mathbf{w},\{y^*_{ii'}\}}\quad\frac{1}{2}\|\mathbf{w}\|_2^2+C_3\sum_{\substack{i,i'=1\\ i\neq i'}}^m\xi_{ii'}+C_4\sum_{\substack{i,i'=1\\ i\neq i'}}^n\xi^*_{ii'}\tag{5}$$

subject to:

$$\begin{aligned}
&\forall(i,i')\in\mathcal{S}_j^S && y_{ii'}\mathbf{w}^\top(\mathbf{a}_i-\mathbf{a}_{i'})\geq 1-\xi_{ii'},\quad \xi_{ii'}\geq 0\\
&\forall(i,i')\in\mathcal{S}_j^T && y^*_{ii'}\mathbf{w}^\top(\mathbf{e}_i-\mathbf{e}_{i'})\geq 1-\xi^*_{ii'},\quad \xi^*_{ii'}\geq 0
\end{aligned}$$

2. Fixing $\{\mathbf{w}_j\}$, $\mathbf{b}$, $\mathbf{\Phi}$, $\tilde{\mathbf{\Phi}}$, and $\{y^*_{ij}\}$, we have two independent sub-problems, that is, for all queries in the source domain $j=1,\dots,J^S$, solve the following for all the documents listed for the query,

$$\min_{\{\mathbf{a}_i\}} C_1 \sum_{i \in \mathcal{S}_j^S} \|\mathbf{z}_i - \mathbf{\Phi}\mathbf{a}_i\|_2^2 + C_3 \sum_{(i,i') \in \mathcal{S}_j^S} \xi_{ii'} \tag{6}$$

subject to:

$$\forall (i,i') \in \mathcal{S}_j^S \; y_{ii'} \mathbf{w}^\top (\mathbf{a}_i - \mathbf{a}_{i'}) \geq 1 - \xi_{ii'}, \;\; \xi_{ii'} \geq 0 \tag{7}$$

Similarly, for all queries in the target domain $j = 1, \ldots, J^T$, we solve a similar problem to update $\{\mathbf{e}_i\}$.

We can see that solving the sub-problems could be very challenging. For example, the problem in eq (5), i.e., the TSVM ranking problem, is a mixed integer programming with solutions known to be slow and unstable (Collobert et al., 2006). In later sections, we present fast optimization algorithms so that it can be scalable to large-scale datasets.

4. Generalization Bounds

In order to obtain a deeper insight into the source of different types of losses in our algorithm, we present a theoretical analysis on the generalization error of our algorithm using the transductive Rademacher complexity bounds (Bartlett and Mendelson, 2003; Shawe-Taylor and Cristianini, 2004). The transductive Rademacher complexity is the generalization of the inductive Rademacher complexity to the transductive learning settings. Similar to the inductive one, it measures the expected correlation of the patterns generated by the algorithm and the noise. In other words, the Rademacher complexity is a measure of how likely an algorithm can detect a pattern in pure noise. Thus, even the numeric value of the Rademacher complexity gives meaningful insight to the performance of an algorithm (Shawe-Taylor and Cristianini, 2004).

We define the loss function of the ranking algorithm $\mathcal{L}^{\mathbf{g}}(\mathbf{x}_i, \mathbf{x}_{i'})$ as ± 1 loss in determining the pairwise preference between two documents $\mathbf{x}_i$ and $\mathbf{x}_{i'}$. The theorem below states an upper bound on the generalization error of LM-HTR:

Theorem 1 ***Risk bound for LM-HTR***

Fix $\delta \in (0, 1)$. Let $(\mathbf{z}_i)_{i=1}^m$ be samples independently drawn from a probability distribution in the source domain. Suppose $(\mathbf{x}_i)_{i=1}^n$ are generated independently according to another distribution in the target domain. Define the constants $c_0 = \sqrt{\frac{32 \ln(4e)}{3}}$, $Q \triangleq \left(\frac{2}{n(n-1)} + \frac{2}{m(m-1)}\right)$ and $S \triangleq \frac{m+n}{(m+n-1/2)(1-1/(2\max(m,n)))}$. With probability at least $1-\delta$, the expected loss for ranking of any sample $(\mathbf{x}_i, \mathbf{x}_j)$ pair for $i, j = 1, \ldots, n$ in the target domain can be bounded as follows:

$$\mathbb{E}[\mathcal{L}^{\mathbf{g}}(\mathbf{x}_i, \mathbf{x}_j)] \leq \frac{2}{m(m-1)} \sum_{\substack{i,j=1 \\ j \neq i}}^{m} \xi_{ij} + \left(\frac{2}{m(m-1)} + \frac{2}{n(n-1)}\right)$$

$$\times \left(\sqrt{R_S} + \sqrt{R_T}\right) + c_0 Q \sqrt{q(q-1)/2} + 2\sqrt{\frac{SQ}{2} \ln \frac{1}{\delta}}.$$

where $R_S = \sum_{i,j=1}^m (\mathbf{a}_i - \mathbf{a}_j)^\top (\mathbf{a}_i - \mathbf{a}_j)$, $R_T = \sum_{i,j=1}^n (\mathbf{e}_i - \mathbf{e}_j)^\top (\mathbf{e}_i - \mathbf{e}_j)$ and $q = \min(m, n)$.

Proof The formal proof is given in the supplementary materials. The proof is established by observing the pairwise ranking of n samples as $\frac{n(n-1)}{2}$ classification tasks. Adapting the procedures for the classification problems in (Shawe-Taylor and Cristianini, 2004) to the transductive settings using the Transductive Rademacher Complexity (El-yaniv and Pechyony, 2007) yields the desired results. In order to capture the effect of dimensionality reduction, it suffices to perform the analysis in the latent domain; we can evaluate the empirical complexity using the values of $\mathbf{a}_i$ and $\mathbf{e}_i$ learned through the experiments. ■

Discussion As we can see, the bound in Theorem 1 involves four terms: (i) the empirical error, (ii) the transductive Rademacher complexity term and (iii)-(iv) residual decaying terms. This reveals the dependency of the error bound to the different factors of our LMTHR algorithm:

1. *Dimensionality Reduction* When the dimensionality reduction algorithm decreases the variance in the unrelated and noisy dimensions, the values of R_S and R_T decrease and the algorithm enjoys lower a generalization bound. Smaller latent space dimensions have lower complexity and better generalization performance. This effect will be numerically verified in the experiments section. Note that reduction of dimensionality to very small dimensions will show its effect on the bound by increasing the empirical error term.
2. *Number of Samples* Theorem 1 indicates that the risk bound decays quadratically as the number of samples increases. This is the direct effect of pairwise ranking in which we create $\mathcal{O}(n^2)$ classification tasks from n available samples for ranking.

Note that the bound above is derived for the algorithm when it is properly initialized with the result of TSVM and contained to the neighborhood of the initialization point. Clearly, with random initialization the complexity of the algorithm, in the worst case, can be 2^r times larger than the above quantity because one can flip the signs of each element of the vectors in the latent space and find another optimal solution.

5. Scalability

In this section, we describe how to efficiently solve the TSVM ranking problem for large datasets by methods such as Stochastic Gradient Descent.

Stochastic Gradient Descent Solution The problem in Eq. (5) is a difference convex program and can be solved using the CCCP procedure as described in (Collobert et al., 2006). In this method, the unlabeled samples are duplicated and each sample receives both $y^*_{ii'} = +1$ and $y^*_{ii'} = -1$ labels. Then, $\mathbf{w}$ is iteratively updated by solving the following problem:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C_3 \sum_{\substack{i,i'=1 \\ i\neq i'}}^{m} \xi_{ii'} + C_4 \sum_{\substack{i,i'=1 \\ i\neq i'}}^{n} \xi^*_{ii'} + \boldsymbol{\alpha}^\top \mathbf{w} \tag{8}$$

subject to:

$$\forall (i,i') \in \mathcal{S}_j^S \quad y_{ii'}\mathbf{w}^\top(\mathbf{a}_i - \mathbf{a}_{i'}) \geq 1 - \xi_{ii'}, \quad \xi_{ii'} \geq 0$$
$$\forall (i,i') \in 2 \times \mathcal{S}_j^T \quad y^*_{ii'}\mathbf{w}^\top(\mathbf{e}_i - \mathbf{e}_{i'}) \geq 1 - \xi^*_{ii'}, \quad \xi^*_{ii'} \geq 0$$

Procedure 1 Fast Algorithms for the LM-HTR Model

Input: Parameters $C_1 - C_4$, r, and β. The precision parameter ϵ.

Input: Source and Target samples $\{\mathbf{z}_i\}_{i=1}^m$, $\{\mathbf{x}_i\}_{i=1}^n$. Ranking information, and query groups $\mathcal{S}_j^S$ and $\mathcal{S}_j^T$

Initialization:

Perform a PCA to map $\mathbf{x}_i$ to $\mathbf{e}_i$ for $i = 1, \ldots, n$.

Perform a PCA on $\mathbf{z}_i$ to initialize $\mathbf{a}_i$ for $i = 1, \ldots, m$. Initialize $\mathbf{\Phi}$ and $\tilde{\mathbf{\Phi}}$ with the PCA matrices.

Initialize $\mathbf{w}^0$ with the standard TSVM solution.

repeat

Update $\{\mathbf{a}_i\}^t$ and $\{\mathbf{e}_i\}^t$ by solving Eq. (7). Use smoothed estimation in Eq. (11) for speed boost.

Update $\mathbf{\Phi}^t$ and $\tilde{\mathbf{\Phi}}^t$ by solving the problem in eq(4) via its Lagrange Dual formulation (Lee et al., 2006).

Update $\mathbf{w}^t$ by solving the problem in Eq. (5). Use SGD in Eq. (9-10) for speed boost.

until $\left\|\mathbf{w}^t - \mathbf{w}^{t-1}\right\|_2 \leq \epsilon$

Output: $\mathbf{w}^t$

where $\alpha_k = \sum_{i,i'} \frac{d}{dw_k} H\left(y^*_{i,i'} f(\mathbf{e}_i - \mathbf{e}_{i'})\right)$, for $k = 1, \ldots, d$ and $H(t) = \max(0, 1-t)$. Eq. (8) can be solved by Quadratic Programming (Collobert et al., 2006). However, Quadratic Programming requires $\mathcal{O}((2n+m)^2)$ units of memory and does not scale well as the number of samples increase, especially in the pairwise ranking task where $\mathcal{O}((m+n)^2)$ number of pairs are created. The Stochastic gradient update to solve the problem in Eq. (5) is performed in the following way: choose a pair of samples with probability $p_0 = \frac{m}{m+n}$ from source and with probability $1 - p_0$ from the target domain. For any pair $\mathbf{a}_j, \mathbf{a}_k$ from the source domain perform the following update:

$$\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} - \frac{\eta_0}{1 + C\eta_0 t} \times \left\{ \frac{\boldsymbol{\alpha}}{n+m} + C_3 \frac{\partial}{\partial \mathbf{w}} H_1\left(y_{kj}(\mathbf{w}^\top(\mathbf{a}_k - \mathbf{a}_j))\right) \right\} \tag{9}$$

For any pair $\mathbf{e}_j, \mathbf{e}_k$ from the target domain perform the following update:

$$\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} - \frac{\eta_0}{1 + C\eta_0 t} \times \left\{ \frac{\boldsymbol{\alpha}}{n+m} + C_4 \frac{\partial}{\partial \mathbf{w}} \left[H_1\left(+\mathbf{w}^\top(\mathbf{e}_k - \mathbf{e}_j)\right) + H_1\left(-\mathbf{w}^\top(\mathbf{e}_k - \mathbf{e}_j)\right) \right] \right\} \tag{10}$$

The proposed Stochastic Gradient Descent solution requires only $\mathcal{O}(n+m)$ storage units. The run-time advantage of the SGD solution is demonstrated in the experiments section. Finally, Algorithm 1 summarizes the steps to achieve fast solutions to LM-HTR.

The Representation Learning Sub-Problems The problem in Eq. (7) is a standard quadratic programming problem, whose solution can be extremely expensive. To achieve scalability, we use the following smooth approximation $H_s(t)$

$$H_s(t) = \begin{cases} \frac{1}{2} - t & \text{if } t \leq 0 \\ \frac{1}{2}(1-t)^2 & \text{if } 0 < t < 1 \\ 0 & \text{if } t \geq 1 \end{cases} \tag{11}$$

Table 1: The average ranking error measured in NDCG@n by different ranking algorithms.

Dataset	US → Cn1		US → Cn2		Cn1 → Cn2		Cn2 → Cn1		OHSUMED	
NDCG@n	@1	@3	@1	@3	@1	@3	@1	@3	@1	@3
LM-HTR	0.5810	0.8724	0.3340	0.8602	0.3373	0.8927	0.5788	0.8972	0.7762	0.8959
TSVM	0.5794	0.8701	0.3320	0.8583	0.3366	0.8897	0.5787	0.9007	0.7762	0.8957
SVM	0.5690	0.8626	0.3275	0.8572	0.3372	0.8896	0.5784	0.9022	0.7770	0.8953
Duh'08	0.5702	0.8541	0.3566	0.8575	0.3354	0.8722	0.5561	0.8670	0.6837	0.8551
Gao'10	0.5684	0.8340	0.3274	0.8444	0.3385	0.8996	0.5805	0.8589	0.6512	0.8915

as an alternative to the hinge loss $H_1(t) = \max\{0, 1-t\}$, and apply the Gradient Descent algorithm (Boyd and Vandenberghe, 2004) for solution.

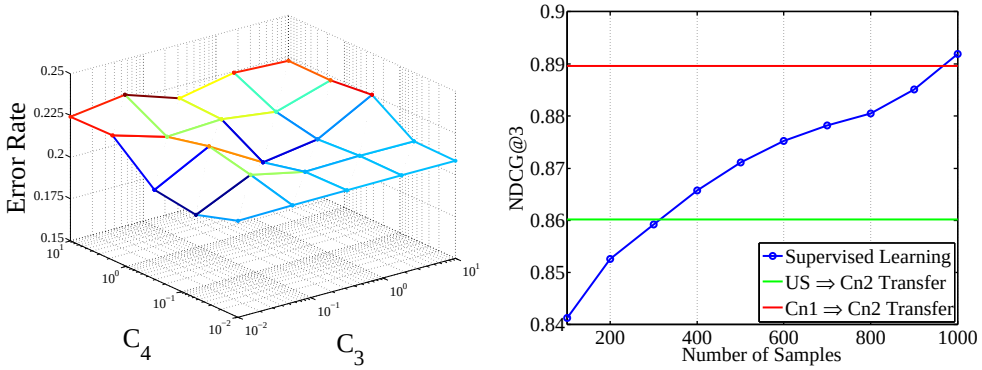
6. Experiments

In order to demonstrate the effectiveness of our algorithm, we compare the performance of LM-HTR with SVMrank, TSVMrank and the algorithms proposed in (Duh and Kirchhoff, 2008) (later referred to as Duh'08) and (Gao et al., 2010) (later referred to as Gao'10) in the Yahoo Search and OHSUMED datasets.

The Yahoo Search Data has been used for verification of performance of the LM-HTR algorithm. The dataset contains the web search data in the United States and two non-US countries, denoted by Cn1 and Cn2. Each data instance is for a query-url pair. For each query, documents are ranked as bad, fair, good, excellent and perfect match. The features generally fall into the following three categories: query features, document features and query-document features. The Query-document features comprise features dependent on the relation of the query with respect to the document, for example, the number of times each term in the query appears in the document, the number of times each term in the query appears in the anchor-texts of the document, etc. We defined four transfer ranking tasks (US → Cn1, US → Cn2, Cn1 → Cn2 and Cn2 → Cn1) and performed transfer ranking experiments on them.

The OHSUMED document ranking dataset is a set of 348,566 references from MEDLINE, an on-line database of medical information. Extracted features include title, abstract, MeSH indexing terms, author, source, and publication type of the journals published during 1987-1991. We use the cleaned version of the dataset available in the LETOR3.0 collection (Liu et al., 2007) which has 16140 documents ranked for 106 queries. Similar to (Gao et al., 2010), we create the source and target datasets by splitting the dataset into two parts, each with 53 queries. Notice that for the OHSUMED dataset, the queries are different for the source and target domain. In other words, the ranking algorithm cannot practically learn per query ranking information and the domain adaptation can help transferring the learning from one query to another one. This is a heterogeneous transfer learning in a loose sense. Unfortunately there are very limited data publicly available for this new learning scenario. To make the results convincing, we use this publicly available dataset (OHSUMED) so that the results can be repeatable.

Accuracy Comparison For accuracy comparison, We report the Normalized Discounted Cumulative Gain (NDCG) values, (Järvelin and Kekäläinen, 2002). LM-HTR has six hy-



(*a*) Sensitivity (*b*) Transduction

Figure 1: (a) Sensitivity of LM-HTR performance with respect to hyper-parameters C_3 and C_4. (b) Illustration of the transduction gain: Learning curve of SVM on the `Cn2` dataset and the accuracy of the LM-HTR obtained by transfering information from the `US` and `Cn1` datasets.

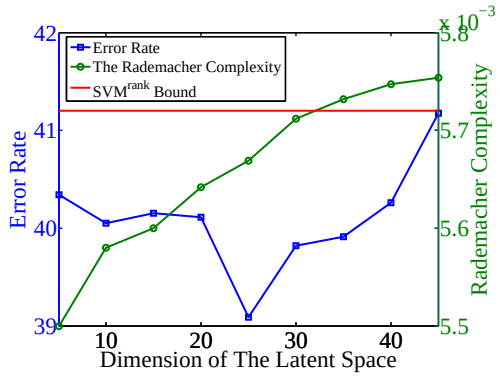
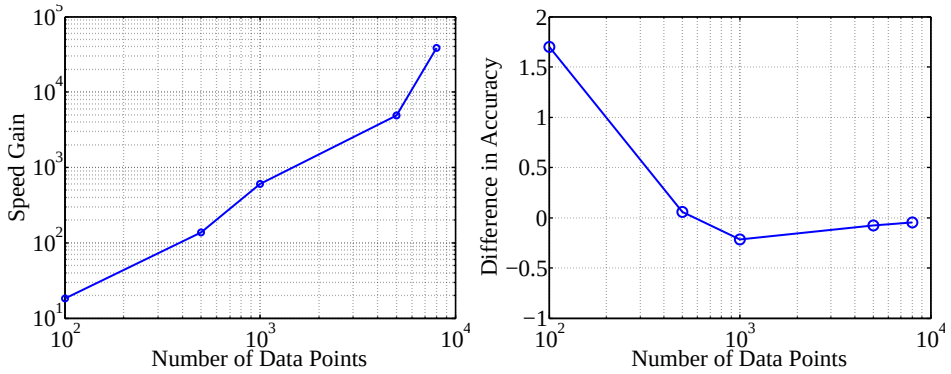


Figure 2: The error rate and transductive Rademacher Complexity of LM-HTR obtained in different latent space dimension on the `OHSUMED` dataset. The red horizontal line shows the value of Empirical Rademacher complexity of SVM^{rank}.

perparameters $(C_1 - C_4, \beta, r)$ to be tuned during the performance evaluation which can be impractical for some large datasets. However, our sensitivity analysis showed that the parameters C_1, C_2 and β do not significantly impact the performance of the algorithm. Thus we set them to a small number and tune the rest of the parameters. To have a more fair comparison, we use linear kernels for all baselines. We use 5-fold cross validation for tuning the values of C_3, C_4 and r.

As it is shown in Table 1, LM-HTR outperforms other algorithms in the transfer ranking tasks. The SVM and TSVM algorithms are not developed for transfer learning tasks, however better performance of TSVM hints the possibility of information transfer in the datasets. The `Duh'08` algorithm attempts to find a pattern in the target samples and projects the source samples on that pattern. However, in our datasets the number of documents listed for a query is small (for e.g. on average 50 documents per query in the



(a) (b)

Figure 3: (a) The ratio of run time of the TSVM implemented via Quadratic Programming over the SGD implementation, while the error rate is kept the same. The horizontal axis represents the number of labeled data points (and unlabeled too). (b) The difference in achieved accuracy (in percentage).

Yahoo dataset.) and finding patterns in 50 data points is usually error prone. Our algorithm relies on all the samples and avoids over-fitting that happens in the `Duh'08` algorithm. `Gao'10` uses weights based on the similarity of the samples to the target samples. Our results confirm that the proposed weighting degrades the performance of algorithm in the studied datasets; because it discards many of the samples that are differently distributed from the target samples. LM-HTR uses all the samples by mapping them into a suitable latent space.

Transduction Gain In order to quantify the amount of gain achieved by LM-HTR in terms of number of equivalent labeled samples in the target domain, we design another experiment as follows: we provide SVM-rank labeled examples from the target domain and train SVM-rank with the labeled examples. We obtain the learning curve by increasing the number of labeled examples. Fig. 1(b) shows the learning curve on the `Cn2` dataset. We also plot the corresponding ranking performance by LM-HTR obtained by transferring information from the `US` and `Cn1` datasets. As we can see, to achieve the same performance of LM-HTR with transfer from `Cn1` to `Cn2` and `US` to `Cn2`, the SVM-rank algorithm requires more than 300 and 900 labeled examples from the target domain, respectively. Note that providing 900 labeled examples from the target domain can be challenging, especially in the scenarios that the characteristics of the datasets change rapidly with time.

Parameter Impact Study The parameters C_3 and C_4 are the coefficients of the ranking loss terms, which can impact the performance of our model. Figure 2 shows the pairwise labeling error for a set of values of C_3 and C_4 in an experiment on the synthetic dataset. While the results confirm the robustness of the performance of LM-HTR with respect to the change in hyper-parameter values, higher values of C_3 are slightly more favorable, which suggests the importance of the labeling information in the source domain.

Empirical Rademacher Complexity In order to study the effect of latent space dimension, we resort to the risk analysis in Theorem 1 to provide more insight. Figure 2 shows

the mean Rademacher complexity values with different latent space dimensionality r on the OHSUMED dataset. As expected, the Rademacher complexity of LM-HTR is upper bounded by SVM, suggesting the superior generalization performance of LM-HTR. The Rademacher complexity plot shows that as we map the data points to a higher dimensional latent space, the generalization error increases. However, considering the accuracy plot, we can deduce that, as expected, the prediction task can become easier as we increase the dimensions of the latent space. As the plot suggests, $r = 25$ is the dimension in which the trade-off point.

Speed Boost by the SGD Algorithm We perform an experiment on a synthetic classification dataset with different number of samples to demonstrate the speed advantage of SGD over Quadratic programming in solving the TSVM problem. The running time of the SGD TSVM algorithm is compared with the Quadratic Programming version in which QP is solved by invoking the Gurobi optimization package (Gu et al., 2011). Figure 3(a) shows the running time enhancement by SGD while the percentage of difference in accuracy is plotted in Figure 3(b). The Quadratic Programming quickly became impractical for datasets with more than $8,000$ labeled samples, when we stopped the experiment. Meanwhile SGD converged in less than 0.1 seconds in all of the experiments.

The speed gain for the TSVM ranking should be much greater because the number of variables grow with $\mathcal{O}((m+n)^2)$ and the QP becomes impractical for datasets as small as 100 data points.

7. Conclusion

In this paper, we proposed a general frame work to solve the heterogeneous ranking problem by mapping the input features in both the source domain and target domain into a shared latent space and simultaneously minimizing the feature reconstruction loss and prediction loss. Under the framework, we designed a transfer ranking algorithm, called LM-HTR. Theoretic bounds of the prediction loss are provided. We also developed fast algorithms via stochastic gradient descent so that they are scalable for large-scale applications. For future work, we are interested in investigating theoretical analysis on general heterogeneous transfer learning algorithms.

Acknowledgment

The research was sponsored by in part by NSF research grants IIS-1134990 and Yahoo! Faculty Award. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agency, or the U.S. Government.

References

Andrew Arnold, Ramesh Nallapati, and William W. Cohen. A comparative study of methods for transductive transfer learning. In *ICDMW*, 2007.

Mohammad Taha Bahadori, Yan Liu, and Dan Zhang. Learning with minimum supervision: A general framework for transductive transfer learning. In *ICDM'11*, 2011.

Jing Bai, Fernando Diaz, Yi Chang, Zhaohui Zheng, and Keke Chen. Cross-market model adaptation with pairwise preference data for web search ranking. In *COLING*, 2010.

Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: risk bounds and structural results. *JMLR*, 2003.

John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, 2007.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

David M. Bradley and J. Andrew Bagnell. Convex coding. In *UAI*, 2009.

C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *ICML*, 2005.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *ICML*, 2007.

Depin Chen, Jun Yan, Gang Wang, Yan Xiong, Weiguo Fan, and Zheng Chen. Transrank: A novel algorithm for transfer of rank learning. ICDMW, 2008.

Ronan Collobert, Fabian Sinz, Jason Weston, and Léon Bottou. Large Scale Transductive SVMs. *JMLR*, 2006.

C. Cortes, M. Mohri, and A. Rastogi. Magnitude-preserving ranking algorithms. In *ICML*, 2007.

Hal Daume, III. Frustratingly Easy Domain Adaptation. In *ACL*, 2007.

Lixin Duan, Dong Xu, and Ivor W. Tsang. Learning with augmented features for heterogeneous domain adaptation. In *ICML*, 2012.

Kevin Duh and Katrin Kirchhoff. Learning to rank with partially-labeled data. In *SIGIR*, 2008.

Ran El-yaniv and Dmitry Pechyony. Transductive rademacher complexity and its applications. In *COLT*, 2007.

Wei Gao, Peng Cai, Kam-Fai Wong, and Aoying Zhou. Learning to rank only using training data from related domain. In *SIGIR*, 2010.

Zonghao Gu, Edward Rothberg, and Robert Bixby. Gurobi 4.6.2, 2011. URL http://www.gurobi.com/.

J. Guiver and E. Snelson. Learning to rank with SoftRank and Gaussian processes. In *SIGIR*, 2008.

Jingrui He, Yan Liu, and Richard Lawrence. Graph-based transfer learning. In *CIKM*, 2009.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 2002.

T. Joachims. Optimizing search engines using clickthrough data. In *KDD*, 2002.

Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y. Ng. Efficient sparse coding algorithms. In *NIPS*, 2006.

Tie-Yan Liu, Jun Xu, Tao Qin, Wenying Xiong, and Hang Li. LETOR: Benchmark Dataset for Research on Learning to Rank for Information Retrieval. In *LR4IR*, 2007.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. on Know. and Data Eng.,*, 2010.

Brian Quanz and Jun Huan. Large margin transductive transfer learning. In *CIKM '09*, 2009.

Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y. Ng. Self-taught learning: transfer learning from unlabeled data. ICML, 2007.

John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

S. Thrun and L.Y. Pratt. *Learning To Learn*. Kluwer Academic Publishers, 1998.

Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag, 1995.

J. Xu and H. Li. Adarank: a boosting algorithm for information retrieval. In *SIGIR*, 2007.

Qiang Yang, Yuqiang Chen, Gui-Rong Xue, Wenyuan Dai, and Yong Yu. Heterogeneous transfer learning for image clustering via the socialweb. In *ACL*, 2009.

Zhaohui Zheng, Keke Chen, Gordon Sun, and Hongyuan Zha. A regression framework for learning ranking functions using relative relevance judgments. In *SIGIR*, 2007.

Yin Zhu, Yuqiang Chen, Zhongqi Lu, Sinno Jialin Pan, Gui-Rong Xue, Yong Yu, and Qiang Yang. Heterogeneous transfer learning for image classification. In *AAAI*, 2011.

Appendix A. Proofs

Proof of Theorem 4.1

Let us consider the classification task for one of the classes. Denote $\bar{\mathbf{f}}_i = (f_i(1), \cdots, f_i(n+m)) \in \mathbb{R}^{m+n}$ the *soft-labels* (before sign function) of all points in the set $A_{m+n} \triangleq (\mathbf{z}_1, \ldots, \mathbf{z}_m, \mathbf{x}_1, \ldots, \mathbf{x}_n)$. We can define $\mathcal{F} \subseteq \mathbb{R}^{m+n}$ as the set of all possible soft classification points that are generated by our learning algorithm. The goal of the learning algorithm is to minimize the test error $\mathcal{L}_n(\bar{\mathbf{f}}_j) \triangleq \frac{1}{n}\sum_{i=m+1}^{m+n} loss(f_j(i), y_i)$ where $loss(.,.)$ is the 0/1 loss function. The function $\hat{\mathcal{L}}_m(\bar{\mathbf{f}}_j) \triangleq \frac{1}{m}\sum_{i=1}^{m} loss(f_j(i), y_i)$ is the empirical error of the algorithm on the training set.

According to (El-yaniv and Pechyony, 2007), we have the following definition of transductive Rademacher complexity and the next theorem showing its application in bounding the expected loss of transductive SVM.

Definition 2 ***Transductive Rademacher Complexity*** *Let $\mathcal{F} \subseteq \mathbb{R}^{m+n}$ and $p \in [0, 1/2]$. Let $\boldsymbol{\sigma} = (\sigma_1, \cdots, \sigma_{m+n})$ be a vector of iid random variables such that*

$$\sigma_i \triangleq \begin{cases} 1, & \textit{with probability} \quad p; \\ -1, & \textit{with probability} \quad p; \\ 0, & \textit{with probability} \quad 1-2p. \end{cases}$$

The (empirical) transductive Rademacher complexity with parameter p is

$$R_{m+n}(\mathcal{F}, p) \triangleq \left(\frac{1}{m} + \frac{1}{n}\right) \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{\mathbf{v} \in \mathcal{F}} \langle \boldsymbol{\sigma}, \mathbf{v} \rangle\right] \tag{12}$$

Risk bound for Transductive SVM Let $\mathcal{F}$ be the set of full-sample soft labellings of the algorithm, generated by operating it on all possible training and test set partitions and $\bar{\mathbf{f}} \in [-1, 1]^{m+n}$. Let $p_0 = \frac{mn}{(m+n)^2}$, $c_0 = \sqrt{\frac{32 \ln(4e)}{3}}$, $Q \triangleq \left(\frac{1}{n} + \frac{1}{m}\right)$ and $S \triangleq$

$\frac{m+n}{(m+n-1/2)(1-1/(2\max(m,n)))}$. With probability at least $1-\delta$ over the choice of the training set from A_{m+n}, for all $\bar{\mathbf{f}} \in \mathcal{F}$,

$$\mathcal{L}_n(\bar{\mathbf{f}}) \leq \hat{\mathcal{L}}_m(\bar{\mathbf{f}}) + R_{m+n}(\mathcal{F}, p_0) + c_0 Q \sqrt{\min(m,n)} + 2\sqrt{\frac{SQ}{2} \ln \frac{1}{\delta}}. \tag{13}$$

Proof: The proof is followed by bounding the empirical Rademacher complexity of the class of functions produced by our algorithm ($R_{m(m-1)/2+n(n-1)/2}(\mathcal{F}, p_0)$) and an application of the bound above. We can bound $R_{m(m-1)/2+n(n-1)/2}(\mathcal{F}, p_0)$ as following (For cleanness of notation, let $\frac{1}{N} = \frac{1}{m(m-1)} + \frac{1}{n(n-1)}$):

$$\begin{aligned}
&R_{m(m-1)/2+n(n-1)/2}(\mathcal{F}, p) = \frac{1}{N}\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{\mathbf{v}\in\mathcal{F}} \left\langle \boldsymbol{\sigma}, \mathbf{v} \right\rangle\right] \\
&= \frac{1}{N}\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{f\in\mathcal{F}} \sum_{\substack{i,i'=1 \\ i\neq i'}}^{m} \sigma_{ii'} f(\mathbf{z}_i - \mathbf{z}_{i'}) + \sum_{\substack{i,i'=1 \\ i\neq i'}}^{n} \hat{\sigma}_{ii'} f(\mathbf{x}_i - \mathbf{x}_{i'})\right] \\
&\leq \frac{1}{N}\mathbb{E}_{\boldsymbol{\sigma}}\left[\max_{\|\mathbf{w}\|\leq 1} \sum_{\substack{i,i'=1 \\ i\neq i'}}^{m} \sigma_{ii'} \mathbf{w}^\top(\mathbf{a}_i - \mathbf{a}_{i'})\right] + \frac{1}{N}\mathbb{E}_{\hat{\boldsymbol{\sigma}}}\left[\max_{\|\mathbf{w}\|\leq 1} \sum_{\substack{i,i'=1 \\ i\neq i'}}^{n} \hat{\sigma}_{ii'} \mathbf{w}^\top(\mathbf{e}_i - \mathbf{e}_{i'})\right] \\
&\leq \frac{2}{N}\mathbb{E}_{\boldsymbol{\sigma}}\left[\left\| \sum_{\substack{i,i'=1 \\ i\neq i'}}^{m} \sigma_{ii'}(\mathbf{a}_i - \mathbf{a}_{i'})\right\|_2\right] + \frac{2}{N}\mathbb{E}_{\hat{\boldsymbol{\sigma}}}\left[\left\| \sum_{\substack{i,i'=1 \\ i\neq i'}}^{n} \hat{\sigma}_{ii'}(\mathbf{e}_i - \mathbf{e}_{i'})\right\|_2\right] \\
&= \frac{2}{N}\mathbb{E}_{\boldsymbol{\sigma}}\left[\left(\left\langle \sum_{\substack{i,i'=1 \\ i\neq i'}}^{m} \sigma_{ii'}(\mathbf{a}_i - \mathbf{a}_{i'}), \sum_{\substack{i,i'=1 \\ i\neq i'}}^{m} \sigma_{ii'}(\mathbf{a}_i - \mathbf{a}_{i'})\right\rangle\right)^{1/2}\right] \\
&\quad + \frac{2}{N}\mathbb{E}_{\hat{\boldsymbol{\sigma}}}\left[\left(\left\langle \sum_{\substack{i,i'=1 \\ i\neq i'}}^{n} \hat{\sigma}_{ii'}(\mathbf{e}_i - \mathbf{e}_{i'}), \sum_{\substack{i,i'=1 \\ i\neq i'}}^{n} \hat{\sigma}_{ii'}(\mathbf{e}_i - \mathbf{e}_{i'})\right\rangle\right)^{1/2}\right] \\
&\leq \frac{2}{N}\left(\mathbb{E}_{\boldsymbol{\sigma}}\left[\sum_{\substack{i,i'=1 \\ i\neq i'}}^{m} \sum_{\substack{j,j'=1 \\ i\neq i'}}^{m} \sigma_{ii'}\sigma_{jj'}(\mathbf{a}_i - \mathbf{a}_{i'})^\top(\mathbf{a}_j - \mathbf{a}_{j'})\right]\right)^{1/2} \\
&\quad + \frac{2}{N}\left(\mathbb{E}_{\hat{\boldsymbol{\sigma}}}\left[\sum_{\substack{i,i'=1 \\ i\neq i'}}^{n} \sum_{\substack{j,j'=1 \\ i\neq i'}}^{n} \sigma_{ii'}\sigma_{jj'}(\mathbf{e}_i - \mathbf{e}_{i'})^\top(\mathbf{e}_j - \mathbf{e}_{j'})\right]\right)^{1/2} \\
&\leq \frac{2}{N}\left(\sum_{\substack{i,i'=1 \\ i\neq i'}}^{m} (\mathbf{a}_i - \mathbf{a}_{i'})^\top(\mathbf{a}_i - \mathbf{a}_{i'})\right)^{1/2} + \frac{2}{N}\left(\sum_{\substack{i,i'=1 \\ i\neq i'}}^{n} (\mathbf{e}_i - \mathbf{e}_{i'})^\top(\mathbf{e}_i - \mathbf{e}_{i'})\right)^{1/2} \\
&= \frac{2}{N}\left(\sqrt{R_S} + \sqrt{R_T}\right).
\end{aligned}$$

An application of the result in (El-yaniv and Pechyony, 2007) with $Q \triangleq \left(\frac{2}{n(n-1)} + \frac{2}{m(m-1)}\right)$ and $S \triangleq \frac{m(m-1)+n(n-1)}{2(m(m-1)/2+n(n-1)-1/2)(1-1/(2\max(m,n)))}$ concludes the proof. □