Learning Similarity Function between Objects in Heterogeneous Spaces

Wei Wu[a], Jun Xu[b], Hang Li[b]

[a]*Department of Probability and Statistics, Peking University, No.5 Yiheyuan Road, Beijing, 100871, P. R. China*
[b]*Microsoft Research Asia, 4F Sigma Building, No. 49 Zhichun Road, Beijing, 100190, P. R. China*

Abstract

Many application problems such as data visualization, document retrieval, image annotation, collaborative filtering, and machine translation can be formalized as a task that utilizes a similarity function between objects in two heterogeneous spaces. In this paper, we address the problem of automatically learning such a similarity function using labeled training data. Conventional metric learning can be viewed as learning of similarity function over one single space, while the 'metric learning' problem in this paper can be regarded as learning of similarity function over two different spaces. We assume that the objects in the two original spaces are linearly mapped into a new space and dot product in the new space is defined as the similarity function. The metric learning problem then becomes that of learning the two linear mapping functions from training data. We then give a general and theoretically sound solution to the learning problem. Specifically, we prove that although the learning problem is non-convex, the global optimal solution exists and one can find the optimal solution using Singular Value Decomposition (SVD). We also show that the solution is 'generalizable' to unobserved data and it is possible to kernelize the method. We conducted two experiments; one experiment shows that keywords and images can be visualized in the same space based on the similarity function learned with our method, and the other experiment shows that the accuracy of document retrieval can be improved with the similarity function (relevance function) learned with our method.

Key words: Metric Learning, Similarity Function, Heterogeneous Spaces

1. Introduction

Many application problems can be viewed as a task utilizing a similarity function between objects in two spaces. For example, in document retrieval, queries belong to the query space and documents belong to the document space[1]. Given a query, we want to retrieve the most similar (relevant) documents with respect to the query. The retrieval is actually performed with the similarity function. Similarly, in image annotation, collaborative filtering, and machine translation, there are two spaces and given an object in one space, we want to find the most similar (relevant) objects in the other space. In those cases, the spaces are defined over keyword and image,

[1] Although sometimes query and document can be represented by bag of words (elements in the same space), it is more natural to assume that they belong to different spaces, because they have different characteristics (length, conciseness of representation).

user and item, and source language and target language, respectively. In all the problems, the similarity function plays an important role.

In this paper, we address the problem of automatically learning the similarity function over two heterogeneous spaces using labeled training data. Let us take image data visualization as example. Suppose that we are given some labeled training data consisting of keyword and image pairs and their similarity labels (similar or dissimilar). Our goal is to learn a similarity function which can represent the similarity between any keyword and image pairs. With the learned similarity function, we can plot keywords and images into the same space, in which distance in the space represents the similarity, as shown in Figure 1.

We formalize the problem of learning similarity function over two spaces and propose a general and theoretically sound method for addressing the learning task. Conventional metric learning is conducted over a single space. The learning problem in this paper is actually an extension of it: from one space to two spaces.

Given two spaces of objects, we assume that there are two transformation functions mapping the objects into a new space. The dot product between the images of the objects in the new space is then defined as similarity function. Given training data which contains information on similarities between objects, we aim to maximize the agreement between the training data and the learned similarity function.

One technical challenge for solving the learning (optimization) issue is that it is not convex. We prove, however, that for the current problem, the global optimal solution exists and it is possible to find the optimal solution by using Singular Value Decomposition (SVD). We show that the solution is 'generalizable' to unobserved data. We also kernelize the method, which can be employed when non-linear features are utilized through kernels, and when the number of training instances is smaller than the dimensions of input spaces.

We have conducted experiments to verify the effectiveness of our method in two tasks. The first experiment is about image data visualization, in which we learn the similarities between keywords and images with some labeled instances, and plot them in the same space based on the similarity function learned. We give an example of the learned results to show that the learned similarity function can indeed put similar keywords and images (from two different spaces) together. The second experiment is about document retrieval, in which we learn the relevance (similarity) function between queries and documents. We demonstrate that the learned similarity function can indeed enhance the relevance ranking in terms of Mean Average Precision (MAP), when compared with several other baseline methods including a conventional metric learning method.

The remaining part of the paper is organized as follows. In section 2, we introduce existing work on metric (similarity) function learning. In section 3, we give the formalization of the new metric (similarity) learning problem. In section 4 we explain our solution to the problem, and describe property of the solution and kernelization of the method. We show our experiments in section 5, and conclude the paper in section 6.

2. Related Work

Our work in this paper can be viewed as an extension of conventional metric learning. In metric learning, a distance metric is automatically learned from training data. The metric is usually defined as Euclidean distance in a new space into which the data is mapped by a linear transformation. There are two approaches to distances metric learning. The first approach such

as Principal Component Analysis (PCA) [12], Linear Discriminant Analysis (LDA) [5], Relevant Component Analysis (RCA) [19], and Neighborhood Components Analysis (NCA) [6] manages to directly learn a linear transformation from the data and then defines a metric based on the transformation. LDA, RCA, and NCA are supervised learning methods, and PCA is an unsupervised learning method. For example, the objective function of PCA is to maximize variance of data in the new space with respect to linear transformation, while the objective function of LDA is to maximize between-class variance of data and minimize within-class variance of data in the new space with respect to linear transformation. The other approach [23, 17, 22, 4, 11, 24, 18] tries to directly learn the metric as a Mahalanobis distance which can be represented as a positive semi-definite matrix. The approach takes advantage of the fact that the optimization is convex and employs efficient algorithms to solve the problem. For example, in [23], information on similarity or dissimilarity between pairs of objects is utilized to learn a Mahalanobis distance for clustering. In [22], a metric is learned for k-nearest neighbor by pulling neighboring objects closer and pushing objects with different labels farther away. Both optimization problems can be solved efficiently. In the existing work, learning of metric is conducted over one single space, while in this paper learning of metric (similarity) is performed over two spaces.

Recently, methods of learning similarity functions are proposed in some specific settings. In [7], for example, a similarity function is learned to match text queries to images. The key idea is to learn a transformation from the image space to the text space and measure similarity between texts and images with dot product in the text space. The proposed model called PAMIR is actually a special case of our model. In [1], the authors learn a relevance (similarity) function from relevant (or irrelevant) query and document pairs in which the relevance function is formalized as a low rank model. The most significant difference between their work and our work is that their method does not learn a metric function. For other related work, see [13, 3].

Canonical Correlation Analysis (CCA) [20] or its kernelized version KCCA [9, 10] is also related to our work. CCA is a method to learn two mappings from training data to capture correlations between pairs of objects. Specifically, it attempts to maximize correlations between object pairs which are labeled as similar (relevant). In our work, we also try to learn two mappings from training data. There are stark differences between our method and CCA, however. CCA only takes positive instances as input, while our method takes both positive and negative instances as input. CCA learns correlation coefficients while our method learns a similarity function.

3. Formulation of Learning Problem

Suppose that there are two spaces $\mathcal{X} \subset \mathbb{R}^m$ and $\mathcal{Y} \subset \mathbb{R}^n$. Let x and y be elements (objects) in the two spaces, respectively, and $f(x, y)$ be a function which measures the similarity between x and y. Further assume that labeled training data $\{(x_i, y_i, r_i)\}_{i=1}^N$ is given, where (x_i, y_i) denotes a pair of objects, and r_i denotes its label from $\{+1, -1\}$. $+1$ means that the two objects are similar, and -1 means that they are dissimilar.[2] Our goal is to automatically learn the similarity function $f(x, y)$ from the training data.

The challenge in the above problem is that objects x and y are from two heterogeneous spaces $\mathcal{X}$ and $\mathcal{Y}$ with different features (e.g. dimensions). It is not trivial how to define the similarity function. Our proposal is to define the similarity function as dot product based on two linear

[2]Note that it is also possible to extend r from binary values to multiple values.

transformations. More specifically, we linearly map objects x and y into a new space, and define dot product in the space as similarity function between them. (In this paper, we restrict ourselves to linear mapping and take the study of non-linear mapping as future work.) Suppose that the two linear mapping functions for $\mathcal{X}$ and $\mathcal{Y}$ are represented as $L_{\mathcal{X}}$ and $L_{\mathcal{Y}}$, respectively. Suppose that the new space is a subspace of $\mathbb{R}^k$. Then, $L_{\mathcal{X}}$ is an $m \times k$ matrix and $L_{\mathcal{Y}}$ is an $n \times k$ matrix. Given a pair of objects (x, y), we first transform x and y to $L_{\mathcal{X}}^\top x$ and $L_{\mathcal{Y}}^\top y$ respectively. Next, we define the similarity between x and y as $\langle L_{\mathcal{X}}^\top x, L_{\mathcal{Y}}^\top y\rangle = x^\top L_{\mathcal{X}} L_{\mathcal{Y}}^\top y$, where $\langle\cdot,\cdot\rangle$ denotes dot product in $\mathbb{R}^k$.

Our formulation naturally extends conventional distance metric learning from one single space to two different spaces. In conventional distance metric learning, one linear transformation is (either explicitly or implicitly) learned and usually Euclidean distance in the new space is utilized as metric function. In our formulation, two linear transformations are learned and dot product $\langle L_{\mathcal{X}}^\top x, L_{\mathcal{Y}}^\top y\rangle$ in the new space is taken as similarity function. In our case, a metric function can be further defined based on the similarity function, $d(x, y) = \langle L_{\mathcal{X}}^\top x, L_{\mathcal{X}}^\top x\rangle + \langle L_{\mathcal{Y}}^\top y, L_{\mathcal{Y}}^\top y\rangle - 2\langle L_{\mathcal{X}}^\top x, L_{\mathcal{Y}}^\top y\rangle$, where $d(x, y)$ denotes a metric function[3].

We aim to learn $L_{\mathcal{X}}$ and $L_{\mathcal{Y}}$ by using the training data $\{(x_i, y_i, r_i)\}_{i=1}^n$. We formalize the learning problem as the following optimization problem :

$$\begin{aligned} &\arg\max_{L_{\mathcal{X}}, L_{\mathcal{Y}}} \sum\nolimits_{r_i=+1} x_i^\top L_{\mathcal{X}} L_{\mathcal{Y}}^\top y_i - \sum\nolimits_{r_i=-1} x_i^\top L_{\mathcal{X}} L_{\mathcal{Y}}^\top y_i \\ &\text{subject to} \quad L_{\mathcal{X}}^\top L_{\mathcal{X}} = I_{k\times k}, L_{\mathcal{Y}}^\top L_{\mathcal{Y}} = I_{k\times k}. \end{aligned} \tag{1}$$

The intuitive explanation is that with the optimization similar objects will become closer and dissimilar objects will become farther apart. The constraint of the optimization problem requires that the mapping functions (transformations) are orthonormal. Such a constraint is natural and widely used in machine learning. We will show in the following sections that with this constraint we can find the *global optimal* solution and guarantee the *generalizability* of the solution.

4. Our Learning Method

4.1. Solution by SVD

The optimization problem is not convex with respect to $L_{\mathcal{X}}$ and $L_{\mathcal{Y}}$. Nonetheless we can prove that the global optimal solution exists for the problem. The optimal solution can be obtained using Singular Value Decomposition (SVD).

The objective function in (1) can be re-written as:

$$Trace(L_{\mathcal{X}} L_{\mathcal{Y}}^\top \sum_{r_i=1} y_i x_i^\top) - Trace(L_{\mathcal{X}} L_{\mathcal{Y}}^\top \sum_{r_i=-1} y_i x_i^\top) = Trace\left(L_{\mathcal{Y}}^\top (M_S - M_D) L_{\mathcal{X}}\right),$$

where M_S and M_D are defined as $\sum_{r_i=1} y_i x_i^\top$ and $\sum_{r_i=-1} y_i x_i^\top$, respectively. Equivalently, we need to solve the following optimization problem:

$$\begin{aligned} &\arg\max_{L_{\mathcal{X}}, L_{\mathcal{Y}}} Trace\left(L_{\mathcal{Y}}^\top (M_S - M_D) L_{\mathcal{X}}\right) \\ &\text{subject to} \quad L_{\mathcal{X}}^\top L_{\mathcal{X}} = I_{k\times k}, L_{\mathcal{Y}}^\top L_{\mathcal{Y}} = I_{k\times k}. \end{aligned} \tag{2}$$

With regard to the new optimization problem, the following theorem holds. Note that in optimization problem (1), $k \leqslant min(n, m)$ holds, since $L_{\mathcal{X}}^\top L_{\mathcal{X}} = I_{k\times k}$ and $L_{\mathcal{Y}}^\top L_{\mathcal{Y}} = I_{k\times k}$ are required.

[3]It is easy to verify that it is a metric.

Algorithm 1 Algorithm of prime problem (2)

1: Input: training data $\{(x_i, y_i, r_i)\}_{i=1}^N$, parameter $k \leqslant min(n, m)$.
2: Calculate M_S and M_D through $\sum_{r_i=1} y_i x_i^\top$ and $\sum_{r_i=-1} y_i x_i^\top$, respectively.
3: Calculate SVD of $M_S - M_D$.
4: Choose left and right singular vectors $(u_1, \cdots, u_k)$ and $(v_1, \cdots, v_k)$ w.r.t the top k singular values.
5: Output: $L_{\mathcal{Y}} = (u_1, \cdots, u_k)$ and $L_{\mathcal{X}} = (v_1, \cdots, v_k)$.

Theorem 4.1. *$\forall k \leqslant min(n, m)$, the global optimal solution of the optimization problem (2) exists. Furthermore, suppose that $M_S - M_D = U\Sigma V^\top$, where Σ is an $n \times m$ diagonal matrix with singular values $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \lambda_p \geqslant 0$, $p = min(n, m)$, $U = (u_1, u_2, \cdots, u_n)$ where $\{u_i\}$ are left singular vectors, and $V = (v_1, v_2, \cdots, v_m)$ where $\{v_i\}$ are right singular vectors. The global optimal $\hat{L}_{\mathcal{X}}$ and $\hat{L}_{\mathcal{Y}}$ are given by $\hat{L}_{\mathcal{X}} = (v_1, v_2, \cdots, v_k)$ and $\hat{L}_{\mathcal{Y}} = (u_1, u_2, \cdots, u_k)$.*

The proof is given in Appendix. The algorithm of finding global optimal using SVD is summarized in Algorithm 1.

4.2. Generalizability of Solution

We show that the similarity function learned by our method is generalizable to unobserved object pairs.

Intuitively, given an object $x \in \mathcal{X}$, for any $x' \in \mathcal{X}$ that is similar to x, for any $y \in \mathcal{Y}$, similarity functions $f(x, y)$ and $f(x', y)$ should be close to each other. This is also true for objects $y, y' \in \mathcal{Y}$, if y and y' are similar in $\mathcal{Y}$ then for any $x \in \mathcal{X}$, the difference between similarity functions $f(x, y)$ and $f(x, y')$ should be small. Consequently, if x is similar to x' and y is similar to y', then the similarity between x and y will be close to the similarity between x' and y'. We call this property generalizability of similarity function.

Suppose that $\forall x \in \mathcal{X}$ and $\forall y \in \mathcal{Y}$, $\|x\|_{\mathcal{X}} = \|y\|_{\mathcal{Y}} = 1$, where $\|\cdot\|_{\mathcal{X}}$ and $\|\cdot\|_{\mathcal{Y}}$ are the norms defined in $\mathcal{X}$ and $\mathcal{Y}$, respectively. If $\|x - x'\|_{\mathcal{X}} < \epsilon$ where $\epsilon > 0$ is a small number, we have

$$|f(x, y) - f(x', y)| = |(x - x')L_{\mathcal{X}} L_{\mathcal{Y}}^\top y| \leqslant \|L_{\mathcal{X}}^\top (x - x')\| \cdot \|L_{\mathcal{Y}}^\top y\|.$$

Since $\|L_{\mathcal{X}}^\top (x - x')\|^2 = \sum_{i=1}^k \langle l_i^{\mathcal{X}}, (x - x')\rangle^2$ and $L_{\mathcal{X}}^\top L_{\mathcal{X}} = I_{k\times k}$, then $\|L_{\mathcal{X}}^\top (x - x')\|^2 \leqslant \|x - x'\|_{\mathcal{X}}^2$. Similarly, $\|L_{\mathcal{Y}}^\top y\|^2 \leqslant \|y\|_{\mathcal{Y}}^2 = 1$. Thus, we have

$$|f(x, y) - f(x', y)| \leqslant \|x - x'\|_{\mathcal{X}} < \epsilon.$$

Similarly, if $\|y - y'\|_{\mathcal{Y}} < \epsilon$, then $|f(x, y) - f(x, y')| < \epsilon$.

That is to say, we can estimate the similarity between any object pair x and y, if we know the similarity between the object pairs in its neighborhood.

4.3. Kernelization

We can also derive the dual problem of the optimization problem (1) and employ the kernel trick [16]. Solving the dual problem (kernelized version) is more preferable when the dimensions of input spaces are high compared with number of training instances. It is also necessary, when non-linear features that can composite a kernel are used. Time complexity of the dual problem is a function of number of instances.

We use $\phi_{\mathcal{X}}(x)$ and $\phi_{\mathcal{Y}}(y)$ to respectively denote the feature vectors of objects $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. The corresponding kernels are denoted by $k_{\mathcal{X}}(\cdot,\cdot)$ and $k_{\mathcal{Y}}(\cdot,\cdot)$, respectively. Thus, the optimization problem (1) becomes

$$\begin{aligned}&\arg\max_{L_{\mathcal{X}},L_{\mathcal{Y}}} \sum\nolimits_{r_i=1} \phi_{\mathcal{X}}(x_i)^\top L_{\mathcal{X}} L_{\mathcal{Y}}^\top \phi_{\mathcal{Y}}(y_i) - \sum\nolimits_{r_i=-1} \phi_{\mathcal{X}}(x_i)^\top L_{\mathcal{X}} L_{\mathcal{Y}}^\top \phi_{\mathcal{Y}}(y_i)\\ &\text{subject to} \quad L_{\mathcal{X}}^\top L_{\mathcal{X}} = I_{k\times k}, L_{\mathcal{Y}}^\top L_{\mathcal{Y}} = I_{k\times k}.\end{aligned}$$

Suppose that $L_{\mathcal{X}} = (l_1^{\mathcal{X}}, \cdots, l_k^{\mathcal{X}})$ and $L_{\mathcal{Y}} = (l_1^{\mathcal{Y}}, \cdots, l_k^{\mathcal{Y}})$. With the property of Hilbert space, we have $l_i^{\mathcal{X}} = z_i^{\mathcal{X}} + v_i^{\mathcal{X}}$ and $l_i^{\mathcal{Y}} = z_i^{\mathcal{Y}} + v_i^{\mathcal{Y}}$, $\forall i$, where $z_i^{\mathcal{X}}$ and $z_i^{\mathcal{Y}}$ are from the spaces spanned by $\{\phi_{\mathcal{X}}(x_i)\}_{i=1}^N$ and $\{\phi_{\mathcal{Y}}(y_i)\}_{i=1}^N$, respectively, and $v_i^{\mathcal{X}}$ and $v_i^{\mathcal{Y}}$ are orthogonal to the spaces spanned by $\{\phi_{\mathcal{X}}(x_i)\}_{i=1}^N$ and $\{\phi_{\mathcal{Y}}(y_i)\}_{i=1}^N$, respectively. Since for any $x_j, v_i^{\mathcal{X}}$, $\langle \phi_{\mathcal{X}}(x_j), v_i^{\mathcal{X}} \rangle = 0$, and for any $y_j, v_i^{\mathcal{Y}}$, $\langle \phi_{\mathcal{Y}}(y_j), v_i^{\mathcal{Y}} \rangle = 0$, the optimal $L_{\mathcal{X}}$ and $L_{\mathcal{Y}}$ become $(z_1^{\mathcal{X}}, \cdots, z_k^{\mathcal{X}})$ and $(z_1^{\mathcal{Y}}, \cdots, z_k^{\mathcal{Y}})$, respectively. Thus, $L_{\mathcal{X}}$ can be represented as $\Phi_{\mathcal{X}} \cdot \alpha_{\mathcal{X}}$ and $L_{\mathcal{Y}}$ can be represented as $\Phi_{\mathcal{Y}} \cdot \alpha_{\mathcal{Y}}$, where $\Phi_{\mathcal{X}} = (\phi_{\mathcal{X}}(x_1), \cdots, \phi_{\mathcal{X}}(x_N))$ and $\Phi_{\mathcal{Y}} = (\phi_{\mathcal{Y}}(y_1), \cdots, \phi_{\mathcal{Y}}(y_N))$. In this case, the optimization problem becomes

$$\begin{aligned}&\arg\max_{\alpha_{\mathcal{X}},\alpha_{\mathcal{Y}}} Trace\left(\alpha_{\mathcal{Y}}^\top (\tilde{M}_S - \tilde{M}_D)\alpha_{\mathcal{X}}\right)\\ &\text{subject to} \quad \alpha_{\mathcal{X}}^\top K_{\mathcal{X}} \alpha_{\mathcal{X}} = I_{k\times k}, \alpha_{\mathcal{Y}}^\top K_{\mathcal{Y}} \alpha_{\mathcal{Y}} = I_{k\times k},\end{aligned}$$

where

$$\tilde{M}_S = \sum\nolimits_{r_i=1} \left(k_{\mathcal{Y}}(y_i, y_1), \cdots, k_{\mathcal{Y}}(y_i, y_N)\right)^\top \left(k_{\mathcal{X}}(x_i, x_1), \cdots, k_{\mathcal{X}}(x_i, x_N)\right), \tag{3}$$

$$\tilde{M}_D = \sum\nolimits_{r_i=-1} \left(k_{\mathcal{Y}}(y_i, y_1), \cdots, k_{\mathcal{Y}}(y_i, y_N)\right)^\top \left(k_{\mathcal{X}}(x_i, x_1), \cdots, k_{\mathcal{X}}(x_i, x_N)\right), \tag{4}$$

$K_{\mathcal{X}} = \left(k_{\mathcal{X}}(x_i, x_j)\right)_{N\times N}$, and $K_{\mathcal{Y}} = \left(k_{\mathcal{Y}}(y_i, y_j)\right)_{N\times N}$.

Let us further suppose that both $K_{\mathcal{X}}$ and $K_{\mathcal{Y}}$ are full-rank matrices. Constraint $\alpha_{\mathcal{X}}^\top K_{\mathcal{X}} \alpha_{\mathcal{X}} = I_{k\times k}$ is equivalent to $\alpha_{\mathcal{X}}^\top K_{\mathcal{X}}^{\frac{1}{2}} K_{\mathcal{X}}^{\frac{1}{2}} \alpha_{\mathcal{X}} = I_{k\times k}$. Thus, we can define $\tilde{\alpha}_{\mathcal{X}} = K_{\mathcal{X}}^{\frac{1}{2}} \alpha_{\mathcal{X}}$, and the constraint becomes $\tilde{\alpha}_{\mathcal{X}}^\top \tilde{\alpha}_{\mathcal{X}} = I_{k\times k}$. Similarly, we can transform the constraint on $\alpha_{\mathcal{Y}}$ to $\tilde{\alpha}_{\mathcal{Y}}^\top \tilde{\alpha}_{\mathcal{Y}} = I_{k\times k}$, where $\tilde{\alpha}_{\mathcal{Y}} = K_{\mathcal{Y}}^{\frac{1}{2}} \alpha_{\mathcal{Y}}$. Under the new constraints, the optimization problem becomes

$$\begin{aligned}&\arg\max_{\tilde{\alpha}_{\mathcal{X}},\tilde{\alpha}_{\mathcal{Y}}} Trace\left(\tilde{\alpha}_{\mathcal{Y}}^\top K_{\mathcal{Y}}^{-\frac{1}{2}} (\tilde{M}_S - \tilde{M}_D) K_{\mathcal{X}}^{-\frac{1}{2}} \tilde{\alpha}_{\mathcal{X}}\right)\\ &\text{subject to} \quad \tilde{\alpha}_{\mathcal{X}}^\top \tilde{\alpha}_{\mathcal{X}} = I_{k\times k}, \tilde{\alpha}_{\mathcal{Y}}^\top \tilde{\alpha}_{\mathcal{Y}} = I_{k\times k}.\end{aligned} \tag{5}$$

We can follow the result of Theorem 4.1 to solve the optimization problem above.

Given new objects x and y, $L_{\mathcal{X}}^\top \phi_{\mathcal{X}}(x)$ and $L_{\mathcal{Y}}^\top \phi_{\mathcal{Y}}(y)$ are determined by

$$[k_{\mathcal{X}}(x, x_1), \cdots, k_{\mathcal{X}}(x, x_N)] K_{\mathcal{X}}^{-\frac{1}{2}} \cdot \tilde{\alpha}_{\mathcal{X}}. \tag{6}$$

$$[k_{\mathcal{Y}}(y, y_1), \cdots, k_{\mathcal{Y}}(y, y_N)] K_{\mathcal{Y}}^{-\frac{1}{2}} \cdot \tilde{\alpha}_{\mathcal{Y}}. \tag{7}$$

When the number of training instances N is smaller than $min(n, m)$, solving the dual problem (5) will be more efficient than solving the prime problem (2), although we have to calculate two matrix inverses. The algorithm of dual problem (5) is given in Algorithm 2.

Algorithm 2 Algorithm of dual problem (5)

1: Input: training data $K_{\mathcal{X}} = (k_{\mathcal{X}}(x_i, x_j))_{N\times N}$, $K_{\mathcal{Y}} = (k_{\mathcal{Y}}(y_i, y_j))_{N\times N}$, $\{r_i\}_{i=1}^N$, and parameter $k \leqslant N$.
2: Calculate $\tilde{M}_S$ and $\tilde{M}_D$ through Equation (3) and (4), respectively.
3: Calculate $K_{\mathcal{X}}^{-\frac{1}{2}}$ and $K_{\mathcal{Y}}^{-\frac{1}{2}}$.
4: Calculate SVD of $K_{\mathcal{X}}^{-\frac{1}{2}}(\tilde{M}_S - \tilde{M}_D)K_{\mathcal{Y}}^{-\frac{1}{2}}$.
5: Choose left and right singular vectors $(u_1, \cdots, u_k)$ and $(v_1, \cdots, v_k)$ w.r.t the top k singular values.
6: Output: $\alpha_{\mathcal{Y}} = K_{\mathcal{Y}}^{-\frac{1}{2}}(u_1, \cdots, u_k)$ and $\alpha_{\mathcal{X}} = K_{\mathcal{X}}^{-\frac{1}{2}}(v_1, \cdots, v_k)$.

5. Experiment

5.1. Image Data Visualization

We have conducted experiment on image data visualization with our similarity learning method. In the experiment, we learned the similarity function between keywords and images from training data. Based on the similarity function learned, we plotted the data into a two dimensional space using Multidimensional Scaling (MDS) [2]. (MDS is a common tool for embedding data from high dimensional space into low dimensional space.) We use the experimental results to show that our method can put texts and images into the same space and preserve their similarity relations very well. Such a technique is useful for data visualization over heterogeneous data. Similar things can be done on collaborative filtering data, translation data, etc.

We made use of the image data set called "ESP Game", which was used in [8] [4]. It contains more than $20,000$ images and 268 keywords. Each image is annotated with several keywords and the keywords represent the content of the image. Average number of keywords per image is 4.7. The whole data set is split into training and testing data sets. In our experiment, we only used the test data set, which contains $2,081$ images and 268 keywords.

We took the annotated data as training data and learned a similarity function with our method. The keywords are represented in a space in which each dimension corresponds to a word and the dimensionality is 268. The images are represented in another space in which each dimension corresponds to an image feature and the dimensionality is $37,152$. If there is a keyword associated with an image, we viewed them as positive instance (similar keyword and image), otherwise, we viewed them as negative instance (dissimilar keyword and image).

To tune the parameter k in our method, we conducted 5-fold cross validation. We evaluated each parameter value using the method in [8]. That is, we assigned each image with its most similar 5 keywords and calculated average precision of the assignments.

We then applied our method (Algorithm 1) to learn the similarity function. The keywords and images were then positioned in the new space with the similarity function. Note that in the original space of keywords, the dimensions are orthogonal, and thus there is no explicit similarity relation between keywords. In contrast, in the new space, the similarities between keyword and keyword, image and image, and keyword and image are all incorporated.

To give a rough idea on how the result looks like, we choose 3 groups of 7 keywords and top 5 similar images for each group, and plot the keywords and images in two dimensional space with

[4] http://lear.inrialpes.fr/people/guillaumin/data_iccv09.php

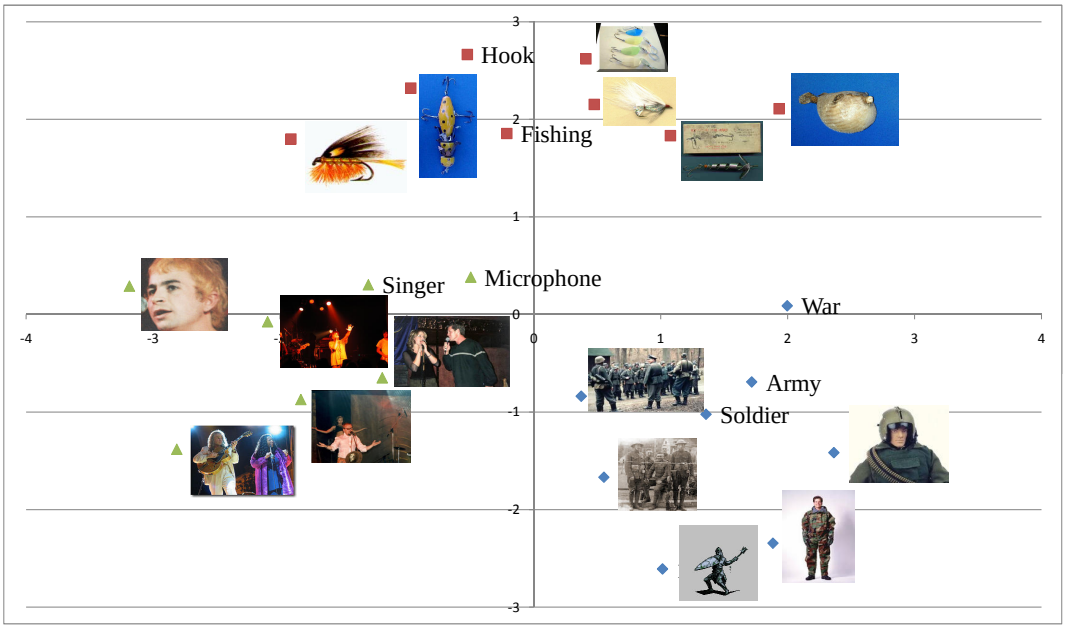

Figure 1: Keywords and images are plotted in the same space with the learned similarity function as 'distance '.

MDS. Figure 1 shows the result. Although certain information has been lost during application of MDS, we can still see that similar images and keywords are really positioned close to each other. That is to say, similarity between heterogeneous data can be accurately learned by our method.(Other example results are given in the supplementary material).

5.2. Document Retrieval

We have also carried out experiment on document retrieval using our similarity learning method. Given query-document pairs and their relevance labels (+1/-1), we learned a similarity (relevance) function with the data using our method. We then ranked documents for other queries with our learned function. We use the experimental results to show that our method can also be used in relevance ranking in document retrieval.

We adopted Mean Average Precision (MAP) [21] as evaluation measure. As baselines, we chose Vector Space Model (VSM) [15], BM25 [14], and Metric Learning for Clustering (MLC) [23]. The former two methods are state-of-the-art methods in document retrieval. In VSM and BM25, no learning is performed and queries and documents are matched with predefined models. In MLC, queries and documents are put into the same space, a single transformation is learned for both queries and documents, and queries and documents are matched through dot product after the transformation.

We made use of the TREC AP data set in the experiment. We utilized queries of year 1992 (query number $51 - 100$), their associated documents, and the related judgments as our data set. There are 50 queries and $6,913$ documents in total. The average number of documents per query is about 156. We used tf-idf of unigrams (words) as features of queries and documents. The

Table 1: MAP values of VSM, BM25, MLC, our method, MLC combined with BM25 and our method combined with BM25 on testing data. Results are averaged over 10 trials. The improvements of our final method over the baselines are statistically significant (t-test, p-value=0.037).

Method	VSM	BM25	MLC	Our Method	MLC+BM25	Our Method+BM25
MAP	0.408	0.413	0.384	0.392	0.415	**0.426**

total number of unigrams is $5,634$, among which 138 words appear in queries and $5,627$ words appear in documents.

In our method, we treated query space and document space as two different spaces, in each of which only words that appear in the space are defined as dimensions. The dimensionality of query space becomes 138, while the dimensionality of document space becomes $5,627$. As a result, we only need to perform SVD on a 138×5627 matrix and this helps to significantly improve efficiency. In contrast, in MLC, the query space and document space are treated as the same space, and thus learning needs to be performed on a much larger matrix ($5,634 \times 5,634$). To efficiently run MLC, we had to first conduct PCA to reduce the dimensionality. We tuned parameter k in our method and the rank of PCA in MLC through 5-fold cross validation.

From (6) and (7), we can see that if a testing query or document is orthogonal to all training queries or documents, then the query or document will be mapped to 0 vector and no similarity can be calculated. To deal with the problem which all the learning based methods have, we linearly combine a learning based model and BM25 as the final model. In other words, for those queries or documents that are orthogonal to the training data, we rely on the conventional relevance model.

Finally, we conducted relevance ranking experiments with the methods. We randomly chose 80% queries as training data and the remaining 20% as testing data. We repeated the processes of learning and ranking 10 times, and took average of the results for each method, as summarized in Table 1. We can see that our final method significantly outperforms the baselines.

6. Conclusion and Future Work

We have proposed a new metric learning problem and its solution in this paper. The metric learning problem is unique in that the similarity function is defined over two heterogeneous spaces. We assume to linearly map the objects in the two heterogeneous spaces into a new space and define dot product in the new space as similarity function. We formalize the learning problem as that of learning the two mapping functions given training data. Our metric learning is a natural extension of conventional metric learning. We have then developed a general and theoretically sound method to solve the new metric learning problem. Although the learning (optimization) problem is not convex, we prove that the global optimal solution exists and we can find the optimal solution through Singular Value Decomposition. We also show the generalizable property of the solution and kernelization of the method. Experiments on image data visualization and document retrieval have demonstrated the effectiveness of our method. As future work, we plan to study extensions of the problem defined in this paper and possible solutions to them, for example, when the mapping functions are non-linear.

References

[1] Bai, B., Weston, J., Collobert, R., Grangier, D., 2009. Supervised semantic indexing. Advances in Information Retrieval, 761–765.

[2] Borg, I., Groenen, P., 1997. Modern multidimensional scaling: Theory and applications. Springer Verlag.

[3] Chechik, G., Sharma, V., Shalit, U., Bengio, S., 2009. An Online Algorithm for Large Scale Image Similarity Learning. NIPS.

[4] Davis, J., Kulis, B., Jain, P., Sra, S., Dhillon, I., 2007. Information-theoretic metric learning. In: ICML. p. 216.

[5] Fisher, R., 1936. The use of multiple measures in taxonomic problems. Ann. Eugenics 7, 179–188.

[6] Goldberger, J., Roweis, S., Hinton, G., Salakhutdinov, R., 2005. Neighbourhood components analysis. NIPS 17, 513–520.

[7] Grangier, D., Bengio, S., 2008. A Discriminative Kernel-Based Model to Rank Images from Text Queries. IEEE transactions on pattern analysis and machine intelligence 30 (8), 1371–1384.

[8] Guillaumin, M., Mensink, T., Verbeek, J., Schmid, C., LEAR, I., Kuntzmann, L., 2009. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. ICCV.

[9] Hardoon, D., Shawe-Taylor, J., 2003. KCCA for different level precision in content-based image retrieval. In: Proceedings of Third International Workshop on Content-Based Multimedia Indexing, IRISA, Rennes, France.

[10] Hardoon, D., Shawe-Taylor, J., 2009. Sparse canonical correlation analysis. stat 1050, 19.

[11] Jain, P., Kulis, B., Dhillon, I., Grauman, K., 2008. Online metric learning and fast similarity search. NIPS, 761–768.

[12] Jolliffe, I., 2002. Principal component analysis. Springer verlag.

[13] Liu, Y., Qin, T., Liu, T., Zhang, L., Ma, W., 2005. Similarity space projection for Web image search and annotation. In: Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval. ACM, p. 56.

[14] Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M., Gatford, M., 1996. Okapi at TREC-4. In: Proceedings of the Fourth Text Retrieval Conference. pp. 73–97.

[15] Salton, G., Wong, A., Yang, C., 1975. A vector space model for automatic indexing. Communications of the ACM 18 (11), 613–620.

[16] Schölkopf, B., Smola, A., 2002. Learning with kernels: Support vector machines, regularization, optimization, and beyond. the MIT Press.

[17] Shalev-Shwartz, S., Singer, Y., Ng, A., 2004. Online and batch learning of pseudo-metrics. In: ICML. ACM, p. 94.

[18] Shent, C., Kimt, J., Wang, L., van den Hengel, A., 2009. Positive Semidefinite Metric Learning with Boosting. NIPS.

[19] Shental, N., Weinshall, D., 2003. Learning distance functions using equivalence relations. In: ICML. pp. 11–18.

[20] Timm, N., 2002. Applied multivariate analysis. Springer Verlag.

[21] Turpin, A., Scholer, F., 2006. User performance versus precision measures for simple search tasks. In: SIGIR. ACM, p. 18.

[22] Weinberger, K., Saul, L., 2009. Distance metric learning for large margin nearest neighbor classification. JLMR 10, 207–244.

[23] Xing, E., Ng, A., Jordan, M., Russell, S., 2003. Distance metric learning with application to clustering with side-information. NIPS, 521–528.

[24] Ying, Y., Huang, K., Campbell, C., 2009. Sparse Metric Learning via Smooth Optimization. NIPS, 521–528.

A. Proof of Theorem 4.1

Proof. Suppose $\mathcal{F} = \{(L_\mathcal{X}, L_\mathcal{Y}) \mid L_\mathcal{X} = \left(l_1^\mathcal{X}, \cdots, l_k^\mathcal{X}\right), L_\mathcal{Y} = \left(l_1^\mathcal{Y}, \cdots, l_k^\mathcal{X}\right), L_\mathcal{X}^\top L_\mathcal{X} = I_{k\times k}, L_\mathcal{Y}^\top L_\mathcal{Y} = I_{k\times k}\}$, where $\forall i, 0 \leqslant i \leqslant k$, $l_i^\mathcal{X}$ and $l_i^\mathcal{Y}$ are the i^{th} column vectors of $L_\mathcal{X}$ and $L_\mathcal{Y}$, respectively. Under the norm defined as $\|(L_\mathcal{X}, L_\mathcal{Y})\|^2 = \|L_\mathcal{X}\|_F^2 + \|L_\mathcal{Y}\|_F^2$, where $\|\cdot\|_F$ is Frobenius norm, it is easy to see that $\mathcal{F}$ is compact, and the objective function (2)(or (1)) is continuous. Therefore, there exist maximum and minimum.

Objective function (2) can be re-written as $\sum_{j=1}^k {l_j^\mathcal{Y}}^\top (M_S - M_D) l_j^\mathcal{X}$, and $\forall j, {l_j^\mathcal{Y}}^\top (M_S - M_D) l_j^\mathcal{X} = \langle l_j^\mathcal{Y}, (M_S - M_D) l_j^\mathcal{X}\rangle$. Since $M_S - M_D = \sum_{i=1}^p \lambda_i u_i v_i^\top$, we have

$$\begin{aligned}
&\langle l_j^\mathcal{Y}, (M_S - M_D) l_j^\mathcal{X}\rangle = \langle l_j^\mathcal{Y}, \sum_{i=1}^p \lambda_i u_i v_i^\top l_j^\mathcal{X}\rangle = \sum_{i=1}^p \lambda_i \langle u_i^\top l_j^\mathcal{Y}, v_i^\top l_j^\mathcal{X}\rangle \leqslant \sum_{i=1}^p \lambda_i |\langle u_i, l_j^\mathcal{Y}\rangle||\langle v_i, l_j^\mathcal{X}\rangle| \\
&= \lambda_k \sum_{i=1}^p |\langle u_i, l_j^\mathcal{Y}\rangle||\langle v_i, l_j^\mathcal{X}\rangle| + \sum_{i=1}^k (\lambda_i - \lambda_k)|\langle u_i, l_j^\mathcal{Y}\rangle||\langle v_i, l_j^\mathcal{X}\rangle| + \sum_{i=k+1}^p (\lambda_i - \lambda_k)|\langle u_i, l_j^\mathcal{Y}\rangle||\langle v_i, l_j^\mathcal{X}\rangle| \\
&\leqslant \lambda_k \sum_{i=1}^p |\langle u_i, l_j^\mathcal{Y}\rangle||\langle v_i, l_j^\mathcal{X}\rangle| + \sum_{i=1}^k (\lambda_i - \lambda_k)|\langle u_i, l_j^\mathcal{Y}\rangle||\langle v_i, l_j^\mathcal{X}\rangle|
\end{aligned}$$

Since $\|l_j^\mathcal{X}\| = \|l_j^\mathcal{Y}\| = 1$ and $\{u_i\}_{i=1}^p$ and $\{v_i\}_{i=1}^p$ are orthonormal, we have $\sum_{i=1}^p |\langle u_i, l_j^\mathcal{Y}\rangle||\langle v_i, l_j^\mathcal{X}\rangle| \leqslant \left[(\sum_{i=1}^p \langle u_i, l_j^\mathcal{Y}\rangle^2)(\sum_{i=1}^p \langle v_i, l_j^\mathcal{X}\rangle^2)\right]^{\frac{1}{2}} \leqslant \|l_j^\mathcal{X}\| \cdot \|l_j^\mathcal{Y}\| \leqslant 1$. Thus, we know $\langle l_j^\mathcal{Y}, (M_S - M_D) l_j^\mathcal{X}\rangle \leqslant \lambda_k + \sum_{i=1}^k (\lambda_i - \lambda_k)|\langle u_i, l_j^\mathcal{Y}\rangle||\langle v_i, l_j^\mathcal{X}\rangle|$. By taking summation on both sides, we obtain

$$\sum_{j=1}^k \langle l_j^\mathcal{Y}, (M_S - M_D) l_j^\mathcal{X}\rangle \leqslant k\lambda_k + \sum_{i=1}^k (\lambda_i - \lambda_k)(\sum_{j=1}^k |\langle u_i, l_j^\mathcal{Y}\rangle||\langle v_i, l_j^\mathcal{X}\rangle|).$$

From this inequality, we obtain

$$\sum_{i=1}^k \lambda_i - \sum_{j=1}^k \langle l_j^\mathcal{Y}, (M_S - M_D) l_j^\mathcal{X}\rangle \geqslant \sum_{i=1}^k (\lambda_i - \lambda_k)(1 - \sum_{j=1}^k |\langle u_i, l_j^\mathcal{Y}\rangle||\langle v_i, l_j^\mathcal{X}\rangle|)$$

Since $L_\mathcal{X}^\top L_\mathcal{X} = I_{k\times k}$ and $L_\mathcal{Y}^\top L_\mathcal{Y} = I_{k\times k}$, we have

$$\sum_{j=1}^k |\langle u_i, l_j^\mathcal{Y}\rangle||\langle v_i, l_j^\mathcal{X}\rangle| \leqslant \left[(\sum_{j=1}^k \langle u_i, l_j^\mathcal{Y}\rangle^2)(\sum_{j=1}^k \langle v_i, l_j^\mathcal{X}\rangle^2)\right]^{\frac{1}{2}} \leqslant \|u_i\| \cdot \|v_i\| \leqslant 1.$$

Thus, $\sum_{j=1}^k \langle l_j^\mathcal{Y}, (M_S - M_D) l_j^\mathcal{X}\rangle \leqslant \sum_{j=1}^k \lambda_j$.

Particularly, letting $l_j^\mathcal{Y} = u_j$ and $l_j^\mathcal{X} = v_j$, we can obtain the global maximum. □