Manifold Alignment Preserving Global Geometry

Chang Wang
IBM Watson Research Center
Yorktown Heights, NY
chwang@cs.umass.edu

Sridhar Mahadevan
Department of Computer Science
University of Massachusetts, Amherst
mahadeva@cs.umass.edu

September 5, 2012

Abstract

This paper proposes a novel algorithm for manifold alignment preserving global geometry. This approach constructs mapping functions that project data instances from different input domains to a new lower-dimensional space, simultaneously matching the instances in correspondence and preserving global distances between instances within the original domains. In contrast to previous approaches, which are largely based on preserving local geometry, the proposed approach is suited to applications where the global manifold geometry needs to be respected. We evaluate the effectiveness of our algorithm for transfer learning in two real-world cross-lingual information retrieval tasks.

1 Introduction

Knowledge transfer is becoming increasingly popular in machine learning and data mining [10, 13]. This area draws inspiration from the observation that people can often apply knowledge learned previously to new problems. Some previous work in transfer assumes the training data and test data are originally represented in the same space. However, many real-world applications like cross-lingual information retrieval [4], or matching words and pictures [1], require transfer of knowledge across domains defined by different features. A key step in addressing such transfer learning problems is to find a common underlying latent space shared by all input high-dimensional data sets that may be defined by different features.

Manifold alignment [5, 9, 15] provides a geometric framework to construct such a latent space. The basic idea of manifold alignment is to map all input data sets to a new space preserving the local geometry (neighborhood relationship) of each data set and matching instances in correspondence. This framework makes use of unlabeled data instances, and can be consequently highly effective when the given correspondence information is limited. In the new space, all input domains are defined by the same features, so manifold alignment can be combined with a variety of existing transfer learning approaches [10, 13] to solve real-world knowledge transfer challenges. Manifold alignment can be done at two levels: instance-level and feature-level. In text mining, examples of instances can be documents in English, Arabic, etc; examples of features can be English words/topics, Arabic words/topics, etc. Work on instance-level alignment, such as [5], computes nonlinear embeddings for alignment, but such an alignment result is defined only on known instances, and difficult to generalize to new instances. Feature-level alignment [15] builds mappings between features, and is more suited for knowledge transfer applications than instance-level alignment. Feature-level alignment can be accomplished by computing "linear" mapping functions, where the mappings can be easily generalized to new instances and provide a "dictionary" representing direct mappings between features in different spaces.

Many existing approaches to manifold alignment are designed to only preserve local geometries of the input manifolds. This objective is not desirable in many applications where the global geometries of the input data sets also need to be respected. One such example is from text mining. Documents in different languages can be aligned in a new space, where direct comparison and knowledge transfer between documents (in different languages) is possible. Local geometry preserving manifold alignment [5, 15] does not prevent distinct documents in the original space from being neighbors in the new space (it only encourages similar documents in the original space to be neighbors in the new space). This could lead to poor performance in some tasks, and needs to be corrected. In some other applications, the distance between instances also provides us with valuable information. For example, in a robot navigation problem, we may be given distances between locations recorded by different sensors, which are represented in distinct high-dimensional feature spaces. We want to align these locations based on a partial correspondence, where we also want to preserve the pairwise distance score. Clearly, manifold alignment based on local geometry may not be sufficient for such tasks.

To address the problems mentioned above, we describe a novel framework that constructs functions mapping data instances from different high dimensional data sets to a new lower dimensional space, simultaneously matching the instances in correspondence and preserving pairwise distances between instances within the original data set. Our algorithm has several other added benefits. For example, its solution involves computing the eigenvectors associated with the largest eigenvalues, which are more stable numerically than computing the smallest eigenvectors used by many other manifold alignment methods. It also has fewer parameters that need to be specified. The effectiveness of our algorithm is demonstrated and validated in two real-world cross-lingual information retrieval tasks.

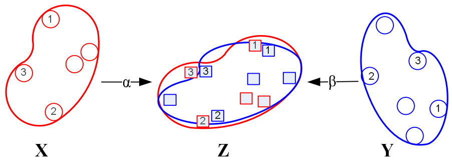

Figure 1: This figure illustrates global distance preserving alignment. X and Y are two input data sets. Three corresponding pairs are given: red i corresponds to blue i for $i \in [1, 3]$. α and β are mapping functions that we want to construct. They project instances from X and Y to a new space Z, where instances in correspondence are projected near each other and pairwise distance within each input set is also respected.

2 Theoretical Analysis

2.1 High Level Explanation

We begin with a brief review of manifold alignment preserving local geometry. Given two data sets X, Y along with l additional pairwise correspondences between a subset of the training instances, local geometry preserving manifold alignment computes the mapping results of x_i and y_j to minimize the following cost function:

$$C(f,g) = \mu \sum_{i,j} (f_i - g_j)^2 W^{i,j} + 0.5 \sum_{i,j} (f_i - f_j)^2 W_x^{i,j} + 0.5 \sum_{i,j} (g_i - g_j)^2 W_y^{i,j},$$

where f_i is the embedding of x_i, g_j is the embedding of y_j, $W^{i,j}$ represents the correspondence between x_i and y_j, $W_x^{i,j}$ is the similarity of x_i and x_j, $W_y^{i,j}$ is the similarity of y_i and y_j, and μ is the weight of the first term. The first term penalizes the differences between X and Y in terms of the embeddings of the corresponding instances. The second and third terms encourage the neighborhood relationship (local geometry) within X and Y to be preserved. There are two types of solutions to this problem: either instance-level [5], when there is no constraint on the mapping functions; or feature-level [15], when the mapping functions are linear.

It can be verified that the optimal (in terms of the above metric) instance-level solution is given by Laplacian eigenmaps [2] on a graph Laplacian matrix modeling the joint manifold that involves X, Y and the correspondence information, whereas the optimal feature-level solution is given by locality preserving projections (LPP) [6] on the same graph Laplacian matrix. As discussed in the introduction, preserving neighborhood relationship may not be sufficient for many applications, like text mining. To solve this problem, we propose a novel framework for manifold alignment, simultaneously matching corresponding instances and preserving global pairwise distances. Our approach uses a distance matrix $\mathcal{D}$ rather than a Laplacian matrix to represent the joint manifold. Our contributions are two-fold: (a) our approach provides a way to construct a distance matrix to model the joint manifold; (b) it enables learning a mapping function for each input dataset (treated as a manifold), such that the mapping functions can work together to project the input manifolds to the same latent space preserving global geometry of each manifold. Some ideas used in (b) are based on MDS/ISOMAP [12] and Isometric projections [3]. Similar to local geometry preserving approaches, there are two solutions to this problem: instance-level and feature-level. In this paper, we focus the latter, which is technically more challenging. The instance-level alignment can be solved in a similar manner. The high level idea is illustrated in Figure 1.

2.2 Notation

Data sets and correspondences:
$X = [x_1 \cdots x_m]$ is a $p \times m$ matrix, where x_i is defined by p features. X represents one high-dimensional data set. $Y = [y_1 \cdots y_n]$ is a $q \times n$ matrix, where y_i is defined by q features. Y represents another high-dimensional data set. The correspondence between X and Y is given as follows: $x_{a_i} \longleftrightarrow y_{b_i}$, where $i \in [1, l]$, $a_i \in [1, m]$ and $b_i \in [1, n]$. Here, the correspondence can be many to many correspondence.
Matrices for re-scaling factor computation:
D_a is a $l \times l$ matrix, where $D_a(i, j)$ is the distance between x_{a_i} and x_{a_j}. D_b is a $l \times l$ matrix, where $D_b(i, j)$ is the distance between y_{b_i} and y_{b_j}.
Distance matrices modeling the joint graph:
$D_{x,x}$ is an $m \times m$ matrix, where $D_{x,x}(i, j)$ is the distance between x_i and x_j. $D_{x,y} = D^T_{y,x}$ is an $m \times n$ matrix, where $D_{x,y}(i, j)$ represents the distance between x_i and y_j. $D_{y,y}$ is an $n \times n$ matrix, where $D_{y,y}(i, j)$ is the distance between y_i and y_j. $\mathcal{D} = \begin{pmatrix} D_{x,x} & D_{x,y} \\ D_{y,x} & D_{y,y} \end{pmatrix}$ is a $(m + n) \times (m + n)$ matrix, modeling a joint graph used in our algorithm.
Mapping functions and distances:
We construct mapping functions α and β to map X and Y to the same d-dimensional space. α is a $p \times d$ matrix, β is a $q \times d$ matrix. In this paper, $\|.\|_2$ represents Frobenius norm, $tr(.)$ represents trace. *T*he τ operator [12] converts distances to inner products, which uniquely characterize the geometry of the data. Given an $m \times m$ distance matrix D, where $D_{i,j}$ represents the distance between instance i and j, $\tau(D) = -HSH/2$. Here, $S_{i,j} = D^2_{i,j}$, $H_{i,j} = \pi_{i,j} - 1/m$ and $\pi_{i,j} = 1$ when $i = j$; 0, otherwise.

2.3 The Problem

We assume the $(m+n) \times (m+n)$ distance matrix $\mathcal{D}$, representing the pairwise distance between any two instances from $\{x_1, \cdots, x_m, y_1, \cdots, y_n\}$, is already given (we will discuss how to construct $\mathcal{D}$ later). The τ *operator* converts distances to inner products, which uniquely characterize the geometry of the data. To preserve global geometry, we define the cost function that needs to be minimized as follows:

$$C(\alpha, \beta, k) = \|\tau(\mathcal{D}) - \tau(\mathcal{D}_{X,Y,\alpha,\beta,k})\|^2_2 = \|\tau(\mathcal{D}) - k \begin{bmatrix} \alpha^T X, & \beta^T Y \end{bmatrix}^T \begin{bmatrix} \alpha^T X, & \beta^T Y \end{bmatrix} \|^2_2,$$

where α, β and k are to be determined: α is a $d \times p$ matrix, β is a $d \times q$ matrix, k is a positive number to rescale mapping functions.

2.4 Construct $\mathcal{D}$ to Represent the Joint Manifold

When data sets X and Y are given, $D_{x,x}$ and $D_{y,y}$ are easily computed using the geodesic distance measure. However, the scales of $D_{x,x}$ and $D_{y,y}$ could be quite different. To create a joint manifold of both X and Y, we need to learn an optimal rescale factor η such that $D_{x,x}$ and $\eta D_{y,y}$ are rescaled to the same space. To compute η, we first create distance matrices D_a, D_b using the instances in correspondence. Obviously D_a and D_b are both $l \times l$ matrices. The algorithm to compute η is given in Theorem 1.

Theorem 1: Given two $l \times l$ matrices D_a and D_b, the solution to η that minimizes $\|D_a - \eta D_b\|_2^2$ is given by $\eta = tr(D_b^T D_a)/tr(D_b^T D_b)$
Proof:

$$\|D_a - \eta D_b\|_2^2 = tr(D_a^T D_a) - 2\eta tr(D_b^T D_a) + \eta^2 tr(D_b^T D_b).$$

$tr(D_a^T D_a)$ is constant, so $\arg_\eta \min \|D_a - \eta D_b\|_2^2 = \arg_\eta \min \eta^2 tr(D_b^T D_b) - 2\eta tr(D_b^T D_a)$.

Differentiating $\eta^2 tr(D_b^T D_b) - 2\eta tr(D_b^T D_a)$ with respect to η, we have $\eta = tr(D_b^T D_a)/tr(D_b^T D_b)$.
□

To construct a distance matrix $\mathcal{D}$ representing the joint manifold, we need to compute distances between instances across datasets. We use $D_{x,x}$, $D_{y,y}$ and the correspondence information to compute these distances. We know $D_{x,x}$ and $D_{y,y}$ model the distance between instances within each given data set. The corresponding pairs can then be treated as "bridges" to connect the two data sets. For any pair (x_i and y_j), we compute the distances between them through all possible "bridges", and set $D_{x,y}(i,j)$ to be the minimum of them. i.e.

$$D_{x,y}(i,j) = \min_{u \in [1,l]} (D_{x,x}(x_i, x_{a_u}) + D_{y,y}(y_j, y_{b_u})).$$

In the approach shown above, we provide one way to compute the distance matrix $\mathcal{D}$ using geodesic distance. Depending on the application, we can also use other approaches to create $\mathcal{D}$. For example, $\mathcal{D}$ could be constructed using Euclidean distance.

2.5 Find Correspondence Across Data Sets

Given X, Y, and the correspondence information, we want to learn mapping functions α for X, β for Y and rescale parameter k, such that $C(\alpha, \beta, k)$ is minimized. The optimal solution will encourage the corresponding instances to be mapped to similar locations in the new space, and the pairwise distance between instances within each set to be respected. To remove the arbitrary scalings, we add one more constraint to guarantee the generated lower dimensional data is sphered, :

$$\begin{bmatrix} \alpha^T X & \beta^T Y \end{bmatrix} \begin{bmatrix} X^T \alpha \\ Y^T \beta \end{bmatrix} = I_d.$$

Theorem 2: Let $Z = \begin{bmatrix} X & 0 \\ 0 & Y \end{bmatrix}$. Then, the eigenvectors corresponding to the d maximum eigenvalues of $Z\tau(\mathcal{D})Z^T\gamma = \lambda ZZ^T\gamma$ provide optimal mappings to

minimize $C(\alpha, \beta, k)$**.**
Proof:

$$C(\alpha, \beta, k) = \|\tau(\mathcal{D}) - k \cdot \left[\begin{array}{cc} X^T & 0 \\ 0 & Y^T \end{array}\right] \left[\begin{array}{c} \alpha \\ \beta \end{array}\right] \left[\begin{array}{c} \alpha^T \beta^T \end{array}\right] \left[\begin{array}{cc} X & 0 \\ 0 & Y \end{array}\right] \|_2^2.$$

Let $f = \left[\begin{array}{c} \alpha \\ \beta \end{array}\right]$, then we have

$$C(\alpha, \beta, k) = \|\tau(\mathcal{D}) - k \cdot Z^T f f^T Z\|_2^2 = tr((\tau(\mathcal{D}) - k \cdot Z^T f f^T Z)(\tau(\mathcal{D}) - k \cdot Z^T f f^T Z)^T)$$
$$= tr(\tau(\mathcal{D})\tau(\mathcal{D})^T) - k \cdot tr(Z^T f f^T Z \tau(\mathcal{D})^T) - k \cdot tr(\tau(\mathcal{D}) Z^T f f^T Z) + k^2 \cdot tr(Z^T f f^T Z Z^T f f^T Z).$$

Given the property that $tr(AB) = tr(BA)$, we have

$$C(\alpha, \beta, k) = tr(\tau(\mathcal{D})\tau(\mathcal{D})^T) - 2k \cdot tr(f^T Z \tau(\mathcal{D}) Z^T f) + k^2 \cdot tr(I_d).$$

Differentiating $C(\alpha, \beta, k)$ with respect to k, we have

$$2 \cdot tr(f^T Z \tau(\mathcal{D}) Z^T f) = 2k \cdot d.$$

This implies $k = tr(f^T Z \tau(\mathcal{D}) Z^T f)/d$. So

$$C(\alpha, \beta, k) = tr(\tau(\mathcal{D})\tau(\mathcal{D})^T) - 2/d \cdot (tr(f^T Z \tau(\mathcal{D}) Z^T f))^2 + 1/d \cdot (tr(f^T Z \tau(\mathcal{D}) Z^T f))^2.$$

Since both $tr(\tau(\mathcal{D})\tau(\mathcal{D})^T)$ and d are constant, we have

$$\arg\min C(\alpha, \beta, k) = \arg\max(tr(f^T Z \tau(\mathcal{D}) Z^T f))^2.$$

It can be verified that $f^T Z \tau(\mathcal{D}) Z^T f$ is positive semi-definite, so $tr(f^T Z \tau(\mathcal{D}) Z^T f) \geq 0$.

$$\text{Then, } \arg\min C(\alpha, \beta, k) = \arg\max tr(f^T Z \tau(\mathcal{D}) Z^T f).$$

It can be shown that the solution to

$$\arg\max tr(f^T Z \tau(\mathcal{D}) Z^T f), \quad s.t. \quad f^T Z Z^T f = I_d.$$

is given by the eigenvectors corresponding to the d largest eigenvalues of $Z \tau(\mathcal{D}) Z^T \gamma = \lambda Z Z^T \gamma$. □

3 The Algorithms

3.1 The Algorithmic Procedure

Notation used in this section is defined in the previous section. Given two high dimensional data sets X, Y along with additional pairwise correspondences between a subset of the instances, the algorithmic procedure is as follows:

1. **Rescale data set** Y**:** $Y = \eta Y$, where $\eta = tr(D_b^T D_a)/tr(D_b^T D_b)$.

2. **Construct distance matrix $\mathcal{D}$, modeling the joint graph:**

$$\mathcal{D} = \begin{pmatrix} D_{x,x} & D_{x,y} \\ D_{y,x} & D_{y,y} \end{pmatrix}, \textbf{ where } D_{y,x}(j,i) = D_{x,y}(i,j) = \min_{u\in[1,l]}(D_{x,x}(x_i, x_{a_u}) + D_{y,y}(y_j, y_{b_u})).$$

3. **Find the correspondence between X and Y:**

   Compute the eigenvectors $[\gamma_1, \cdots, \gamma_d]$ corresponding to d maximum eigenvalues of

$$\begin{bmatrix} X & 0 \\ 0 & Y \end{bmatrix} \tau(\mathcal{D}) \begin{bmatrix} X^T & 0 \\ 0 & Y^T \end{bmatrix} \gamma = \lambda \begin{bmatrix} X & 0 \\ 0 & Y \end{bmatrix} \begin{bmatrix} X^T & 0 \\ 0 & Y^T \end{bmatrix} \gamma.$$

4. **Construct α and β to map X and Y to the same d-dimensional space:**

   The d-dimensional representations of X and Y are columns of $\alpha^T X$ and $\beta^T Y$, where

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = [\gamma_1, \cdots, \gamma_d].$$

3.2 Added Benefits

The cost function for local geometry preserving manifold alignment shown in the previous section uses a scalar real-valued parameter μ to balance the conflicting objectives of matching corresponding instances and preserving manifold topologies. μ is usually manually specified by trial and error. In the new approach, μ is not needed. The usage of μ is replaced by setting the distance between corresponding instances across domains to 0. In contrast to local geometry preserving alignment approaches that use eigenvectors corresponding to the smallest eigenvalues, the new approach is based on eigenvectors corresponding to the largest eigenvalues. Numerically, eigensolvers are less stable in computing the smallest eigenvectors than when they compute the largest eigenvectors. In this paper, we illustrate our approach using the linear feature-level framework, but it is straightforward to generalize it to the non-linear case: replace $\alpha^T X$ with $\mathcal{A}$ and $\beta^T Y$ with $\mathcal{B}$ in the cost function. The solution is then given by the maximum eigenvalue solution to $\tau(D)\gamma = \lambda\gamma$.

3.3 Time Complexity Analysis

Given two input domains X with m instances and Y with n instances, our approach first needs to construct a local neighborhood graph, which is an $O(N^2)$ problem assuming $N = m + n$. Then it requires the shortest paths between every pair of points resulting in $O(N^2 \log N)$ computation. The step of eigenvalue decomposition takes $O(P^3)$ time when all eigenvectors are needed, where P is the total number of features from the input domains. Similar to the other manifold learning approaches, techniques like Nÿstrom approximation [11] are needed to address the computational challenges when N is very large. Recent work [11] shows that a problem involving 18M instances represented over 2,304 features can be handled under a similar setting as ours.

4 Experimental Results

In the first experiment, we compare our global geometry preserving approach to local geometry preserving approaches at finding both instance-level [5] and feature-level [15] alignments using parallel data in two languages: English and Arabic. In the second experiment, we use three input datasets, since our approach can be generalized to handle more than two domains. This ability to process multiple datasets is useful for the situations when we have knowledge from multiple related sources. We compare our approach against local geometry preserving manifold alignment and the other state of the art approaches, including Canonical Correlation Analysis (CCA) [7], Affine matching based alignment [9] and Procrustes alignment [14]. In this paper, we use kNN ($k = 10$) to build adjacency graphs.

4.1 English Arabic Cross-Lingual Retrieval

The first experiment is to find exact correspondences between documents in different languages. This application is useful, since it allows users to input queries in their native language and retrieve results in a foreign language. The data set used below was originally studied in [4]. It includes two collections: one in English and one in Arabic (manually translated). The features are constructed by the language model. The topical structure of each collection is treated as a manifold over documents. Each document is an instance sampled from the manifold. To learn correspondences between the two collections, we are also given some training correspondences between documents that are exact translations of each other. The task is to find the most similar document in the other corpus for each English or Arabic document in the untranslated set. In this experiment, each of the two document collections has 2,119 documents. We tried two different settings: (1) Correspondences between 25% of them were given; (2) Correspondences between 10% of them were given. The remaining instances were used in both training (as unlabeled data) and testing. Our testing scheme is as follows: for each given English document, we retrieve its top k most similar Arabic documents. The probability that the true match is among the top k documents is used to show the goodness of the method. We use this data to compare our framework with the local geometry preserving framework. Both frameworks map the data to a 100 dimensional latent space ($d = 100$), where documents in different languages can be directly compared. A baseline approach was also tested. The baseline method is as follows: assume that we have l correspondences in the training set, then document x is represented by a vector V with length l, where $V(i)$ is the similarity of x and the i^{th} document in the training correspondences. The baseline method maps the documents from different collections to the same embedding space R^l.

When 25% instances are used as training correspondences, the results are in Figure 2. In our global geometry preserving approach, for each given English document, if we retrieve the most relevant Arabic document, then the true match has a 35% probability of being retrieved. If we retrieve the 10 most similar documents, the probability increases to 80%. For feature-level local geometry preserving manifold alignment [15], the corresponding numbers are 26% and 68%. Instance-level local geometry preserving manifold alignment [5] results in a very poor alignment. One reason for this is

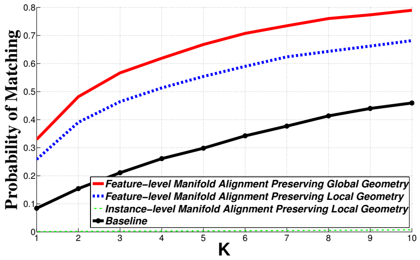

Figure 2: Test on cross-lingual data (25% instances are in the given correspondence).

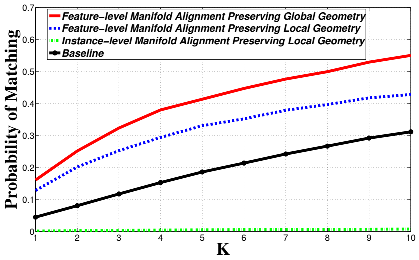

Figure 3: Test on cross-lingual data (10% instances are in the given correspondence).

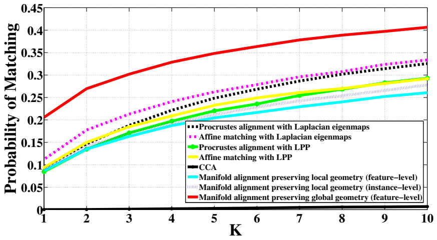

Figure 4: EU parallel corpus data with 1,500 English-Italian-German test triples.

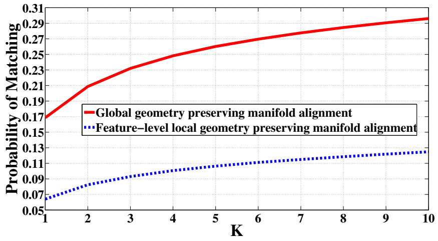

Figure 5: EU parallel corpus data with 69,458 English-Italian test pairs.

that instance-level alignment learns non-linear mapping functions for alignment. Since the mapping function can be any function, it might overfit the training data and does not generalize well to the test data. To verify this, we also examined a case where the training instances lie on the new space and found out that the training instances were perfectly aligned. When 10% instances are used as training correspondences, similar results are reported in Figure 3.

4.2 European Parliament Proceedings Test

Eight approaches are tested in this experiment. Three of them are instance-level approaches: Procrustes alignment with Laplacian eigenmaps, Affine matching with Laplacian eigenmaps, and instance-level manifold alignment preserving local geometry. The other five are feature-level approaches: Procrustes alignment with LPP, Affine matching with LPP, CCA, feature-level manifold alignment preserving local geometry and our feature-level manifold alignment preserving global geometry. Procrustes alignment and Affine matching can only handle pairwise alignment, so when we align two collections the third collection is not taken into consideration. The other manifold alignment approaches and CCA align all input data simultaneously.

In this experiment, we make use of the proceedings of European Parliament [8], dating from 04/1996 to 10/2009. The corpus includes versions in 11 European languages. Altogether, the corpus comprises of about 55 million words for each language. The data for our experiment comes from English, Italian and German collections. The dataset has many files, each file contains the utterances of one speaker in turn. We treat an utterance as a document. We filtered out stop words, and extracted English-Italian-German document triples where all three documents have at least 75 words. This resulted in 70,458 document triples. We then represented each English document with the most commonly used 2,500 English words, each Italian document with the most commonly used 2,500 Italian words, and each German document with the most commonly used 2,500 German words. The documents were represented as bags of words, and no tag information was included. The topical structure of each collection can be thought as a manifold over documents. Each document is a sample from the manifold.

Instance-level manifold alignment cannot process a very large collection since it needs to do an eigenvalue decomposition of an $(m_1 + m_2 + m_3)\times(m_1 + m_2 + m_3)$ matrix, where m_i represents the number of examples in the i^{th} input dataset. Approaches based on Laplacian eigenmaps suffer from a similar problem. In this experiment, we use a small subset of the whole dataset to test all seven approaches. $1,000$ document triples were used as corresponding triples in training and $1,500$ other document triples were used as unlabeled documents for both training and testing, i.e. $p_1 = p_2 = p_3 = 2,500$, $m_1 = m_2 = m_3 = 2,500$. $x_1^i \longleftrightarrow x_2^i \longleftrightarrow x_3^i$ for $i \in [1, 1000]$. Similarity matrices W_1, W_2 and W_3 were all $2,500 \times 2,500$ adjacency matrices constructed by nearest neighbor approach, where $k = 10$. To use Procrustes alignment and Affine matching, we ran a pre-processing step with Laplacian eigenmaps and LPP to project the data to a $d = 100$ dimensional space. In CCA and feature-level manifold alignment, d is also 100. The procedure for the test is quite similar to the previous test. The only difference is that we consider three different scenarios in the new setting: English $\leftrightarrow$ Italian, English $\leftrightarrow$ German and Italian $\leftrightarrow$ German. Figure 4 summarizes the average performance of these three scenarios.

Our new global preserving approach outperforms all the other approaches. Given a document in one language, it has a 21% probability of finding the true match if we retrieve the most similar document in another language. If we retrieve 10 most similar documents, the probability of finding the true match increases to more than 40%. Our approach results in three mapping functions to construct the new latent space: $\mathcal{F}_1$ (for English), $\mathcal{F}_2$ (for Italian) and $\mathcal{F}_3$ (for German). These three mapping functions project documents from the original English/Italian/German spaces to the same 100 dimensional space. Each column of $\mathcal{F}_i$ is a $2,500 \times 1$ vector. Each entry on this vector corresponds to a word. To illustrate how the alignment is achieved using our approach, we show the words that make the largest contributions to 2 selected corresponding columns from $\mathcal{F}_1$, $\mathcal{F}_2$ and $\mathcal{F}_3$ in Tabel 1. From this table, we can see that the mapping functions can automatically project the documents with similar contents but in different languages to similar locations in the new space.

The second result shown in Figure 4 is that all three instance-level approaches outperform the corresponding feature-level approaches. There are two possible reasons for this. One is that feature-level approaches use linear mapping functions to compute lower dimensional embedding or alignment. Instance-level approaches are based on non-linear mapping functions, which are more powerful than linear mappings. Another

	Top Terms
English 1	policy gentlemen foreign committee behalf security eu defence rights development
English 2	programme administrative turkey process answer ministers adoption conclusions created price
Italian 1	politica chiusa estera nome sicurezza sapere modifica chiarezza dobbiamo diritti
Italian 2	programma turchia processo paese chiusa disoccupazione cambiamenti obiettivi milioni potra
German 1	politik ausschusses gemeinsame bereich man namen eu menschenrechte herren insgesamt
German 2	programm turkei prozess meines programms britischen linie aufmerksam menschenrechte zweitens

Table 1: 2 selected mapping functions in English, Italian and German.

reason is that the number of training samples in this experiment is smaller than the number of features. So the training data is not sufficient to determine the mapping functions for feature-level approaches. Feature-level approaches have two advantages over instance-level approaches. Firstly, feature-level approaches learn feature feature correlations, so they can be applied to a very large dataset and directly generalize to new test data. Secondly, their chance of getting into overfitting problems is much lower than instance-level approaches due to the “linear” constraint on mapping functions.

The third result is that CCA does a very poor job in aligning the test documents. CCA can be shown as a special case of feature-level manifold alignment preserving local geometry when manifold topology is not respected. When the training data is limited, CCA has a large chance of overfitting the given correspondences. Feature-level manifold alignment does not suffer from this problem, since the manifold topology also needs to be respected in the alignment.

Our new approach and the feature-level local geometry preserving manifold alignment algorithm can handle problems at a large scale. In our second setting, we apply these two approaches to process all 69,458 test English-Italian document pairs represented over the most popular 1,000 English/Italian words. The results are summarized in Figure 5. For any English document, if we retrieve the most similar Italian document, the new approach has a 17% chance of getting the true match. If we retrieve 10 most similar Italian documents, the new approach has a 30% probability of getting the true match. Feature-level local geometry preserving approach performs much worse than the new approach. This shows that global geometry preservation is quite important for applications like text mining. This test under the second setting is in fact very hard, since we have thousands of features, roughly 70,000 documents in each input dataset but only 1,000 given corresponding pairs. In contrast to most approaches in cross-lingual knowledge transfer, we are not using any specialized pre-processing technique from information retrieval to tune our framework to this task.

5 Conclusions

This paper proposes a novel framework for manifold alignment, which maps data instances from different high dimensional data sets to a new lower dimensional space, simultaneously matching the instances in correspondence and preserving global distances between instances within the original data set. Unlike previous approaches based on local geometry preservation, the proposed approach is better suited to applications where the global geometry of manifold needs to be respected like cross-lingual retrieval. Our algorithm can also be used as a knowledge transfer framework for transfer learning, providing direct feature-feature translation across domains.

References

[1] K. Barnard, P. Duygulu, D. Forsyth, N. Freitas, D. Blei, and M. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, pages 1107–1135, 2003.

[2] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396, 2003.

[3] D. Cai, X. He, and J. Han. Isometric projections. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 2006–2747, 2007.

[4] F. Diaz and D. Metzler. Pseudo-aligned multilingual corpora. In *Proceedings of The International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2727–2732, 2007.

[5] J. Ham, D. Lee, and L. Saul. Semisupervised alignment of manifolds. In *Proceddings of the International Workshop on Artificial Intelligence and Statistics*, pages 120–127, 2005.

[6] X. He and P. Niyogi. Locality preserving projections. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 2003.

[7] H. Hotelling. Relations between two sets of variates. *Biometrika*, 10:321–377, 1936.

[8] P. Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT Summit*, 2005.

[9] S. Lafon, Y. Keller, and R. Coifman. Data fusion and multicue data matching by diffusion maps. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 28(11):1784–1797, 2006.

[10] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[11] A. Talwalkar, S. Kumar, and H. Rowley. Large scale manifold learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[12] J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[13] L. Torrey and J. Shavlik. *Transfer learning*. Handbook of Research on Machine Learning Applications, IGI Global, 2009.

[14] C. Wang and S. Mahadevan. Manifold alignment using procrustes analysis. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1120–1127, 2008.

[15] C. Wang and S. Mahadevan. Manifold alignment without correspondence. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1273–1278, 2009.