Multiscale Manifold Alignment

Chang Wang and Sridhar Mahadevan

Computer Science Department
University of Massachusetts Amherst
Amherst, Massachusetts 01003
{chwang, mahadeva}@cs.umass.edu

Abstract. We introduce a novel approach to multiscale manifold alignment. Our approach goes beyond the previously studied approaches in that it yields a hierarchical alignment that preserves the local geometry of each manifold and matches the corresponding instances across manifolds at different temporal and spatial scales. The proposed approach is non-parametric, data-driven, and automatically generates multilevel alignments by analyzing the intrinsic (or latent) hierarchical shared structure of the given data sets. We describe and evaluate our approach both theoretically and experimentally, and present results showing useful knowledge transfer in several real-world tasks.

Key words: Transfer Learning, Domain Adaptation, Manifold Alignment, Manifold Learning

1 Introduction

In many situations, we want to adapt the knowledge learned in a source domain for use in a target domain, where the source and target domains may be different but related. This problem arises in a variety of applications in information retrieval, e-commerce, computer vision, and many other areas.

The area of transfer learning in general, and domain adaptation in particular, has recently seen a surge of activity [1–5]. However, one limitation that has not been fully addressed is that most existing domain adaptation approaches assume that the source and target domains are defined by the same features, and the difference between domains primarily arises due to the difference between data distributions. This assumption is not valid in many scenarios such as cross-lingual retrieval, where the source and target domains are represented in different languages and do not share any features. Recently, a new approach to transfer learning called manifold alignment was proposed to address the problem of learning correlations across domains defined by different features. Manifold alignment builds mappings between two or more disparate data sets by aligning their underlying manifolds and provides a geometric framework for knowledge transfer across data sets. More formally, given data sets $X = \{x_1, \cdots, x_m\}$ (from manifold $\mathcal{X}$) and $Y = \{y_1, \cdots, y_n\}$ (from manifold $\mathcal{Y}$) together with a small fraction of samples labeled with known correspondences, we want to find a

correspondence between them. Directly working with the original data instances can be quite difficult, since they are in high dimensional spaces and might be defined by different features. The solution is to map X and Y to a new latent space $\mathcal{Z}$, where instances x_i and y_j can be directly compared.

Existing manifold alignment approaches can be categorized into two types. In two-step alignment approaches, such as diffusion-maps based alignment [6] and Procrustes alignment [7], the first step maps the data sets to low dimensional spaces reflecting their intrinsic geometries using standard dimensionality reduction approaches. Then, a subsequent step eliminates some components (like rotational and scaling components) from one set so that the alignment of two sets can be achieved. In one-step alignment methods, such as semi-supervised alignment [8], semi-definite alignment [9], and manifold projections [10], the embedding projection and alignment steps are combined into one. Semi-supervised alignment first creates a joint manifold representing the union of both manifolds $\mathcal{X}$ and $\mathcal{Y}$. Then it maps the joint manifold to a lower dimensional *latent* space preserving local geometries of both $\mathcal{X}$ and $\mathcal{Y}$, and matching instances in correspondence. Semi-supervised alignment is based on eigenvalue decomposition. Semi-definite alignment solves a similar problem using a semi-definite programming framework. Manifold projections extends semi-supervised alignment by considering many to many correspondences, and assumes that linear mapping functions are used to compute alignments. Manifold projections directly builds connections between features rather than instances and can naturally handle new test instances.

Many real-world data sets exhibit non-trivial regularities at *multiple* levels, which correspond to their underlying intrinsic structure. For example, in the NIPS conference paper data set (www.cs.toronto.edu/∼roweis/data.html), at the most abstract level, the set of all papers can be categorized into two main topics: machine learning and neuroscience. At the next level, the papers can be categorized into a number of areas, such as dimensionality reduction, reinforcement learning, etc. To transfer knowledge across domains taking consideration of their intrinsic multilevel structures, we need to provide ways to do multiscale manifold alignment, which has not been studied yet. We formulate the problem of multiscale alignment using the framework of multiresolution wavelet analysis [11]. Compared to single-level alignment, multiscale alignment automatically generates alignment results at different levels by discovering the shared intrinsic multilevel structures of the given data sets. In contrast to previous "flat" alignment methods, where users need to specify the dimensionality of the new space, the multilevel approach automatically finds alignments of varying dimensionality. In addition to the theoretical analysis of the algorithm, we also evaluate our approach in several real-world domains, including cross-lingual information retrieval and corpora alignment.

The rest of this paper is as follows. In Section 2 we describe the problem and the main algorithm. In Section 3 we provide a theoretical analysis of our approach. We describe some applications and summarize our experimental results in Section 4. Section 5 provides some concluding remarks.

2 Multiscale Manifold Alignment

In this section, we introduce the framework of multiscale alignment. The notation used in this paper is summarized in Figure 1.

$x_i \in R^p$; $X = \{x_1, \cdots, x_m\}$ is a $p \times m$ matrix; $X_l = \{x_1, \cdots, x_l\}$ is a $p \times l$ matrix.
$y_i \in R^q$; $Y = \{y_1, \cdots, y_n\}$ is a $q \times n$ matrix; $Y_l = \{y_1, \cdots, y_l\}$ is a $q \times l$ matrix .
X_l and Y_l are in correspondence: $x_i \in X_l \longleftrightarrow y_i \in Y_l$.
W_x is a similarity matrix, e.g. $W_x^{i,j} = e^{-\frac{||x_i - x_j||^2}{2\sigma^2}}$.
D_x is a full rank diagonal matrix: $D_x^{i,i} = \sum_j W_x^{i,j}$;
$L_x = D_x - W_x$ is the combinatorial Laplacian matrix.
W_y, D_y and L_y are defined similarly.
Ω_i is a diagonal matrix having μ on the top l elements of the diagonal (the other elements are 0);
Ω_1 is an $m \times m$ matrix; Ω_2 and Ω_3^T are $m \times n$ matrices; Ω_4 is an $n \times n$ matrix.
$Z = \begin{pmatrix} X & 0 \\ 0 & Y \end{pmatrix}$ is a $(p+q) \times (m+n)$ matrix.
$D = \begin{pmatrix} D_x & 0 \\ 0 & D_y \end{pmatrix}$ and $L = \begin{pmatrix} L_x + \Omega_1 & -\Omega_2 \\ -\Omega_3 & L_y + \Omega_4 \end{pmatrix}$ are both $(m+n) \times (m+n)$ matrices.
F is a $(p+q) \times r$ matrix, where r is the rank of ZDZ^T and $FF^T = ZDZ^T$. F can be constructed by SVD.
$(\cdot)^+$ represents the Moore-Penrose pseudoinverse.
At level k: α_k is a mapping from any $x \in X$ to a p_k dimensional space: $\alpha_k^T x$ (α_k is a $p \times p_k$ matrix).
At level k: β_k is a mapping from any $y \in Y$ to a p_k dimensional space: $\beta_k^T y$ (β_k is a $q \times p_k$ matrix).

Fig. 1. Notation used in this paper.

2.1 Single-Level Manifold Alignment

We review two approaches to single-level manifold alignment: semi-supervised alignment learns an instance-level alignment by constructing nonlinear embeddings; manifold projections learns a feature-level alignment by constructing linear embedding functions. In both cases, we are given two data sets X, Y along with additional pairwise correspondences.

Semi-supervised alignment [8] finds the best alignment mapping for instances x_i and y_i by minimizing the following cost function:

$$C(f, g) = \mu \sum_{i=1}^{l} (f_i - g_i)^2 + 0.5 \sum_{i,j} (f_i - f_j)^2 W_x^{i,j} + 0.5 \sum_{i,j} (g_i - g_j)^2 W_y^{i,j},$$

where f_i is the embedding of x_i, g_i is the embedding of y_i and μ is the weight of the first term. The first term penalizes the differences between X and Y on the embeddings of the corresponding instances. The second and third terms ensure that the local geometries within X and Y will be preserved. To remove

an arbitrary scaling factor in the embedding, an extra constraint is imposed: $f^T D_x f + g^T D_y g = \gamma^T D \gamma = I$. Then, the d dimensional alignment result is given by

$$\begin{bmatrix} f \\ g \end{bmatrix} = [\gamma_1 \cdots \gamma_d],$$

where $\gamma_1 \cdots \gamma_d$ are eigenvectors of $L\gamma = \lambda D\gamma$ corresponding to the d smallest non-zero eigenvalues.

Manifold projections [10] learns mapping functions α and β for alignment. When the correspondence is given, its cost function is given as: [1]:

$$C(\alpha, \beta) = \mu \sum_{i}^{l} (\alpha^T x_i - \beta^T y_i)^2 + 0.5 \sum_{i,j} (\alpha^T x_i - \alpha^T x_j)^2 W_x^{i,j} + 0.5 \sum_{i,j} (\beta^T y_i - \beta^T y_j)^2 W_y^{i,j}.$$

To remove an arbitrary scaling factor in the embedding, an extra constraint is needed: $\alpha^T X D_x X^T \alpha + \beta^T Y D_y Y^T \beta = \gamma^T Z D Z^T \gamma = I$. Then, the d dimensional mapping function is given by

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

where $\gamma_1 \cdots \gamma_d$ are the eigenvectors of $ZLZ^T\gamma = \lambda ZDZ^T\gamma$ corresponding to the d smallest non-zero eigenvalues. Manifold projections builds connections between features (rather than instances) across manifolds, so it can handle new test instances and makes direct knowledge transfer possible.

Semi-supervised alignment and manifold projections are based on a similar idea. Given samples from two manifolds $\mathcal{X}$ and $\mathcal{Y}$, they first create a joint manifold represented by the Laplacian matrix L. L has the information from both L_x and L_y. Such information models the local geometries of both $\mathcal{X}$ and $\mathcal{Y}$. The submatrices $\Omega_1 - \Omega_4$ in L play a key role in joining the two manifolds. They force the instances in correspondence (from different manifolds) to be neighbors in the joint manifold. The joint manifold is then mapped to a lower dimensional space preserving its local geometry.

2.2 Multiscale Manifold Alignment Problem

Both semi-supervised manifold alignment [8] and manifold projections [10] start with a cost function. It can be shown that the solution to minimize the cost function is also the best solution to achieve alignment of the input manifolds. In this section, we present another way to understand those algorithms and explain how that helps achieve multiscale manifold alignment. Let

$$W_{joint} = \begin{pmatrix} W_x & W_{x,y} \\ W_{x,y}^T & W_y \end{pmatrix},$$

[1] When no correspondence is given or when one instance can match multiple instances in another data set, the loss function can be specified in a more general manner.

where $W^{i,j}_{x,y} = \mu$, if x_i and y_j are in correspondence; 0, otherwise. If we define a diagonal matrix D_{joint} as $D^{i,i}_{joint} = \sum_j W^{i,j}_{joint}$, then the matrix L defined in Figure 1 can also be written as

$$L = D_{joint} - W_{joint}.$$

Obviously, L is the graph Laplacian matrix corresponding to the joint weight matrix W_{joint}, reflecting the joint manifold constructed from two input manifolds and the given corresponding pairs. From the joint manifold, we can learn lower dimensional embedding of each instance using Laplacian eigenmaps [12] resulting in semi-supervised manifold alignment [8], or locality preserving projections [13] resulting in manifold projections [10]. A key problem in manifold alignment that has not been addressed so far is that the dimensionality of the alignment needs to be decided by the users. In previous approaches, this quantity is specified using an arbitrary number. In this paper, we design an algorithm to **simultaneously** find the most appropriate scales (dimensionalities) for alignments and the corresponding alignment results. This approach is based on an intrinsic structure analysis of the joint manifold (represented by L). Note that multiscale manifold alignment does not use eigenvalue decomposition. Rather, it uses a multiresolution method called diffusion wavelets [11]. Given an input dataset, diffusion wavelets (DWT) is able to automatically identify the multilevel intrinsic structure of the data. If the input data is a joint manifold, then those levels will correspond to the most appropriate scales to align the input manifolds.

In this paper, we explain how manifold projections problem can be solved at multiple scales. The same idea can be easily generalized to semi-supervised alignment problem. The multiscale manifold alignment problem is formally defined as follows: given two data sets X, Y along with partial correspondence information $x_i \in X_l \longleftrightarrow y_i \in Y_l$, compute mapping functions $\mathcal{A}_k$ and $\mathcal{B}_k$ at level k that project X and Y to a new space preserving local geometry of each set and matching instances in correspondence. Here $k = 1, \ldots, h$ represents each level of the joint manifold hierarchy.

To apply DWT to multiscale analysis of the given manifold, we need to address the following challenge: DWT can only handle regular eigenvalue decomposition in the form of $A\gamma = \lambda\gamma$, where A is the given matrix, γ is an eigenvector and λ is the corresponding eigenvalue. However, the problem we are interested in is a generalized eigenvalue decomposition: $A\gamma = \lambda B\gamma$, where we have two input matrices A and B. It is non-trivial to convert the latter problem to the former such that two problems have the same solution. We prove three theorems in Section 3 to make it happen.

2.3 An Overview of Diffusion Wavelets

The diffusion wavelets algorithm constructs a compressed representation of the dyadic powers of a (symmetric or non-symmetric) square matrix by representing the associated matrices at each scale not in terms of the original (unit vector)

$\{[\phi_j]_{\phi_0}, [\psi_j]_{\phi_0}\} = \mathcal{DWT}(T, \varepsilon)$
//**INPUT**:
//T: The input matrix.
//ε: Desired precision, which can be set to a small number or simply machine precision.
//**OUTPUT**:
//$[\phi_j]_{\phi_0}$: extended diffusion scaling functions at scale j.
//$[\psi_j]_{\phi_0}$: extended diffusion wavelet functions at scale j.
$\phi_0 = I$;
For $j = 0$ *to* $J - 1$
{
$([\phi_{j+1}]_{\phi_j}, [T^{2^j}]_{\phi_j}^{\phi_{j+1}}) \leftarrow \mathcal{QR}([T^{2^j}]_{\phi_j}^{\phi_j}, \varepsilon)$;
$[\phi_{j+1}]_{\phi_0} = [\phi_{j+1}]_{\phi_j}[\phi_j]_{\phi_0}$;
$[\psi_j]_{\phi_j} \leftarrow \mathcal{QR}(I_{<\phi_j>} - [\phi_{j+1}]_{\phi_j}[\phi_{j+1}]_{\phi_j}^T, \varepsilon)$;
$[\psi_{j+1}]_{\phi_0} = [\psi_{j+1}]_{\phi_j}[\phi_j]_{\phi_0}$;
$[T^{2^{j+1}}]_{\phi_{j+1}}^{\phi_{j+1}} = ([T^{2^j}]_{\phi_j}^{\phi_{j+1}}[\phi_{j+1}]_{\phi_j})^2$;
}

Fig. 2. Diffusion Wavelets constructs multiscale representations of the input matrix at different scales. $\mathcal{QR}$ is a modified QR decomposition. J is the max step number (this is optional, since the algorithm automatically terminates when it reaches a matrix of size 1×1). The notation $[T]_{\phi_a}^{\phi_b}$ denotes matrix T whose column space is represented using basis ϕ_b at scale b, and row space is represented using basis ϕ_a at scale a. The notation $[\phi_b]_{\phi_a}$ denotes basis ϕ_b represented on the basis ϕ_a. At an arbitrary scale j, we have p_j basis functions, and length of each function is l_j. The number of p_j is determined by the intrinsic structure of the given dataset in $\mathcal{QR}$ routine. $[T]_{\phi_a}^{\phi_b}$ is a $p_b \times l_a$ matrix, and $[\phi_b]_{\phi_a}$ is an $l_a \times p_b$ matrix.

basis, but rather using a set of custom generated bases [11]. Figure 2 summarizes the procedure to generate diffusion wavelets. Given a matrix T, the $\mathcal{QR}$ (a modified QR decomposition) subroutine decomposes T into an orthogonal matrix Q and a triangular matrix R such that $T \approx QR$, where $|T_{i,k} - (QR)_{i,k}| < \varepsilon$ for any i and k. Columns in Q are orthonormal basis functions spanning the column space of T at the finest scale. RQ is the new representation of T with respect to the space spanned by the columns of Q (this result is based on the matrix invariant subspace theory). At an arbitrary level j, $\mathcal{DWT}$ learns the basis functions from T^{2^j} using $\mathcal{QR}$. Compared to the number of basis functions spanning T^{2^j}'s original column space, we usually get **fewer** basis functions, since some high frequency information (corresponding to the "noise" at that level) can be filtered out. $\mathcal{DWT}$ then computes $T^{2^{j+1}}$ using the low frequency representation of T^{2^j} and the procedure repeats. This procedure is illustrated in Figure 3.

Running $\mathcal{DWT}$ is equivalent to running a Markov chain on the input data forward in time, integrating the local geometry and therefore revealing the relevant geometric structures of data at different scales. At scale j, the representation of $T^{2^{j+1}}$ is compressed based on the amount of remaining information and the desired precision. Two sets of basis functions are constructed: "scaling" functions

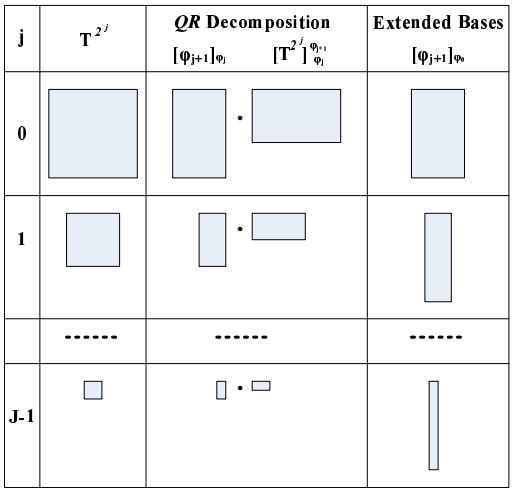

Fig. 3. Construction of Diffusion Wavelets.

span the column space of the input matrix at a given level; "wavelet" functions (not discussed in this paper) span the orthogonal complement of the matrix column space. These terms are motivated by analogy to the regular wavelet transform.

2.4 The Main Algorithm

Given X, X_l, Y, Y_l, using the notation defined in Figure 1, the algorithm is as follows:

1. **Construct a matrix representing the joint manifold:** $T = F^+ZLZ^T(F^T)^+$.
2. **Use diffusion wavelets to explore the intrinsic structure of the joint manifold:**
   $[\phi_k]_{\phi_0} = \mathcal{DWT}(T^+)$, where $\mathcal{DWT}()$ is described in Section 2, $[\phi_k]_{\phi_0}$ are the scaling function bases at level k represented as an $r \times p_k$ matrix, $k = 1, \cdots, h$ represents the level in the joint manifold hierarchy. The value of p_k is determined in $\mathcal{DWT}()$ based on the intrinsic structure of the given dataset.
3. **Compute mapping functions for manifold alignment (at level k):**
   $\begin{bmatrix} \alpha_k \\ \beta_k \end{bmatrix} = (F^T)^+[\phi_k]_{\phi_0}$ is a $(p+q) \times p_k$ matrix.
4. **At level k: apply α_k and β_k to find correspondences between X and Y:**
   For any i and j, $\alpha_k^T x_i$ and $\beta_k^T y_j$ are in the same p_k dimensional space and can be directly compared.

To use the multiscale framework to solve the semi-supervised alignment problem, we need to minimize the cost function $C(f, g)$ instead. This requires making two changes to our main algorithm. Step 1: $T = H^+L(H^T)^+$, where $D = HH^T$. Step 4: At level k, row i of α_k and row j of β_k are in the same p_k dimensional space and can be directly compared.

3 Theoretical Analysis

One significant advantage of wavelet analysis is that it directly generalizes to non-symmetric matrices, which are often encountered when constructing graphs using k-nearest neighbor relationships, in directed citation and web graphs, and Markov decision processes. If the matrix is symmetric, there is an interesting connection between our algorithm and manifold projections. Theorem 3 below proves that the proposed alignment result at level k and the result from manifold projections (with top p_k eigenvectors) are both optimal with respect to the loss function $C(\alpha, \beta)$ described in Sec 2.1. Theorems 1 and 2 prove some intermediate results, which are subsequently used in Theorem 3.

Theorem 1. The solution to the generalized eigenvalue decomposition $ZLZ^T\gamma = \lambda ZDZ^T\gamma$ is given by $((F^T)^+x, \lambda)$, where x and λ are eigenvector and eigenvalue of $F^+ZLZ^T(F^T)^+x = \lambda x$.
Proof: Using the notation summarized in Figure 1, $ZDZ^T = FF^T$, where F is a $(p+q) \times r$ matrix of rank r and can be constructed by singular value decomposition. It is obvious that ZDZ^T is positive semi-definite.
Case 1: when ZDZ^T is positive definite:
It is trivial to see that $r = p + q$. This implies that F is a $(p+q) \times (p+q)$ full rank matrix: $F^{-1} = F^+$.
$ZLZ^T\gamma = \lambda ZDZ^T\gamma \Longrightarrow ZLZ^T\gamma = \lambda FF^T\gamma \Longrightarrow ZLZ^T\gamma = \lambda F(F^T\gamma)$
$\Longrightarrow ZLZ^T(F^T)^{-1}(F^T\gamma) = \lambda F(F^T\gamma) \Longrightarrow F^{-1}ZLZ^T(F^T)^{-1}(F^T\gamma) = \lambda(F^T\gamma)$
$\Longrightarrow$ Solution to $ZLZ^T\gamma = \lambda ZDZ^T\gamma$ is given by $((F^T)^+x, \lambda)$, where x and λ are eigenvector and eigenvalue of $F^+ZLZ^T(F^T)^+x = \lambda x$.
Case 2: when ZDZ^T is positive semi-definite but not positive definite:
In this case, $r < p + q$ and F is a $(p+q) \times r$ matrix of rank r.
Since $ZD^{0.5}$ is a $(p+q)\times(m+n)$ matrix, F is a $(p+q)\times r$ matrix, there exits a matrix G such that $ZD^{0.5} = FG$. This implies $Z = FGD^{-0.5}$ and $GD^{-0.5} = F^+Z$.
$ZLZ^T\gamma = \lambda ZDZ^T\gamma$
$\Longrightarrow FGD^{-0.5}LD^{-0.5}G^TF^T\gamma = \lambda FF^T\gamma \Longrightarrow FGD^{-0.5}LD^{-0.5}G^T(F^T\gamma) = \lambda F(F^T\gamma)$
$\Longrightarrow (F^+F)GD^{-0.5}LD^{-0.5}G^T(F^T\gamma) = \lambda(F^T\gamma)$
$\Longrightarrow GD^{-0.5}LD^{-0.5}G^T(F^T\gamma) = \lambda(F^T\gamma) \Longrightarrow F^+ZLZ^T(F^T)^+(F^T\gamma) = \lambda(F^T\gamma)$
$\Longrightarrow$ One solution to $ZLZ^T\gamma = \lambda ZDZ^T\gamma$ is $((F^T)^+x, \lambda)$, where x and λ are eigenvector and eigenvalue of $F^+ZLZ^T(F^T)^+x = \lambda x$. Note that eigenvector solution to Case 2 is not unique.

Theorem 2. The matrix L is positive semi-definite.
Proof:
Assume $s = [s_{1:p}, s_{p+1:p+q}]$ is an arbitrary vector, where $s_{1:p} = [s_1, \cdots, s_p]$, $s_{p+1:p+q} = [s_{p+1}, \cdots, s_{p+q}]$. Let $L_1 = \begin{pmatrix} L_x & 0 \\ 0 & L_y \end{pmatrix}$, $L_2 = \begin{pmatrix} \Omega_1 & -\Omega_2 \\ -\Omega_3 & \Omega_4 \end{pmatrix}$, then $sLs^T = sL_1s^T + sL_2s^T$.

Firstly, $sL_1s^T = s_{1:p}L_xs_{1:p}^T + s_{p+1:p+q}L_ys_{p+1:p+q}^T \geq 0$.
The reason is as follows: L_x is a graph Laplacian matrix, so it is positive semi-

definite. This implies that $s_{1:p}L_x s_{1:p}^T \geq 0$. Similarly, $s_{p+1:p+q}L_y s_{p+1:p+q}^T \geq 0$. Considering the fact that $sL_2s^T = \mu \sum_{i=1}^{l}(s_i - s_{i+p})^2$, we have $sL_2s^T \geq 0$. So $sLs^T = sL_1s^T + sL_2s^T \geq 0$.
Since s is an arbitrary vector, L is positive semi-definite. □

It is well known that the alignment result from manifold projections (using p_k eigenvectors corresponding to the smallest non-zero eigenvalues of $ZLZ^T\gamma = \lambda ZDZ^T\gamma$) is optimal with respect to the loss function $C(\alpha, \beta)$. Theorem 3 shows that the proposed multiscale algorithm also achieves the optimal result.

Theorem 3: At level k, the multiscale manifold alignment algorithm achieves the optimal p_k dimensional alignment result with respect to the cost function $C(\alpha, \beta)$.
Proof: Let $T = F^+ZLZ^T(F^T)^+$. Since L is positive semi-definite (Theorem 2), T is also positive semi-definite. This means all eigenvalues of $T \geq 0$, and eigenvectors corresponding to the smallest non-zero eigenvalues of T are the same as the eigenvectors corresponding to the largest eigenvalues of T^+. From Theorem 1, we know the solution to generalized eigenvalue decomposition $ZLZ^T\gamma = \lambda ZDZ^T\gamma$ is given by $((F^T)^+x, \lambda)$, where x and λ are eigenvector and eigenvalue of $Tx = \lambda x$. Let columns of P_X denote the eigenvectors corresponding to the p_k largest non-zero eigenvalues of T^+. Then the manifold projections solution is given by $(F^T)^+P_X$.

Let columns of P_Y denote $[\phi_k]_{\phi_0}$, the scaling functions of T^+ at level k and p_k be the number of columns of $[\phi_k]_{\phi_0}$. In our multiscale algorithm, the solution at level k is provided by $(F^T)^+P_Y$.

From [11], we know P_X and P_Y span the same space. This means $P_XP_X^T = P_YP_Y^T$. Since the columns of both P_X and P_Y are orthonormal, we have $P_X^TP_X = P_Y^TP_Y = I$, where I is an $p_k \times p_k$ identity matrix. Let $Q = P_Y^TP_X$, then $P_X = P_XI = P_XP_X^TP_X = P_YP_Y^TP_X = P_Y(P_Y^TP_X) \Longrightarrow P_X = P_YQ$.

$Q^TQ = QQ^T = I$ and $det(Q^TQ) = (det(Q))^2 = 1$, $det(Q) = 1$. So Q is a rotation matrix.

Combining the results shown above, multiscale alignment algorithm at level k and manifold projections with p_k smallest non-zero eigenvectors achieve the same alignment results up to a rotation Q. □

4 Experimental Results

In this section, we apply our approach to transfer knowledge from one domain to another in three real-world problems on corpora alignment and cross-lingual information retrieval. $\mu = 1$ for all experiments except EU parallel corpus test, where $\mu = 10$.

The criterion for success can be defined in several ways, e.g., the interpretability of the constructed multiscale alignment or the performance at some ultimate task. For an example of the first criterion of interpretability to be satisfied, we can check to see if the proposed approach returns a multilevel alignment result that matches our prior knowledge. Section 4.1 is an example of this. For the

second criterion of performance at some task, we can test to see if the approach helps in finding the correspondence between the unlabeled data. In Sections 4.2 and 4.3, we compare the performance of the proposed method on the alignment task to other state-of-the-art manifold alignment approaches.

4.1 Multiscale Alignment of Corpora/Topics

One application of manifold alignment in information retrieval is corpora alignment, where corpora can be aligned so that knowledge transfer between different collections is possible. In this test, we applied our approach to align corpora represented in different topic spaces. Interestingly, our approach was also shown to be useful in finding topics shared by multiple collections.

Given two collections: X_1 (a $|W_1| \times |D_1|$ matrix) and X_2 (a $|W_2| \times |D_2|$ matrix), where $|W_i|$ is the size of the vocabulary set and $|D_i|$ is the number of the documents in collection X_i. Assume the topics learned from the two collections are given by S_1 and S_2, where S_i is a $W_i \times r_i$ matrix and r_i is the number of the topics in X_i. Then the representations of X_i in the topic space is $S_i^T X_i$. Following our main algorithm, $S_1^T X_1$ and $S_2^T X_2$ can be aligned in the latent space at level k by using mapping functions α_k and β_k. The representations of X_1 and X_2 after alignment become $\alpha_k^T S_1^T X_1 = (S_1\alpha_k)^T X_1$ and $\beta_k^T S_2^T X_2 = (S_2\beta_k)^T X_2$. Obviously, the document contents (X_1 and X_2) are not changed. The only thing that has been changed is S_i- the topic matrix. Recall that the columns of S_i are topics of X_i. The alignment algorithm changes S_1 to $S_1\alpha_k$ and S_2 to $S_2\beta_k$. The columns of $S_1\alpha_k$ and $S_2\beta_k$ are still of the length $|W_i|$. Such columns are in fact the new "aligned" topics.

The data set we used is the NIPS (1-12) full paper data set, which includes 1,740 papers and 2,301,375 tokens in total. We first represented this data set using two different topic spaces: LSI space [14] and LDA space [15]. In other words, $X_1 = X_2$, but $S_1 \neq S_2$ for this set. The reasons for aligning these two data sets is that while they define different features, they are constructed from the same data, and hence admit a correspondence under which the resulting data sets should be aligned well. Also, LSI and LDA topics can be mapped back to the English words, so the mapping functions are semantically interpretable. This helps us understand how the alignment of two collections is achieved (by aligning their underlying topics). We extracted 400 topics from the data set with both LDA and LSI models ($r_1 = r_2 = 400$). The top 8 words of topic 1-5 from each model are shown in Figure 4 and Figure 5. It is clear that none of those topics are similar across the two sets. Following the main algorithm using 20% uniformly selected documents as correspondences, we identified a 3 level hierarchy of mapping functions and the number of basis functions spanning each level was: 800, 91, 2. These numbers correspond to the intrinsic structure of the underlying joint manifold. At the finest scale, the manifold is spanned by 800 vectors. This makes sense, since the joint manifold is definitely spanned by 400 LSI topics+ 400 LDA topics. At level 3, the joint manifold is spanned by 2 vectors. To see how the original topics were changed can help us better understand the alignment algorithm. In Figure 6 and 7, we show 5 corresponding

Top 8 Terms
generalization function generalize shown performance theory size shepard
hebbian hebb plasticity activity neuronal synaptic anti hippocampal
grid moore methods atkeson steps weighted start interpolation
measure standard data dataset datasets results experiments measures
energy minimum yuille minima shown local university physics

Fig. 4. Topic 1-5 (LDA) before alignment.

Top 8 Terms
fish terminals gaps arbor magnetic die insect cone
learning algorithm data model state function models distribution
model cells neurons cell visual figure time neuron
data training set model recognition image models gaussian
state neural network model time networks control system

Fig. 5. Topic 1-5 (LSI) before alignment.

Top 8 Terms
road car vehicle autonomous lane driving range unit
processor processors brain ring computation update parallel activation
hopfield epochs learned synapses category modulation initial pulse
brain loop constraints color scene fig conditions transfer
speech capacity peak adaptive device transition type connections

Fig. 6. 5 LDA topics at level 2 after alignment.

Top 8 Terms
road autonomous vehicle range navigation driving unit video
processors processor parallel approach connection update brain activation
hopfield pulse firing learned synapses stable states network
brain color visible maps fig loop elements constrained
speech connections capacity charge type matching depth signal

Fig. 7. 5 LSI topics at level 2 after alignment.

Top 8 Terms
recurrent direct events pages oscillator user hmm oscillators
false chain protein region mouse human proteins roc

Fig. 8. 2 LDA topics at level 3 after alignment.

Top 8 Terms
recurrent belief hmm filter user head obs routing
chain mouse region human receptor domains proteins heavy

Fig. 9. 2 LSI topics at level 3 after alignment.

topics (corresponding columns of $S_1\alpha_2$ and $S_2\beta_2$) at level 2. From these figures, we can see that the new topics in correspondence are very similar to each other across the data sets, and interestingly the new aligned topics are semantically meaningful to represent some areas in either machine learning or neuroscience. At level 3, there are only two aligned topics (Figure 8 and 9). Clearly, one of them is about machine learning and another is about neuroscience. These two topics are the most abstract topics of NIPS conference. From these results, we can see that our algorithm can automatically align the given data sets at different scales following the intrinsic structure of the joint manifold. Since the alignment of collections is done via topic alignment, the new approach is also useful to find the common topics shared by the given collections. We also ran a test to directly compare the embedding of x_i (a document defined in LDA space) and

Top Terms
february gender violence april ngos equality mechanisms obstacles
copenhagen china barcelona swedish balkans secretary wording
ratification petitions pillar hundreds applause barcelona prosperity seven
racism secretary dignity globalisation pact everybody portugal meetings
examples credible prosperity sovereignty texts seven users sincerely

Fig. 10. 5 selected mapping functions at level 2 (English)

Top Terms
febbraio violenza aprile dichiarazione ong genere donne definiti
cina copenaghen barcellona svedese asia maastricht discarico sostenibilita
ratifica applausi barcellona sette centinaia petizioni pilastro rurali
razzismo riunioni segretario dignita globalizzazione convenzioni portogallo
prosperita esempi deplorevole sette organizzata sovranita ottobre utenti

Fig. 11. 5 selected mapping functions at level 2 (Italian)

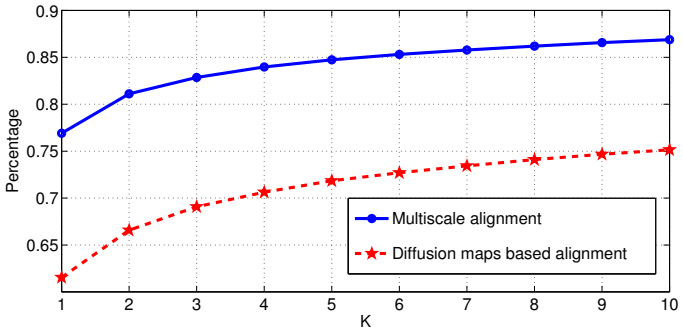

Fig. 12. EU parallel corpus test.

Italian Words	francese	francesi	governo	bovina	fiscale	carne	imposta
English Translations	French	French	government	beef	fiscal/tax	meat	impose
Contributions	0.494416	0.33942	0.305621	0.208814	0.198853	0.185615	0.143339

Fig. 13. 7 Italian words that make the largest contributions to the Italian query generated by $\alpha_2\beta_2^+$ from English query "French government beef tax".

y_j (a document defined in LSI space) at level 2 and found that the true match of x_i has a 86% probability of being the nearest neighbor of x_i in that new latent space.

4.2 European Parliament Proceedings Parallel Corpus Test

The data we use in this test is a collection of the proceedings of the European Parliament [16], dating from 04/1996 to 10/2006. The corpus includes versions in 11 European languages: French, Italian, Spanish, Portuguese, English, Dutch, German, Danish, Swedish, Greek and Finnish. Altogether, the corpus comprises of about 30 million words for each language.

The data for our experiments comes from the English-Italian parallel corpora, each of which has more than 36,000,000 words. The data set has many files, each file contains the utterances of one speaker in turn. We treat an utterance as a document. We first extracted English-Italian document pairs where both documents have at least 100 words. This resulted in 59,708 document pairs. 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. The documents were represented as bags of words, and no tag information was included. 10% resulting document pairs were used for training and the remaining 90% document pairs were held for testing.

We applied our algorithm to this data set and determined a 4 level hierarchy of mapping functions. The number of basis functions at each level was: 5000, 845, 2, 1. In Figure 10 and 11 we show the words that make the largest contributions to each of the 5 selected corresponding mapping functions at level 2. Our resulting tables resemble inter-language dictionaries, since they have roughly the same contents but in different languages. Note that we did not use any dictionary or ad hoc information retrieval technique in this alignment process. To compare our approach to the other state of the art approaches, we also tried diffusion maps based alignment [17] in the same setting, where 845 dimensional embeddings were used for comparison. Our testing scheme was as follows: for each given English document, we retrieved its top K most similar Italian documents. The probability that the true match is among the top K documents was used to show the goodness of the method. The results are summarized in Figure 12. The proposed multiscale approach beats diffusion maps based alignment by a large margin.

Interestingly, $\alpha_k \beta_k^+$ can automatically translate any unseen instance from domain X (English) to domain Y (Italian), where β_k^+ is the inverse of β_k. Such a translation is via the latent space, so the information that is only useful for domain X will not be transferred. To illustrate how $\alpha_k \beta_k^+$ works, we randomly generate an English query "French government beef tax", and use $\alpha_2 \beta_2^+$ to translate this query into Italian. The English query is represented by a vector of length 2,500, corresponding to 2,500 English words. Only 4 entries on that vector are 1s, all the other entries are 0s. The resulting Italian query is also a vector of length 2,500, corresponding to 2,500 Italian words. The numbers on the resulting vector show the contribution from each Italian word to the query. We print out top 7 Italian words in Figure 13. The result shows that the resulting Italian query can be treated as a translation of the English query, and used for cross-lingual information retrieval.

4.3 Cross-Lingual IR (English-Arabic)

In this section, we compare our multiscale approach with Procrustes alignment and semi-supervised alignment using a real world cross-lingual information retrieval data set. The task here is to find exact correspondences between the documents in different languages, enabling users to query a document in their native language and retrieve documents in a foreign language. The data set used

below was originally from [18]. It includes two document collections: one in English and one in Arabic (manually translated). The topical structure of each collection can be thought as a manifold over documents. Each document is a sample from the manifold. In this experiment, each of the two document collections has 2,119 documents. Correspondences between 20% of them are given and used to learn the alignment. The remaining 80% are held for testing. Our testing scheme is the same as that used in EU parallel test. We first applied multiscale approach to this problem and achieved the alignment results at 6 levels. The dimensionality of each level was: 240, 39, 6, 3, 2, 1. We chose level 2 for comparison. We tested Procrustes alignment using this data. Procrustes alignment consists of two steps: learning low dimensional embeddings of the two manifolds and aligning the low dimensional embeddings. In the first step, we tried both Laplacian eigenmaps [12] and LPP [13], where the top 39 eigenvectors were used to construct the embeddings. We also used the same data set to test the semi-supervised manifold alignment method [8], where top 39 eigenvectors were used for low dimensional embeddings. In Procrustes alignment (with Laplacian eigenmaps): for each given Arabic document, if we retrieve 5 most relevant English documents, then the true match has a 35% probability of being among the 5. In Procrustes alignment (with LPP): if we retrieve the 5 most relevant English documents, then we have a 40% probability of getting the true match. The performance of semi-supervised alignment is not very good compared to the other approaches. Semi-supervised alignment can map instances in correspondence to the similar locations in the new space, but the instances outside of the correspondence are not aligned well. The multiscale approach performs the best on this task, achieving roughly 5% improvement over Procrustes alignment when $K = 5$ and 1.

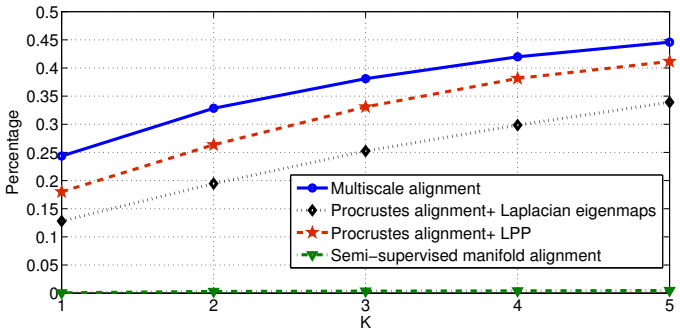

Fig. 14. Cross-lingual information retrieval test.

4.4 Discussions

In previous manifold alignment approaches, the users need to specify the dimensionality of the intended alignment. Finding an appropriate value for this is quite difficult. The proposed approach constructs multilevel alignment results based on the common underlying intrinsic structures of the given data sets, leaving the users with a small number of levels to consider (often < 10) even when the underlying problem may be defined by tens of thousands of features. Also, some levels are defined by either too many or too few features. This eliminates from consideration additional levels, usually resulting a handful of levels as possible candidates. The users can select the level that is the most appropriate for their applications. For example, in parallel corpus test presented in Section 4.2, we only have alignment results at 4 levels involving 5000, 845, 2, 1 dimensional spaces. Choosing the space defined by 845 features is a natural choice, since the levels below and above this have too few or too many features, respectively. A user can also select the most appropriate level by testing his/her data at different levels.

5 Conclusions

In this paper, we introduce a novel approach to multiscale manifold alignment based on multiresolution wavelet analysis. Our approach extends previously studied approaches in that it produces a hierarchical alignment that preserves the local geometry of each given manifold and matches the corresponding instances across manifolds at multiple scales. In addition to a theoretical analysis, we also presented real-world applications of our approach to corpora alignment and cross-lingual information retrieval.

References

1. J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *The 2006 Conference on Empirical Methods on Natural Language Processing*, pages 120–128, 2006.
2. H. Daumé III and D. Marcu. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, pages 101–126, 2006.
3. L. Duan, I. Tsang, D. Xu, and T. Chua. Domain adaptation from multiple sources via auxiliary classifiers. In *International Conference on Machine Learning*, 2009.
4. Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation with multiple sources. In *Neural Information Processing Systems (NIPS)*, 2009.
5. S. J. Pan and Q. Yang. A survey on transfer learning. Technical report, http://www.cse.ust.hk/∼sinnopan/SurveyTL.htm, 2008.
6. 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.
7. C. Wang and S. Mahadevan. Manifold alignment using procrustes analysis. In *Proceedings of the 25th International Conference on Machine Learning*, 2008.

8. 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.
9. L. Xiong, F. Wang, and C. Zhang. Semi-definite manifold alignment. In *Proceedings of the 18th European Conference on Machine Learning*, 2007.
10. C. Wang and S. Mahadevan. Manifold alignment without correspondence. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, 2009.
11. R. Coifman and M. Maggioni. Diffusion wavelets. *Applied and Computational Harmonic Analysis*, 21:53–94, 2006.
12. M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15, 2003.
13. X. He and P. Niyogi. Locality preserving projections. In *Proceedings of the Advances in Neural Information Processing Systems*, 2003.
14. S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.
15. D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
16. P. Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT Summit*, 2005.
17. R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21:5–30, 2006.
18. F. Diaz and D. Metzler. Pseudo-aligned multilingual corpora. In *Proceedings of The International Joint Conference on Artificial Intelligence*, pages 2727–2732, 2007.