Manifold Alignment without Correspondence

Chang Wang and **Sridhar Mahadevan**
Computer Science Department
University of Massachusetts
Amherst, Massachusetts 01003
{chwang, mahadeva}@cs.umass.edu

Abstract

Manifold alignment has been found to be useful in many areas of machine learning and data mining. In this paper we introduce a novel manifold alignment approach, which differs from "semi-supervised alignment" and "Procrustes alignment" in that it does not require predetermining correspondences. Our approach learns a projection that maps data instances (from two different spaces) to a lower dimensional space simultaneously matching the local geometry and preserving the neighborhood relationship within each set. This approach also builds connections between spaces defined by different features and makes direct knowledge transfer possible. The performance of our algorithm is demonstrated and validated in a series of carefully designed experiments in information retrieval and bioinformatics.

1 Introduction

In many areas of machine learning and data mining, one is often confronted with situations where the data is in a high dimensional space. Directly dealing with such high dimensional data is usually intractable, but in many cases, the underlying manifold structure may have a low intrinsic dimensionality. Manifold alignment builds connections between two or more disparate data sets by aligning their underlying manifolds and provides knowledge transfer across the data sets. Real-world applications include automatic machine translation [Diaz & Metzler, 2007], representation and control transfer in Markov decision processes, bioinformatics [Wang & Mahadevan, 2008], and image interpretation. Two previously studied manifold alignment approaches are Procrustes alignment [Wang & Mahadevan, 2008] and semi-supervised alignment [Ham *et al.*, 2005].

Procrustes alignment (illustrated in Figure 1(A)) is a two step algorithm leveraging pairwise correspondences between a subset of the instances. In the first step, the entire data sets are mapped to low dimensional spaces reflecting their intrinsic geometries using a standard (linear like LPP [He & Niyogi, 2003] or nonlinear like Laplacian eigenmaps [Belkin & Niyogi, 2003]) dimensionality reduction approach. In the second step, the translational, rotational and scaling components are removed from one set so that the optimal alignment between the instances in correspondence is achieved. Procrustes alignment learns a mapping defined everywhere, when a suitable dimensionality reduction method is used, so it can handle the new test points. In Procrustes alignment, the computation of lower dimensional embeddings is done in a unsupervised way (without considering the purpose of alignment), so the resulting embeddings of the two data sets might be quite different. Semi-supervised alignment (illustrated in Figure 1(B)) also uses a set of correspondences to align the manifolds. In this approach, the points of the two data sets are mapped to a new space by solving a constrained embedding problem, where the embeddings of the corresponding points from different sets are constrained to be close to each other. A significant disadvantage of this approach is that it directly computes the embedding results rather than the mapping functions, so the alignment is defined only on the known data points, and it is hard to handle the new test points.

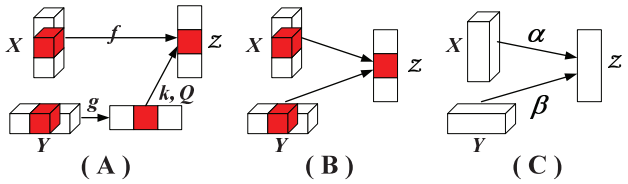

Figure 1: Comparison of different manifold alignment approaches. X and Y are the spaces where manifolds are defined on. $\mathcal{Z}$ is the new lower dimensional space. The red regions represent the subsets that are in correspondence. (A) Procrustes manifold alignment; (B) Semi-supervised manifold alignment; (C) The new approach: α and β are mapping functions.

A more general manifold alignment problem arises in many real world applications, where two manifolds (defined by totally different features) need to be aligned with no correspondence information available to us. Solving this problem is rather difficult, if not impossible, since there are two unknown variables in this problem: the correspondence and the transformation. One such example is control transfer between different Markov decision processes (MDPs), where we want to align state spaces of different tasks. Here, states are usually

defined by different features for different tasks and it is hard to find correspondences between them. This problem can be more precisely defined as follows: suppose we have two data sets $X = \{x_1, \cdots, x_m\}$ and $Y = \{y_1, \cdots, y_n\}$ for which we want to find correspondence, our aim is to compute functions α and β to map x_i and y_j to the same space such that $\alpha^T x_i$ and $\beta^T y_j$ can be directly compared.

To solve the problem mentioned above, the new algorithm (illustrated in Figure 1(C)) needs to go beyond the regular manifold alignment in that it should be able to map the data instances (from two different spaces) to a new lower dimensional space without using correspondence information. We also want the resulting alignment to be defined everywhere rather than just on the training instances, so that it can handle new test points. In this paper, we propose a novel approach to learn such mapping functions α and β to project the data instances to a new lower dimensional space by simultaneously matching the local geometry and preserving the neighborhood relationship within each set. In addition to the theoretical analysis of our algorithm, we also report on several real-world applications of the new alignment approach in information retrieval and bioinformatics. Notation used in this paper is defined and explained in Figure 3.

The rest of this paper is as follows. In Section 2 we describe the main algorithm. In Section 3 we explain the rationality underlying our approach. We describe some novel applications and summarize experimental results in Section 4. Section 5 provides some concluding remarks.

2 The Main Algorithm

2.1 The Problem

As defined in Figure 3, X is a set of samples collected from manifold $\mathcal{X}$; Y is a set of samples collected from manifold $\mathcal{Y}$. We want to learn mappings α and β to map X and Y to a new space $\mathcal{Z}$, where the neighborhood relationships inside of X and Y will be preserved, and if local geometries of x_i and y_j are matched in the original spaces, they will be neighbors in the new space.

2.2 High Level Explanation

The data sets X and Y are represented by different features. Thus, it is difficult to directly compare x_i and y_j. To build connections between them, we use the relation between x_i and its neighbors to characterize x_i's local geometry. Using relations rather than features to represent local geometry makes the direct comparison of x_i and y_j be possible. However, x_i might be similar to more than one instance in Y, and it is hard to identify which one is the true match (in fact, for many applications, there is more than one true match).

An interesting fact is that solving the original coupled problem could be easier than only finding the true match. The reason is the structure of both manifolds need to be preserved in the alignment. This helps us get rid of many false positive matches. In our algorithm, we first identify all the possible matches for each instance leveraging its local geometry. Then we convert the alignment problem to an embedding problem with constraints. The latter can be solved by solving a generalized eigenvalue decomposition problem.

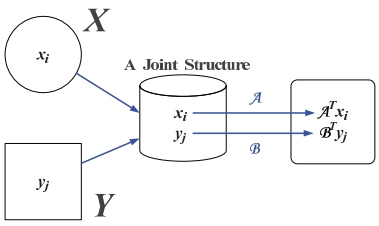

Figure 2: Illustration of the main algorithm.

2.3 The Algorithm

Assume the kernels for computing the similarity between data points in each of the two data sets are already given (for example, heat kernel). The algorithm is as follows:

1. **Create connections between local geometries:**
   - $W^{ij} = e^{-dist(R_{x_i}, R_{y_j})/\delta^2}$, where R_{x_i}, R_{y_j}, and W are defined and explained in Figure 3, $dist(R_{x_i}, R_{y_j})$ is defined in Sec 3.2.
   - The definition of W^{ij} could be application oriented. Using other ways to define W^{ij} does not affect the other parts of the algorithm.
2. **Join the two manifolds:**
   - Compute the matrices L, Z and D, which are used to model the joint structure.
3. **Compute the optimal projection to reduce the dimensionality of the joint structure:**
   - The d dimensional projection is computed by d minimum eigenvectors $\gamma_1 \cdots \gamma_d$ of the generalized eigenvalue decomposition $ZLZ^T\gamma = \lambda ZDZ^T\gamma$.
4. **Find the correspondence between X and Y:**
   - Let $\mathcal{A}$ be the top p rows of $[\gamma_1 \cdots \gamma_d]$, and $\mathcal{B}$ be the next q rows. For any i and j, $\mathcal{A}^T x_i$ and $\mathcal{B}^T y_j$ are in the same space and can be directly compared.

The algorithm is illustrated in Figure 2. $x_i \in X$ and $y_j \in Y$ are from different manifolds, so they cannot be directly compared. Our algorithm learns a mapping $\mathcal{A}$ for X and a mapping $\mathcal{B}$ for Y to map the two manifolds to one space so that instances (from different manifolds) with similar local geometry will be mapped to similar locations and the manifold structures will also be preserved. Computing $\mathcal{A}$ and $\mathcal{B}$ is tricky. Steps 1 and 2 are in fact joining the two manifolds so that their underlying structures in common can be explored. Step 3 computes a mapping to map the joint structure to a lower dimensional space, where $\begin{pmatrix} \mathcal{A} \\ \mathcal{B} \end{pmatrix} = [\gamma_1 \cdots \gamma_d]$ is used for manifold alignment. Section 3 explains the rationale underlying the approach. Once $\mathcal{A}$ and $\mathcal{B}$ are available to us, $\mathcal{A}\mathcal{B}^+$ and $\mathcal{B}\mathcal{A}^+$ can be used as "keys" to translate instances between spaces defined by totally different features (for example, one is in English, another is in Chinese). The algorithm can also be used when partial correspondence information is available (see Sec 4.2 for more details).

x_i is defined in a p dimensional space (manifold $\mathcal{X}$), and the p features are $\{f_1, \cdots, f_p\}$;
$X = \{x_1, \cdots, x_m\}$, X is a $p \times m$ matrix.
$W_x^{i,j}$ is the similarity of x_i and x_j (could be defined by heat kernel).
D_x is a diagonal matrix: $D_x^{ii} = \sum_j W_x^{ij}$.
$L_x = D_x - W_x$.

y_i is defined in a q dimensional space (manifold $\mathcal{Y}$), and the q features are $\{g_1, \cdots, g_q\}$;
$Y = \{y_1, \cdots, y_n\}$, Y is a $q \times n$ matrix.
$W_y^{i,j}$ is the similarity of y_i and y_j (could be defined by heat kernel).
D_y is a diagonal matrix: $D_y^{ii} = \sum_j W_y^{ij}$.
$L_y = D_y - W_y$.

$$Z = \begin{pmatrix} X & 0 \\ 0 & Y \end{pmatrix}, D = \begin{pmatrix} D_x & 0 \\ 0 & D_y \end{pmatrix}.$$

k: k in k-nearest neighbor method.

R_{x_i} is a $(k+1) \times (k+1)$ matrix representing the local geometry of x_i. $R_{x_i}(a,b) = distance(z_a, z_b)$, where $z_1 = x_i, \{z_2, \cdots z_{k+1}\}$ are x_i's k nearest neighbors.
Similarly, R_{y_j} is a $(k+1) \times (k+1)$ matrix representing the local geometry of y_i.

W is an $m \times n$ matrix, where $W^{i,j}$ is the similarity of R_{x_i} and R_{y_j}.

The order of y_j's k nearest neighbors have $k!$ permutations, so R_{y_j} has $k!$ variants. Let $\{R_{y_j}\}_h$ denote its h^{th} variant.

α is a mapping to map x_i to a scalar: $\alpha^T x_i$ (α is a $p \times 1$ matrix).
β is a mapping to map y_i to a scalar: $\beta^T y_i$ (β is a $q \times 1$ matrix).
$\gamma = (\alpha^T, \beta^T)^T$.

$C(\alpha, \beta)$ is the cost function (defined in Sec 3.1).
μ is the weight of the first term in $C(\alpha, \beta)$.

Ω_1 is an $m \times m$ diagonal matrix, and $\Omega_1^{ii} = \sum_j W^{i,j}$.
Ω_2 is an $m \times n$ matrix, and $\Omega_2^{i,j} = W^{i,j}$.
Ω_3 is an $n \times m$ matrix, and $\Omega_3^{i,j} = W^{j,i}$.
Ω_4 is an $n \times n$ diagonal matrix, and $\Omega_4^{ii} = \sum_j W^{j,i}$.

$$L = \begin{pmatrix} L_x + \mu\Omega_1 & -\mu\Omega_2 \\ -\mu\Omega_3 & L_y + \mu\Omega_4 \end{pmatrix}.$$

$\|\cdot\|_F$ denotes Frobenius norm.
$()^+$ denotes Pseudo Inverse.

Figure 3: Notation used in this paper.

3 Justification

3.1 The Big Picture

Let's begin with semi-supervised manifold alignment [Ham *et al.*, 2005]. Given two data sets X, Y along with additional pairwise correspondences between a subset of the training instances $x_i \longleftrightarrow y_i$ for $i \in [1, l]$, semi-supervised alignment directly computes the mapping results of x_i and y_i for alignment 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 mapping result of x_i, g_i is the mapping result of y_i and μ is the weight of the first term. The first term penalizes the differences between X and Y on the mapping results of the corresponding instances. The second and third terms guarantee that the neighborhood relationship within X and Y will be preserved.

Our approach has two fundamental differences compared to semi-supervised alignment. First, since we do not have correspondence information, the correspondence constraint in $C(f,g)$ is replaced with a soft constraint induced by local geometry similarity. Second, we seek for linear mapping functions α and β rather than direct embeddings, so that the mapping is defined everywhere. The cost function we want to minimize is as follows:
$C(\alpha, \beta) = \mu \sum_{i,j} (\alpha^T x_i - \beta^T y_j)^2 W^{i,j}$
$+0.5 \sum_{i,j} (\alpha^T x_i - \alpha^T x_j)^2 W_x^{i,j} + 0.5 \sum_{i,j} (\beta^T y_i - \beta^T y_j)^2 W_y^{i,j}$.

The first term of $C(\alpha, \beta)$ penalizes the differences between X and Y on the matched local patterns in the new space. Suppose that R_{x_i} and R_{y_j} are similar, then W^{ij} will be large. If the mapping results in x_i and y_j are being far away from each other in the new space, the first term will be large. The second and third terms preserve the neighborhood relationship within X and Y. In this section, we first explain how local patterns are computed, matched and why this is valid (Theorem 1). Then we convert the alignment problem to an embedding problem with constraints. An optimal solution to the latter is provided in Theorem 2.

3.2 Matching Local Geometry

Given X, we first construct an $m \times m$ distance matrix $Distance_x$, where $Distance_x(i,j) =$ Euclidean distance between x_i and x_j. We then decompose it into elementary contact patterns of fixed size $k+1$. As defined in Figure 3, each local contact pattern R_{x_i} is represented by a submatrix, which contains all pairwise distances between local neighbors around x_i. Such a submatrix is a 2D representation of a high dimensional substructure. It is independent of the coordinate frame and contains enough information to reconstruct the whole manifold. Y is processed similarly and distance between R_{x_i} and R_{y_j} is defined as follows:
$dist(R_{x_i}, R_{y_j}) = \min_{1 \le h \le k!} \min(dist_1(h), dist_2(h))$,
where
$dist_1(h) = \|\{R_{y_j}\}_h - k_1 R_{x_i}\|_F$,
$dist_2(h) = \|R_{x_i} - k_2 \{R_{y_j}\}_h\|_F$,
$k_1 = trace(R_{x_i}^T \{R_{y_j}\}_h) / trace(R_{x_i}^T R_{x_i})$,
$k_2 = trace(\{R_{y_j}\}_h^T R_{x_i}) / trace(\{R_{y_j}\}_h^T \{R_{y_j}\}_h)$.

Theorem 1: Given two $(k+1)\times(k+1)$ **distance matrices** R_1 **and** R_2**,** $k_2 = trace(R_2^T R_1)/trace(R_2^T R_2)$ **minimizes** $\|R_1 - k_2 R_2\|_F$ **and** $k_1 = trace(R_1^T R_2)/trace(R_1^T R_1)$ **minimizes** $\|R_2 - k_1 R_1\|_F$**.**
Proof:
Finding k_2 is formalized as $k_2 = \arg\min_{k_2} \|R_1 - k_2 R_2\|_F$, where $\|\cdot\|_F$ represents Frobenius norm. (1)

It is easy to verify that $\|R_1 - k_2 R_2\|_F =$
$trace(R_1^T R_1) - 2k_2 trace(R_2^T R_1) + k_2^2 trace(R_2^T R_2)$. (2)

Since $trace(R_1^T R_1)$ is a constant, the minimization problem is equal to
$k_2 = \arg\min_{k_2} k_2^2 trace(R_2^T R_2) - 2k_2 trace(R_2^T R_1)$. (3)

Differentiating with respect to k_2, (3) implies
$2k_2 trace(R_2^T R_2) = 2trace(R_2^T R_1)$. (4)
(4) implies $k_2 = trace(R_2^T R_1)/trace(R_2^T R_2)$. (5)

Similarly, $k_1 = trace(R_1^T R_2)/trace(R_1^T R_1)$. □

To compute matrix W (defined in Figure 3), we need to compute the comparison of all pairs. When comparing local pattern R_{x_i} and R_{y_j}, we assume x_i matches y_j. However, how x_i's k neighbors match y_j's k neighbors is not known to us. To find the best possible match, we have to consider all $k!$ possible permutations. This is tractable, since we are comparing local patterns and k is always small. R_{x_i} and R_{y_j} are from different manifolds, so the their sizes could be quite different. In Theorem 1, we show how to find the best re-scaler to enlarge or shrink one of them to match the other. It is straightforward to show that $dist(R_{x_i}, R_{y_j})$ considers all the possible matches between two local patterns and returns the distance computed from the best possible match.

$dist(\cdot)$ defined in this section provides a general way to compare local patterns. In fact, the local pattern generation and comparison can also be application oriented. For example, many existing kernels based on the idea of convolution kernels [Haussler, 1999] can be applied here. Choosing another way to define $dist(\cdot)$ will not affect the other parts of the algorithm. Similarity $W^{i,j}$ is directly computed from $dist(i,j)$ of neighboring points with either heat kernel $e^{-dist(i,j)/\delta^2}$ or something like $v_{large} - dist(i,j)$, where v_{large} is larger than $dist(i,j)$ for any i and j.

3.3 Manifold Alignment without Correspondence

In this section, we show that the solution to minimize $C(\alpha,\beta)$ provides the optimal mappings to align X and Y. The solution is achieved by solving a generalized eigenvalue decomposition problem.

Theorem 2: The minimum eigenvectors of the generalized eigenvalue decomposition $ZLZ^T\gamma = \lambda ZDZ^T\gamma$ **provide optimal mappings to align** X **and** Y **regarding the cost function** $C(\alpha,\beta)$**.**
Proof:
The key part of the proof is that $C(\alpha,\beta) = \gamma^T ZLZ^T\gamma$.
This result is not trivial, but can be verified by expanding the right hand side of the equation. The matrix L is in fact used to join two graphs such that two manifolds can be aligned and the underlying structure in common can be explored.

To remove an arbitrary scaling factor in the embedding, we impose an extra constraint $\alpha^T X D_x X^T \alpha + \beta^T Y D_y Y^T \beta = \gamma^T ZDZ^T\gamma = 1$. The matrices D_x and D_y provide a natural measure on the vertices (instances) of the graph. If the value D_x^{ii} or D_y^{ii} is large, it means x_i or y_i is more important. Without this constraint, all instances could be mapped to the same location in the new space. A similar constraint is also used in Laplacian eigenmaps [Belkin & Niyogi, 2003].

Finally, the optimization problem can be written as:
$\arg\min_{\gamma:\gamma^T ZDZ^T\gamma=1} C(\alpha,\beta) =$
$\arg\min_{\gamma:\gamma^T ZDZ^T\gamma=1} \gamma^T ZLZ^T\gamma$
By using the Lagrange trick, it is easy to see that solution to this equation is the same as the minimum eigenvector solution to $ZLZ^T\gamma = \lambda ZDZ^T\gamma$.

Standard methods show that the solution to find a d dimensional alignment is provided by the eigenvectors corresponding to the d lowest eigenvalues of the same generalized eigenvalue decomposition equation. □

4 Experimental Results

In this section, we test and illustrate our approach using a bioinformatics data set and an information retrieval data set. In both experiments, we set $k = 4$ to define local patterns, and $\delta = 1$ in heat kernel. We also tried $k = 3$ and 5, which performed as well in both tests. We did not try $k \le 2$ or $k \ge 6$, since either there were too many false positive matches or the learning was too time consuming.

4.1 Alignment of Protein Manifolds

In this test, we directly align two manifolds to illustrate how our algorithm works. The two manifolds are from bioinformatics domain, and were first created and used in [Wang & Mahadevan, 2008].

Protein 3D structure reconstruction is an important step in Nuclear Magnetic Resonance (NMR) protein structure determination. It is a process to estimate 3D structure from partial pairwise distances. In practice, researchers usually combine the partial distance matrix with other techniques, such as angle constraints and human experience to determine protein structures. With the information available to us, NMR techniques might find multiple estimations (models), since more than one configurations can be consistent with the distance matrix and the constraints. Thus, the construction result is an ensemble of models, rather than a single structure. Models related to the same protein are similar to each other but not exactly the same, so they can be used to test manifold alignment approaches. The comparison between models in the ensemble provides some information on how well the protein conformation was determined by NMR.

To align such two manifolds A and B, [Wang & Mahadevan, 2008] used roughly 25% uniformly selected amino acids in the protein as correspondences. Our approach does not require correspondence and can be directly applied to the data.

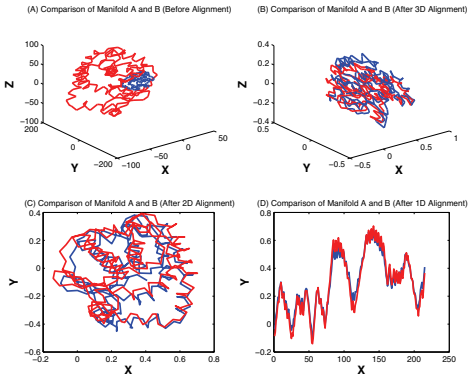

Figure 4: Alignment of protein manifolds: (A) Manifold A and B; (B) 3D alignment; (C) 2D alignment; (D) 1D alignment.

For the purpose of comparison, we plot both manifolds on the same figure (Figure 4A). It is clear that manifold A is much larger than B, and the orientations of A and B are quite different. To show how our approach works, we plot 3D (Figure 4B), 2D (Figure 4C) and 1D (Figure 4D) alignment results in Figure 4. nD alignment result is achieved by applying top n minimum eigenvectors. These figures clearly show that the alignment of two different manifolds is achieved by projecting the data (represented by the original features) onto a new space using our carefully generated mapping functions.

4.2 Alignment of Document Collection Manifolds

Another application field of manifold alignment is in information retrieval, where corpora can be aligned to identify correspondences between relevant documents. Results on cross-lingual document alignment (identifying documents with similar contents but in different languages) have been reported in [Diaz & Metzler, 2007] and [Wang & Mahadevan, 2008]. In this section, we apply our approach to compute correspondences between documents represented in different topic spaces. This application is directly related to a new research area: topic modeling, which is to extract succinct descriptions of the members of a collection that enable efficient generalization and further processing. A topic could be thought as a multinomial word distribution learned from a collection of textual documents. The words that contribute more to each topic provide keywords that briefly summarize the themes in the collection. In this paper, we consider two topic models: Latent Semantic Indexing (LSI) [Deerwester *et al.*, 1990] and diffusion model [Wang & Mahadevan, 2009].

LSI: Latent semantic indexing (LSI) is a well-known linear algebraic method to find topics in a text corpus. The key idea is to map high-dimensional document vectors to a lower dimensional representation in a latent semantic space. Let the singular values of an $n \times m$ term-document matrix A be $\delta_1 \geq \cdots \geq \delta_r$, where r is the rank of A. The singular value decomposition of A is $A = U\Sigma V^T$, where $\Sigma = diag(\delta_1, \cdots \delta_r)$, U is an $n \times r$ matrix whose columns are orthonormal, and V is an $m \times r$ matrix whose columns are also orthonormal. LSI constructs a rank-k approximation of the matrix by keeping the k largest singular values in the above decomposition, where k is usually much smaller than r. Each of the column vectors of U is related to a concept, and represents a topic in the given collection of documents.

Diffusion model: Diffusion model based topic modeling builds on diffusion wavelets [Coifman & Maggioni, 2006] and is introduced in [Wang & Mahadevan, 2009]. It is completely data-driven, largely parameter-free and can automatically determine the number of levels of the topical hierarchy, as well as the topics at each level.

Representing Documents in Topic Spaces: If a topic space $\mathcal{S}$ is spanned by a set of r topic vectors, we write the set as $S = (t(1), \cdots, t(r))$, where topic $t(i)$ is a column vector $(t(i)_1, t(i)_2 \cdots, t(i)_n)^T$. Here n is the size of the vocabulary set, $\|t(i)\| = 1$ and the value of $t(i)_j$ represents the contribution of term j to $t(i)$. Obviously, S is an $n \times r$ matrix. We know the term-document matrix A (an $n \times m$ matrix) models the corpus, where m is the number of the documents and columns of A represent documents in the "term" space. The low dimensional embedding of A in the "topic" space $\mathcal{S}$ is then $S^T A$ (an $r \times m$ matrix), whose columns are the new representations of documents in $\mathcal{S}$.

Data Set: In this test, we extract LSI and diffusion model topics from the NIPS paper data set, which includes 1,740 papers. The original vocabulary set has 13,649 terms. The corpus has 2,301,375 tokens in total. We used a simplified version of this data set, where the terms that appear ≤ 100 times in the corpus were filtered out, and only 3,413 terms were kept. Size of the collection did not change too much. The number of the remaining tokens was 2,003,017.

Results: We extract the top 37 topics from the data set with both LSI and diffusion model. The diffusion model approach can automatically identify number of topics in the collection. It resulted in 37 topics being automatically created from the data. The top 10 words of topic 1-5 from each model are shown in Table 1 and Table 2. It is clear that only topic 3 is similar across the two sets, so representations of the same document in different topic spaces will "look" quite different. We denote the data set represented in diffusion model topic space manifold as A, and in LSI topic space manifold as B. Even A and B do not look similar, they should still be somehow aligned well after some transformations, since they are from the same data set. To test the performance of alignment. We apply our algorithm to map A and B to a new space (dimension=30), where the direct comparison between instances is possible. For each instance in A, we consider its top a most similar instances in B as match candidates. If the true match is among these a candidates, we call it a hit. We tried $a = 1, \cdots, 4$ in experiment and summarized the results in Figure 5. From this figure, we can see that the true match has a roughly 65% probability of being the top 1 candidate and 80% probability of being among the top 4 retrieved candidates. We also tested using local patterns only to align two manifolds (without considering the manifold structure). The overall performance (Figure 5) is worse, but we can see that our local pattern comparison approach does well at modeling the local similarity. For 40% corresponding instances, their local patterns are close to each other. As mentioned above, LSI topics and diffusion model topics are quite dif-

Table 1: Topic 1-5 (diffusion model)

Top 10 Terms
network learning model neural input data time function figure set
cells cell neurons firing cortex synaptic visual cortical stimulus response
policy state action reinforcement actions learning reward mdp agent sutton
mouse chain proteins region heavy receptor protein alpha human domains
distribution data gaussian density bayesian kernel posterior likelihood em regression

Table 2: Topic 1-5 (LSI)

Top 10 Terms
perturbed terminals bus monotonicity magnetic quasi die weiss mostafa leibler
algorithm training error data set learning class algorithms examples policy
policy state action reinforcement learning actions mdp reward sutton policies
distribution gaussian data kernel spike density regression functions bound bayesian
chip analog circuit voltage synapse gate charge vlsi network transistor

ferent (shown in Table 1 and 2). In Table 3 and 4, we show how those top 5 topics are changed in the alignment. From the two new tables, we can see that all corresponding topics are similar to each other across the data sets now. This change shows our algorithm aligns the two document manifolds by aligning their features (topics). The new topics are in fact the topics shared by the two given collections.

In this paper, we show that when no correspondence is given, alignment can still be learned by considering local geometry and the manifold topology. In real world applications, we might have more or less correspondence information that may significantly help alignment. Our approach is designed to be able to use both information sources. As defined in Figure 3, matrix W characterizes the similarity between x_i and y_j. To do alignment without correspondence, we have to generate W leveraging local geometry. When correspondence information is available, W is then predetermined. For example, if we know x_i matches y_i for $i = 1, \cdots, l$, then W is a matrix having 1 on the top l elements of the diagonal, and 0 on all the other places. So our algorithm is in fact very general and can also work with correspondence information by changing W. We plot the alignment result with 15% corresponding points in Figure 5. The performance of alignment without correspondence is just slightly worse than that. We also tried semi-supervised alignment using the same data and correspondence. The performance (not shown here) was poor compared to the other approaches. Semi-supervised alignment can map instances in correspondence to the same location in the new space, but the instances outside of the correspondence were not aligned well.

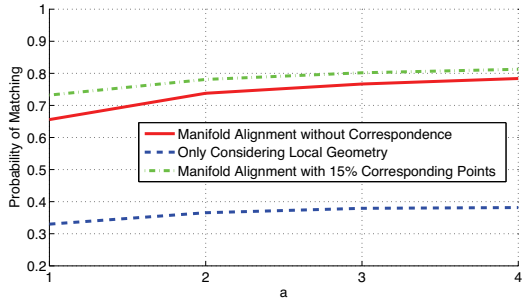

Figure 5: Results of document manifold alignment

Table 3: Topic 1-5 after alignment (diffusion model)

Top 10 Terms
som stack gtm hme expert adaboost experts date boosting strings
hint hints monotonicity mostafa abu market financial trading monotonic schedules
instructions instruction obs scheduling schedule dec blocks execution schedules obd
actor critic pendulum iiii pole tsitsiklis barto stack instructions instruction
obs obd pruning actor stork hessian critic pruned documents retraining

Table 4: Topic 1-5 after alignment (LSI)

Top 10 Terms
som stack gtm skills hme automaton strings giles date automata
hint hints monotonicity mostafa abu market financial trading monotonic schedules
instructions instruction obs scheduling schedule dec blocks execution schedules obd
actor critic pendulum pole signature tsitsiklis barto iiii sutton control
obs obd pruning actor critic stork hessian pruned documents retraining

5 Conclusions

In this paper, we introduce a novel approach to manifold alignment. Our approach goes beyond Procrustes alignment and semi-supervised alignment in that it does not require correspondence information. It also results in mappings defined everywhere rather than just on the training data points and makes knowledge transfer between domains defined by different features possible. In addition to theoretical validations, we also presented real-world applications of our approach to bioinformatics and information retrieval.

Acknowledgments

Support for this research was provided in part by the National Science Foundation under grants IIS-0534999 and IIS-0803288.

References

[Belkin & Niyogi, 2003] Belkin, M., Niyogi, P. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation* 15.

[Coifman & Maggioni, 2006] Coifman, R., and Maggioni, M. 2006. Diffusion wavelets. *Applied and Computational Harmonic Analysis* 21:53–94.

[Deerwester *et al.*, 1990] Deerwester, S.; Dumais, S. T.; Furnas, G. W.; Landauer, T. K.; and Harshman, R. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41(6):391–407.

[Diaz & Metzler, 2007] Diaz, F., Metzler, D. 2007. Pseudo-aligned multilingual corpora. *The International Joint Conference on Artificial Intelligence(IJCAI)* 2727-2732.

[Ham *et al.*, 2005] Ham, J., Lee, D., Saul, L. 2005. Semisupervised alignment of manifolds. 10^{th} *International Workshop on Artificial Intelligence and Statistics.* 120-127.

[Haussler, 1999] Haussler, D. 1999. Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10.

[He & Niyogi, 2003] He, X., Niyogi, P. 2003. Locality preserving projections. *NIPS 16.*

[Wang & Mahadevan, 2008] Wang, C., Mahadevan, S. 2008. Manifold alignment using Procrustes analysis. In *Proceedings of the 25th International Conference on Machine Learning*.

[Wang & Mahadevan, 2009] Wang, C., Mahadevan, S. 2009. Multiscale analysis of document corpora based on diffusion models. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*.