Manifold Alignment using Procrustes Analysis

Chang Wang CHWANG@CS.UMASS.EDU
Sridhar Mahadevan MAHADEVA@CS.UMASS.EDU
Computer Science Department, University of Massachusetts, Amherst, MA 01003 USA

Abstract

In this paper we introduce a novel approach to manifold alignment, based on Procrustes analysis. Our approach differs from "semi-supervised alignment" in that it results in a mapping that is defined everywhere – when used with a suitable dimensionality reduction method – rather than just on the training data points. We describe and evaluate our approach both theoretically and experimentally, providing results showing useful knowledge transfer from one domain to another. Novel applications of our method including cross-lingual information retrieval and transfer learning in Markov decision processes are presented.

1. Introduction

Manifold alignment is very useful in a variety of applications since it provides knowledge transfer between two seemingly disparate data sets. Sample applications include automatic machine translation, representation and control transfer between different Markov decision processes (MDPs), image comparison, and bioinformatics. More precisely, suppose we have two data sets $\mathcal{S}_1 = \{x_1, \cdots, x_m\}$ and $\mathcal{S}_2 = \{y_1, \cdots, y_n\}$ for which we want to find a correspondence. Working with the data in its original form can be very difficult as the data might be in high dimensional spaces and the two sets might be represented by different features. For example, $\mathcal{S}_1$ could be a collection of English documents, whereas $\mathcal{S}_2$ is a collection of Arabic documents. Thus, it may be difficult to directly compare documents from the two collections.

Even though the processing of high-dimensional data sets is challenging, for many cases, the data source may only have a limited number of degrees of freedom, implying the data set has a low intrinsic dimensionality. Similar to current work in the field, we assume kernels for computing the similarity between data points in the original space are already given. In the first step, we map the data sets to low dimensional spaces reflecting their intrinsic geometries using a standard (nonlinear or linear) dimensionality reduction approach. For example, using a graph-based nonlinear dimensionality reduction method provides a discretized approximation to the manifolds, so the new representations characterize the relationships between points but not the original features. By doing this, we can compare the embeddings of the two sets instead of their original representations. Generally speaking, if two data sets $\mathcal{S}_1$ and $\mathcal{S}_2$ have similar intrinsic geometry structures, they have similar embeddings. In our second step, we apply Procrustes analysis to align the two low dimensional embeddings of the data sets based on a number of landmark points. Procrustes analysis, which has been used for statistical shape analysis and image registration of 2D/3D data (Luo et al., 1999), removes the translational, rotational and scaling components from one set so that the optimal alignment between the two sets can be achieved.

There is a growing body of work on manifold alignment. Ham et al. (Ham et al., 2005) align the manifolds leveraging a set of correspondences. In their approach, they map the points of the two data sets to the same space by solving a constrained embedding problem, where the embeddings of the corresponding points from different sets are constrained to be identical. The work of Lafon et al. (Lafon et al., 2006) is based on a similar framework as ours. They use Diffusion Maps to embed the nodes of the graphs corresponding to the aligned sets, and then apply affine matching to align the resulting clouds of points.

Our approach differs from semi-supervised alignment (Ham et al., 2005) in that it results in a mapping that is defined everywhere rather than just on the known data points (provided a suitable dimensionality

Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008.

reduction method like LPP (He et al., 2003) or PCA is used). Recall that semi-supervised alignment is defined only on the known data points and it is hard to handle the new test points (Bengio et al., 2004). Our method is also faster, since it requires computing eigendecompositions of much smaller matrices. Compared to affine matching, which changes the shape of one given manifold to achieve alignment, our approach keeps the manifold shape untouched. This property preserves the relationship between any two data points in each individual manifold in the process of alignment. The computation times for affine matching and Procrustes analysis are similar, both run in $O(N^3)$ (where N is the number of instances).

Given the fact that dimensionality reduction approaches play a key role in our approach, we provide a theoretical bound for the difference between subspaces spanned by low dimensional embeddings of the two data sets. This bound analytically characterizes when the two data sets can be aligned well. In addition to the theoretical analysis of our algorithm, we also report on several novel applications of our alignment approach.

The rest of this paper is as follows. In Section 2 we describe the main algorithm. In Section 3 we explain the rationale underlying our approach, and prove a bound on the difference between the subspaces spanned by low dimensional embeddings of the two data sets being aligned. We describe some novel applications and summarize our experimental results in Section 4. Section 5 provides some concluding remarks.

2. Manifold Alignment

2.1. The Problem

Given two data sets along with additional pairwise correspondences between a subset of the training instances, we want to determine a correspondence between the remaining instances in the two data sets. Formally speaking, we have two sets: $\mathcal{S}_1 = \mathcal{S}_1^l \bigcup \mathcal{S}_1^u = \{x_1, \cdots, x_m\}$, $\mathcal{S}_2 = \mathcal{S}_2^l \bigcup \mathcal{S}_2^u = \{y_1, \cdots, y_n\}$, and the subsets $\mathcal{S}_1^l$ and $\mathcal{S}_2^l$ are in pairwise alignment. We want to find a mapping f, which is more precisely defined in Section 3.1, to optimally match the points between $\mathcal{S}_1^u$ and $\mathcal{S}_2^u$.

2.2. The Algorithm

Assume the kernel K_i for computing the similarity between data points in each of the two data sets is already given. The algorithmic procedure is stated below. For the sake of concreteness, in the procedure, Laplacian eigenmap (Belkin et al., 2003) is used for dimensionality reduction.

1. **Constructing the relationship matrices:**
   - Construct the weight matrices W_1 for $\mathcal{S}_1$ and W_2 for $\mathcal{S}_2$ using K_i, where $W_1(i,j) = K_1(x_i, x_j)$ and $W_2(i,j) = K_2(y_i, y_j)$.
   - Compute Laplacian matrices $\mathcal{L}_1 = I - D_1^{-0.5} W_1 D_1^{-0.5}$ and $\mathcal{L}_2 = I - D_2^{-0.5} W_2 D_2^{-0.5}$, where D_k is a diagonal matrix ($D_k(i,i) = \sum_j W_k(i,j)$) and I is the identity matrix.
2. **Learning low dimensional embeddings of the data sets:**
   - Compute selected eigenvectors of $\mathcal{L}_1$ and $\mathcal{L}_2$ as the low dimensional embeddings of the data sets $\mathcal{S}_1$ and $\mathcal{S}_2$. Let X, X_U be the d dimensional embeddings of $\mathcal{S}_1^l$ and $\mathcal{S}_1^u$, Y, Y_U be the d dimensional embeddings of $\mathcal{S}_2^l$ and $\mathcal{S}_2^u$, where $\mathcal{S}_1^l$, $\mathcal{S}_2^l$ are in pairwise alignment and $|\mathcal{S}_1^l| = |\mathcal{S}_2^l|$.
3. **Finding the optimal alignment of X and Y:**
   - Translate the configurations in X, X_U, Y and Y_U, so that X, Y have their centroids $(\sum_{i=1}^{|\mathcal{S}_1^l|} X_i/|\mathcal{S}_1^l|, \sum_{i=1}^{|\mathcal{S}_2^l|} Y_i/|\mathcal{S}_2^l|)$ at the origin.
   - Compute the singular value decomposition (SVD) of $Y^T X$, that is $U\Sigma V^T = SVD(Y^T X)$.
   - $Y^* = kYQ$ is the optimal mapping result that minimizes $\|X - Y^*\|_F$, where $\|.\|_F$ is Frobenius norm, $Q = UV^T$ and $k = trace(\Sigma)/trace(Y^T Y)$.
4. **Apply Q and k to find correspondences between $\mathcal{S}_1^u$ and $\mathcal{S}_2^u$.**
   - $Y_U^* = kY_U Q$.
   - For each element x in X_U, its correspondence in $Y_U^* = \arg\min_{y^* \in Y_U^*} \|y^* - x\|$.

Depending on the approach that we want to use, there are several variations of Step 1. For example, if we are using PCA, then we use the covariance matrices instead of Laplacian matrices; similarly, if we are using LPP (He et al., 2003), then we construct the weight matrices W_1^l for $\mathcal{D}_1^l$, W_2^l for $\mathcal{D}_2^l$ using K_i and then learn the projections. Note that when PCA or LPP is used, then the low dimensional embedding will be defined everywhere rather than just on the training points.

3. Justification

In this section, we prove two theorems. Theorem 1 shows why the algorithm is valid. Given the fact that dimensionality reduction approaches play a key role in our approach, Theorem 2 provides a theoretical bound for the difference between subspaces spanned by low dimensional embeddings of the two data sets. This bound analytically characterizes when the two data sets can be aligned well.

3.1. Optimal Manifold Alignment

Procrustes analysis seeks the isotropic dilation and the rigid translation, reflection and rotation needed to best match one data configuration to another (Cox et al., 2001). Given low dimensional embeddings X and Y (defined in Section 2), the most convenient way to do translation is to translate the configurations in X and Y so that their centroids are at the origin. Then the problem is simplified as: finding Q and k so that $\|X - kYQ\|_F$ is minimized, where $\|\cdot\|_F$ is Frobenius norm. The matrix Q is orthonormal, giving a rotation and possibly a reflection, k is a re-scale factor to either stretch or shrink Y. Below, we show that the optimal solution is given by the *SVD* of Y^TX. A detailed review of Procrustes analysis can be found in (Cox et al., 2001).

Theorem 1: Let X and Y be low dimensional embeddings of the points with known correspondences in data set $\mathcal{S}_1$, $\mathcal{S}_2$, and X_i matches Y_i for each i. If Singular Value Decomposition (*SVD*) of Y^TX is $U\Sigma V^T$, then $Q = UV^T$ and $k = trace(\Sigma)/trace(Y^TY)$ minimize $\|X - kYQ\|_F$.
Proof:
The problem is formalized as:
$\{k_{opt}, Q_{opt}\} = \arg\min_{k,Q} \|X - kYQ\|_F$. (1.1)

It is easy to verify that
$\|X - kYQ\|_F^2 = trace(X^TX) + k^2 \cdot trace(Y^TY) - 2k \cdot trace(Q^TY^TX)$. (1.2)

Since $trace(X^TX)$ is a constant, the minimization problem is equivalent to $\{k_{opt}, Q_{opt}\} = \arg\min_{k,Q}(k^2 \cdot trace(Y^TY) - 2k \cdot trace(Q^TY^TX))$. (1.3)

Differentiating with respect to k, we have
$2k \cdot trace(Y^TY) = 2 \cdot trace(Q^TY^TX)$,
i.e. $k = trace(Q^TY^TX)/trace(Y^TY)$. (1.4)

(1.3) and (1.4) show that the minimization problem reduces to $Q_{opt} = \arg\max_Q (trace(Q^TY^TX))^2$. (1.5)

Case 1:
If $trace(Q^TY^TX) \geq 0$, then the problem becomes $Q_{opt} = \arg\max_Q \ trace(Q^TY^TX)$. (1.6)

Using Singular Value Decomposition, we have $Y^TX = U\Sigma V^T$, where U and V are orthonormal, and Σ is a diagonal matrix having as its main diagonal all the positive singular values of Y^TX. So $\max_Q trace(Q^TY^TX) = \max_Q trace(Q^TU\Sigma V^T)$. (1.7)

It is well known that for two matrices A and B, $trace(AB) = trace(BA)$, so $\max_Q trace(Q^TU\Sigma V^T) = \max_Q trace(V^TQ^TU\Sigma)$. (1.8)

For simplicity, we use Z to represent V^TQ^TU. We know Q, U and V are all orthonormal matrices, so Z is also orthonormal. It is well known that any element in an orthonormal matrix, say B, is in [-1,1] (otherwise B^TB is not an identity matrix). So we know $trace(Z\Sigma) = Z_{1,1}\Sigma_{1,1} + \cdots + Z_{c,c}\Sigma_{c,c} \leq \Sigma_{1,1} + \cdots + \Sigma_{c,c}$ (1.9) , which implies $Z = I$ maximizes $trace(Z\Sigma)$, where I is an identity matrix. (1.10)

Obviously, the solution to $Z = I$ is $Q = UV^T$. (1.11)

Case 2:
If $trace(Q^TY^TX) < 0$, then the problem becomes $Q_{opt} = \arg\min_Q \ trace(Q^TY^TX)$. (1.12)

Following the similar procedure shown above, we have $trace(Z\Sigma) = Z_{1,1}\Sigma_{1,1} + \cdots + Z_{c,c}\Sigma_{c,c} \geq -\Sigma_{1,1} - \cdots - \Sigma_{c,c}$ (1.13) , which implies that $Z = -I$ minimizes $trace(Z\Sigma)$. (1.14)

Obviously, the solution to $Z = -I$ is $Q = -UV^T$. (1.15)

Considering (1.5), it is easy to verify that $Q = UV^T$ and $Q = -UV^T$ return the same results, so $Q = UV^T$ is always the optimal solution to (1.5), no matter whether $trace(Q^TY^TX)$ is positive or not. Further, we can simplify (1.4), and have $k = trace(\Sigma)/trace(Y^TY)$. (1.16) □

3.2. Theoretical Analysis

Many dimensionality reduction approaches first compute a relationship matrix, and then project the data onto a subspace spanned by the "top" eigenvectors of the matrix. The "top" eigenvectors mean some subset of eigenvectors that are of interest. They might be eigenvectors corresponding to largest, smallest, or

A is a $N \times N$ relationship matrix computed from $\mathcal{S}_1$.
B is a $N \times N$ relationship matrix computed from $\mathcal{S}_2$.
$E = B - A$.

$\mathcal{X}$ denotes a subspace of the column space of A spanned by top M eigenvectors of A.
$\mathcal{Y}$ denotes a subspace of the column space of B spanned by top M eigenvectors of B.
X is a matrix whose columns are an orthonormal basis of $\mathcal{X}$.
Y is a matrix whose columns are an orthonormal basis of $\mathcal{Y}$.

δ_A^1 is the set of top M eigenvalues of A, δ_A^2 includes all eigenvalues of A except those in δ_A^1.
δ_B^1 is the set of top M eigenvalues of B, δ_B^2 includes all eigenvalues of B except those in δ_B^1.

d_1 is the eigengap between δ_A^1 and δ_A^2, i.e. $d_1 = \min_{\lambda_i \in \delta_A^1, \lambda_j \in \delta_A^2} |\lambda_i - \lambda_j|$.
$d = \delta_A^1 - \delta_B^2$.

P denotes the orthogonal projection onto subspace $\mathcal{X}$.
Q denotes the orthogonal projection onto subspace $\mathcal{Y}$.

$\| \cdot \|$ denotes *Operator Norm*, i.e. $\|L\|_{\mu,\nu} = \max_{\nu(x)=1} \mu(Lx)$, where μ, ν are simply $\| \cdot \|_2$.

Figure 1. Notation used in Theorem 2.

even arbitrary eigenvalues. One example is Laplacian eigenmap, where we project the data onto the subspace spanned by the "smoothest" eigenvectors of the graph Laplacian. Another example is PCA, where we project the data onto the subspace spanned by the "largest" eigenvectors of the covariance matrix. In this section, we study the general approach, which provides a general framework for each individual algorithm such as Laplacian eigenmap. We assume the two given data sets $\mathcal{S}_1$ and $\mathcal{S}_2$ do not differ significantly, so the related relationship matrices A and B are "very similar". We study the difference between the embedding subspaces corresponding to the two relationship matrices. Notation used in the proof is in Figure 1. The difference between orthogonal projections $\|Q-P\|$ characterizes the distance between the two subspaces. The proof of the theorem below is based on the perturbation theory of spectral subspaces, where $E = B-A$ can be thought as the perturbation to A. The only assumption we need to make is for any i and j, $|E_{i,j}| = |B_{i,j} - A_{i,j}| \leq \tau$.

Theorem 2: If the absolute value of each element in E is bounded by τ, and $\tau \leq 2\varepsilon d_1/(N(\pi + 2\varepsilon))$, then the difference between the two embedding subspaces $\|Q - P\|$ is at most ε.

Proof:
From the definition of operator norm, we know $\|E\| = \max_{k_1,k_2,\cdots,k_N} \sqrt{\sum_i (\sum_j k_j E_{i,j})^2}$, given $\sum_i k_i^2 = 1$. (2.1)

We can verify the following inequality always holds: $\sum_i (\sum_j k_j E_{i,j})^2 \leq N \sum_j k_j^2 \sum_i E_{i,j}^2$. (2.2)

From (2.1) and (2.2), we have $\sum_i (\sum_j k_j E_{i,j})^2 \leq N^2\tau^2 \sum_j k_j^2 = N^2\tau^2$. (2.3)

Combining (2.1) and (2.3), we have: $\|E\| \leq N\tau$. (2.4)

It can be shown that if A and E are bounded self-adjoint operators on a separable Hilbert space, then the spectrum of $A+E$ is in the closed $\|E\|$-neighborhood of the spectrum of A (Kostrykin et al., 2003). From (Kostrykin et al., 2003), we also have the following inequality: $\|Q^\perp P\| \leq \pi\|E\|/2d$. (2.5)

We know A has an isolated part δ_A^1 of the spectrum separated from its remainder δ_A^2 by gap d_1. To guarantee $A + E$ also has separated components, we need to assume $\|E\| < d_1/2$. Thus (2.5) becomes $\|Q^\perp P\| \leq \pi\|E\|/2(d_1 - \|E\|)$. (2.6)

Interchanging the roles of δ_A^1 and δ_A^2, we have the analogous inequality: $\|QP^\perp\| \leq \pi\|E\|/2(d_1 - \|E\|)$. (2.7)

Since $\|Q - P\| = max\{\|Q^\perp P\|, \|QP^\perp\|\}$ (2.8), we have $\|Q - P\| \leq \pi\|E\|/2(d_1 - \|E\|)$. (2.9)

We define $R = Q - P$, and from (2.9), we get $\|R\| \leq \pi\|E\|/2(d_1 - \|E\|)$. (2.10)

(2.10) implies that if $\|E\| \leq 2d_1\varepsilon/(2\varepsilon + \pi)$, then $\|R\| \leq \varepsilon$. (2.11)

So we have the following conclusion: if the absolute value of each element in E is bounded by τ, and $\tau \leq 2\varepsilon d_1/(N(\pi + 2\varepsilon))$, then the difference of the subspaces spanned by top M eigenvectors of A and B is at most ε. □

Theorem 2 tells us that if the eigengap (between δ_A^1 and δ_A^2) is large, then the subspace corresponding to the top M eigenvectors of A is insensitive to perturbations. In other words, the algorithm can tolerate larger differences between A and B. So when we are selecting eigenvectors to form a subspace, the eigengap is an important factor to be considered. The reasoning behind this is that if the magnitudes of the relevant eigenval-

ues do not change too much, the top M eigenvectors will not be overtaken by other eigenvectors, thus the related space is more stable. Our result in essence connects the difference between the two relationship matrices to the difference between the subspaces spanned by their low dimensional embeddings.

4. Applications and Results

In this section, we first use a toy example to illustrate how our algorithm works, then we apply our approach to transfer knowledge from one domain to another. We present results applying our approach to two real world problems: cross-lingual information retrieval and transfer learning in Markov decision processes (MDPs).

4.1. A Toy Example

In this example, we directly align two manifolds and use some pictures to illustrate how our algorithm works. The two manifolds come from real protein tertiary structure data.

Protein 3D structure reconstruction is an important step in Nuclear Magnetic Resonance (NMR) protein structure determination. Basically, it finds a map from distances to coordinates. A protein 3D structure is a chain of amino acids. Let n be the number of amino acids in a given protein and $C_1, \cdots, C_n$ be the coordinate vectors for the amino acids, where $C_i = (C_{i,1}, C_{i,2}, C_{i,3})^T$ and $C_{i,1}, C_{i,2}$, and $C_{i,3}$ are the x, y, z coordinates of amino acid i (in biology, one usually uses atom but not amino acid as the basic element in determining protein structure. Since the number of atoms is huge, for simplicity, we use amino acid as the basic element). Then the distance $d_{i,j}$ between amino acids i and j can be defined as $d_{i,j} = \|C_i - C_j\|$. Define $A = \{d_{i,j}, i, j = 1, \cdots, n\}$, and $C = \{C_i, i = 1, \cdots, n\}$. It is easy to see that if C is given, then we can immediately compute A. However, if A is given, it is non-trivial to compute C. The latter problem is called Protein 3D structure reconstruction. In fact, the problem is even more tricky, since only the distances between neighbors are reliable, and this makes A an incomplete distance matrix. The problem has been proved to be NP-complete for general sparse distance matrices (Hogben, 2006). In real life, people use other techniques, such as angle constraints and human experience, together with the partial distance matrix to determine protein structures.

With the information available to us, NMR techniques might find multiple estimations (models), since more than one configuration can be consistent with the distance matrix and the constraints. Thus, the result is an ensemble of models, rather than a single structure. Most usually, the ensemble of structures, with perhaps 10 - 50 members, all of which fit the NMR data and retain good stereochemistry is deposited with the Protein Data Bank (PDB) (Berman et al., 2000). Models related to the same protein should be similar and comparisons between the models in this ensemble provides some information on how well the protein conformation was determined by NMR.

In this test, we study a Glutaredoxin protein PDB-1G7O (this protein has 215 amino acids in total), whose 3D structure has 21 models. Since such models are already low dimensional (3D) embeddings of the distance matrices, we skip Step 1 and 2 in our algorithm. We pick up Model 1 and Model 21 for test. These two models are related to the same protein, so it makes sense to treat them as manifolds to test our techniques. We denote Model 1 by Manifold A, which is represented by matrix S_1. We denote Model 21 by Manifold B, which is represented by matrix S_2. Obviously, both S_1 and S_2 are 215×3 matrices. To evaluate our re-scale factor, we manually stretch manifold A by letting $S_1 = 4 \cdot S_1$. Manifold A and B (row vectors of S_1 and S_2 represent points in the 3D space) are shown in Figure 2(A) and Figure 2(B). In biology, such chains are called protein backbones. For the purpose of comparison, we also plot both manifolds on the same graph (Figure 2(C)). It is clear that manifold A is much larger than B, and the orientations of A and B are quite different.

To align the two manifolds, we uniformly selected 1/4 amino acids as correspondence resulting in matrix X and Y, where row i of X (from S_1) matches row i of Y (from S_2) and both X and Y are 54×3 matrices. We run our algorithm from Step 3. Our algorithm identifies the re-scale factor k as 4.2971, and the rotation matrix Q as

$$\mathbf{Q} = \begin{pmatrix} 0.56151 & -0.53218 & 0.63363 \\ 0.65793 & 0.75154 & 0.048172 \\ -0.50183 & 0.38983 & 0.77214 \end{pmatrix}.$$

S_2^*, the new representation of S_2, is computed as $S_2^* = kS_2Q$. We plot S_2^* and S_1 in the same graph (Figure 2(D)). The result shows that Manifold B is rotated and enlarged to the similar size as A, and now the two manifolds are aligned very well.

4.2. Cross-lingual Information Retrieval

In information retrieval, manifold alignment can be used to find correspondences between documents. One example is finding the exact correspondences between documents in different languages. Such systems are quite useful, since they allow users to query a docu-

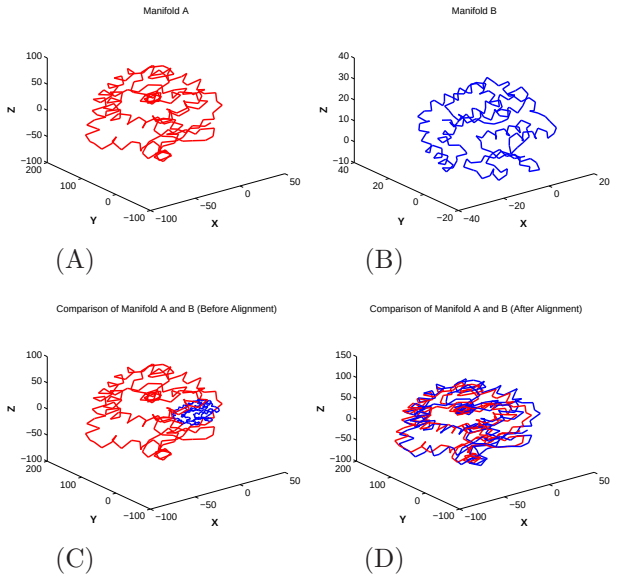


Figure 2. (A): Manifold A; (B): Manifold B; (C): Comparison of Manifold A(red) and B(blue) before alignment; (D): Comparison of Manifold A(red) and B(blue) after alignment.

ment in their native language and retrieve documents in a foreign language. Assume that we are given two document collections. For example, one in English and one in Arabic. We are also given some training correspondences between documents that are exact translations of each other. The task is: for each English or Arabian document in the untranslated set, to find the most similar document in the other corpus.

We apply our manifold alignment approach to this problem. The topical structure of each collection can be thought as a manifold over documents. Each document is a sample from the manifold. We are interested in the case where the underlying topical manifolds of two languages are similar. Our procedure for aligning collections consists of two steps: learning low dimensional embeddings of the two manifolds and aligning the low dimensional embeddings. To compute similarity of two documents in the same collection, we assume that document vectors are language models (multinomial term distributions) estimated using the document text. By treating documents as probability distributions, we can use distributional affinity to detect topical relatedness between documents. More precisely, a multinomial diffusion kernel is used for this particular application. The kernel used here is the same as the one used in (Diaz et al., 2007), where more detailed description is provided. Dimensionality reduction approaches are then used to learn the low dimensional embeddings. After shifting the centroids of the documents in each collection to the origin point, we apply our approach to learn the re-scale factor k and rotation Q from the training correspondences and then apply them to the untranslated set.

In our experiments, we used two document collections (one in English, one in Arabic, manually translated), each of which has 2119 documents. Correspondences between 25% of them were given and used to learn the mapping between them. The remaining 75% were used for testing. We used Laplacian eigenmap and LPP (the projection was learned from the data points in the correspondence) to learn the low dimensional embeddings, where top 100 eigenvectors were used to construct the embeddings. Our testing scheme is as follows: for each given Arabic document, we retrieve its top j most similar English documents. The probability that the true match is among this top j documents is used to show the goodness of the method. We also used the same data set to test the semi-supervised manifold alignment method proposed in (Ham et al., 2005), where top 100 eigenvectors were used for low dimensional embeddings. A fourth method (called baseline method) was also tested. The baseline method is as follows: assume that we have m correspondences in the training set, then document x is represented by a vector V with length m, 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 - $\mathcal{R}^m$. Experiment results are shown in Figure 3.

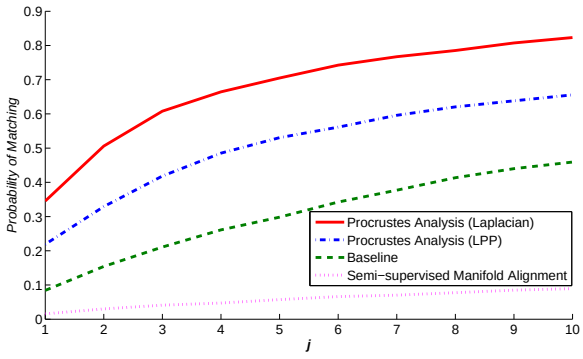


Figure 3. Cross-lingual information retrieval test.

Compared to semi-supervised manifold alignment method, the performance of Prucrustes (with Laplacian eigenmap) is significantly better. For each given Arabic document, if we retrieve 3 most relevant English documents, then the true match has a 60% probability of being among the 3. If we retrieve 10 most relevant English documents, then we have about 80% probability of getting the true match. Further, our method is much faster. Semi-supervised manifold

alignment method requires solving an eigenvalue problem over a $(n_1 + n_2 - m) \times (n_1 + n_2 - m)$ matrix, where n_i is the total number of the documents in collection i, and m is the number of training correspondences. Using our approach, the most time consuming step is finding the low dimensional embeddings with Laplacian eigenmap, which requires solving eigenvalue problems over a $n_1 \times n_1$ matrix and a $n_2 \times n_2$ matrix. We also compute the *SVD* over a $d \times d$ matrix, where d is the dimension of the low dimensional embeddings and is usually much smaller than n. In the experiments, Procrustes (with Laplacian eigenmap) is roughly 2 times faster than semi-supervised manifold alignment. Procrustes (with LPP) also returns reasonably good results: if we retrieve 10 most relevant English documents, then we have a 60% probability of getting the true match. Procrustes (with LPP) results in a mapping that is defined everywhere rather than just on the training data points and it also requires less time. Another interesting result is that the baseline algorithm also performs quite well, and better than semi-supervised alignment method. One reason that semi-supervised manifold alignment method is not working well is that mappings of the corresponding points are constrained to be identical. This might lead to "over fitting" problems for some applications.

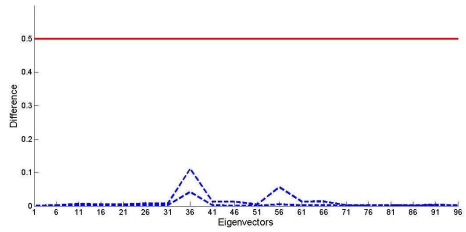
4.3. Transfer Learning in Markov Decision Process

Transfer learning studies how to re-use knowledge learned from one domain or task to a related domain or task. In this section, we investigate transfer learning in Markov decision processes (MDPs) following the approach of "proto-value functions" (PVFs), where the Laplacian eigenmap method is used to construct basis functions (Mahadevan, 2005). In a MDP, a value function is a mapping from states to real numbers, where the value of a state represents the long-term reward achieved starting from that state, and executing a particular policy. PVFs are an orthonormal basis spanning all value functions of an MDP on a state space manifold. They are computed as follows: First, create a weight matrix that reflects the topology of the state space using a series of random walks; Second, compute the graph Laplacian of the weight matrix; Third, select the smoothest k eigenvectors of this graph Laplacian as PVFs. If the state space is the same and only the reward function is changed, then the PVFs can be directly transferred to the new domain. One interesting question related to PVFs is how to transfer the old PVFs to a new domain when the new state space is only slightly different from the old one. In this section, we answer this question with our techniques.

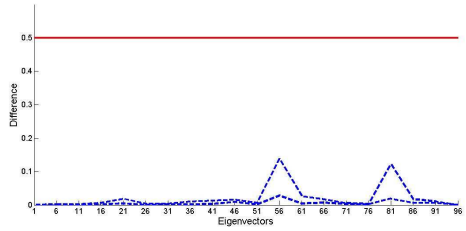
Let columns of Y denote PVFs of the current MDP. Given the procedure on how to generate PVFs, we know the rows of Y are also the low dimensional representations of the data points on the current state space manifold. Let rows of X represent the low dimensional embedding of the new manifold. Assume centroids of both X and Y are at the origin. By using isotropic dilation, reflection and rotation to align the two state space manifolds, we may find the optimal k and Q such that the two manifolds are aligned well. Our argument is that the new PVFs are YQ. The reason is as follows: suppose we have already found the optimal k and Q that minimize $\|X - kYQ\|_F$, then Y will be changed to kYQ in the process of alignment. k can be skipped, since it is well known that kYQ and YQ span the same space. The only thing that we need to show is the columns of YQ are orthonormal to each other (a requirement of PVFs). The proof is quite simple: $(YQ)^T YQ = Q^T Y^T YQ = Q^T IQ = I$, where I is an identity matrix. This means different columns of YQ are orthogonal to each other and norm of each column is 1, so YQ is orthonormal.

The conclusion shown above works when two state space manifolds are similar. Here, we still need to answer one more question: "under what conditions are the two manifolds similar?". Theorem 2 provides an answer to this question. Theorem 2 numerically bounds the difference between two spaces given the difference between the relevant relationship matrices. For this case, the relationship matrices are the Laplacian matrices used to model the state spaces. In this test, we run experiments to verify the bound. We investigate two reinforcement learning tasks. The inverted pendulum task requires balancing a pendulum of unknown mass and length by applying force to a cart attached to the pendulum. The state space is defined by two variables: the vertical angle of the pendulum, and the angular velocity of the pendulum. The mountain car task is to get a simulated car to the top of a hill as quickly as possible. The car does not have enough power to get there immediately, and so must oscillate on the hill to build up the necessary momentum. The state space is the position and velocity of the car.

We first generate two different sets of sampled states for the pendulum task and compute their related normalized graph Laplacian matrices A and B. We compute the top i non-trivial eigenvectors of A and B, and directly compute the difference between the spaces spanned by them. Theorem 2 says if the absolute value of each element in $A - B$ is bounded by τ, and $\tau \leq 2\varepsilon d_1/(N(\pi+2\varepsilon))$, then the difference of the spaces spanned by top i eigenvectors of A and B is at most



(A) Pendulum Task



(B) Mountain Car Task

Figure 4. (A): Bound for Pendulum task. (B): Bound for Mountain car task. For both tasks, ε is 0.5, true values (Max and Min in 5 tests) of the difference between two spaces are in dotted lines.

ε. We set ε be 0.5, and let τ be $\varepsilon d_1/(N(\pi+2\varepsilon))$. Here d_1 is the eigengap between top i eigenvectors and the other eigenvetors, N is 500. Based on our theorem, the difference between spaces should not be larger than ε. In our experiments, we tried 20 different values for i=1, 6, 11, $\cdots$, 96. For each i, we ran 5 tests. We carried out the same experiment on the Mountain Car task. Figure 4(A) and 4(B) respectively show the results from Pendulum task and Mountain car task. For each figure, we plot ε and the maximum and minimum difference values of the 5 tests for various values of i. For this application, the bound is loose, but the bound given in Theorem 2 is a general theoretical bound and for other applications, it might be tight. We also empirically evaluate the PVFs transfer performance. The results (not included) show that we can learn a good policy by using PVFs from a similar domain.

5. Conclusions

In this paper we introduce a novel approach to manifold alignment based on Procrustes Analysis. When used with a suitable dimensionality reduction method, our approach results in a mapping defined everywhere rather than just on the training data points. We also study the conditions under which low dimensional embeddings of two data sets can be aligned well. We presented novel applications of our approach, including cross-lingual information retrieval and transfer learning in Markov decision processes.

Acknowledgments

We thank the reviewers for their helpful comments. This project was supported in part by the National Science Foundation under grant IIS-0534999.

References

Belkin, M., Niyogi, P. (2003) Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation* 15.

Bengio, Y. et al. (2004) Out-of-Sample Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering. *NIPS* 16.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N. Weissig, H., Shindyalov, I. N., Bourne, P. E. (2000) The protein data bank. *Nucleic Acids Research*, 28:235–242.

Cox, M. F., Cox, M. A. A. (2001) Multidimensional scaling. Chapman and Hall.

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

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

He, X., Niyogi, P. (2003) Locality preserving projections. *The Annual Conference on Neural Information Processing Systems (NIPS) 16.*

Hogben, L. (2006) Handbook of linear algebra. Chapman/Hall CRC Press.

Kostrykin, V., Makarov, K. A., Motovilov, A. K. (2003) On a subspace perturbation problem. *Proc. of the American Mathematical Society.* 131:3469-3476.

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

Luo, B., Hancock, W.R. (1999) Feature matching with Procrustes alignment and graph editing. *7th International Conference on Image Processing and its Applications.*

Mahadevan, S. (2005) Proto-value functions: developmental reinforcement learning. *The 22nd International Conference on Machine Learning (ICML).*