Semisupervised alignment of manifolds

Jihun Ham
Department of Electrical and
Systems Engineering,
University of Pennsylvania,
Philadelphia, PA 19104

Daniel D. Lee
Department of Electrical and
Systems Engineering,
University of Pennsylvania,
Philadelphia, PA 19104

Lawrence K. Saul
Department of Computer and
Information Science,
University of Pennsylvania,
Philadelphia, PA 19104

Abstract

In this paper, we study a family of semisupervised learning algorithms for "aligning" different data sets that are characterized by the same underlying manifold. The optimizations of these algorithms are based on graphs that provide a discretized approximation to the manifold. Partial alignments of the data sets—obtained from prior knowledge of their manifold structure or from pairwise correspondences of subsets of labeled examples—are completed by integrating supervised signals with unsupervised frameworks for manifold learning. As an illustration of this semisupervised setting, we show how to learn mappings between different data sets of images that are parameterized by the same underlying modes of variability (e.g., pose and viewing angle). The curse of dimensionality in these problems is overcome by exploiting the low dimensional structure of image manifolds.

1 Introduction

Examples of very high-dimensional data such as high-resolution pixel images or large vector-space representations of text documents abound in multimodal data sets. Learning problems involving these data sets are difficult due to the curse of dimensionality and associated large computational demands. However, in many cases, the statistical analysis of these data sets may be tractable due to an underlying low-dimensional manifold structure in the data. Recently, a series of learning algorithms that approximate data manifolds have been developed, such as Isomap [15], locally linear embedding [13], Laplacian eigenmaps [3], Hessian eigenmaps [7], and charting [5]. While these algorithms approach the problem of learning manifolds from an unsupervised perspective; in this paper, we address the problem of establishing a regression between two or more data sets by aligning their underlying manifolds. We show how to align the low-dimensional representations of the data sets given some additional information about the mapping between the data sets. Our algorithm relies upon optimization over a graphical representation of the data, where edges in the graphs are computed to preserve local structure in the data. This optimization yields a common low-dimensional embedding which can then be used to map samples between the disparate data sets.

Two main approaches for alignment of manifolds are presented. In the first approach, additional knowledge about the intrinsic embedding coordinates of some of the samples are used to constrain the alignment. This information about coordinates may be available given knowledge about the data generating process, or when some coordinates are manually assigned to correspond to certain labeled samples. Our algorithm yields a graph embedding where these known coordinates are preserved. Given multiple data sets with such coordinate labels, we show how the underlying data manifolds can be aligned to each other through a common set of coordinates.

In the second approach, we assume that there is no prior knowledge of explicit coordinates, but that we know the pairwise correspondences of some of the samples [11, 16]. These correspondences may be apparent from temporal conjunction, such as simultaneously obtained images and sounds from cameras and microphones. Correspondences may also be obtained from hand-labeled matches among samples in different data sets. We demonstrate how these correspondences allow implicit alignment of the different data manifolds. This is achieved by joining the graph representations of the different data sets and estimating a common low-dimensional embedding over the joined graph.

In Section 2 we first review a graph-based framework for manifold learning algorithms. Section 3 describes our algorithms for manifold alignment using either prior coordinate knowledge or paired correspondences. Section 4 demonstrates the application of our approach to aligning the pose manifolds of images of different objects. Finally, the utility and future direction of this approach is discussed in Section 5.

2 Unsupervised manifold learning with graphs

Let X and Y be two data sets in high dimensional vector spaces

$$X = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m\} \subset \mathbb{R}^{D_X}, \;\; Y = \{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_n\} \subset \mathbb{R}^{D_Y},$$

with $D_X, D_Y \gg 1$. When the data lie close to a low-dimensional manifold embedded in a high dimensional Euclidean space, manifold learning algorithms such as [3] can successfully learn low-dimensional embeddings by constructing a weighted graph that captures local structure in the data. Let $G(\mathcal{V}, \mathcal{E})$ be the graph where the vertices $\mathcal{V}$ correspond to samples in the data and the undirected edges $\mathcal{E}$ denote neighborhood relationships between the vertices. These neighborhood relations can be defined in terms of k-nearest neighbors or an ϵ-ball distance criterion in the Euclidean space of original data. The similarities between points are summarized by a weight matrix W where $W_{ij} \neq 0$ when the ith and jth data points are neighbors ($i \sim j$), otherwise $W_{ij} = 0$. The matrix W is typically symmetric, and has nonnegative weights $W_{ij} = W_{ji} \geq 0$. The generalized graph Laplacian L is then defined as:

$$L_{ij} := \begin{cases} d_i, & \text{if } i = j, \\ -W_{ij}, & \text{if } i \sim j, \\ 0, & \text{otherwise} \end{cases}$$

where $d_i = \sum_{j \sim i} W_{ij}$ is the degree of the ith vertex. If the graph is connected, L will have a single zero eigenvalue associated with the uniform vector $\boldsymbol{e} = [11 \cdots 1]^T$.

A low-dimensional embedding of the data can be computed from the graph Laplacian in the following manner. A real valued function $\boldsymbol{f} : \mathcal{V} \mapsto \mathbb{R}$ on the vertices of the graph is associated with the cost:

$$\boldsymbol{f}^T L \boldsymbol{f} = \frac{1}{2} \sum_{i,j} (f_i - f_j)^2 W_{ij}. \quad (1)$$

An optimal embedding is given by functions $\boldsymbol{f}$ that minimize (1), subject to scale and translation constraints $\boldsymbol{f}^T \boldsymbol{f} = 1$ and $\boldsymbol{f}^T \boldsymbol{e} = 0$. These solutions are then the eigenvectors of L with the smallest non-zero eigenvalues [8]. These solutions may also be interpreted as the kernel principal components of a Gram matrix given by the pseudoinverse of L [10]. This interpretation defines a metric over the graph which is related to the commute times of random walks on the graph [1], and resistance distance in electrical networks [6].

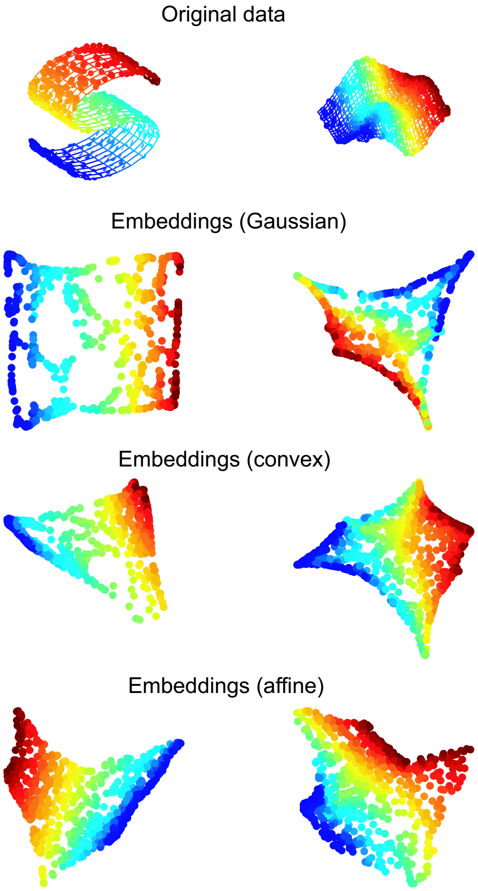

Figure 1: Two-dimensional embeddings of surfaces in $\mathbb{R}^3$. The embeddings are computed from diagonalizing the graph Laplacians. Different edge weightings yield qualitative differences in the embeddings. Only 600 and 800 points were sampled from the two manifolds, making it difficult for the algorithms to find a faithful embedding of the data.

Choice of weights

Within this graph framework, different algorithms may employ different choices for the weights W. For example, W can be defined according to the Gaussian process $W_{ij} = e^{-|\boldsymbol{x}_i - \boldsymbol{x}_j|^2/2\sigma^2}$, and is related to a diffusion process on the graph [3, 12]. The symmetric, nonnegative assumptions on the weights $W_{ij} = W_{ji} \geq 0$ can be relaxed. For a directed graph structure, such as when the neighborhoods are determined by k-nearest neighbors, the matrix W is not symmetric. Nonnegativity constraints may also be lifted. Consider the least-squares approach to optimize weights W_{ij} :

$$W_{ij} = \arg\min_W \ |\boldsymbol{x}_i - \sum_{j\sim i} W_{ij}\boldsymbol{x}_j|^2, \tag{2}$$

that is, W_{ij} are the coefficients of the neighbors of $\boldsymbol{x}_i$ that best approximates $\boldsymbol{x}_i$, and are in general asymmetric. Locally linear embedding determines weights from minimizing (2) subject to $\sum_j W_{ij} = 1$, yielding possibly negative coefficients that best approximates $\boldsymbol{x}_i$ from an affine combination of its neighbors [14]. This is in contrast to minimizing (2) over a set of convex coefficients that are nonnegative: $W_{ij} \geq 0$. As noted in [14], a possible disadvantage of convex approximation is that a point on the boundary may not be reconstructed from the convex hull of its neighbors. Consequently, the corners of the resultant embedding with convex weights tend to be rounded.

Graph Laplacians with negative weights have been recently studied [9, 10]. Although it is difficult to properly generalize spectral graph theory, we can define a new cost function analogous to (1) for graphs with asymmetric, negative weights as:

$$\boldsymbol{f}^T L^T L \boldsymbol{f} = \sum_i |f_i - \sum_{j\sim i} W_{ij} f_j|^2, \tag{3}$$

where $L = D - W$. Since $L^T L$ is positive semidefinite and satisfies $L^T L\boldsymbol{e} = 0$, the eigenvectors of $L^T L$ can be used to construct a low-dimensional embedding of the graph that minimizes the cost (3).

Figure 1 shows the unsupervised graph embedding of two artificial data sets using three different weighting schemes: a symmetric Gaussian, asymmetric convex reconstruction, and asymmetric affine reconstruction weights. 600 points were sampled from an S-shaped two-dimensional manifold, and 800 points were sampled from a wavy two-dimensional manifold. The data was intentionally undersampled, and the unsupervised learning algorithms have difficulty in faithfully reconstructing the proper embedding. In the next sections, we will show how semisupervised approaches can greatly improve on these embeddings with the same data.

3 Semisupervised alignment of manifolds

We now consider aligning disparate data manifolds, given some additional information about the data samples. In the following approaches, we consider this additional information to be given for only a partial subset of the data. We denote the samples with this additional "labeled" information by the ordinal index l, and the samples without extra information by the index u. We also use the same notation for the sets X and Y; for example, X_l and Y_l refer to the "labeled" parts of X and Y.

This additional information about the data samples may be of two different types. In the first algorithm, the labels refer to prior information about the intrinsic real-valued coordinates within the manifold for particular data samples. In the second algorithm, the labels indicate pairwise correspondences between samples $\boldsymbol{x}_i \in X$ and $\boldsymbol{y}_j \in Y$. These two types of additional information are quite different, but we show how each can be used to align the different data manifolds.

3.1 Alignment with given coordinates

In this approach, we are given desired coordinates for certain labeled samples. Similar to regression models, we would like to find a map defined on the vertices of the graph $f : V \mapsto \mathbb{R}$ that matches known target values for the labeled vertices. This can be solved by finding $\arg\min_f \ |f_i - s_i|^2 \ (i \in l)$ where $\boldsymbol{s}$ is the vector of target values. With a small number of labeled examples, it is crucial to exploit manifold structure in the data when constructing the class of admissible functions $\boldsymbol{f}$. The symmetric graph Laplacian $L = L^T$ provides this information. A regularized regression cost on a graph is defined as:

$$C(\boldsymbol{f}) = \sum_i \mu|f_i - s_i|^2 + \boldsymbol{f}^T L\boldsymbol{f}. \tag{4}$$

The first term in (4) is the fitting error, and the second term enforces smoothness along the manifold by $\boldsymbol{f}^T L\boldsymbol{f} \approx \sum_i |\nabla_i \boldsymbol{f}|^2$ [2, 17]. The relative weighting of these terms is given by the coefficient μ. The optimum $\boldsymbol{f}$ is then obtained by the linear solution:

$$\boldsymbol{f} = \begin{pmatrix} \mu I + L_{ll} & L_{lu} \\ L_{ul} & L_{uu} \end{pmatrix}^{-1} \begin{pmatrix} \mu I \\ 0 \end{pmatrix} \boldsymbol{s}, \tag{5}$$

where L consists of labeled and unlabeled partitions:

$$L = \begin{bmatrix} L_{ll} & L_{lu} \\ L_{ul} & L_{uu} \end{bmatrix}.$$

In the limit $\mu \to \infty$, i.e. there is no uncertainty in the labels $\boldsymbol{s}$, the solution becomes

$$\boldsymbol{f}_u = -(L_{uu})^{-1} L_{ul}\boldsymbol{s} = (L_{uu})^{-1} W_{ul}\ \boldsymbol{s}. \tag{6}$$

This result is directly related to harmonic functions [18], which are smooth functions on the graph such that f_i is determined by the average of its neighbors:

$$f_i = \frac{\sum_j W_{ij} f_j}{\sum_j W_{ij}}. \tag{7}$$

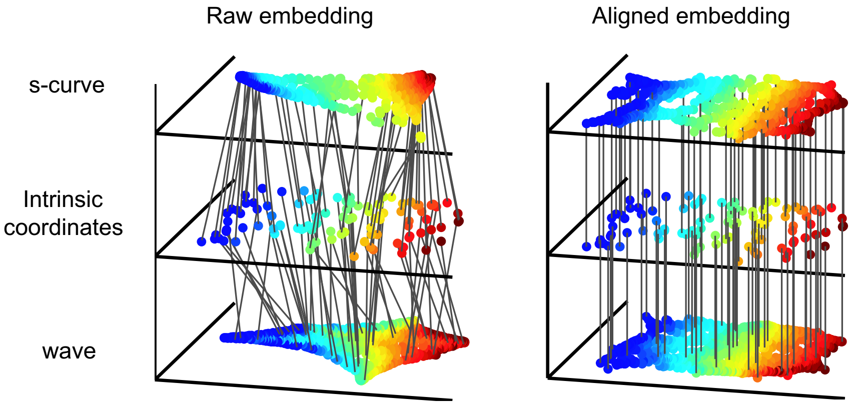

Figure 2: Graph embeddings for the s-curve and wave surface are aligned with given coordinates, and compared to the unaligned embeddings. The lines indicate samples whose known coordinates are used to estimate a common embedding space.

The solution in (6) is a linear superposition of harmonic functions which directly interpolate the labeled data.

Given r-dimensional coordinate vectors $S = [\boldsymbol{s}_1 \cdots \boldsymbol{s}_r]$ as desired embedding coordinates, solutions $\boldsymbol{f}_i$ of (5) or (6) can be used as estimated coordinates of unlabeled data. This "stretches" the embedding of the graph so that the labeled vertices are at the desired coordinates. Figure 3 shows the results of this algorithm applied to an image manifold with two-dimensional pose parameters as coordinates. Simultaneous alignment of two different data sets is performed by simply mapping each of the data sets into a common space with known coordinates. Given two data sets X and Y, where subsets X_l and Y_l are given coordinates $\boldsymbol{s}$ and $\boldsymbol{t}$ respectively, we let $\boldsymbol{f}$ and $\boldsymbol{g}$ denote real-valued functions, and L^x and L^y the graph Laplacians of X and Y respectively. Since there is no explicit coupling between X and Y, we use (6) to get the two solutions:

$$\boldsymbol{f}_u = -(L^x_{uu})^{-1} L^x_{ul}\, \boldsymbol{s}, \text{ and } \boldsymbol{g}_u = -(L^y_{uu})^{-1} L^y_{ul}\, \boldsymbol{t}.$$

Figure 2 shows the semisupervised algorithm applied to the synthetic data used in the previous section. Among the 600 and 800 points, 50 labeled points are randomly chosen from each, and the two-dimensional coordinates are provided for $\boldsymbol{s}$ and $\boldsymbol{t}$. The graph weights are chosen by the best convex reconstruction from 6 and 10 neighbors. As can be seen from the figure, the two curves are automatically aligned to each other by sharing a common embedding space. From this common embedding, a point on the s-curve can be mapped to the corresponding point on the wave surface using nearest neighbors, without inferring a direct transformation between the two data spaces.

In [18, 17] the authors assumed symmetric and nonnegative weights. With an asymmetric L, the quadratic term in (4) is no longer valid, and the smoothness term may be replaced with the squared error cost (3). However, there is a difference in the resulting aligned embeddings using a different choice of edge weights on the graph. This is illustrated in the right side of Figure 3 where convex and affine weights are used. With convex weights, the aligned embedding of unlabeled points lies within the convex hull of labeled points. In contrast, the affine weights can extrapolate to points outside the convex hull of the labeled examples. If we consider the matrix of coefficients $M = -(L_{uu})^{-1}L_{ul}$ in (6), it is not difficult to see $\sum_j M_{ij} = 1$ for all i because $\sum_{j\in u} L_{ij} + \sum_{j\in l} L_{ij} = \sum_j L_{ij} = 0$ for all i. Consequently, each row of M are affine coefficients. With an additional constraint $W_{ij} \geq 0$, the M satisfies $M_{ij} \geq 0$ as well, (refer to [4] for proofs) rendering each row of M convex coefficients.

3.2 Alignment by pairwise correspondence

Given multiple data sets containing no additional information about intrinsic coordinates, it is still possible to discover common relationships between the data sets using pairwise correspondences. In particular, two data sets X and Y may have subsets X_l and Y_l which are in pairwise alignment. For example, given sets of images of different persons, we may select pairs with the same pose, facial expression, etc. With this additional information, it is possible to then determine how to match the unlabeled examples using an aligned manifold embedding.

The pairwise correspondences are indicated by the indices $\boldsymbol{x}_i \leftrightarrow \boldsymbol{y}_i$, $(i \in l)$, and $\boldsymbol{f}$ and $\boldsymbol{g}$ denote real-valued functions defined on the respective graphs of X and Y. $\boldsymbol{f}$ and $\boldsymbol{g}$ represent embedding coordinates that are extracted separately for each data set, but they should

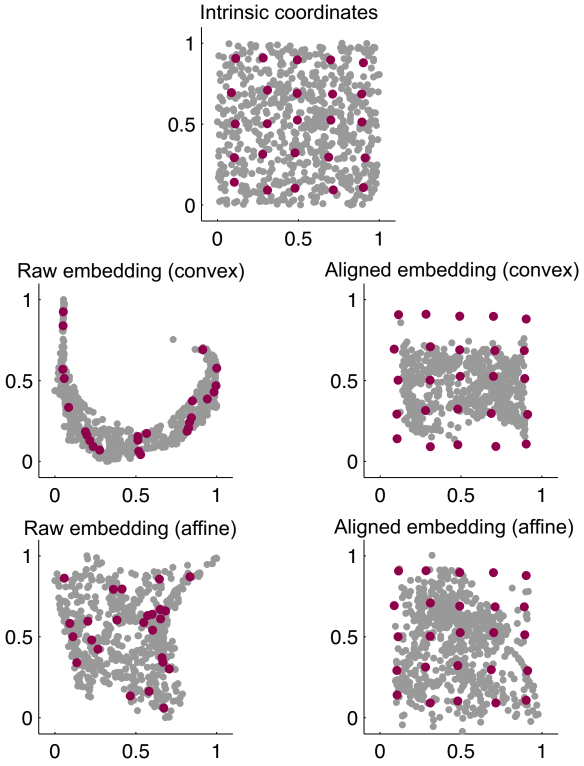

Figure 3: Embedding a data manifold with given coordinates. A set of 698 images of a statue was taken by a camera with varying tilt and pan angles as pose parameters. These pose parameters are provided as labeled coordinates for chosen imges (large dots). This information is used to infer the two-dimensional coordinates corresponding to poses of the unlabeled images. Different conditions for weights in the graph Laplacian result in quite different embeddings.

take similar values for corresponding pairs. Generalizing the single graph embedding algorithm, the dual embedding can be defined by optimizing:

$$C(\boldsymbol{f},\boldsymbol{g}) = \mu \sum_{i\in l} |f_i - g_i|^2 + \boldsymbol{f}^T L^x \boldsymbol{f} + \boldsymbol{g}^T L^y \boldsymbol{g}, \tag{8}$$

where L^x and L^y are the graph Laplacian matrices. The first term penalizes discrepancies between $\boldsymbol{f}$ and $\boldsymbol{g}$ on the corresponding vertices, and the second term imposes smoothness of $\boldsymbol{f}$ and $\boldsymbol{g}$ on the respective graphs.

However, unlike the regression in (4), the optimization in (8) is ill-defined because it is not invariant to simultaneous scaling of $\boldsymbol{f}$ and $\boldsymbol{g}$. We instead should minimize the Rayleigh quotient:

$$\tilde{C}(\boldsymbol{f},\boldsymbol{g}) := \frac{C(\boldsymbol{f},\boldsymbol{g})}{\boldsymbol{f}^T\boldsymbol{f} + \boldsymbol{g}^T\boldsymbol{g}} \tag{9}$$

This quotient can be written in terms of the augmented vector: $\boldsymbol{h} = [\boldsymbol{f}^T \boldsymbol{g}^T]^T$. Minimizing (9) is then equivalent to

$$\min_h \ \tilde{C}(\boldsymbol{h}) := \frac{\boldsymbol{h}^T L^z \boldsymbol{h}}{\boldsymbol{h}^T\boldsymbol{h}}, \quad \text{s.t.} \quad \boldsymbol{h}^T\boldsymbol{e} = 0, \tag{10}$$

where L^z is defined as

$$L^z = \begin{bmatrix} L^x + U^x & -U^{xy} \\ -U^{yx} & L^y + U^y \end{bmatrix} \geq 0, \tag{11}$$

and U^x,U^y,U^{xy}, and U^{yx} are matrices having non-zero elements only on the diagonal

$$U_{ij} = \begin{cases} \mu, & i = j \in l \\ 0, & \text{otherwise} \end{cases}$$

The r-dimensional embedding is obtained by the r-nonzero eigenvectors of L^z. A slightly different embedding results from using the normalized cost function (9):

$$\tilde{C}(\boldsymbol{f},\boldsymbol{g}) := \frac{C(\boldsymbol{f},\boldsymbol{g})}{\boldsymbol{f}^T D^x \boldsymbol{f} + \boldsymbol{g}^T D^y \boldsymbol{g}},$$

where D^x and D^y are diagonal matrices corresponding to the vertex degrees $D^x_{ii} = \sum_j W^x_{ij}$ and $D^y_{ii} = \sum_j W^y_{ij}$. This optimization is solved by finding the generalized eigenvectors of L^z and $D^z = \mathrm{diag}(D^x, D^y)$.

In (8) the coefficient μ weights the importance of the correspondence term relative to the smoothness term. In the limit $\mu \to \infty$, the result is equivalent to imposing hard constraints $f_i = g_i$ for $i \in l$. In this limit, the optimization is given by the eigenvalue problem:

$$C\tilde{(}\boldsymbol{h}) := \frac{\boldsymbol{h}^T L^z \boldsymbol{h}}{\boldsymbol{h}^T\boldsymbol{h}}, \quad \text{s.t.} \quad \boldsymbol{h}^T\boldsymbol{e} = 0, \tag{12}$$

where $\boldsymbol{h}$ and L^z are defined as

$$\boldsymbol{h} = \begin{bmatrix} \boldsymbol{f}_l = \boldsymbol{g}_l \\ \boldsymbol{f}_u \\ \boldsymbol{g}_u \end{bmatrix}, \ L^z = \begin{bmatrix} L^x_{ll} + L^y_{ll} & L^x_{lu} & L^y_{lu} \\ L^x_{ul} & L^x_{uu} & 0 \\ L^y_{ul} & 0 & L^y_{uu} \end{bmatrix}. \tag{13}$$

This formulation results in a smaller eigenvalue problem than (10), and the parameter μ need not be explictly determined.

The two methods in (11) and (13) of constructing a new graph Laplacian L^z can be interpreted as joining two disparate graphs. The former definition of L^z links two graphs by adding edges between paired vertices of the graphs with weights μ, whereas the latter L^z “short-circuits” the paired vertices. In either case, the embedding of the joint graph automatically aligns the two constituent graphs.

Figure 4 shows the alignment of the embeddings of s-curve and wave surfaces via the hard coupling of the graphs. Joining the two graphs not only aligns each other, but also highlights the underlying structure in common, yielding slightly more uniform embeddings than the unsupervised ones.

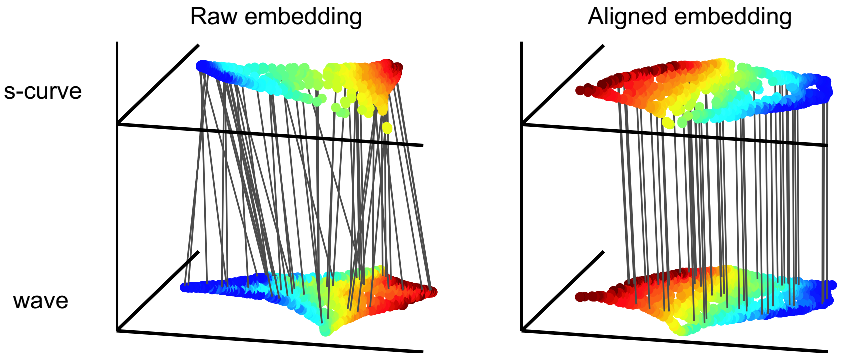

Figure 4: The graph embeddings of the s-curve and wave surface are aligned by pairwise correspondence. 100 pairs of points in one-to-one correspondence are indicated by lines (only 50 shown).

4 Applications

The goal of aligning manifolds was to find an bi-continuous map between the manifolds. A common embedding space is first learned by incorporating additional information about the data samples. We can use this common low-dimensional embedding space to address the following matching problems. What is the most relevant sample $\boldsymbol{y}_i \in Y$ that corresponds to a $\boldsymbol{x}_j \in X$? or the most relevant sample $\boldsymbol{x}_i \in X$ that corresponds to a $\boldsymbol{y}_j \in Y$?

The Euclidean distance of samples in the common embedding space can provide a relevant measure for matching. Let $F = [\boldsymbol{f}_1 \boldsymbol{f}_2 \cdots \boldsymbol{f}_r]$ and $G = [\boldsymbol{g}_1 \boldsymbol{g}_2 \cdots \boldsymbol{g}_r]$ be the r-dimensional representations of aligned manifolds of X and Y. If the coordinates in F and G are aligned from known coordinates, the distance between $\boldsymbol{x}_i \in X$ and $\boldsymbol{y}_j \in Y$ is defined by the usual distance:

$$d(\boldsymbol{x}_i, \boldsymbol{y}_j)^2 := \sum_k |F_{ik} - G_{jk}|^2.$$

If F and G are computed from normalized eigenvectors of a graph Laplacian, the coordinates should be properly scaled. We use the eigenvalues $\lambda_1, \lambda_2, \cdots$, to scale the distance between $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ [10]:

$$d(\boldsymbol{x}_i, \boldsymbol{y}_j)^2 := \sum_k |F_{ik} - G_{jk}|^2 / \lambda_k.$$

Then the best match $\boldsymbol{y}_i \in Y$ to $\boldsymbol{x} \in X$ is given by finding $\arg\min_i d(\boldsymbol{x}, \boldsymbol{y}_i)$.

We demonstrate matching image examples with three sets of high-dimensional images. The three data sets X, Y, and Z consist of 841 images of a person, 698 images of a statue, and 900 images of the earth, available at *http://www.seas.upenn.edu/∼jhham* and *http://isomap.stanford.edu/datasets.html*. Data set X consists of 120×100 images obtained by varying the pose of a person's head with a fixed camera. Data set Y are 64×64 computer generated images of a 3-D model with varying light sources and pan and tilt angles for the observer. Data set Z are 100×100 rendered images of the globe by rotating its azimuthal and elevation angles. For Y and Z we know the intrinsic parameters of the variations: Y varies through -75 to 75 degrees of pan and -10 to 10 degrees of tilt, and -75 to 75 degrees of light source angles. Z contains of -45 to 45 degrees of azimuth and -45 to 45 degrees for elevation changes. We use the pan and tilt angles of Y and Z as the known 2-D coordinates of the embeddings. Affine weights are determined with 12,12, and 6 nearest neighbors to construct the graphs of data X, Y, and Z.

We describe how both known pose coordinates as well as pairwise correspondences are used to align the image manifolds from the three different data sets.

Matching two sets with correspondence and known coordinates

The task is to align X and Y using both the correspondences of $X \leftrightarrow Y$, and the known pose coordinates of Y. First, 25 matching pairs of images in X and Y are manually chosen. The joint graph of X and Y is formed by fusing the corresponding vertices as in (13). Then the joint graph is aligned to the 25 sample coordinates of Y by (6). The best matching images in X and Y that correspond to various pose parameters are found by nearest image samples in the embedding space. Figure 5 shows the result when 16 grid points in the pose parameter embedding are given and the best matching images in X and Y are displayed.

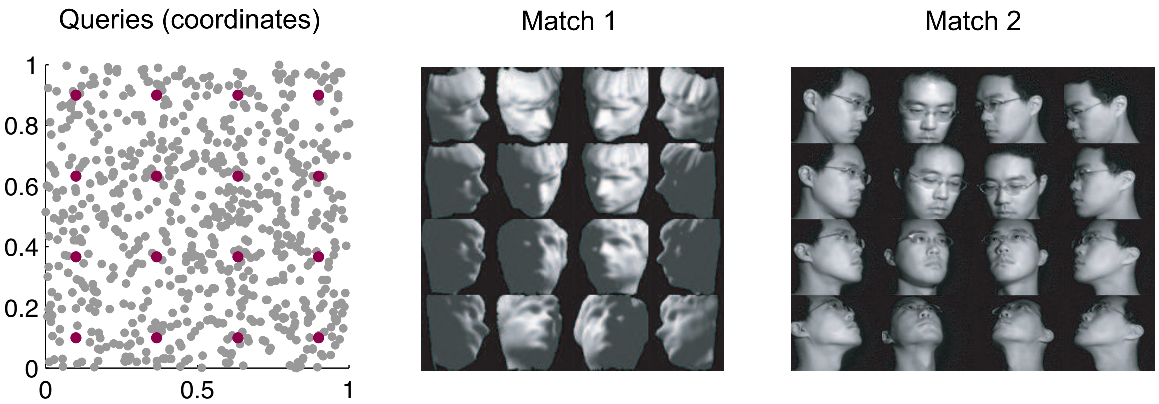

Figure 5: Matching two data sets with correspondence and external coordinates. 25 images of a statue are parameterized by its tilt/pan angles (gray dots on the left). Additionally, 25 corresponding pairs of images of the statue and person are manually matched. Given 16 queries (dark dots on the left) in the embedding space, the best matching images of statue (middle) and person (right) are found by aligning the two data sets and pose parameters simultaneously.

Matching three sets with correspondence

We also demonstrate the simultaneous matching of three data sets. Among the three data sets, we have pairwise correspondence between example images in $X \leftrightarrow Y$ and examples images in $Y \leftrightarrow Z$ separately. 25 pairs of corresponding images between X and Y are used, and an additional 25 pairs of images in Y and Z are chosen manually. The joint graph of X, Y, and Z is formed by the straightforward extension of (12) to handle three sets. A joint graph Laplacian is formed and the final aligned embeddings of the three sets are computed by diagonalizing the graph Laplacian. Given unlabeled sample images from Z as input, the best matching data for Y and X are determined and shown in Figure 6.

5 Discussion

The main computational cost of the graph algorithm lies in finding the spectral decomposition of a large matrix. We employ methods for calculating eigenvectors of large sparse matrices to efficiently speed computation of the embeddings. The graph algorithm is able to quite robustly align the underlying manifold structure in these data sets. Even with the small number of training samples provided, the algorithm is able to estimate a common low-dimensional embedding space which can be used to map samples from one data set to another. Even in situations where the unsupervised manifold learning algorithm suffers from a lack of samples, additional knowledge from the known coordinates and/or pairwise correspondences can be used to discover a faithful embedding. We are currently working on extending these results on additional real-world data sets such as video streams and audio signals.

Finally, we would like to acknowledge support from the U.S. National Science Foundation, Army Reserach Office, and Defense Advanced Research Projects Agency.

References

[1] D. Aldous and J. Fill. Reversible Markov chains and random walks on graphs, 2002. In preparation.

[2] M. Belkin, I. Matveeva, and P. Niyogi. Regularization and regression on large graphs. In *Proceedings of 17th Annual Conference on Learning Theory*, pages 624–638, 2004.

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

[4] A. Berman and R. J. Plemmons. *Nonnegative Matrices in the Mathematical Science*. Academic Press, New York, 1996.

[5] M. Brand. Charting a manifold. In *Advances in Neural Information Processing Systems 15*, pages 961–968, Cambridge, MA, 2003. MIT Press.

[6] A. K. Chandra, P. Raghavan, W. L. Ruzzo, and R. Smolensky. The electrical resistance of a graph captures its commute and cover times. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 574–586. ACM Press, 1989.

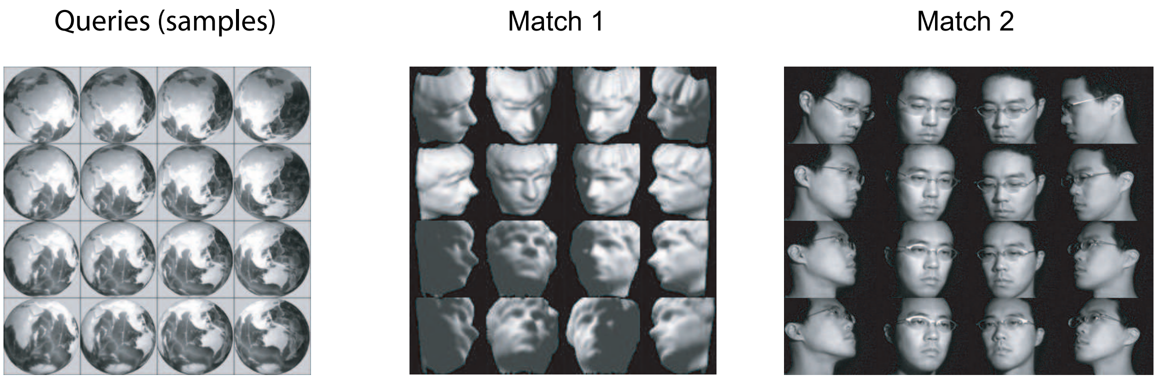

Figure 6: Matching three data sets using correspondence between 25 image pairs of statue and person, and 25 additional image pairs of statue and earth. After the aligned embedding of the joint graph is computed, it is possible to match images across the three data sets. Given the left images of the earth as queries, the right figures show the best matching images of the statue (middle) and person (right).

[7] D. L. Donoho and C. Grimes. Hessian eigenmaps: locally linear embedding techniques for high-dimensional data. In *Proceedings of National Academy of Science, 100 (10)*, pages 5591–5596, 2003.

[8] M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its applications to graph theory. *Czechoslovak Math Journal, 25 (100)*, pages 619–633, 1975.

[9] S. Guattery. Graph embeddings, symmetric real matrices, and generalized inverses. Technical Report NASA/CR-1998-208462 ICASE Report No. 98-34, Institute for Computer Applications in Science and Engineering, August 1998.

[10] J. Ham, D. D. Lee, S. Mika, and B. Schölkopf. Kernel view of the dimensionality reduction of manifolds. In *Proceedings of International Conference on Machine Learning*, 2004.

[11] J. Ham, D. D. Lee, and L. K. Saul. Learning high-dimensional correspondences from low-dimensional manifolds. In *Workshop on The Continuum from Labeled to Unlabled Data in Machine Learning and Data Mining at Twentieth International Conference on Machine Learning*, pages 34–39, 2003.

[12] I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete structures. In *Proceedings of International Conference on Machine Learning*, 2002.

[13] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science, 290*, pages 2323–2326, 2000.

[14] L. K. Saul and S. T. Roweis. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research, 4*, pages 119–155, 2003.

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

[16] J. J. Verbeek, S. T. Roweis, and N. Vlassis. Nonlinear CCA and PCA by alignment of local models. In *Advances in Neural Information Processing Systems 16*, 2004.

[17] D. Zhou and B. Schölkopf. A regularization framework for learning from graph data. In *Workshop on Statistical Relational Learning at Twenty-first International Conference on Machine Learning*, 2004.

[18] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of International Conference on Machine Learning*, pages 912–919, 2003.