Semi-definite Manifold Alignment

Liang Xiong, Fei Wang, and Changshui Zhang

Department of Automation,
Tsinghua University,
Beijing, China

Abstract. In this paper, we study the problem of *manifold alignment*, which aims at "aligning" different data sets which share a similar intrinsic manifold provided some supervision. Unlike traditional methods that rely on pairwise correspondences between the two data sets, our method only needs some relative comparison information like "A is more similar to B than A is to C". This method provides a more flexible way to acquire the prior knowledge for alignment, thus is able to handle situations where corresponding pairs are hard or impossible to identify. We optimize our objective based on the graphs that give discrete approximations of the manifold. Further, the problem is formulated as a *semi-definite programming* (SDP) problem which can readily be solved. Finally the experimental results on aligning several different types of manifolds are presented to show the effectiveness of our method.

1 Introduction

In the field of machine learning, we are often faced with data that have very high dimensionality (*e.g.* images and vector-space representations of text documents). Directly dealing with these data is usually intractable due to the high computational load and *the curse of dimensionality*. In recent years, researchers have realized that in many applications the samples of interest are actually confined to a low-dimensional manifold embedded in the high dimensional feature space [1, 2]. This intrinsic structure bears a great amount of information hence algorithms that can effectively explore and exploit this structure is highly desired to facilitate the analysis and learning of data.

Consequently, many dimensionality reduction methods have been developed to characterize data manifolds, such as *Locally Linear Embedding* [2], *Laplacian Eigenmaps* [3] and *Maximum Variance Unfolding* [4]. However, all these algorithms are *unsupervised*, that is, no prior knowledge is used to guide the dimensionality reduction process. As a result, the low-dimensional representations of data achieved by these methods usually failed to explicitly reflect samples' parameters, which is often of central interest during analysis and learning (For example, the pose parameters for head images). Fortunately, provided some kind of *supervised* information, we are able to develop methods that can both discover manifold structures and reveal their underlying parameters.

In this paper, we will focus on the problem of correspondence learning on manifolds, which is also called *manifold alignment*. More concretely, assuming we are given some data sets sharing the same underlying manifold, we seek to learn the correspondences between samples from different data sets (*e.g.* Finding different persons' face images with the same pose). Besides its usage in data analysis and visualization, this problem also have wide potential applications in various fields. For instance, in *facial expression recognition*, one may have a set of standard labeled images with known expressions, such as *happiness, sadness, surprise, anger and fear*, of a particular person. Then we can recognize the expressions of another person just by aligning his/her facial images to the standard image set. Its application can also be found directly in pose estimation. One can refer to [5] for more details.

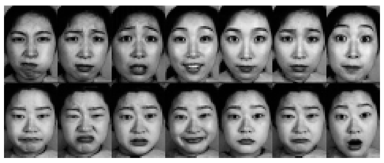

Fig. 1. An example of data sharing the same manifold. Above is facial expression images from the *JAFFE* data set [6]. The top and bottom rows show selected pairs with the same underlying facial expression parameters.

There have already been some traditional methods proposed to align manifolds in a *semi-supervised* way [7, 8, 5, 9, 10]. Specifically, they usually assumed that some pairwise correspondences of samples in different data sets were already known, and then those information would be used to guide the alignment. However, in practice it might be difficult to obtain and use such information since:

1. The sizes of data sets can be very large, then finding high-quality correspondences between them can be very time consuming and even intractable.
2. There may exist some ambiguities in the images (see figure 2 for an example), which makes explicit matching a hard task. Brutally determine and enforce these unreliable constraints may lead to poor results;
3. Sometimes it may be hard to find the exact correspondences when the available samples are scarce. This situation may happen when the data source is restricted and users are only allowed to access a small subset, or at the early stage of an experiment when samples are still to be collected.

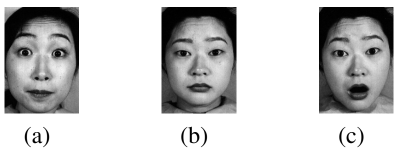

Fig. 2. Three facial expression images. (a) and (c) is *surprised* and (b) is *neutral*. It is hard to make a confident decision that (a) and (c) possess the same expression parameters. However, it is obvious that (c) is more similar to (a) than (b) is.

To solve the above problems, we propose to apply another type of supervised information to guide the process of manifold alignment. In particular, we consider a relative and qualitative supervision of the form "*A is closer to B than A is to C*". We believe that this type of information is more easily available in practice than traditional correspondence-based information. With the help of such information, we show that the manifold alignment problemcan be formulated as a *Quadratically Constrained Quadratic Programming* (*QCQP*) [11] problem. To make the optimization tractable, we further relax it to a *Semi-Definite Programming* (*SDP*) [11] problem, which can be readily solved by popular software packages such as *Sedumi* [12]. Besides, under this formulation we are able to incorporate both relative relations and correspondences to align manifolds in a very flexible way. Finally experimental results on aligning several different types of manifolds are presented to show the effectiveness of our method.

The rest of this paper is organized as follows. Section 2 will introduce some basic notations and related works, the detailed algorithm will be presented in section 3. The experimental results will be provided in section 4, followed by the discussions in section 5 and conclusions in section 6.

2 Notations and Related Works

As stated in the introduction, in this paper we will study the problem of *aligning* different data sets that are characterized by the same underlying manifold. For the convenience of presentation, first let us consider the case of two data sets.

More formally, let $\mathcal{X}$ and $\mathcal{Y}$ be two data sets in high-dimensional vector spaces

$$\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\} \subset \mathbb{R}^{D_x}, \ \ \mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N\} \subset \mathbb{R}^{D_y} \tag{1}$$

with $D_x, \ D_y \gg 1$. When the data lie close to a low-dimensional manifold embedded in a high dimensional *Euclidean* space, the manifold learning algorithms such as *Laplacian eigenmaps* [3] can effectively learn the low-dimensional embeddings by constructing an *undirected weighted graph* that captures the local structure of data. For example, for data set $\mathcal{X}$, we can construct a graph $\mathcal{G}_\mathcal{X} = (\mathcal{V}_\mathcal{X}, \mathcal{E}_\mathcal{X})$, where $\mathcal{V}_\mathcal{X} = \mathcal{X}$ corresponds to the vertices of $\mathcal{G}_\mathcal{X}$, and $\mathcal{E}_\mathcal{X}$ represents the edge set of $\mathcal{G}_\mathcal{X}$. Generally there is a nonnegative weight W_{ij} associated with each edge $e_{ij} \in \mathcal{E}_\mathcal{X}$, and we can aggregate all the edge weights to form an $N \times N$ *weight matrix* $\mathbf{W}_\mathcal{X}$ with its (i,j)-th entry $\mathbf{W}_\mathcal{X}(i,j) = W_{ij}$. The *degree matrix* $\mathbf{D}_\mathcal{X}$ is an $N \times N$ diagonal matrix with the i-th entry on its diagonal line $\mathbf{D}_\mathcal{X}(i,i) = \sum_j W_{ij}$, and the *combinatorial graph Laplacian* is defined as an $N \times N$ matrix $\mathbf{L}_\mathcal{X} = \mathbf{D}_\mathcal{X} - \mathbf{W}_\mathcal{X}$.

The low-dimensional embeddings of the data in $\mathcal{X}$, say $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_N] \in \mathbb{R}^{D \times N}$ $(D \ll D_\mathcal{X})$ can be achieved by minimizing the following criterion under certain constraints (*e.g.* scale and translational invariances)

$$\mathcal{S}_\mathcal{X} = tr(\mathbf{F}\mathbf{L}_\mathcal{X}\mathbf{F}^T), \tag{2}$$

where $tr(\cdot)$ represents the trace of a matrix. According to [3], $\mathcal{S}_\mathcal{X}$ measures the smoothness of the low-dimensional embeddings of $\mathcal{X}$ over the its underlying manifold.

Similarly, we can also define a graph $\mathcal{G}_\mathcal{Y} = (\mathcal{V}_\mathcal{Y}, \mathcal{E}_\mathcal{Y})$ for data set $\mathcal{Y}$ with its combinatorial graph Laplacian $\mathbf{L}_\mathcal{Y} = \mathbf{D}_\mathcal{Y} - \mathbf{W}_\mathcal{Y}$. Then the low-dimensional embeddings of $\mathcal{Y}$, say $\mathbf{G} = [\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_N] \in \mathbb{R}^{D\times N}$ can also be achieved by minimizing $\mathcal{S}_\mathcal{Y} = tr(\mathbf{G}\mathbf{L}_\mathcal{Y}\mathbf{G})^T$ under certain constraints, and we can minimize the following combined smoothness criterion to achieve the common embeddings of both $\mathcal{X}$ and $\mathcal{Y}$

$$\mathcal{S} = tr(\mathbf{F}\mathbf{L}_\mathcal{X}\mathbf{F}^T) + tr(\mathbf{G}\mathbf{L}_\mathcal{Y}\mathbf{G}^T). \tag{3}$$

Now let's return to our *manifold alignment* problem. Assuming that we have known some pairwise correspondences $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^l$ between $\mathcal{X}$ and $\mathcal{Y}$, we can embed $\mathcal{X}$ and $\mathcal{Y}$ into a common low-dimensional space by minimizing [8]

$$\mathcal{J} = \mu \sum\nolimits_{i=1}^{l} \|\mathbf{f}_i - \mathbf{g}_i\|^2 + tr(\mathbf{F}\mathbf{L}_\mathcal{X}\mathbf{F}^T) + tr(\mathbf{G}\mathbf{L}_\mathcal{Y}\mathbf{G}^T), \tag{4}$$

where μ is a regularization parameter to tradeoff the embedding smoothness and the matching precision. When $\mu = \infty$, the pairwise correspondences will become hard constraints which impose $\mathbf{f}_i = \mathbf{g}_i$ after embedding [7, 10]. Finally the matched sample for $\mathbf{x}_i \in \mathcal{X}$ is $\mathbf{y}_i \in \mathcal{Y}$ with the minimum Euclidean distance to in the embedded space.

However, as we explained in the introduction, sometimes it may be difficult to obtain the pairwise correspondences. Hence we propose a novel scheme for manifold alignment in this paper, which is based on relative comparisons among the data points.

3 Manifold Alignment via Semi-definite Programming

In this section we will introduce our *Semi-Definite Manifold Alignment* (*SDMA*) algorithm in detail. First let's see the objective and problem formulation.

3.1 The Quadratic Formulation

As we have stated in section 2, the manifold alignment process is composed of two steps: (1) embedding the data points into a common low-dimensional space with the guidance of some prior knowledge; (2) finding the one-to-one correspondences based on the Euclidean distances in the embedding space. The main challenge is the first step, *i.e.*, how to effectively embed the data from different data sets into a common low-dimensional space.

Co-Embedding without Prior Knowledge Following [8], we also adopt the graph based criterion as our optimization objective. We construct weighted undirected graphs $\mathcal{G}_\mathcal{X}, \mathcal{G}_\mathcal{Y}$ for data sets $\mathcal{X}$ and $\mathcal{Y}$ respectively, and then seek a embedding which minimize Eq.(3). To avoid the illness of the optimization problem, we further impose the scale and translational invariance constraints

$$\begin{aligned} &tr(\mathbf{F}\mathbf{F}^T) = 1, tr(\mathbf{G}\mathbf{G}^T) = 1, \\ &\mathbf{F}\mathbf{e} = \mathbf{0}, \mathbf{G}\mathbf{e} = \mathbf{0}, \end{aligned} \tag{5}$$

to the optimization objective. Then the *co-embedding* problem can be formulated as

$$\begin{aligned} \min_{\mathbf{F},\mathbf{G}} \; & tr(\mathbf{F}\mathbf{L}_{\mathcal{X}}\mathbf{F}^T) + tr(\mathbf{G}\mathbf{L}_{\mathcal{Y}}\mathbf{G}^T) \\ s.t. \quad & tr(\mathbf{F}\mathbf{F}^T) = 1, tr(\mathbf{G}\mathbf{G}^T) = 1 \\ & \mathbf{Fe} = \mathbf{0}, \mathbf{Ge} = \mathbf{0} \end{aligned} \tag{6}$$

which is a co-dimensionality reduction problem without any prior knowledge about the relationship between $\mathcal{X}$ and $\mathcal{Y}$.

Another issue that should be addressed here is the construction of the *Laplacian* matrices $\mathbf{L}_{\mathcal{X}}$ and $\mathbf{L}_{\mathcal{Y}}$. A common choice is to first compute the weight matrices by the Gaussian functions, *i.e.* the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ can be computed by

$$\mathbf{W}_{\mathcal{X}}(i,j) = \exp\left(-\beta\|\mathbf{x}_i - \mathbf{x}_j\|^2\right), \tag{7}$$

and then calculate the Laplacian matrices in their standard ways. However, a problem is that the free parameter β is usually set empirically and its value may affect the final results significantly [13]. Therefore we propose to use the *iterated Laplacian* here [3]. Mathematically, the *iterated Laplacian* for data set $\mathcal{X}$ is defined as

$$\mathbf{M}_{\mathcal{X}} = (\mathbf{I} - \mathbf{Q}_{\mathcal{X}})^T(\mathbf{I} - \mathbf{Q}_{\mathcal{X}}), \tag{8}$$

where $\mathbf{Q}_{\mathcal{X}}$ is an $N \times N$ square matrix with its (i,j)-th entry $\mathbf{Q}_{\mathcal{X}}(i,j) = q_{ij}$ being optimized by

$$\begin{aligned} \min_{q_{ij}} \; & \left\|\mathbf{x}_i - \sum\nolimits_{\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i)} q_{ij}\mathbf{x}_j\right\|^2 \\ s.t. \quad & \sum\nolimits_j q_{ij} = 1, \end{aligned} \tag{9}$$

where $\mathcal{N}(\mathbf{x}_i)$ is the neighborhood of $\mathbf{x}_i$ (*e.g. k-nearest neighborhood* or ε*-ball neighborhood*), and for $\mathbf{x}_j \notin \mathcal{N}(\mathbf{x}_i)$, $\mathbf{Q}_{\mathcal{X}}(i,j) = 0$. Similarly, we can define the iterated Laplacian $\mathbf{M}_{\mathcal{Y}}$ for data set $\mathcal{Y}$.

Manifold Alignment by Incorporating the Prior Knowledge Now let's take the relative comparison based constraints into account. As we have introduced in section 2, the knowledge "$\mathbf{y}_i$ *is closer to* $\mathbf{x}_j$ *than* $\mathbf{x}_k$" can be translated into the relative distance constraint

$$\|\mathbf{g}_i - \mathbf{f}_j\|^2 \leq \|\mathbf{g}_i - \mathbf{f}_k\|^2 \tag{10}$$

in the embedded space. In the rest of this paper, for notational convenience, we will denote the constraint shown in Eq.(10) as an ordered 3-tuple

$$t_c = \{\mathbf{y}_i, \mathbf{x}_j, \mathbf{x}_k\}.$$

We use $\mathcal{T} = \{t_c\}_{c=1}^C$ to denote the set of constraints. By incorporating those constraints, our optimization problem can be formulated as

$$\begin{aligned} \min_{\mathbf{F},\mathbf{G}} \; & tr(\mathbf{F}\mathbf{M}_{\mathcal{X}}\mathbf{F}^T) + tr(\mathbf{G}\mathbf{M}_{\mathcal{Y}}\mathbf{G}^T) \\ s.t. \quad & \forall\{\mathbf{y}_i, \mathbf{x}_j, \mathbf{x}_k\} \in \mathcal{T}, \|\mathbf{g}_i - \mathbf{f}_j\|^2 \leq \|\mathbf{g}_i - \mathbf{f}_k\|^2 \\ & tr(\mathbf{F}\mathbf{F}^T) = 1, tr(\mathbf{G}\mathbf{G}^T) = 1, \\ & \mathbf{Fe} = \mathbf{0}, \mathbf{Ge} = \mathbf{0} \end{aligned} \tag{11}$$

Let $\mathbf{H} = [\mathbf{F}, \mathbf{G}]$, $\mathbf{M} = \begin{bmatrix} \mathbf{M}_{\mathcal{X}} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{\mathcal{Y}} \end{bmatrix}$, then Eq.(11) can be simplified to

$$\begin{aligned} &\min\nolimits_{\mathbf{H}} \ tr(\mathbf{H}\mathbf{M}\mathbf{H}^T) \\ &\quad s.t. \ \ \forall\{\mathbf{y}_i, \mathbf{x}_j, \mathbf{x}_k\} \in \mathcal{T}, \|\mathbf{h}_{i+N} - \mathbf{h}_j\|^2 \leq \|\mathbf{h}_{i+N} - \mathbf{h}_k\|^2 \\ &\qquad\quad tr(\mathbf{H}_F\mathbf{H}_F^T) = 1, tr(\mathbf{H}_G\mathbf{H}_G^T) = 1 \\ &\qquad\quad \mathbf{H}_{\mathbf{F}}\mathbf{e} = \mathbf{0}, \mathbf{H}_{\mathbf{G}}\mathbf{e} = \mathbf{0}, \end{aligned} \tag{12}$$

where $\mathbf{H}_F$ and $\mathbf{H}_F$ are the sub-matrices of $\mathbf{H}$ corresponding to $\mathbf{F}$ and $\mathbf{G}$.

Now we have formulated our tuple-constrained optimization as a *Quadratically Constrained Quadratic Programming* (*QCQP*) [11] problem. However, since the relative distance constraints in Eq.(12) is not *convex*, then 1) computationally the solution is difficult to derive and 2) the solution is trapped in local minima. Therefore, a reformulation is needed to make this problem tractable.

3.2 A Semi-Definite Approach

In this section we will present the details on how to relax the QCQP problem Eq.(12) to a SDP problem. Note that

$$\begin{aligned} &\|\mathbf{h}_{i+N} - \mathbf{h}_j\|^2 \leq \|\mathbf{h}_{i+N} - \mathbf{h}_k\|^2 \\ \Longleftrightarrow\ &-2\mathbf{h}_{i+N}^T\mathbf{h}_j + 2\mathbf{h}_{i+N}^T\mathbf{h}_k + \mathbf{h}_j^T\mathbf{h}_j - \mathbf{h}_k^T\mathbf{h}_k \leq 0 \end{aligned} \tag{13}$$

and

$$tr(\mathbf{H}\mathbf{M}\mathbf{H}^T) = tr(\mathbf{M}\mathbf{H}^T\mathbf{H}). \tag{14}$$

These two facts motivate us to relax the problem and deal with the *Gram matrix* instead of manipulating the data coordinates directly. The Gram matrix of data is $\mathbf{K} = \mathbf{H}^T\mathbf{H}$ with its (p, q)-th entry $\mathbf{K}_{pq} = \mathbf{h}_\mathbf{p}^\mathbf{T}\mathbf{h}_\mathbf{q}$, which can be divided into four blocks as

$$\mathbf{K} = \left[\begin{array}{c:c} \mathbf{F}^T\mathbf{F} & \mathbf{F}^T\mathbf{G} \\ \hdashline \mathbf{G}^T\mathbf{F} & \mathbf{G}^T\mathbf{G} \end{array}\right] = \left[\begin{array}{c:c} \mathbf{K}^{FF} & \mathbf{K}^{FG} \\ \hdashline \mathbf{K}^{GF} & \mathbf{K}^{GG} \end{array}\right]. \tag{15}$$

Using $\mathbf{K}$, we are able to convert the formulas in Eq.(12) into linear forms as follows:

- **The objective function** is now

$$\min_{\mathbf{K}} tr(\mathbf{M}\mathbf{K}). \tag{16}$$

- **The relative distance constraints** is

$$\forall\{\mathbf{y}_i, \mathbf{x}_j, \mathbf{x}_k\} \in \mathcal{T}, -2\mathbf{K}_{i+N,j} + 2\mathbf{K}_{i+N,k} + \mathbf{K}_{j,j} - \mathbf{K}_{k,k} \leq 0 \tag{17}$$

- **The scale invariance** can be achieved by constraining the traces of diagonal blocks of $\mathbf{K}$ *i.e.*

$$1 = trace(\mathbf{F}\mathbf{F}^T) = trace(\mathbf{K}^{FF}), \tag{18}$$

$$1 = trace(\mathbf{G}\mathbf{G}^T) = trace(\mathbf{K}^{GG}). \tag{19}$$

– **The translation invariance** is achieved by constraints

$$\sum_{i,j} \mathbf{K}^{FF}_{i,j} = 0, \sum_{i,j} \mathbf{K}^{GG}_{i,j} = 0. \tag{20}$$

To see this, consider the following fact for $\mathbf{F}$ (and similar for $\mathbf{G}$)

$$0 = \sum_i \mathbf{f}_i \Leftrightarrow 0 = \left|\sum_i \mathbf{f}_i\right|^2 = \sum_{i,j} \mathbf{f}_i^T \mathbf{f}_j = \sum_{i,j} \mathbf{K}^{FF}_{ij} \tag{21}$$

Finally, to be a valid Gram matrix, $\mathbf{K}$ must be *positive semi-definite*, resulting that

$$\mathbf{K} \succeq 0. \tag{22}$$

Combining Eq.(16) to Eq.(22), we can write our new optimization problem as

$$\begin{aligned}
\min_{\mathbf{K}}\ & tr(\mathbf{MK}) \\
s.t.\ \ & \forall\{\mathbf{y}_i, \mathbf{x}_j, \mathbf{x}_k\} \in \mathcal{T}, -2\mathbf{K}_{i+N,j} + 2\mathbf{K}_{i+N,k} + \mathbf{K}_{j,j} - \mathbf{K}_{k,k} \leq 0 \\
& trace(\mathbf{K}^{FF}) = 1, trace(\mathbf{K}^{GG}) = 1, \\
& \sum_{i,j} \mathbf{K}^{FF}_{i,j} = 0, \sum_{i,j} \mathbf{K}^{GG}_{i,j} = 0, \\
& \mathbf{K} \succeq 0.
\end{aligned} \tag{23}$$

In addition, to avoid the case where the feasible set is empty and encourage the influence of prior knowledge, we relax the constraints by introducing the slack variables $\mathcal{E} = \{\varepsilon_c\}_{c=1}^{C}$ and reformulate the optimization problem as follows

$$\begin{aligned}
\min_{\mathbf{K},\varepsilon}\ & tr(\mathbf{MK}) + \alpha \sum_{c=1}^{C} \varepsilon_c \\
s.t.\ \ & \forall\{\mathbf{y}_i, \mathbf{x}_j, \mathbf{x}_k\} \in \mathcal{T}, -2\mathbf{K}_{i+N,j} + 2\mathbf{K}_{i+N,k} + \mathbf{K}_{j,j} - \mathbf{K}_{k,k} \leq \varepsilon_c \\
& trace(\mathbf{K}^{FF}) = 1, trace(\mathbf{K}^{GG}) = 1, \\
& \sum_{i,j} \mathbf{K}^{FF}_{i,j} = 0, \sum_{i,j} \mathbf{K}^{GG}_{i,j} = 0, \\
& \mathbf{K} \succeq 0, \\
& \forall \varepsilon_c \in \mathcal{E},\ \varepsilon_c \leq 0,
\end{aligned} \tag{24}$$

where α is a parameter to balance the data's inherent structure and the supervised information. When α is small, the embedding result is dominated by the manifold structure, otherwise, the prior knowledge (*i.e.*, the distance relation constraints) plays a more important role. The constraint $\varepsilon \leq 0$ can be removed if we allow the relative distance relations to be violated.

Since Eq.(24) is a *Semi-Definite Programming* (*SDP*) problem [11], we call our method *Semi-Definite Manifold Alignment* (*SDMA*). Clearly, Eq.(24) is convex and thus is free of local minima. Besides, various software packages are available for efficient solutions, and we have preferred the *Sedumi* [12] package in this paper. When the Gram matrix $\mathbf{K}$ is solved, the embedded coordinates $\mathbf{F}, \mathbf{G}$ can be recovered from $\mathbf{K}^{FF}$ and $\mathbf{K}^{\mathbf{GG}}$'s dominant eigenvectors. The number of embedded dimensions can be determined either by prior knowledge or the eigenstrucutre of $\mathbf{K}$.

Finally, we emphasize that SDMA can serve as a very flexible framework for manifold alignment and embedding. More concretely, Eq.(24) can be generalized (or degenerated) in the following ways. 1) Flexible supervision. First, the form of tuple constraints can be changed from $t_c = \{\mathbf{y}_i, \mathbf{x}_j, \mathbf{x}_k\}$ to $t_c = \{\mathbf{h}_i, \mathbf{h}_j, \mathbf{h}_k\}$. This means that we do not have to specify which manifold the samples are chosen from. In fact, SDMA accepts relative distance constraints between *any* three samples (*e.g.* all from the same manifold, or from 3 different manifolds). Moreover, although we only present how to align manifolds based on relative comparisons, our formulation is also able to incorporate the traditional correspondence information by adding constraints "$\mathbf{K}_{i,i} = \mathbf{K}_{i,j} = \mathbf{K}_{j,j}$". 2) Multi-manifold alignment. This can be done straightforwardly by adding more manifold components into $\mathbf{H}$ and $\mathbf{M}$ along with corresponding constraints. 3) Semi-supervised embedding. When there is only one manifold component, SDMA provides a way to embed it with the guide of flexible supervisions.

3.3 Correspondence Acquisition

Now the only remaining problem is how to get the pairwise correspondences (*i.e.*, align the manifolds) using the low-dimensional embeddings. More formally, given a sample $\mathbf{x}_i \in \mathcal{X}$, we want to find its corresponding $\mathbf{y}_i \in \mathcal{Y}$. If we are finding a closest match, a straightforward choice is to find $\mathbf{x}_i$'s nearest neighbor within $\mathcal{Y}$ in the embedded space, *i.e.*, $j = \arg\min_k d(\mathbf{x}_i, \mathbf{x}_k)$, where $d(\mathbf{x}_i, \mathbf{x}_k) = \|\mathbf{f}_i - \mathbf{g}_k\|$.

On the other hand, if we are looking for a bijective mapping between two data sets *i.e.*, if we constrain that each sample from one set is coupled with one and only one sample from the other set, the problem can be solved by minimize the total distance between matched pairs. Specifically, we find matches by

$$\arg\min_{\mathcal{P}} \sum\nolimits_{(\mathbf{x}_i, \mathbf{y}_j) \in \mathcal{P}} d(\mathbf{x}_i, \mathbf{y}_j), \tag{25}$$

where $\mathcal{P} = \{(\mathbf{x}_i, \mathbf{y}_j) | \mathbf{x}_i \text{ and } \mathbf{y}_j \text{ is coupled}\}$. (25) is an *NP-hard* problem. Thus, again, we do relaxation to make it tractable. Eq.(25) can be written as

$$\begin{aligned} \min\nolimits_{\{C_{ij}\}} \; & \sum\nolimits_{i,j} C_{ij} d(\mathbf{x}_i, \mathbf{y}_j) \\ s.t. \quad & \sum\nolimits_i C_{ij} = 1, \sum\nolimits_j C_{ij} = 1, \\ & C_{ij} >= 0, \end{aligned} \tag{26}$$

where $C_{ij} \in \{0, 1\}, C_{ij} = 1$ if $(\mathbf{x}_i, \mathbf{y}_j) \in \mathcal{P}$, and $C_{ij} = 0$ otherwise. To relax, we use continuous C_{ij} as fuzzy indicators. The larger C_{ij} is, the more probable that $\mathbf{x}_i$ and $\mathbf{y}_j$ is coupled. Then (26) becomes a *linear programming* (LP) [11]. When $\{C_{ij}\}$ are obtained, the match $\mathbf{y}_j$ for $\mathbf{x}_i$ is found by solving $\arg\max_j C_{ij}$.

4 Experiments

In this section, we will demonstrate the performance of our SDMA algorithm on several real world data sets. First let's see the basic information of those data sets.

4.1 Data and Settings

- **COIL-20** [14]: This data set contains images of 20 objects. Each object has 72 images from different view points. The underlying parameter is the view point angle.
- **Head pose** [15]. This data set contains head images of 15 sets, each of which has 2 series of 93 images of the same person at 93 different poses. In we use a subset in which the horizontal pose angles vary from -90^o to 90^o and the vertical pose angles very from -30^o to 30^o. The underlying parameters are the horizontal and vertical pose angle of the head.
- **Facial expression** [6]. This data set contains 213 images of 7 facial expressions posed by 10 Japanese female models. The actual underlying parameter is unknown.

The relative distance constraints are obtained as follows. First, samples are drawn randomly to form a tuple $T = \{\mathbf{y}_i, \mathbf{x}_j, \mathbf{x}_k\}$, which means that "$\mathbf{y}_i$ *is more similar to* $\mathbf{x}_j$ *than* $\mathbf{x}_k$". Then a user, or a computer, will judge if this tuple is valid. Finally, valid tuples are collected into $\mathcal{T}$ as the constraints. Since only "yes/no" questions are involved, this procedure is very easy for the users compared to the task of searching for correspondence.

Specifically, $\mathcal{T}$ is determined as follows. For COIL data, we use the difference of view angle to determine the similarity. For the head pose data, similarity is determined by the sum of horizontal and vertical angle differences. For the facial expression data, if y_i and x_j have the same expression that is different from x_j, then T is valid. This strategy gives a conservative yet reliable supervision.

In SDMA, the parameter α tunes the influence of relative distance constraints. In our experiments it is chosen manually from the grid $\{10^{-5}, \cdots, 10^{-1}, 1\}$. After co-embedding, we find correspondences by solving Eq.(26). For all the data sets, the graphs are constructed using Eq.(8). All the results are obtained using only relative distance constraints.

4.2 Results

First we align the COIL data. We construct the graph using neighborhood size 4, and 60 tuples are provided to the algorithm. α is set to 10^{-4}. 144 images of 2 objects are embedded onto a 2-D plane. The pure co-embedding by solving Eq.(6)) and alignment results are both shown in figure 3. By comparison we can see that the embedded coordinates by co-embedding are not related to the manifold's underlying parameters directly, while by alignment we can infer those parameters based on samples' positions. In addition, we also demonstrate how SDMA aligns multiple manifolds in figure 4, where 216 images of 3 objects are embedded onto a 2-D plane. We use 150 tuples for supervision with other settings unchanged. It is show that SDMA can also finds most of the correspondences correctly.

Figure 5 shows the experimental result of SDMA on head pose data. We construct the graph using neighborhood size 7, and use 500 tuples, $\alpha = 10^{-3}$. 130 samples from 2 subjects are embedded onto a 2-D plane. It can be seen that both of the underlying manifold parameters are successfully captured and aligned.

Figure 6 shows the alignment of facial expression data. The graph is constructed with neighborhood size 5, and 50 tuples are used. Since its manifold structure is not

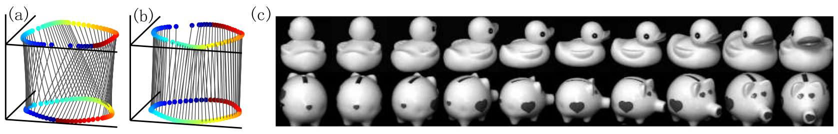

Fig. 3. Alignment of COIL data. (a) is the result of *co-embedding*. (b) is the alignment by SDMA. Lines indicate the true correspondences. Lines in (a) are skew, while in (b) they are nearly straight which implies that correspondance are found. (c) shows some samples of matched images by SDMA.

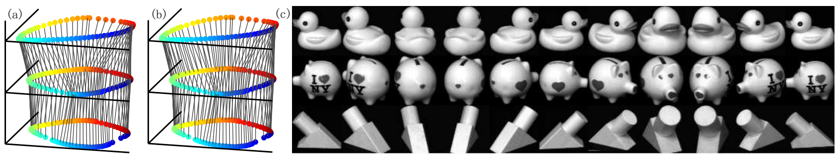

Fig. 4. Alignment of 3 manifolds. (a) shows the true correspondences and (b) shows the matched pairs found by SDMA. (c) shows some samples of matched image.

evident, we set $\alpha = 10^{-1}$ to strengthen the influence of relative distance constraints. 40 samples are embedded onto a 2-D plane since only two eigenvalues of $\mathbf{K}$ are not zero.

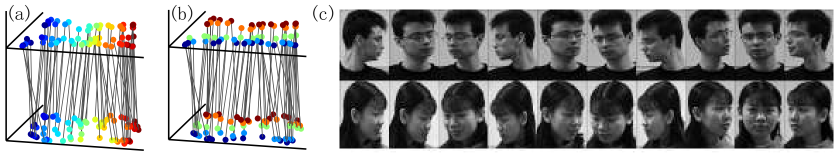

Fig. 5. Alignment of head pose images. (a) and (b) shows the embedding by SDMA and the correspondences found. Points are colored according to horizontal angles in (a) and vertical angles in (b). (c) shows some samples of matched pairs.

5 Discussions

In machine learning problems, we usually need supervisions on individual samples like "$x = a, y = b$". However, this information may be difficult to obtain in some situations. To slack, the *ordinal* approaches (*e.g.* [16]) only require binary order relations like "$x > y$". Further as a generalization, methods using relative comparison information are introduced[1]. These methods enable us to supervise learning problems in a more flexible way.

[1] Order relations is a special form of relative comparisons, considering that the relation "$x > y$" is equivalent to "$|x - r| > |y - r|$" where r is the reference point at the negative infinity.

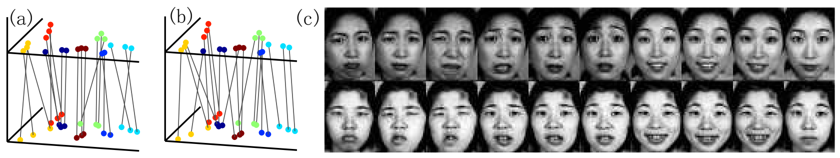

Fig. 6. Alignment of facial expression images. (a) and (b) shows the embedding, with points colored according to the true expressions. (a) shows the true correspondences, while (b) shows those found by SDMA. (c) shows some matched pairs.

The idea of learning with relative comparisons has also been used in other problems. [17] treat relative distance relations as the character of data, and use *AdaBoost* to seek for an embedding where this character is preserved. However, they did not utilize the data's intrinsic structure. [18] and [19] propose to learn distance metrics from relative comparisons. They both seek a distance measure $d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})^T \mathbf{A}(\mathbf{x}-\mathbf{y})}$ and use the relative relations to constrain the feasible region of $\mathbf{A}$ through *mathematical programming*. In spirit, our method is similar to [18]. They learn a distance measure that preserves global distance relations, and we learn an embedding that preserves local manifold structures.

SDMA are closely related to *kernel* methods. By the semi-definite relaxation, we first derive the data's Gram matrix (*a.k.a. Kernel matrix*), and then calculate the low-dimensional coordinates by eigen-decomposition. This procedure is similar to *kernel principal component analysis* (KPCA) [20] except that our kernel matrix is learnt by aligning manifolds. Therefore, SDMA can be considered a *kernel learning* method. From this perspective, SDMA is similar to [21]. The difference is that they only use a single manifold's structure, while we exploit multiple manifolds and their correspondences.

In SDMA, finding correspondent pairs is generally more difficult than in traditional methods, since relative distance constraints are "weaker" than the correspondence constraints, which are directly imposed on the coordinates. Thus the embedded coordinates by SDMA is not very accurate if the number of constraints is small. To achieve similar performances, more relative distance constraints are required, or we need to add a few correspondence constraints. Nevertheless, we believe that this cost is acceptable considering this type of constraint's wide availability and applicability.

One drawback of SDMA is that its computational cost is high when dealing with large data sets. Although the semi-definite relaxation makes the problem tractable, it inevitably increases the number of variables. In the future we are aiming at finding more efficient solutions.

6 Conclusion

Traditional align algorithms rely on the knowledge of high-quality pairwise correspondence, which is difficult to acquire in many situations. In this paper, we study a new way of aligning manifold based on the smoothness on graphs. To achieve maximum applicability and minimum user effort, we introduce the novel *relative distance constraint* to guide the alignment. Alignment using this type of prior knowledge is first formulized as a *quadratically constrained quadratic programming* (QCQP) problem. Further, by manipulating the *Gram matrix* of data instead of the coordinates, we relax this problem to a *semi-definite programming* (SDP) problem, which can be solved readily. Besides, we show that this semi-definite formulation can serve as a general framework for semi-supervised manifold alignment and embedding. Experiments on aligning various data demonstrate the effectiveness of our method.

References

1. Seung, H.S., Lee, D.D.: The manifold ways of perception. Science **290** (2000) 2268–2269
2. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290** (2000) 2323–2326
3. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation **15** (2003) 1373–1396
4. Weinberger, K.Q., Saul, L.K.: Unsupervised learning of image manifolds by semidefinite programming. International Journal of Computer Vision **70(1)** (2006) 77–90
5. Ham, J., Ahn, I., Lee, D.: Learning a manifold-constrained map between image sets: Applications to matching and pose estimation. In: CVPR-06. (2006)
6. Lyons, M.J., Kamachi, M., Gyoba, J., Akamatsu, S.: Coding facial expressions with gabor wavelets. In: Procedings of the 3rd IEEE Aut. Face and Gesture Recog. (1998)
7. Ham, J., Lee, D., Saul, L.: Learning high dimensional correspondence from low dimensional manifolds. In: Workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining, ICML-03. (2003)
8. Ham, J., Lee, D., Saul, L.: Semisupervised alignment of manifolds. In: Proceedings of the 8th International Workshop on Artificial Intelligence and Statics (*AISTATS 2005*). (2005)
9. Verbeek, J., Roweis, S., Vlassis, N.: Non-linear cca and pca by alignment of local models. In: Advances in NIPS-04. (2004)
10. Verbeek, J., Vlassis, N.: Gaussian fields for semi-supervised regression and correspondence learning. Pattern Recognition **39(10)** (2006) 1864–1875
11. Boyd, S.P., Vandenberghe, L.: Convex Optimization. Cambridge, UK (2004)
12. Sturm, J.F.: Using sedumi 1.02, a matlab toolbox for optimization overy symmetric cones. Optimization Methods and Software **11-12** (1999) 625C653
13. Wang, F., Zhang, C.: Label propagation through linear neighborhoods. In: ICML-06. (2006)
14. Nene, S.A., Nayar, S.K., Murase, H.: Columbia object image library (coil-20). Technical report, Technical Report CUCS-005-96 (1996)
15. N. Gourier, D. Hall, J.L.C.: Estimating face orientation from robust detection of salient facial features. In: Proceedings of Pointing 2004, ICPR, International Workshop on Visual Observation of Deictic Gestures, Cambridge, UK. (2004)
16. Cohen, W.W., Schapire, R.E., Singer, Y.: Learning to order things. In: Advances in NIPS-98. (1998)
17. Athitsos, V., Alon, J., Sclaroff, S., Kollios, G.: Boostmap: A method for efficient approximate similarity rankings. In: CVPR-04. (2004)
18. Schultz, M., Joachims, T.: Learning a distance metric from relative comparisons. In: Advances in NIPS-03. (2003)
19. Rosales, R., Fung, G.: Learning sparse metrics via linear programming. In: KDD-06. (2006)
20. Bernhard Schölkopf, Alexander Smola, K.R.M.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation **10** (1998) 1299–1319
21. Weinberger, K., Fei, S., Saul, L.K.: Learning a kernel matrix for nonlinear dimensionality reduction. In: ICML-04. (2004)