Margin Based Semi-Supervised Elastic Embedding for Face Image Analysis

F. Dornaika[1,2]
[1] University of the Basque Country UPV/EHU, San Sebastian, Spain
[2] IKERBASQUE, Basque Foundation for Science, Bilbao, Spain
fadi.dornaika@ehu.es

Y. El Traboulsi [1]
University of the Basque Country UPV/EHU, San Sebastian, Spain
youssoftraboulsi@gmail.com

Abstract

This paper introduces a graph-based semi-supervised elastic embedding method as well as its kernelized version for face image embedding and classification. The proposed frameworks combines Flexible Manifold Embedding and non-linear graph based embedding for semi-supervised learning. In both proposed methods, the nonlinear manifold and the mapping (linear transform for the linear method and the kernel multipliers for the kernelized method) are simultaneously estimated, which overcomes the shortcomings of a cascaded estimation. Unlike many state-of-the art non-linear embedding approaches which suffer from the out-of-sample problem, our proposed methods have a direct out-of-sample extension to novel samples. We conduct experiments for tackling the face recognition and image-based face orientation problems on four public databases.These experiments show improvement over the state-of-the-art algorithms that are based on label propagation or graph-based semi-supervised embedding.

Keywords: Manifold learning, semi-supervised learning, graph-based embedding, out-of-sample extension, classification

1. Introduction

Feature extraction with dimensionality reduction is an important step and essential process in embedding face images. Although the supervised feature extraction methods had been successfully applied to many pattern recognition applications, they require a full labeling of data samples. It is well-known that it is much easier to collect unlabeled data than labeled samples. The labeling process is often expensive, time consuming, and requires intensive human involvement. As a result, partially labeled datasets are more frequently encountered in real-world problems.

In the last decade, semi-supervised learning algorithms have been developed to effectively utilize limited number of labeled samples and a large amount of unlabeled samples for real-world applications [29, 5, 30, 18, 11]. In the past years, many graph-based methods for semi-supervised learning have been developed. The main advantage of graph-based methods is their ability to identify classes of arbitrary distributions. The use of data-driven graphs has led to many progresses in the field of semi-supervised learning (e.g., [3, 26, 16, 8, 7]). Toward classification, an excellent subspace should be smooth as well as discriminative. Hence, a graph-theoretic learning framework is usually deployed to simultaneously meet the smoothness requirement among nearby samples and the discriminative requirement among differently labeled samples (e.g.,[13]). In [10], the authors propose a joint learning of labels and distance metric approach, which is able to optimize the labels of unlabeled samples and a Mahalanobis distance metric in a unified scheme. It was shown that a good distance metric can be constructed with only very few training samples.

In addition to the use of partial labeling in semi-supervised learning, many researchers use pairwise constraints which can be seen as another form of side information [4]. These constraints are simply indicating if two instances are similar (must-link) or dissimilar (cannot-link). These constraints are usually used for getting a linear or non-linear embedding by adding these constraints to the criterion derived from unlabelled data samples [23]. The final application is to help spectral clustering recover from an undesirable partition.

Semi-supervised learning has benefited from many advances proposed for supervised and unsupervised manifold learning which aims to represent data samples in appropriate subspaces according to some criteria (e.g., [2, 25, 22]). Despite the success of many graph-based algorithms in dealing with partially labeled problems, there are still some problems that are not properly addressed. Almost all semi-supervised feature extraction techniques can suffer from one of the following limitations:

- The non-linear semi-supervised approaches do not have, in general, a function that can map unseen data samples. In other words, the non-linear methods provide embedding for only the training data. This is the transductive setting, i.e., the test set coincides with the set of unlabeled samples in the training dataset. Indeed, solving the out-of-sample extension is still an open problem for non-linear embedding techniques.

- Almost all proposed semi-supervised approaches target the estimation of a linear transform that maps original data into a low dimensional subspace. While this simplifies the learning processes and gets rid of the out-of-sample problem, there is no guarantee that such approaches will have optimal performances for all datasets. The main reason behind this is that the criterion used is already a rigid constraint that is based on solving a linear mapping. Any coordinate in the low-dimensional subspace is supposed to be a linear combination of the original features. Thus, the adopted criterion that derives the linear mapping has not the flexibility to adapt it to a given non-linear model.

In this paper, we propose a graph-based semi-supervised elastic embedding method as well as its kernelized version. The dimension of the final embedding obtained by the two proposed methods is not limited to the number of classes and they can be used by any kind of classifiers. Unlike many state-of-the art non-linear embedding approaches which suffer from the out-of-sample problem, our proposed methods have a direct out-of-sample extension to novel samples, and are thus easily generalized to the entire high-dimensional input space. The paper is structured as follows. In section II, we briefly review the main methods for semi-supervised learning including the graph-based label propagation and the semi-supervised embedding methods. In section III, we introduce our graph-based semi-supervised elastic embedding method. In Section IV, we present its kernel version. Section V contains the experimental results obtained with seven real datasets. This section compares the performance of the proposed methods with that of the competing methods. Finally, in section VI, we present our conclusions. In the sequel, capital bold letters denote matrices and small bold letters denote vectors.

2. Related work

2.1. Notation and preliminaries

We define the training data matrix as $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_l, \mathbf{x}_{l+1}, ..., \mathbf{x}_{l+u}] \in \mathbb{R}^{D\times(l+u)}$, where $\mathbf{x}_i|_{i=1}^{l}$ and $\mathbf{x}_i|_{i=l+1}^{l+u}$ are the labeled and unlabeled samples, respectively, with l and u being the total numbers of labeled and unlabeled samples, respectively, and D being the sample dimension. For face image analysis, the sample $\mathbf{x}_i$ can refer to a raw face image (or its descriptor). Let $N = l + u$ be the total number of training samples and n_c be the total number of labeled samples in the c^{th} class. We represent the labeled samples as $\mathbf{X}_l = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_l] \in \mathbb{R}^{D\times l}$ with the label of $\mathbf{x}_i$ as $y_i \in 1, 2, ..., C$, where C is the total number of classes. Let $\mathbf{S} \in \mathbb{R}^{(l+u)\times(l+u)}$ be the graph similarity matrix with $S(i,j)$ representing the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$, i.e., $S(i,j) = sim(\mathbf{x}_i, \mathbf{x}_j)$. In a supervised context, one can also consider two similarity matrices $\mathbf{S}_w$ and $\mathbf{S}_b$ that encode the within class and between class graphs, respectively. $\mathbf{S}_w$ encodes the pairwise similarities among samples having the same label. Thus, $S_w(i,j) = sim(\mathbf{x}_i, \mathbf{x}_j)$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ have the same class label; $S_w(i,j) = 0$, otherwise. Similarly, $\mathbf{S}_b$ encodes the pairwise similarities among samples having different labels. Thus, $S_b(i,j) = sim(\mathbf{x}_i, \mathbf{x}_j)$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ have different labels; $S_b(i,j) = 0$, otherwise. The function $sim(.,.)$ can be any symmetric function that measures the similarity between two samples. This can be given by the cosine or the Gaussian kernel.

For each similarity matrix, a Laplacian matrix can be computed. For the similarity matrix $\mathbf{S}$, the Laplacian matrix is given by $\mathbf{L} = \mathbf{D} - \mathbf{S}$ where $\mathbf{D}$ is a diagonal matrix whose elements are the row (or column since the similarity matrix is symmetric) sums of $\mathbf{S}$ matrix. Similar expression can be found for $\mathbf{L}_b$ and $\mathbf{L}_w$. The normalized Laplacian $\hat{\mathbf{L}}$ is defined by $\hat{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-1/2}\,\mathbf{S}\,\mathbf{D}^{-1/2}$ where $\mathbf{I}$ denotes the identity matrix.

We also define a binary label matrix $\mathbf{Y} \in \mathbb{B}^{N\times C}$ associated with the samples with $Y(i,j) = 1$ if $\mathbf{x}_i$ has label $y_i = j$; $Y(i,j) = 0$, otherwise. In addition to $\mathbf{Y}$, we can define an unknown label matrix denoted by $\mathbf{F} \in \mathbb{R}^{N\times C}$. In a semi-supervised setting, $\mathbf{F} = \begin{pmatrix} \mathbf{F}_{\mathcal{L}} \\ \mathbf{F}_{\mathcal{U}} \end{pmatrix}$ where $\mathbf{F}_{\mathcal{L}} = \mathbf{Y}_{\mathcal{L}}$.

2.2. Graph-based label propagation methods

In the last decade, the semi-supervised learning methods using graph-based label propagation attracted much attention. All of them impose that samples with high similarity should share similar labels. They differ by the regularization term as well as by the loss function used for fitting label information associated with the labeled samples. All of these methods take as input the similarity matrix $\mathbf{S}$ associated with data and the label matrix $\mathbf{Y}$ associated with the labeled samples. The state-of-the art label propagation methods (can also be called classifiers [19]) can be: Gaussian Fields and Harmonic Functions **(GFHF)** [31], Local and Global Consistency **(LGC)** [28], Laplacian Regularized Least Square **(LapRLS)** [1], Robust Multi-class Graph Transduction **(RMGT)** [12], Flexible Manifold Embedding (**FME**) [14], and Manifold Adaptive Label Propagation (**MALP**) [15].

2.3. Graph-based embedding methods

Many of the existing label propagation algorithms and non-linear embedding techniques can only work on transductive setting, which requires that both the training and test set are available during the learning process. Therefore, they are not always suitable for recognition applications where the test set is generally not available during the training phase. Unlike label propagation techniques that seek label inference, the embedding techniques seek a general coordinate representation where the dimension of the mapped data is not necessarily limited to the number of classes. Cai et al. extended Linear Discriminant Analysis (LDA) to Semi-supervised Discriminant Analysis (SDA) [2] by adding a geometrically-based regularization term in the objective function of LDA. The core assumption in SDA is still the manifold smoothness assumption, namely, nearby points will have similar representations in the lower-dimensional space. Semi-supervised Discriminant Embedding (SDE) [9, 25] can be seen as the semi-supervised variant of the Local Discriminant Embedding (LDE) method [6]. In order to discover both geometrical and discriminant structure of the data manifold, SDE relies on three graphs: the within-class graph G_w (intrinsic graph), the between-class graph G_b (penalty), and the graph defined over the whole set (labeled and unlabeled samples).

3. Proposed method

3.1. Margin based Discriminant Embedding for supervised case

The concept of margin among classes has been already used in the literature in order to get discriminant projections. For instance, the work of [21] used a margin that is defined in two local neighborhoods. This is a sample based margin that relies on a specific neighborhood size for intra-class and inter-class samples.

We proceed as follows. Let $\mathbf{w}$ be a projection vector. The linear representation of a sample $\mathbf{x}_i$ on that axis is $z_i = \mathbf{w}^T\, \mathbf{x}_i$. Let us denote the 1-D projections of labeled samples onto the axis $\mathbf{w}$ as $\{z_i = \mathbf{w}^T\, \mathbf{x}_i\}_{i=1}^{l}$. In matrix form, the latter equations can be written as $\mathbf{z}^T = (z_1, z_2, ..., z_l) = \mathbf{w}^T\, \mathbf{X}_l$.

For each labeled sample $\mathbf{x}_i$, we define a samplewise margin. This is given by the difference between two types of distances (along the projection axis $\mathbf{w}$): one is the distance between $\mathbf{x}_i$ and samples taking different labels, the other is the distance between $\mathbf{x}_i$ and samples sharing the same label. At $\mathbf{x}_i$ ($i \in C_k$), there are l_k intra-class distances and $l - l_k$ inter-class distances, where C_k ($k = 1, ..., C$) is the set of indices of samples from the same class and $l_k = |C_k|$. The margin associated with sample $\mathbf{x}_i$, in the projected space, is given by:

$$m(i) = \sum_{j \notin C_k} (z_i - z_j)^2\, \frac{1}{l - l_k} - \sum_{t \in C_k} (z_i - z_t)^2\, \frac{1}{l_k} \quad (1)$$

Note that the above sample-based margin is the one given in [21] in which all homogeneous and heterogeneous samples are used. Let us focus on l labeled examples $\mathbf{x}_1, ..., \mathbf{x}_l$ belonging to C classes, upon which two graphs G_w and G_b are built using label information. Within the intra-class graph G_w, we establish an undirected edge from each sample $\mathbf{x}_i$ in the graph to those sharing the same label as $\mathbf{x}_i$. Within the inter-class graph G_b, we establish a directed edge from $\mathbf{x}_i$ to all samples taking different labels. Therefore, G_w is undirected and G_b is directed, and then we define two similarity matrices $\mathbf{S}^w$ and $\mathbf{S}^b \in \mathbb{R}^{l \times l}$ pertaining to G_w and G_b respectively, by

$$S^w_{ij} \;=\; \begin{cases} \frac{1}{l_k} & if\ i \in C_k \text{ and } j \in C_k \\ 0, & otherwise \end{cases} \quad (2)$$

$$S^b_{ij} \;=\; \begin{cases} \frac{1}{l - l_k} & if\ i \in C_k \text{ and } j \notin C_k \\ 0, & otherwise \end{cases} \quad (3)$$

Note that the sum of each row in $\mathbf{S}^w$ and $\mathbf{S}^b$ is 1 and $\mathbf{S}^w$ is symmetric. We further define a diagonal matrix $\mathbf{D}^b$ with the entries being the column sums of $\mathbf{S}^b$. By using Eqs. (2) and (3), we can write the local margin defined in Eq. (1) as:

$$m(i) = \sum_{j=1}^{l} [(z_i - z_j)^2\, S^b_{ij} - (z_i - z_j)^2\, S^w_{ij}] \quad (4)$$

The average margin $\overline{m} = \frac{1}{l}\sum_{i=1}^{l} m(i)$ over all the labeled samples is given by:

$$\begin{aligned}
\overline{m} &= \frac{1}{l} \sum_{i=1}^{l}\sum_{j=1}^{l} [(z_i - z_j)^2\, S^b_{ij} - (z_i - z_j)^2\, S^w_{ij}] \\
&= \frac{1}{l} \left(\sum_{i=1}^{l} z_i^2 + \sum_{j=1}^{l} z_j^2 D^b_{jj} - 2 \sum_{i=1}^{l}\sum_{j=1}^{l} z_i S^b_{ij} z_j \right) - \\
& \quad \frac{1}{l} \left(2 \sum_{i=1}^{l} z_i^2 - 2 \sum_{i=1}^{l}\sum_{j=1}^{l} z_i S^w_{ij} z_j \right) \\
&= \frac{1}{l}\, \mathbf{z}^T (\mathbf{I} + \mathbf{D}^b - 2\,\mathbf{S}^b)\, \mathbf{z} - \frac{1}{l} \mathbf{z}^T (2\,\mathbf{I} - 2\,\mathbf{S}^w)\, \mathbf{z} \\
&= \frac{1}{l}\, \mathbf{z}^T (\mathbf{I} + \mathbf{D}^b - \mathbf{S}^b - \mathbf{S}^{bT})\, \mathbf{z} - \frac{2}{l}\, \mathbf{z}^T (\mathbf{I} - \mathbf{S}^w)\, \mathbf{z} \\
&= \frac{2}{l}\, \mathbf{w}^T \mathbf{X}_l\, \mathbf{D}_l\, \mathbf{X}_l^T\, \mathbf{w} - \frac{1}{l}\, \mathbf{w}^T \mathbf{X}_l\, \mathbf{M}_l \mathbf{X}_l^T\, \mathbf{w} \qquad (5)
\end{aligned}$$

where $\mathbf{D}_l = \mathbf{I} + \mathbf{D}^b$ and $\mathbf{M}_l = 3\,\mathbf{I} + \mathbf{D}^b + \mathbf{S}^b + \mathbf{S}^{bT} - 2\,\mathbf{S}^w$. In the above derivation, we have used the equalities $\mathbf{z}^T\, \mathbf{S}^b\, \mathbf{z} = \mathbf{z}^T\, \mathbf{S}^{bT}\, \mathbf{z}$ and $\mathbf{z}^T = \mathbf{w}^T\, \mathbf{X}_l$.

The projection axis can be found by maximizing the average margin $\overline{m}$, i.e., $\mathbf{w} = \arg\max_{\mathbf{w}} \overline{m}$. This maximization can be casted into a minimization problem using an equality constraint, i.e.

$$\begin{aligned} &\mathbf{w} = \arg\max_{\mathbf{w}} \overline{m} \Longrightarrow \\ &\mathbf{w} = \arg\min_{\mathbf{w}} [\mathbf{w}^T \mathbf{X}_l\, \mathbf{M}_l \mathbf{X}_l^T\, \mathbf{w}] \quad s.t. \quad \mathbf{w}^T \mathbf{X}_l\, \mathbf{D}_l \mathbf{X}_l^T\, \mathbf{w} = 1 \end{aligned}$$

We can also conclude that the non-linear embedding can be obtained by using the elements of $\mathbf{z} = \mathbf{X}_l^T\, \mathbf{w}$ as unknowns:

$$\mathbf{z} = \arg\min_{\mathbf{z}} \mathbf{z}^T\, \mathbf{M}_l\, \mathbf{z} \quad s.t. \quad \mathbf{z}^T\, \mathbf{D}_l\, \mathbf{z} = 1 \tag{6}$$

3.2. Proposed semi-supervised elastic embedding

In this section, we propose a semi-supervised elastic embedding that combines the merits of Flexible Manifold Embedding idea [14] and the non-linear graph based embedding. It should be noticed that the dimension of the final embedding is not limited to the number of classes. We assume that the non-linear embedding of the seen data samples is given by the matrix $\mathbf{Z} \in \mathbb{R}^{N \times d}$, i.e., the row vector $\mathbf{Z}_{i.}$ is the non-linear representation of the vector $\mathbf{x}_i$. We consider again the within class and between class graphs associated with the labeled data ($\mathbf{S}^w$ and $\mathbf{S}^b$) as well as the graph associated with the labeled and unlabeled data represented by its Laplacian matrix $\mathbf{L}$. We have shown that (6) is the criterion derived from the labeled part of data in order to get the non-linear 1D embedding. Note that, for semi-supervised case that deals with all N samples, the same criterion can be written as (the dimension of $\mathbf{z}$ is now N):

$$\mathbf{z} = \arg\min_{\mathbf{z}} \mathbf{z}^T\, \widetilde{\mathbf{M}}_l\, \mathbf{z} \quad s.t. \quad \mathbf{z}^T\, \widetilde{\mathbf{D}}_l\, \mathbf{z} = 1 \tag{7}$$

where $\widetilde{\mathbf{M}}_l \in \mathbb{R}^{N \times N}$ and $\widetilde{\mathbf{D}}_l \in \mathbb{R}^{N \times N}$ are the augmented form of $\mathbf{M}_l \in \mathbb{R}^{l \times l}$ and $\mathbf{D}_l \in \mathbb{R}^{l \times l}$:

$$\widetilde{\mathbf{M}}_l = \begin{pmatrix} \mathbf{M}_l & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad \widetilde{\mathbf{D}}_l = \begin{pmatrix} \mathbf{D}_l & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$$

A natural way to get a non-linear Semi-supervised Discriminant Embedding for more one dimension is to simultaneously minimize the following criteria:

$$\min_{\mathbf{Z}} trace(\mathbf{Z}^T\, \mathbf{L}\, \mathbf{Z}) \tag{8}$$

$$\min_{\mathbf{Z}} trace(\mathbf{Z}^T\, \widetilde{\mathbf{M}}_l\, \mathbf{Z}) \quad s.t. \quad \mathbf{Z}^T\, \widetilde{\mathbf{D}}_l\, \mathbf{Z} = \mathbf{I} \tag{9}$$

Note that (8) is simply the graph smoothness criterion that imposes locality preserving. We propose to combine the above criteria with regression and regularization terms. We simultaneously recover a non-linear embedding and its linear approximation by minimizing the following criterion that depends on both the non-linear embedding and the regression transform:

$$\begin{aligned} e(\mathbf{Z}, \mathbf{W}, \mathbf{b}) &= trace(\mathbf{Z}^T\, \mathbf{L}\, \mathbf{Z}) + \lambda\, trace(\mathbf{Z}^T\, \widetilde{\mathbf{M}}_l\, \mathbf{Z}) + \\ &\quad \mu\, (\|\mathbf{W}\|^2 + \gamma\, \|\mathbf{X}^T\, \mathbf{W} + \mathbf{1}\, \mathbf{b}^T - \mathbf{Z}\|^2) \\ &= trace(\mathbf{Z}^T\, \mathbf{L}_1\, \mathbf{Z}) + \\ &\quad \mu\, (\|\mathbf{W}\|^2 + \gamma\, \|\mathbf{X}^T\, \mathbf{W} + \mathbf{1}\, \mathbf{b}^T - \mathbf{Z}\|^2) \end{aligned} \tag{10}$$

where $\mathbf{L}_1 = \mathbf{L} + \lambda\, \widetilde{\mathbf{M}}_l$ and $\mathbf{1} \in \mathbb{R}^N$ is a column vector of 1s. μ, γ, and λ are positive balance parameters. Note that in the above criterion $\mathbf{Z}$ is the non-linear embedding and $\mathbf{W}$ and $\mathbf{b}$ are the linear transform that embed data $\mathbf{X}$ such that $\mathbf{Z} \approx \mathbf{X}^T\, \mathbf{W} + \mathbf{1}\, \mathbf{b}^T$.

The non-linear embedding as well as the regression are estimated by minimizing e. To obtain the optimal solution, we vanish the derivatives of the objective function e with respect to $\mathbf{W}$ and $\mathbf{b}$. We have:

$$\mathbf{b} = \frac{1}{N}(\mathbf{Z}^T\, \mathbf{1} - \mathbf{W}^T \mathbf{X}\, \mathbf{1}) \tag{11}$$

$$\mathbf{W} = \gamma\, (\gamma\, \mathbf{X}_c\, \mathbf{X}_c^T + \mathbf{I})^{-1} \mathbf{X}_c\, \mathbf{Z} = \mathbf{A}\, \mathbf{Z} \tag{12}$$

where $\mathbf{A} = \gamma\, (\gamma\, \mathbf{X}_c\, \mathbf{X}_c^T + \mathbf{I})^{-1} \mathbf{X}_c$ and $\mathbf{X}_c$ is the centered data matrix, i.e., $\mathbf{X}_c = \mathbf{X}\, \mathbf{H}_c$ with $\mathbf{H}_c$ being the centering matrix $\mathbf{H}_c = \mathbf{I} - \frac{1}{N} \mathbf{1}\, \mathbf{1}^T$. We use the above expression for $\mathbf{W}$ and $\mathbf{b}$ in the regression function $\mathbf{X}^T\, \mathbf{W} + \mathbf{1}\, \mathbf{b}^T$, we get:

$$\begin{aligned} \mathbf{X}^T\, \mathbf{W} + \mathbf{1}\, \mathbf{b}^T &= \mathbf{X}^T \mathbf{A}\, \mathbf{Z} + \frac{1}{N} \mathbf{1} \mathbf{1}^T \mathbf{Z} - \frac{1}{N} \mathbf{1} \mathbf{1}^T \mathbf{X}^T \mathbf{A}\, \mathbf{Z} \\ &= (\mathbf{I} - \frac{1}{N} \mathbf{1}\, \mathbf{1}^T)\, \mathbf{X}^T \mathbf{A}\, \mathbf{Z} + \frac{1}{N} \mathbf{1} \mathbf{1}^T \mathbf{Z} \\ &= \mathbf{H}_c\, \mathbf{X}^T \mathbf{A}\, \mathbf{Z} + \frac{1}{N} \mathbf{1} \mathbf{1}^T \mathbf{Z} = \mathbf{B}\, \mathbf{Z} \end{aligned}$$

with $\mathbf{B} = \mathbf{H}_c \mathbf{X}^T\, \mathbf{A} + \frac{1}{N} \mathbf{1} \mathbf{1}^T$. Thus, the criterion $e(\mathbf{Z}, \mathbf{W}, \mathbf{b})$ becomes:

$$\begin{aligned} e &= trace(\mathbf{Z}^T\, \mathbf{L}_1\, \mathbf{Z}) + \mu\, [trace(\mathbf{Z}^T\, \mathbf{A}^T \mathbf{A}\, \mathbf{Z}) \\ &\quad + \gamma\, trace((\mathbf{B}\, \mathbf{Z} - \mathbf{Z})^T (\mathbf{B}\, \mathbf{Z} - \mathbf{Z}))] \\ &= trace(\mathbf{Z}^T (\mathbf{L}_1 + \mu\, \mathbf{A}^T \mathbf{A} + \mu\gamma (\mathbf{B} - \mathbf{I})^T\, (\mathbf{B} - \mathbf{I}))\, \mathbf{Z}) \\ &= trace(\mathbf{Z}^T (\mathbf{L}_1 + \mathbf{E})\, \mathbf{Z}) \end{aligned} \tag{13}$$

where $\mathbf{E} = \mu\, \mathbf{A}^T \mathbf{A} + \mu\gamma (\mathbf{B} - \mathbf{I})^T\, (\mathbf{B} - \mathbf{I})$.

Thus, the non-linear embedding $\mathbf{Z}$ is estimated by minimizing the above criterion under the constraint used in the criterion (9):

$$\mathbf{Z}^{\star} = \arg\min_{\mathbf{Z}} trace(\mathbf{Z}^T (\mathbf{L}_1 + \mathbf{E})\, \mathbf{Z}) \quad s.t. \quad \mathbf{Z}^T\, \widetilde{\mathbf{D}}_l\, \mathbf{Z} = \mathbf{I}$$

Thus $\mathbf{Z}^{\star}$ can be solved by generalized eigenvalue decomposition. Once $\mathbf{Z}^{\star}$ is estimated the corresponding regression $\mathbf{W}^{\star}$ and $\mathbf{b}^{\star}$ are estimated by Eqs. (12) and (11), respectively. Given an unseen sample $\mathbf{x}_{test}$, its embedding (a column vector) is given by $\mathbf{z}_{test} = \mathbf{W}^{\star T} \mathbf{x}_{test} + \mathbf{b}^{\star}$.

3.3. Features of the proposed method

Our proposed method has several advantages over existing methods. First, unlike FME [14] which estimates labels, our method estimates a non-linear embedding whose dimension is not limited to the number of classes as it is the case with many frameworks adopting the label propagation paradigm [12, 14]. Second, it is a non-linear feature extractor that lends itself nicely to all machine learning tools that can be used in the output subspace with any dimension. Third, the proposed method is not limited to a transductive setting in the sense that it can work with unseen data. Fourth, it inherits the flexibility of FME [12, 14].

4. Kernelized version

In this section, we propose the kernel version of the proposed method. The motivation behind the use of kernel is that in some cases the non-linearity of data cannot be close to a linear subspace [27, 20]. In such cases, the flexibility introduced by the linear regression term may not lead to good approximation of the embedded data. The proposed kernel version aims at a flexibility in which the regression itself is non-linear. Thus, the role of the kernel trick is to seek an inductive non-linear embedding that is close to a real non-linear subspace.

In [1], it is shown that a kernelized version of label propagation based on the linear LapRLS can be written as:

$$\min \sum_{i=1}^{l} L(\mathbf{d}(\mathbf{x}_i), \mathbf{y}_i) + \lambda_A \, \|\mathbf{d}\|_{\mathcal{K}}^2 + \lambda_I \, \|\mathbf{d}\|_{\mathcal{G}}$$

where $L(.,.)$ is a loss function. $\|\mathbf{d}\|_{\mathcal{K}}$ and $\|\mathbf{d}\|_{\mathcal{G}}$ are the RKHS-norm and the graph-based smoothness of the model $\mathbf{d}$, respectively. The linear homologue of the above criterion is given by the linear LapRLS where the function is given by $\mathbf{d}(\mathbf{x}_i) = \mathbf{W}^T \, \mathbf{x}_i + \mathbf{b}$. According to Belkin [1], the model $\mathbf{d}$ will expand over all the labeled and unlabeled points in the form of (here $\mathbf{d}$ is a row vector having C elements):

$$\mathbf{d}(\mathbf{x}) = \sum_{j=1}^{l+u} \mathbf{v}_j^T \, K(\mathbf{x}, \mathbf{x}_j) = [K(\mathbf{x}, \mathbf{x}_1), \ldots, K(\mathbf{x}, \mathbf{x}_{l+u})] \, \mathbf{V}$$

where $\mathbf{K} \in \mathbb{R}^{(l+u)\times(l+u)}$ is a kernel Gram matrix, and the matrix $\mathbf{V} \in \mathbb{R}^{(l+u)\times C}$ is the matrix of multipliers ($\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{l+u}]^T$). The entries of the Gram matrix are given by $K(\mathbf{x}_i, \mathbf{x}_j)$ that represents a dot product in feature space. This kernel function can be Gaussian or polynomial. For all samples, this condition can be written in matrix form as $\mathbf{Z} = \mathbf{K}\,\mathbf{V}$ where the rows of $\mathbf{Z}$ will be the predicted labels.

In our proposed kernel version, the non-linear embedding of the feature vector $\mathbf{x}_i$ is represented by the row vector $\mathbf{Z}_{i.}$ whose dimension is d. Note that the value of d can be any arbitrary number such that $d \leq l + u$. The non-linear embedding of all training samples $\mathbf{Z}$ should be as close as possible to its kernel embedding given by $\mathbf{K}\,\mathbf{V}$. Thus, the global criterion that allows the simultaneous estimation of the matrix of multipliers $\mathbf{V}$ and the non-linear embedding $\mathbf{Z}$ will be given by:

$$\begin{aligned} &e(\mathbf{Z}, \mathbf{V}) = trace(\mathbf{Z}^T \, \mathbf{L}_1 \, \mathbf{Z}) + \\ &\mu \, [trace(\mathbf{V}^T \, \mathbf{K}\,\mathbf{V}) + \gamma \, trace((\mathbf{K}\,\mathbf{V} - \mathbf{Z})^T \, (\mathbf{K}\,\mathbf{V} - \mathbf{Z}))] \end{aligned} \tag{14}$$

At the extremum of e, the derivative of e w.r.t. $\mathbf{V}$ should vanish. This gives:

$$2\mathbf{K}\,\mathbf{V} + 2\gamma \, \mathbf{K} \, (\mathbf{K}\,\mathbf{V} - \mathbf{Z}) = \mathbf{0}$$

This can written in the following form:

$$\mathbf{V} = \gamma \, (\mathbf{I} + \gamma\mathbf{K})^{-1} \, \mathbf{Z} = \mathbf{A}_1 \, \mathbf{Z}$$

where $\mathbf{A}_1 = \gamma \, (\mathbf{I} + \gamma\mathbf{K})^{-1}$. By plugging the above expression in Eq. (14), this becomes:

$$\begin{aligned} e(\mathbf{Z}) &= trace(\mathbf{Z}^T \, \mathbf{L}_1 \, \mathbf{Z}) + \\ &\quad \mu \, trace(\mathbf{Z}^T \, \mathbf{A}_1^T \, \mathbf{A}_1 \, \mathbf{Z}) + \mu \, \gamma \, trace(\mathbf{Z}^T \, \mathbf{B}_1^T \, \mathbf{B}_1 \, \mathbf{Z}) \\ &= trace(\mathbf{Z}^T \, (\mathbf{L}_1 + \mu\mathbf{A}_1^T \, \mathbf{K}\mathbf{A}_1 + \mu \, \gamma \, \mathbf{B}_1^T \, \mathbf{B}_1) \, \mathbf{Z}) \end{aligned} \tag{15}$$

with $\mathbf{B}_1 = \mathbf{K}\,\mathbf{A}_1 - \mathbf{I}$.

The solution $\mathbf{Z}^\star$ is estimated by minimizing the above criterion under the constraint used in the criterion (9):

$$\begin{aligned} &\arg\min_{\mathbf{Z}} trace(\mathbf{Z}^T (\mathbf{L}_1 + \mu\mathbf{A}_1^T \, \mathbf{K}\mathbf{A}_1 + \mu \, \gamma \, \mathbf{B}_1^T \, \mathbf{B}_1) \, \mathbf{Z}) \\ &s.t. \, \mathbf{Z}^T \, \widetilde{\mathbf{D}}_l \, \mathbf{Z} = \mathbf{I} \end{aligned}$$

Thus $\mathbf{Z}^\star$ can be solved by generalized eigenvalue decomposition. Given an unseen sample $\mathbf{x}_{test}$ its embedding (a column vector) is given by $\mathbf{z}_{test} = \mathbf{V}^{\star T} \, [K(\mathbf{x}_{test}, \mathbf{x}_1), \ldots, K(\mathbf{x}_{test}, \mathbf{x}_N)]^T$.

5. Performance study

We test our proposed methods on four public face datasets: Extended Yale[1], FERET[2], PIE[3], and FacePix[4]. The first three datasets are used for face recognition tasks. However, FacePix dataset is used for coarse face orientation estimation. This dataset includes a set of face images with pose angle variations. It is composed of 181 face images (representing yaw angles from −90° to +90° at 1 degree increments) of 30 different subjects, with a total of 5430 images. We subsampled this dataset so that we have 10 different yaw angles/classes, each with 30 subjects. The classification carried out for this dataset concerned the yaw angle classification.

[1] $www.vision.ucsd.edu/\sim\ leekc/ExtYaleDatabase/ExtYaleB.html$

[2] $www.itl.nist.gov/iad/humanid/feret/$

[3] $http://www.ri.cmu.edu/projects/project\_418.html$

[4] $www.facepix.org/$

We adopt a common preprocessing step for images in which the images of all datasets are resized to 32×32 pixels in order to get small sizes.

Semi-supervised learning and empirical setting We compare our proposed methods with GFHF, RMGT, LapRLS, FME, SDA, SDE, and Transductive Component Analysis (TCA) [13]. For the methods relying on a projection matrix (SDA, SDE, TCA, and the two proposed methods), any classifier can be used with the obtained mapped data in order to classify the unlabeled and unseen data samples. All compared semi-supervised methods use the graph Laplacian, $\mathbf{L}$, associated with the training data. For a fair comparison, we adopt the same graph for all methods. This graph is constructed using the KNN graph (symmetric KNN) and the Gaussian kernel for the edge weights. Thus, the weight associated with each neighboring pair is given by $S(\mathbf{x}_i, \mathbf{x}_j) = exp(-||\mathbf{x}_i - \mathbf{x}_j\|^2/t_0)$ where $t_0 \in \mathbb{R}^+$ is the kernel bandwidth parameter. It is set as in many works to the average of squared distances in the training set. The neighborhood size was set to 10. For SDE method, we need to compute the within-class and the between-class graph (built on the labeled subset). The weights associated are set to ones or zeros, i.e. the corresponding similarity matrices $\mathbf{S}_b$ and $\mathbf{S}_w$ are binary matrices. It is worth noting that all compared methods used the same data graph. This makes sure that the difference in performance is due to the embedding method only and not to the data graph.

We randomly select 50% of data as the training dataset and use the remaining 50% data as the unseen test dataset. Among the training data, we randomly label P samples per class and treat the other training samples as unlabeled data. The above setting is a natural setting to compare different methods. All the training data (labeled and unlabeled samples) are used to learn a subspace (i.e., a projection matrix) for semi-supervised embedding methods or a classifier for the label propagation methods. In all the experiments, PCA is used as a preprocessing step to preserve 98% energy of the data.

Method comparison For LapRLS, FME, TCA, SDA, and SDE, two regularization parameters should be tuned. For our proposed methods three parameters are used, namely λ, μ, and γ. For fair comparison, we set each parameter to a subset of values belonging to $\{10^{-9}, 10^{-6}, 10^{-3}, 1, 10^3, 10^6, 10^9\}$ as in [14].

We then report the top-1 recognition accuracy (best average recognition rate) of all methods from the best parameter configuration.

Tables 1 and 2 report the best average recognition accuracy (for all datasets) over ten random splits on the unlabeled data and the test data, which are referred to as Unlabel and Test, respectively. Note that in Table 2 the classification concerned the yaw angle using face patches (model-less 3D face orientation estimation).

For the embedding methods (SDA, SDE, TCA, and the two proposed methods), the classification was performed using the Nearest Neighbor classifier. For the two proposed methods, the dimension of the embedding is bounded by the number of training samples N. Thus, for each parameter configuration associated with the criterion and for each split we have a curve for the recognition rate that depict the rate at several sampled dimensions. Thus, for each parameter configuration, the performance is set to the best rate in the average curve which was obtained by averaging the rate curves over the ten splits. For the kernel method, we used the Gaussian kernel whose expression is given by $K(\mathbf{x}_i, \mathbf{x}_j) = exp(-||\mathbf{x}_i - \mathbf{x}_j\|^2/(2^m\, t_0))$ in which t_0 is set to the average of squared distances in the training set, and m is an integer chosen in the interval $\{1, 2, 3, 4, 5, 6\}$.

Figure 1 illustrates the average recognition rate curves as a function of feature dimension for Extended Yale and FERET datasets. The used classifier was the Nearest Neighbor (NN) classifier. These curves were obtained for the test part of data using three labeled samples per class. We recall that FME method does not depend on the feature dimension. We stress the fact that the maximum dimensions of all methods are not the same. Indeed, the maximum dimension of SDA method is given by $C - 1$, and the maximum dimension of SDE is given by the dimension of input samples. For the two proposed methods, the maximum dimension is given by the number of training samples. Figure 2 illustrates the average recognition rate curves as a function of feature dimension when the used classifier was a linear Support Vector Machine (SVM). Figure 3 illustrates the average recognition rate curves as a function of feature dimension when the used classifier was the Two Phase Test Sample Sparse Representation (TPTSSR) classifier [24] for which the number of chosen neighboring samples was set to 5%.

We can draw the following conclusions:

(1) In general, the proposed method and its kernelized version have given the best recognition rate.

(2) In general, the kernel version has given better performance than the non-kernel method.

(3) At low dimensions, the rate obtained with the TCA method was poor. Indeed, a competing performance for TCA was obtained whenever enough features were used.

(4) The superiority of the proposed methods holds for three different classifiers: Nearest Neighbor, Support Vector Machine, and the Two Phase Test Sample Sparse Representation. This indicates that the embedding provided by the proposed methods was more discriminative than that provided by the competing graph-based embedding techniques.

(5) As can be seen from the recognition accuracy curves, by increasing the number of features in the projection sub-

space (obtained by the proposed methods) the recognition accuracy of the proposed methods will not necessarily increase. Thus, in practice, the two proposed methods will provide good performance even with few dimensions.

(6) We can observe that unlike many non-linear embedding methods whose performance deteriorates by increasing the number of features (e.g., [17]), ours do not have such disadvantage.

6. Conclusion

This paper presented two novel graph-based semi-supervised embedding methods for classification tasks. More precisely, we propose a graph-based semi-supervised elastic embedding as well as its kernelized version. The proposed schemes retained the merits of Flexible Manifold Embedding and the graph-based non-linear embedding. The proposed methods simultaneously estimate a non-linear embedding as well as its out-of-sample transform that is needed for mapping the unseen samples.

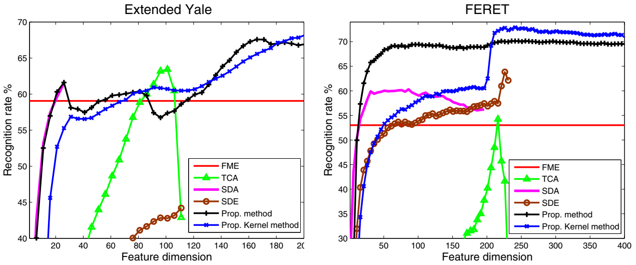

Figure 1. Recognition accuracy vs. feature dimension for Extended Yale and FERET datasets (test evaluation). Three labeled samples per class were used. The classifier used was 1-NN.

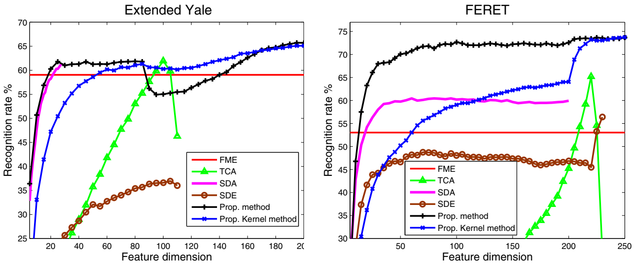

Figure 2. Recognition accuracy vs. feature dimension for Extended Yale and FERET datasets (test evaluation). Three labeled samples per class were used. The classifier used was a linear SVM.

References

[1] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006. 2, 5

[2] D. Cai, X. He, and J. Han. Semi-supervised discriminant analysis. In *IEEE Int. Conf. Comput.Vision*, 2007. 1, 3

Table 1. The best average recognition rates in (%) on ten random splits using several semi-supervised learning frameworks. U and T denote Unlabel and Test, respectively. The three columns correspond to 1, 2, and labeled samples per class.

Ext Yale	**1 sample**		**2 samples**		**3 samples**	
Method	*U*	*T*	*U*	*T*	*U*	*T*
GFHF	19.0	-	37.3	-	42.9	-
RMGT	23.0	-	40.5	-	45.6	-
LapRLS	**44.9**	**41.7**	59.6	56.7	61.3	59.1
SDA	36.6	34.8	57.2	55.0	65.0	61.5
SDE	40.0	37.7	54.5	52.4	50.0	49.0
TCA	40.1	37.3	57.7	56.4	67.4	64.1
FME	38.4	35.6	59.9	56.6	64.8	59.1
Linear method	44.6	41.1	**65.1**	**61.4**	**73.5**	67.8
Kernel version	44.6	41.1	64.3	61.2	73.1	**68.8**

FERET	**1 sample**		**2 samples**		**3 samples**	
Method	*U*	*T*	*U*	*T*	*U*	*T*
GFHF	17.8	-	25.3	-	29.6	-
RMGT	19.2	-	26.4	-	31.1	-
LapRLS	39.0	35.6	50.8	47.9	59.6	60.2
SDA	21.7	21.0	37.7	38.5	46.8	56.2
SDE	24.6	41.2	38.8	54.6	42.3	62.1
TCA	32.0	32.8	41.8	46.4	47.4	58.5
FME	35.5	27.9	47.2	39.5	54.1	53.0
Linear method	46.5	42.9	59.1	57.3	65.5	70.3
Kernel version	**52.9**	**48.2**	**64.7**	**60.7**	**70.6**	**72.9**

PIE	**1 sample**		**2 samples**		**3 samples**	
Method	*U*	*T*	*U*	*T*	*U*	*T*
GFHF	10.3	-	18.4	-	22.5	-
RMGT	11.4	-	19.3	-	22.9	-
LapRLS	29.3	32.4	35.5	40.1	30.3	32.5
SDA	15.3	19.6	35.5	41.2	48.2	52.8
SDE	20.9	29.2	28.6	36.3	31.3	34.9
TCA	23.8	26.1	36.7	40.0	44.8	47.4
FME	23.7	25.7	35.3	41.3	41.6	46.1
Linear method	32.8	34.6	42.3	44.7	49.9	52.0
Kernel version	**33.1**	**35.5**	**45.5**	**47.8**	**52.3**	**53.8**

Table 2. Best average yaw angle classification using 10 classes/poses for FacePix dataset. U and T denote Unlabel and Test, respectively.

Yaw angle	**1 sample**		**2 samples**		**3 samples**	
Method	*U*	*T*	*U*	*T*	*U*	*T*
GFHF	55.0	-	62.1	-	68.9	-
RMGT	61.2	-	65.7	-	72.1	-
LapRLS	62.4	58.5	68.5	66.7	74.2	74.9
SDA	55.2	58.2	65.1	71.2	74.7	75.2
SDE	64.3	68.2	67.8	72.2	74.2	75.3
TCA	59.5	76.3	62.2	**79.0**	67.2	**80.8**
FME	58.5	61.6	66.2	68.3	73.0	74.3
Linear method	69.3	72.5	72.0	76.5	76.3	76.9
Kernel version	**73.8**	**76.3**	**74.9**	**79.0**	**83.0**	**80.8**

[3] G. Camps-Valls, T. B. Marsheva, and D. Zhou. Semi-

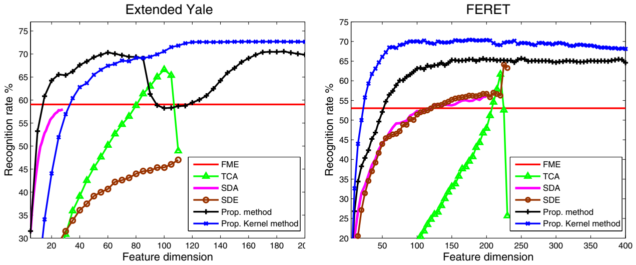

Figure 3. Recognition accuracy vs. feature dimension for Extended Yale and FERET datasets (test evaluation). Three labeled samples per class were used. The classifier used was the Two Phase Test Sample Sparse Representation (TPTSSR).

supervised graph-based hyperspectral image classification. *IEEE Trans. Geoscience and Remote Sensing*, 45(10):3044–3054, 2007. 1

[4] H. Cevikalp. Semi-supervised dimensionality reduction using pairwise equivalence constraints. In *International Conference on Computer Vision Theory and Applications*, 2009. 1

[5] O. Chapelle, B. Scholkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, Cambridge MA, 2006. 1

[6] H. Chen, H. Chang, and T. Liu. Local discriminant embedding and its variants. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2005. 3

[7] F. Dornaika and A. Bosaghzadeh. Adaptive graph construction using data self-representativeness for pattern classification. *Information Sciences*, 325:118 – 139, 2015. 1

[8] F. Dornaika, A. Bosaghzadeh, H. Salmane, and Y. Ruichek. Graph-based semi-supervised learning with local binary patterns for holistic object categorization. *Expert Systems with Applications*, 41(17):7744–7753, 2014. 1

[9] H. Huang, J. Liu, and Y. Pan. Semi-supervised marginal fisher analysis for hyperspectral image classification. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, I-3:377–382, 2012. 3

[10] B. Liu, M. Wang, R. Hong, Z. Zha, and X. Hua. Joint learning of labels and distance metric. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 40(3):973–978, 2010. 1

[11] C.-L. Liu, W.-H. Hsaio, C.-H. Lee, and F.-S. Gou. Semi-supervised linear discriminant clustering. *IEEE Transactions on Cybernetics*, 44(7):989–1000, 2014. 1

[12] W. Liu and S. Chang. Robust multi-class transductive learning with graphs. In *Computer Vision and Pattern Recognition*, 2009. 2, 5

[13] W. Liu, D. Tao, and J. Liu. Transductive component analysis. In *IEEE International Conference on Data Mining*, 2008. 1, 6

[14] F. Nie, D. Xu, I. Tsang, and C. Zhang. Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction. *IEEE Transactions on Image Processing*, 19(7):1921–1932, 2010. 3, 4, 5, 6

[15] X. Pei, Z. Lyu, C. Chen, and C. Chen. Manifold adaptive label propagation for face clustering. *Cybernetics, IEEE Transactions on*, PP(99):1–1, 2014. 3

[16] B. Raducanu, A. Bosaghzadeh, and F. Dornaika. Facial expression recognition based on multi-view observations with application to social robotics. In *Workshop on Computer Vision for Affective Computing*, pages 1–8, november 2014. 1

[17] B. Raducanu and F. Dornaika. A supervised non-linear dimensionality reduction approach for manifold learning. *Pattern Recognition*, 45:2432–2444, 2012. 7

[18] T. Silva and L. Zhao. Network-based stochastic semisupervised learning. *IEEE Transactions on Neural Networks and Learning Systems*, 23(3):451–466, 2012. 1

[19] C. Sousa, S. Rezende, and G. Batista. Influence of graph construction on semi-supervised learning. In *European Conference on Machine Learning*, 2013. 2

[20] J. Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, England, 2006. 5

[21] F. Wang, X. Wang, D. Zhang, C. Zhang, and T. Li. Marginface: A novel face recognition method by average neighborhood margin maximization. *Pattern Recognition*, 42:2863–2875, 2009. 3

[22] G. Wang, F. Wang, T. Chen, D. Yeung, and F. Lochovsky. Solution path for manifold regularized semisupervised classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 42(2):308–319, 2011. 1

[23] X. Wang, B. Qian, and I. Davidson. On constrained spectral clustering and its applications. *Data Mining and Knowledge Discovery*, 28(1):1–30, 2014. 1

[24] Y. Xu, D. Zhang, J. Yang, and J.-Y. Yang. A two-phase test sample sparse representation method for use with face recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(9):1255–1262, 2011. 6

[25] G. Yu, G. Zhang, C. Domeniconi, Z. Yu, and J. You. Semi-supervised classification based on random subspace dimensionality reduction. *Pattern Recognition*, 45:1119–1135, 2012. 1, 3

[26] Y. Zhang, K. Huang, X. Hou, and C. Liu. Learning locality preserving graph from data. *IEEE Transactions on Cybernetics*, 44(11):2088–2098, 2014. 1

[27] W. Zheng, Z. Lin, and H. Wang. L1-norm kernel discriminant analysis via bayes error bound optimization for robust feature extraction. *IEEE Transactions on Neural Networks and Learning Systems*, 25(4):793–803, 2014. 5

[28] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Scholkopf. Learning with local and global consistency. In *Adv. Neural Inf. Process. Syst.*, 2004. 2

[29] D. Zhou, J. Huang, and B. Scholkopf. Learning from labeled and unlabeled data on a directed graph. In *International Conference on Machine Learning*, 2005. 1

[30] X. Zhu. Cross-domain semi-supervised learning using feature formulation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 41(6):1627–1638, 2011. 1

[31] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *International Conference on Machine Learning*, 2003. 2