A Flexible Semi-Supervised Feature Extraction Method for Image Classification

Fadi Dornaika[1,2] and Youssof El Traboulsi[1]

[1] University of the Basque Country (UPV/EHU), San Sebastian, Spain
[2] IKERBASQUE, Basque Foundation for Science, Bilbao, Spain

Abstract. This paper proposes a novel discriminant semi-supervised feature extraction for generic classification and recognition tasks. The paper has two main contributions. First, we propose a flexible linear semi-supervised feature extraction method that seeks a non-linear subspace that is close to a linear one. The proposed method is based on a criterion that simultaneously exploits the discrimination information provided by the labeled samples, maintains the graph-based smoothness associated with all samples, regularizes the complexity of the linear transform, and minimizes the discrepancy between the unknown linear regression and the unknown non-linear projection. Second, we provide extensive experiments on four benchmark databases in order to study the performance of the proposed method. These experiments demonstrate much improvement over the state-of-the-art algorithms that are either based on label propagation or semi-supervised graph-based embedding.

1 Introduction

Feature extraction with dimensionality reduction is an important step and essential process in embedding data analysis. By computing an adequate representation of data that has a low dimension, more efficient learning and inference [1–4] can be achieved. 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.

Recently, semi-supervised learning algorithms were developed to effectively utilize limited number of labeled samples and a large amount of unlabeled samples for real-world applications [5, 6]. 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., [7–13]). 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 points and the discriminative requirement among differently labeled points (e.g., [14]). In addition to the use of partial labelling in

semi-supervised learning, many researchers use pairwise constraints which can be seen as another form of side information [15].

Despite the success of many graph-based algorithms in dealing with partially labeled problems [16], 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:

1. The non-linear semi-supervised approaches do not have, in general, an implicit 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 those techniques adopting non-linear embedding.
2. Almost all proposed semi-supervised approaches target the estimation of a linear transform that maps original data into a low dimensional space. While this simplifies the learning processes and gets rid of the out-of-sample problem, there is no guarantee that such approaches will be optimal for all datasets. The main reason behind this is that the criterion used is already a rigid constraint that contains only the linear mapping. Thus, any coordinate in the low-dimensional space is supposed to be a linear combination of the original features. Thus, the model has not the flexibility to adapt the linear model to a given non-linear model.

In addition to the above limitations, it is not clear what would be the performance of the semi-supervised approaches when minimal labeling is used. In this paper, we propose an Inductive Flexible Semi Supervised Feature Extraction. The aim is to combine the merits of Flexible Manifold Embedding and the non-linear graph-based embedding. The proposed method will be flexible since it estimates a non-linear manifold that is the closest one to a linear embedding. The non-linear manifold and the mapping are simultaneously estimated. The dimension of the final embedding obtained by our proposed method is not limited to the number of classes. This allows the application of any kind of classifiers once the data are embedded in new sub-spaces. Unlike nonlinear dimensionality reduction approaches which suffer from the out-of-sample problem, our proposed method has an obvious advantage that the learnt subspace has 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 2, we briefly review the main methods for semi-supervised learning including the graph-based label propagation and the semi-supervised embedding methods. In section 3, we introduce the IFSSFE method. Section 4 states the differences between the proposed method and the existing ones. Section 5 contains the experimental results obtained with four public datasets. This section compares the performance of the proposed method with the that of the competing methods. Finally, in section 6 we present our conclusions. In the sequel, capital bold letters denote matrices and small bold letters denote vectors.

2 Related work

In order to make the paper self-contained, this section will briefly describe some state-of-the art semi-supervised methods.

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, D being the feature dimension, and $N = l + u$ being the total number of training samples. Let n_c be the total number of labeled samples in the c^{th} class and represent the labeled samples as $\mathbf{X}_{\mathcal{L}} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_l] \in \mathbb{R}^{D\times l}$ with the label of $\mathbf{x}_i$ as $y_i \in 1, 2, ..., C$, where C is the total number of classes. Let $\mathbf{S} \in \mathbb{R}^{(l+u)\times(l+u)}$ as 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. 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 SSL graph-based label propagation methods 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 weighted graph $\mathbf{S}$ associated with data and the label matrix $\mathbf{Y}$. The state-of-the art label propagation algorithms (can also be called classifiers [17]) can be: Gaussian Fields and Harmonic Functions **(GFHF)** [18], Local and Global Consistency **(LGC)** [19], Laplacian Regularized Least Square **(LapRLS)** [20], Robust Multi-class Graph Transduction **(RMGT)** [21], and Flexible Manifold Embedding (**FME**) [22].

Gaussian Fields and Harmonic Functions The GFHF algorithm [18] solves the following optimization problem:

$$\min_{\mathbf{F}} \sum_{i,j} ||\mathbf{F}_{i.} - \mathbf{F}_{j.}||^2 S_{ij} = \min_{\mathbf{F}} trace(\mathbf{F}^T \mathbf{L} \mathbf{F}) \;\; s.t. \;\; \mathbf{F}_{\mathcal{L}} = \mathbf{Y}_{\mathcal{L}}$$

Given the graph affinity matrix $\mathbf{S}$ as well as the known labels $\mathbf{Y}_{\mathcal{L}} \in \mathbb{R}^{l \times C}$, the goal is to derive the labels of unlabeled samples, $\mathbf{F}_{\mathcal{U}} \in \mathbb{R}^{u \times C}$. It can be shown that the matrix of unknown labels is given by:

$$\mathbf{F}_{\mathcal{U}} = -\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1} \mathbf{L}_{\mathcal{U}\mathcal{L}} \mathbf{Y}_{\mathcal{L}} \tag{1}$$

where $\mathbf{L}_{\mathcal{U}\mathcal{U}}$ and $\mathbf{L}_{\mathcal{U}\mathcal{L}}$ are submatrices of the Laplacian matrix $\mathbf{L}$:

$$\mathbf{L} = \begin{pmatrix} \mathbf{L}_{\mathcal{L}\mathcal{L}} & \mathbf{L}_{\mathcal{L}\mathcal{U}} \\ \mathbf{L}_{\mathcal{U}\mathcal{L}} & \mathbf{L}_{\mathcal{U}\mathcal{U}} \end{pmatrix}$$

Local and Global Consistency The Local and Global Consistency algorithm [19] solves the following optimization problem:

$$\min_{\mathbf{F}} \left[trace(\mathbf{F}^T \widehat{\mathbf{L}} \mathbf{F}) + \mu \, trace((\mathbf{F} - \mathbf{Y})^T (\mathbf{F} - \mathbf{Y})) \right]$$

which gives the closed-form solution:

$$\mathbf{F} = (\mathbf{I} + \widehat{\mathbf{L}}/\mu)^{-1} \mathbf{Y}$$

Robust Multi-class Graph Transduction (RMGT) The RMGT algorithm solves the convex optimization problem $\min_{\mathbf{F}} trace(\mathbf{F}^T \mathbf{L} \mathbf{F})$ s.t. $\mathbf{F}_{\mathcal{L}} = \mathbf{Y}_{\mathcal{L}}$, $\mathbf{F}\,\mathbf{1}_C = \mathbf{1}_N$, $\mathbf{F}^T \mathbf{1}_N = N\boldsymbol{\Omega}$, where the vector $\boldsymbol{\Omega} \in \mathbb{R}^C$ is the class prior probabilities. The solution of this optimization problem is given by:

$$\mathbf{F}_{\mathcal{U}} = -\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1} \mathbf{L}_{\mathcal{U}\mathcal{L}} \mathbf{Y}_{\mathcal{L}} + \frac{\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1} \mathbf{1}_u}{\mathbf{1}_u^T \mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1} \mathbf{1}_u} (N \boldsymbol{\Omega}^T - \mathbf{1}_l^T \mathbf{Y}_{\mathcal{L}} + \mathbf{1}_u^T \mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1} \mathbf{L}_{\mathcal{U}\mathcal{L}} \mathbf{Y}_{\mathcal{L}})$$

Laplacian RLS The linear LapRLS defines a linear regression function that maps a feature vector $\mathbf{x}$ to its label representation $\mathbf{Y}_{i.}$, i.e., $\mathbf{Y}_{i.} = \mathbf{W}^T \mathbf{x}_i + \mathbf{b}$. The term Laplacian is due to the fact that the regularization term contains the classic Laplacian smoothing criterion. The linear LapRLS estimates the linear transform by optimizes the following criterion:

$$g(\mathbf{W}, \mathbf{b}) = \sum_{i=1}^{l} \|\mathbf{W}^T \mathbf{x}_i + \mathbf{b} - \mathbf{Y}_{i.}\|^2 + \lambda_A \|\mathbf{W}\|^2 + \lambda_I \, trace(\mathbf{W}^T \mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{W}) \tag{2}$$

where the two coefficients λ_A and λ_I balance the norm of $\mathbf{W}$, the manifold smoothness and the regression error. The closed-form solution is given by:

$$\begin{aligned} \mathbf{W} &= (\lambda_I \mathbf{X} \mathbf{L} \mathbf{X}^T + \mathbf{X}_{\mathcal{L}} \mathbf{X}_{\mathcal{L}}^T + \lambda_A \mathbf{I})^{-1} \mathbf{X}_{\mathcal{L}} \mathbf{Y}_{\mathcal{L}} \\ \mathbf{b} &= \mathbf{Y}_{\mathcal{L}}^T \mathbf{1}_l - \mathbf{W}^T \mathbf{X}_{\mathcal{L}} \mathbf{1}_l \end{aligned}$$

Flexible Manifold Embedding (FME) Flexible Manifold Embedding can be seen as a flexible variant of non-linear embedding where the embedding is given by the label distribution. FME simultaneously estimates the non-linear embedding of unlabel samples and the linear regression over these non-linear representations. In other words, FME can be seen as a framework that merges LGC and LapRLS in order to solve the out-of-sample extension problem. Compared with LapRLS, FME does not force the prediction labels to lie in the space spanned by all the samples. Therefore, it can be more flexible and it can better cope with the samples which reside on the nonlinear manifold. This framework simultaneously estimates the label matrix as well as a linear mapping by minimizing the following criterion:

$$\begin{aligned} g(\mathbf{F},\mathbf{W},\mathbf{b}) = trace((\mathbf{F}-\mathbf{Y})^T\,\mathbf{U}(\mathbf{F}-\mathbf{Y})) + trace(\mathbf{F}^T\,\mathbf{L}\,\mathbf{F}) \\ +\mu\,(\|\mathbf{W}\|^2 + \gamma\,\|\mathbf{X}^T\,\mathbf{W} + \mathbf{1}_N\,\mathbf{b}^T - \mathbf{F}\|^2) \end{aligned}$$

where μ and γ are two balance parameters, and $\mathbf{U}$ is a diagonal matrix whose first l diagonal elements are set to one and the rest $N-l$ are set to zero. As can be seen, the above criterion has four terms: the first is a fitting term over the labeled sample, the second is the smoothing term over all samples, the third is a regularization term, and the fourth term is the regression term. The sought solution $(\mathbf{F},\mathbf{W},\mathbf{b})$ is found by minimizing the above criterion. By vanishing the derivative of g with respect to $\mathbf{W}$ and $\mathbf{b}$, a relation between $\mathbf{F}$ and $\mathbf{W}$ can be obtained. Then, by vanishing the derivative with respect to $\mathbf{F}$ a closed form solution can be obtained. This is given by:

$$\mathbf{F} = (\mathbf{U} + \mathbf{L} + \mu\,\gamma\,\mathbf{H}_c - \mu\,\gamma^2\,\mathbf{Q})^{-1}\mathbf{U}\,\mathbf{Y}$$

with $\mathbf{Q} = \mathbf{X}_c^T\,\mathbf{X}_c\,(\gamma\,\mathbf{X}_c^T\,\mathbf{X}_c + \mathbf{I})^{-1}$ where $\mathbf{X}_c$ is the centered data matrix and $\mathbf{H}_c = \mathbf{I} - \frac{1}{N}\mathbf{1}_N\,\mathbf{1}_N^T$ is the centering matrix.

2.3 Graph-based embedding methods

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. Two main techniques represent the state-of-the art in semi-supervised graph-based embedding:

Semi-Supervised Discriminant Analysis (SDA) Cai et al. extended LDA to SDA [23] 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. We define as the data matrix of labeled data $\mathbf{X}_{\mathcal{L}} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_l]$. LDA can be seen as a particular case of a graph-based embedding.

Semi-Supervised Discriminant Embedding (SDE) SDE can be seen as the semi-supervised variant of the Local Embedding (LDE) method [24]. In order to discover both geometrical and discriminant structure of the data manifold, SDE [25, 26] 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 Inductive Flexible Semi Supervised Feature Extraction (IFSSFE)

In this section, we propose an Inductive Flexible Semi Supervised Feature Extraction (IFSSFE) that can combine the merits of Flexible Manifold Embedding idea 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 class. 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 as well as the graph associated the labeled and unlabeled data. The expression of the criteria associated with the non-linear Semi-Supervised Discriminant Embedding will be given by $\min_{\mathbf{Z}} trace(\mathbf{Z}^T \widetilde{\mathbf{L}}_w \mathbf{Z})$ $\max_{\mathbf{Z}} trace(\mathbf{Z}^T \widetilde{\mathbf{L}}_b \mathbf{Z})$ $\min_{\mathbf{Z}} trace(\mathbf{Z}^T \mathbf{L} \mathbf{Z})$:

By combining the above criteria together with the regression and regularization terms we can define a criterion that should be minimized. This is given by:

$$\begin{aligned} e(\mathbf{Z}, \mathbf{W}, \mathbf{b}) &= trace(\mathbf{Z}^T \mathbf{L} \mathbf{Z}) + trace(\mathbf{Z}^T \widetilde{\mathbf{L}}_w \mathbf{Z}) - \lambda\, trace(\mathbf{Z}^T \widetilde{\mathbf{L}}_b \mathbf{Z}) + \\ &\quad \mu\,(\|\mathbf{W}\|^2 + \gamma\,\|\mathbf{X}^T \mathbf{W} + \mathbf{1}_N\, \mathbf{b}^T - \mathbf{Z}\|^2) \qquad (3) \\ &= trace(\mathbf{Z}^T \mathbf{L}_1 \mathbf{Z}) + \mu\,(\|\mathbf{W}\|^2 + \gamma\,\|\mathbf{X}^T \mathbf{W} + \mathbf{1}_N\, \mathbf{b}^T - \mathbf{Z}\|^2) \qquad (4) \end{aligned}$$

where $\mathbf{L}_1 = \mathbf{L} + \widetilde{\mathbf{L}}_w - \lambda \widetilde{\mathbf{L}}_b$, μ, γ, and λ are three positive balance parameters.

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

$$\mathbf{b} = \frac{1}{N}(\mathbf{Z}^T\, \mathbf{1}_N - \mathbf{W}^T \mathbf{X}\, \mathbf{1}_N) \qquad (5)$$

$$\mathbf{W} = \gamma\,(\gamma\, \mathbf{X}_c\, \mathbf{X}_c^T + \mathbf{I})^{-1} \mathbf{X}_c\, \mathbf{Z} = \mathbf{A}\, \mathbf{Z} \qquad (6)$$

where $\mathbf{A} = \gamma\,(\gamma\, \mathbf{X}_c\, \mathbf{X}_c^T + \mathbf{I})^{-1} \mathbf{X}_c$. We use the above expression for $\mathbf{W}$ and $\mathbf{b}$ in the regression function $\mathbf{X}^T \mathbf{W} + \mathbf{1}_N\, \mathbf{b}^T$, we get:

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

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

$$e(\mathbf{Z},\mathbf{W},\mathbf{b}) = trace(\mathbf{Z}^T\,\mathbf{L}_1\,\mathbf{Z}) + \mu\,(trace(\mathbf{Z}^T\,\mathbf{A}^T\mathbf{A}\,\mathbf{Z}) + \gamma\,trace((\mathbf{B}\,\mathbf{Z}-\mathbf{Z})^T(\mathbf{B}\,\mathbf{Z}-\mathbf{Z})) \quad (7)$$

$$= trace(\mathbf{Z}^T(\mathbf{L}_1 + \mu\,\mathbf{A}^T\mathbf{A} + \mu\gamma(\mathbf{B}-\mathbf{I})^T\,(\mathbf{B}-\mathbf{I}))\,\mathbf{Z}) \quad (8)$$

$$= trace(\mathbf{Z}^T(\mathbf{L}_1 + \mathbf{E})\,\mathbf{Z}) \quad (9)$$

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 a constraint in order to avoid the trivial solution $\mathbf{Z} = \mathbf{0}$.

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

Thus $\mathbf{Z}^\star$ is given by the eigenvectors of $\mathbf{L}_1 + \mathbf{E}$ associated with the smallest eigenvalues. Once $\mathbf{Z}^\star$ is estimated the corresponding regression $\mathbf{W}^\star$ and $\mathbf{b}^\star$ are estimated by Eqs. (6) and (5).

Given an unseen sample $\mathbf{x}_{test}$ its embedding (a column vector) is given by $\mathbf{z}_{test} = \mathbf{W}^{\star T}\mathbf{x}_{test} + \mathbf{b}^\star$.

4 Difference between the proposed method and existing methods

Obviously, our proposed flexible method has several advantages compared with existing methods. Indeed it can combine the merits of graph-based semi-supervised label propagation and those of graph-based semi-supervised embedding methods. The advantages are as follows. First, unlike the FME method which estimates label distributions, 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 algorithm. Second, the proposed method is a kind of a non-linear feature extractor that lends itself nicely to all machine learning tools that can be used in the output space with any dimension in order to infer the class (classification) or the continuous label (regression). Third, the method is still inductive in the sense that it can work with unseen data. Fourth, it inherits the flexibility of FME in the sense that a non-linear embedding and a regression are found such that the non-linear embedding is close to the linear one obtained by regression (see Figure 1). Thus, the proposed method can better cope with the data sampled from a certain type of nonlinear manifold that is somewhat close to a linear subspace.

5 Performance evaluation

We test our proposed method on four datasets. In our experiments, we use three face datasets Extended Yale, FacePix and FERET, and one object database (COIL-20).

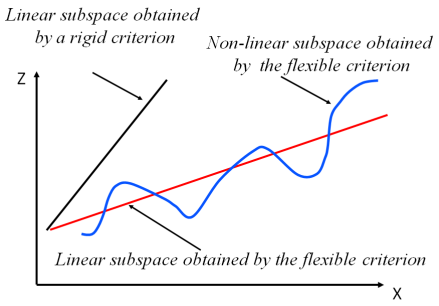

Fig. 1. An illustration of the difference between a rigid linear embedding and the proposed flexible scheme IFSSFE.

5.1 Datasets

- **Extended Yale**[1]**:** We use the cropped version contains 1774 face images of 28 individuals. The images of the cropped version contain illumination variations and facial expression variations. The image size is 192×168 pixels with 256-bit grey scale. The images are rescaled to 32×32 pixels in our experiments.
- **FERET**[2]**:** We use a subset of FERET database, which includes 1400 images of 200 distinct subjects, each subject has seven images. The subset involves variations in facial expression, illumination and pose. In our experiment, the facial portion of each original image is cropped automatically based on the location of eyes and resized to 32×32 pixels.
- **FacePix**[3]**:** This database 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 used a subset of this dataset in which each person has 18 images.
- **COIL-20**[4] This dataset (Columbia Object Image Library) consists of 1440 images of 20 objets. Each object has underwent 72 rotations (each object has 72 images). The objects display a wide variety of complex geometry and reflectance characteristics. We used a subset of the database with 18 images for each object (one image for every 20 degree of rotation).

5.2 Semi-supervised learning and empirical setting

We compare our proposed method with GFHF, Class Mass Normalized GFHF (GFHF+CMN), RMGT, LapRLS, SDA, SDE, FME, and LDA. It should be noted that all these methods are semi-supervised except LDA which is supervised.

[1] $http://vision.ucsd.edu/\sim leekc/ExtYaleDatabase/ExtYaleB.html$
[2] $http://www.itl.nist.gov/iad/humanid/feret/$
[3] $http://www.facepix.org/$
[4] $http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php$:

For the embedding methods (LDA, SDA, SDE, and the proposed method), any classifier can be used with the obtained mapped data in order to classify the unlabeled and unseen data samples. Since all compared semi-supervised methods used the graph Laplacian $\mathbf{L}$ associated with the training data, the graph was constructed using the classic KNN graph (symmetric KNN) and the RBF (or Gaussian) kernel for the edge weights. 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 values of neighborhood size was set to 10. For the proposed method, we need to compute the within-class and in 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 worthy 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% data as the training dataset and use the remaining 50% data as the 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, except that we only use the labeled data for subspace learning in LDA. In all the experiments, PCA is used as a preprocessing step to preserve 98% energy of the data.

5.3 Method comparison

For LapRLS, SDA, SDE, FME, two regularization parameters should be tuned. For our proposed method three parameters are used. Each of these parameters is set to a subset of values belonging to $\{10^{-9}, 10^{-6}, 10^{-3}, 1, 10^3, 10^6, 10^9\}$ as in [22], and then we report the top-1 recognition accuracy (best average recognition rate) from the best parameter configuration. Table 1 reports the best mean recognition accuracy (for the four datasets) over ten random splits on the unlabeled data and the test data, which are referred to as Unlabel and Test, respectively. For the embedding methods (LDA, SDA, SDE, IFSSFE), the classification was performed using the Nearest Neighbor classifier.

For the proposed method (IFSSFE), 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 depicts the rate at several sampled dimensions. Thus, for each parameter configuration, the performance is set to best rate in the mean curve which was obtained by averaging the rate curves over the splits.

Figure 2 illustrates the average recognition rate curves as a function of feature dimensions. These curves were obtained for the test part of data using one labeled sample per class. We recall that FME method does not depend on the dimension, 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 proposed method, the maximum dimension is given by the number of training samples.

From the results depicted in Table 1 and Figure 2, we can draw the following conclusions:

- In general, the proposed method IFSSFE has given the best recognition rate.
- For the non-face dataset (COIL-20), the improvement obtained by the proposed method was very significant compared with the performance of FME and the embedding methods SDA, and SDE. This holds true for all label percentages and for unlabeled and test data.
- For some datasets, the performance obtained with the test part of the data was better than the performance obtained with the unlabeled part. This can be explained by the fact that the captured model has a high generalization capacity.
- The performance of GFHF, GFHF+CMN, and RMGT (direct label propagation methods) was not that good for face datasets.
- More importantly, we can observe that the optimal performance of IFSSFE can be reached with a relatively low dimension. This property makes the proposed method very appealing in practice. Indeed, one needs to find a trade-off between a high recognition rate and a compact representation with a reduced number of dimensions.

5.4 Method performance with fixed dimension

In this section, we compare the performance of the FME method with that of our proposed method for which the dimension of features is fixed to the number of classes, C. Note that the FME method is essentially a label propagation algorithm that uses C features. We will show that even in the case where dimensionality of the embedding is fixed to C, IFSSFE is still superior to FME if the balance parameters were optimized at this fixed dimension. This is explained by the fact that the criterion used by the proposed method was the main reason for this obtained superiority. Table 2 illustrates the average performance of FME and IFSSFE in such conditions. Since the proposed IFSSFE is a generic semi-supervised embedding, we used four classifiers: 1-NN, Support Vector Machines (SVM with RBF Kernel), SVM (with polynomial degree equal to 1 and 3), and the Two Phase Test Sample Sparse Representation (TPTSSR) classifier [27]. As can be seen, even when IFSSFE is restricted to work with only C features, its performance is still better than that of FME. We can also observe that the use of other classifiers such as SVM and TPTSSR has enhanced the performance of the IFSSFE with respect to the Nearest Neighbor classifier.

Figure 3 illustrates the average performance of FME and the proposed IFSSFE as a function of feature dimension for three datasets: Extended Yale, FacePix, and COIL-20. The projection models associated with FME and IFSSFE were optimized on the fixed dimension given by C. In each plot, we show the average curve over ten splits. These curves depicts the performance on the

Table 1. The best average classification results on ten random splits. GFHF, GFHF+CMN, RMGT, LapRLS, and FME are based on label propagation. LDA, SDA, SDE, and the two proposed method are embedding methods for which nearest neighbor classifier was used after the projection.

Ext Yale	**1 labeled sample**		**2 labeled samples**		**3 labeled samples**	
Method	Unlabel(%)	Test(%)	Unlabel(%)	Test(%)	Unlabel(%)	Test(%)
LDA	36.6	34.8	56.9	55.2	64.4	60.1
GFHF	19.0	-	37.3	-	42.9	-
GFHF+CMN	26.3	-	41.0	-	45.9	-
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
FME	38.4	35.6	59.9	56.6	64.8	59.1
IFSSFE	**46.3**	41.2	**65.9**	**62.6**	**75.3**	**69.3**
FacePix	**1 labeled sample**		**2 labeled samples**		**3 labeled samples**	
Method	Unlabel(%)	Test(%)	Unlabel(%)	Test(%)	Unlabel(%)	Test(%)
LDA	26.0	26.9	39.1	39.6	48.2	46.6
GFHF	17.3	-	27.9	-	36.6	-
GFHF+CMN	21.6	-	30.3	-	38.0	-
RMGT	18.1	-	29.4	-	38.8	-
LapRLS	31.3	30.0	43.4	40.9	48.4	45.6
SDA	26.0	26.9	42.4	43.0	53.6	50.9
SDE	40.4	36.0	54.4	49.5	57.1	52.2
FME	28.1	28.0	42.7	40.3	50.5	47.7
IFSSFE	**40.6**	**37.8**	**56.6**	**53.6**	**65.8**	**61.2**
FERET	**1 labeled sample**		**2 labeled samples**		**3 labeled samples**	
Method	Unlabel(%)	Test(%)	Unlabel(%)	Test(%)	Unlabel(%)	Test(%)
LDA	21.7	21.0	35.7	36.4	43.9	55.0
GFHF	17.8	-	25.3	-	29.6	-
GFHF+CMN	23.6	-	30.9	-	38.4	-
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
FME	35.5	27.9	47.2	39.5	54.1	53.0
IFSSFE	**39.7**	**37.4**	**51.3**	50.1	**60.6**	**70.8**
COIL-20	**1 labeled sample**		**2 labeled samples**		**3 labeled samples**	
Method	Unlabel(%)	Test(%)	Unlabel(%)	Test(%)	Unlabel(%)	Test(%)
LDA	43.2	43.5	52.8	59.3	58.5	66.9
GFHF	52.8	-	58.3	-	63.2	-
GFHF+CMN	58.9	-	63.3	-	68.0	-
RMGT	57.1	-	60.4	-	65.1	-
LapRLS	58.9	54.1	64.8	65.7	69.3	71.7
SDA	43.2	43.5	53.3	59.0	60.4	66.9
SDE	56.0	55.5	66.8	65.6	75.4	72.4
FME	62.0	57.9	66.8	64.7	70.7	68.6
IFSSFE	**68.0**	**60.8**	**75.1**	**71.7**	**80.4**	**77.4**

test subset in which the number of labeled samples per class was set to three. For the proposed method IFSSFE, we used four classifiers: 1-NN, RBF SVM, polynomial SVM (degree=3), and the Two Phase Test Sample Sparse Representation (TPTSSR) classifier

6 Conclusion

This paper presented a novel semi-supervised dimensionality reduction method for classification tasks. We propose an Inductive Flexible Semi Supervised Feature Extraction that retained the merits of Flexible Manifold Embedding and the graph based non-linear embedding. The proposed method simultaneously estimates a non-linear embedding as well as a transform needed for mapping the unseen samples. The proposed method was evaluated on four benchmark databases. We have provided a comparison with several competing methods based on label propagation methods as well as on semi supervised graph-based embedding. Our proposed method outperformed the competing methods in most cases.

Acknowledgment This work was supported by the project EHU13/40.

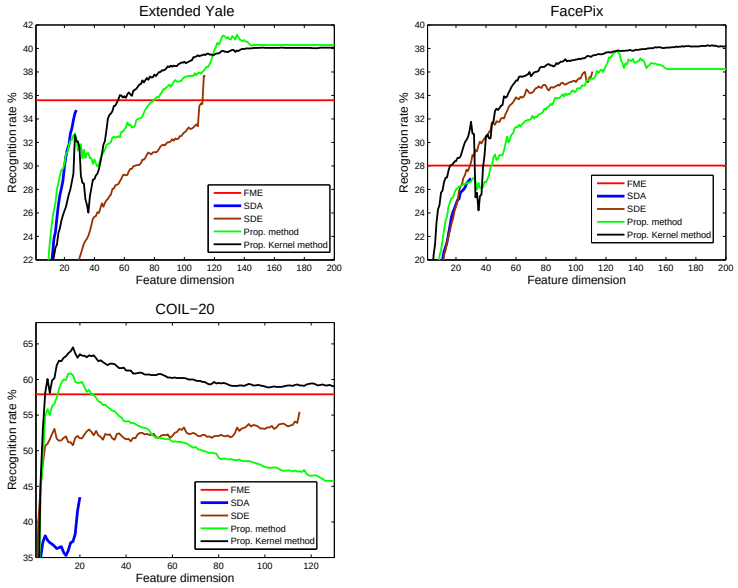

Fig. 2. Recognition accuracy variation as a function of dimensions for Extended Yale, FacePix, and COIL-20 datasets. These curves correspond to the best average curves. One labeled sample per class is used.

Table 2. Comparing the average performance of the FME method and the proposed IFSSFE method obtained at dimension equal to C. For the proposed IFSSFE (a generic semi-supervised embedding), we used four classifiers: 1-NN, RBF SVM, polynomial SVM (degree=3), and the Two Phase Test Sample Sparse Representation (TPTSSR) classifier.

Ext Yale	**1 labeled sample**		**2 labeled samples**		**3 labeled samples**	
Method	Unlabel(%)	Test(%)	Unlabel(%)	Test(%)	Unlabel(%)	Test(%)
FME	38.4	35.6	59.9	56.6	64.8	59.1
IFSSFE (1-NN)	38.5	36.1	56.2	53.5	65.4	60.8
IFSSFE (RBF SVM)	38.5	36.1	53.0	50.6	59.5	54.9
IFSSFE (Poly. SVM)	44.9	41.0	62.2	57.2	67.3	60.8
IFSSFE (TPTSSR)	49.3	44.7	67.2	62.5	71.6	65.9

FacePix	**1 labeled sample**		**2 labeled samples**		**3 labeled samples**	
Method	Unlabel(%)	Test(%)	Unlabel(%)	Test(%)	Unlabel(%)	Test(%)
FME	28.1	28.0	42.7	40.3	50.5	40.3
IFSSFE (1-NN)	31.6	32.1	47.9	45.9	56.8	53.1
IFSSFE (RBF SVM)	31.6	32.1	48.2	45.8	57.1	53.1
IFSSFE (Poly. SVM)	32.1	29.9	48.4	44.1	56.7	50.6
IFSSFE (TPTSSR)	31.9	30.8	50.9	47.3	61.3	55.7

COIL-20	**1 labeled sample**		**2 labeled samples**		**3 labeled samples**	
Method	Unlabel(%)	Test(%)	Unlabel(%)	Test(%)	Unlabel(%)	Test(%)
FME	62.03	57.9	66.8	64.7	70.7	68.6
IFSSFE (1-NN)	63.6	59.5	72.9	68.7	78.2	75.5
IFSSFE (RBF SVM)	60.2	55.7	68.7	67.7	76.9	76.3
IFSSFE (Poly. SVM)	63.7	60.8	72.6	71.9	79.2	77.9
IFSSFE (TPTSSR)	62.0	57.9	66.8	64.7	70.7	68.6

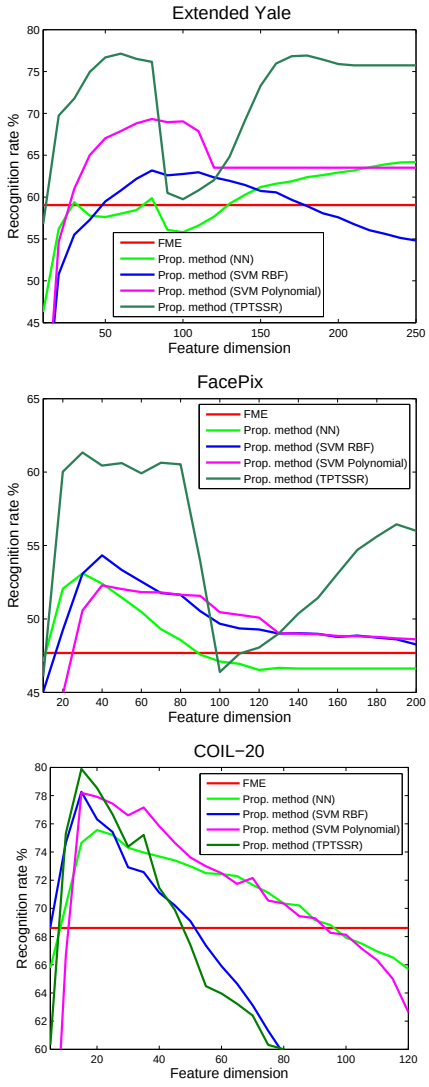

Fig. 3. A performance comparison between FME and the proposed IFSSFE as a function of features for three datasets: Extended Yale, FacePix, and COIL-20. The projection models associated with FME and IFSSFE were optimized on the fixed dimension given by C, namely the number of classes. In each plot, we show the average curve over ten splits. These curves depicts the performance on the test subset in which the number of labeled samples was set to three per class. For the proposed method IFSSFE, we used four classifiers:1-NN, RBF SVM, polynomial SVM (degree 3), and the Two Phase Test Sample Sparse Representation (TPTSSR) classifier.

References

1. Maaten, L., Postma, E., Herik, J.: Dimensionality reduction: A comparative review. Technical Report TiCC TR 2009005, TiCC, Tilburg University (2009)
2. Saul, L., Weinberger, K., Sha, F., Ham, J., Lee, D.: Spectral methods for dimensionality reduction. In: Semisupervised Learning. MIT Press, Cambridge, MA (2006)
3. Yan, S., Xu, D., Zhang, B., Zhang, H., Yang, Q., Lin, S.: Graph embedding and extension: a general framework for dimensionality reduction. IEEE Trans. on Pattern Analysis and Machine Intelligence **29** (2007) 40–51
4. Zhang, T., Tao, D., Li, X., Yang, J.: Patch alignment for dimensionality reduction. IEEE Trans. on Knowledge and Data Engineering **21** (2009) 1299–1313
5. Chapelle, O., Scholkopf, B., Zien, A.: Semi-Supervised Learning. MIT Press, Cambridge MA (2006)
6. Silva, T., Zhao, L.: Network-based stochastic semisupervised learning. IEEE Transactions on Neural Networks and Learning Systems **23** (2012) 451–466
7. Camps-Valls, G., Marsheva, T.B., Zhou, D.: Semi-supervised graph-based hyperspectral image classification. IEEE Trans. Geoscience and Remote Sensing **45** (2007) 3044–3054
8. Huang, H., Li, J., Liu, J.: Enhanced semi-supervised local fisher discriminant analysis for face recognition. Future Generation Computer Systems **28** (2012) 244–253
9. Liu, W., He, J., Chang, S.: Large graph construction for scalable semi-supervised learning. In: International Conference on Machine Learning. (2010)
10. Pan, F., Wang, J., Lin, X.: Local margin based semi-supervised discriminant embedding for visual recognition. Neurocomputing **74** (2011) 812–819
11. Yang, W., Zhang, S., Liang, W.: A graph based subspace semi-supervised learning framework for dimensionality reduction. In: International Conference on Computer Vision. (2008)
12. Z. Xu, I. King, M.R.T.L., Rong, J.: Discriminative semi-supervised feature selection via manifold regularization. IEEE Trans. on Neural Networks **21** (2010) 1033–1047
13. Zhang, T., Ji, R., Liu, W., Tao, D., Hua, G.: Semi-supervised learning with manifold fitted graphs. In: International Joint Conference on Artificial Intelligence. (2013)
14. Liu, W., Tao, D., Liu, J.: Transductive component analysis. In: IEEE International Conference on Data Mining. (2008)
15. Cevikalp, H.: Semi-supervised dimensionality reduction using pairwise equivalence constraints. In: International Conference on Computer Vision Theory and Applications. (2009)
16. Song, Y., Nie, F., Zhang, C., Xiang, S.: A unified framework for semi-supervised dimensionality reduction. Pattern Recognition **41** (2008) 2789–2799
17. Sousa, C., Rezende, S., Batista, G.: Influence of graph construction on semi-supervised learning. In: European Conferene on Machine Learning. (2013)
18. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using gaussian fields and harmonic functions. In: International Conference on Machine Learning. (2003)
19. Zhou, S., Chellappa, R., Mogghaddam, B.: Visual tracking and recognition using appearance-adaptive models in particle filters. IEEE Transactions on Image Processing **13** (2004) 1473–1490

20. Belkin, M., Niyogi, P., Sindhwani, V.: Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. Journal of Machine Learning Research **7** (2006) 2399–2434
21. Liu, W., Chang, S.: Robust multi-class transductive learning with graphs. In: Computer Vision and Pattern Recognition. (2009)
22. Nie, F., Xu, D., Tsang, I., Zhang, C.: Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction. IEEE Transactions on Image Processing **19** (2010) 1921–1932
23. Cai, D., He, X., Han, J.: Semi-supervised discriminant analysis. In: IEEE Int. Conf. Comput.Vision. (2007)
24. Chen, H., Chang, H., Liu, T.: Local discriminant embedding and its variants. In: IEEE International Conference on Computer Vision and Pattern Recognition. (2005)
25. Huang, H., Liu, J., Pan, Y.: Semi-supervised marginal fisher analysis for hyperspectral image classification. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences **I-3** (2012) 377–382
26. Yu, G., Zhang, G., Domeniconi, C., Yu, Z., J, Y.: Semi-supervised classification based on random subspace dimensionality reduction. Pattern Recognition **45** (2012) 1119–1135
27. Xu, Y., Zhang, D., Yang, J., Yang, J.Y.: A two-phase test sample sparse representation method for use with face recognition. IEEE Transactions on Circuits and Systems for Video Technology **21** (2011) 1255–1262