A Supervised Non-linear Dimensionality Reduction Approach for Manifold Learning

B. Raducanu[1] and F. Dornaika[2,3]

[1] *Computer Vision Center, Barcelona, SPAIN*
[2] *Department of Computer Science and Artificial Intelligence*
University of the Basque Country UPV/EHU, San Sebastian, SPAIN
[3] *IKERBASQUE, Basque Foundation for Science, Bilbao, SPAIN*

Abstract

In this paper we introduce a novel supervised manifold learning technique called Supervised Laplacian Eigenmaps (S-LE), which makes use of class label information to guide the procedure of non-linear dimensionality reduction by adopting the large margin concept. The graph Laplacian is split in two components: within-class graph and between-class graph to better characterize the discriminant property of the data. Our approach has two important characteristics: (i) it adaptively estimates the local neighborhood surrounding each sample based on data density and similarity, and (ii) the objective function simultaneously maximizes the local margin between heterogenous samples and pushes the homogeneous samples closer to each other.

Our approach has been tested on several challenging face databases and it has been conveniently compared with other linear and non-linear techniques, demonstrating its superiority. Although we have concentrated in this paper on the face recognition problem, the proposed approach could also be applied to other category of objects characterized by large variations in their appearance (such as hand or body pose, for instance).

Preprint submitted to Pattern Recognition *November 28, 2011*

Keywords: Supervised manifold learning, non-linear dimensionality reduction, discriminant analysis, face recognition

1. Introduction

In recent years, a new family of non-linear dimensionality reduction techniques for manifold learning has emerged. The most known ones are: Kernel Principal Component Analysis (KPCA) [1], Locally Linear Embedding (LLE) [2, 3], Isomap [4], Supervised Isomap [5], Laplacian Eigenmaps (LE)[6, 7]. This family of non-linear embedding techniques appeared as an alternative to their linear counterparts which suffer of severe limitation when dealing with real-world data: i) they assume the data lie in an Euclidean space and ii) they may fail when the number of samples is too small. On the other hand, the non-linear dimensionality techniques are able to discover the intrinsic data structure by exploiting the local topology. In general, they attempt to optimally preserve the local geometry around each data sample while using the rest of the samples to preserve the global structure of the data.

The main contribution of our work is represented by a Supervised LE (S-LE) algorithm, which exploits the class label information for mapping the original data in the embedded space. The use of labels allows us to split graph Laplacian associated to the data in two components: within-class graph and between-class graph. Our proposed approach benefits from two important properties: (i) it adaptively estimates the local neighborhood surrounding each sample based on data density and similarity, and (ii) the objective function simultaneously maximizes the local margin between heterogenous samples and pushes the homogeneous samples closer to each other.

The combination between locality preserving property (inherited from the classical LE[1]) and the discriminative property (due to the large margin concept) represents a clear advantage for S-LE, compared with other non-linear embedding techniques, because it finds a mapping which maximizes the distances between data samples from different classes at each local area. In other words, it maps the points in an embedded space where data with similar labels fall close to each other and where the data from different classes fall far apart.

The adaptive selection of neighbors for the two graphs represents also an added value to our algorithm. It is well known that a sensitive matter affecting non-linear embedding techniques is represented by the proper choice for neighborhood size. Setting a too high value for this parameter would result in a loss of local information, meanwhile a too low value could result in an over-fragmentation of the manifold (problem known as 'short-circuiting). For this reason, setting an adequate value for this parameter is crucial in order to confer the approach topological stability.

The rest of the paper is organized as follows. Section 2 reviews some related work on linear and non-linear dimensionality reduction techniques. In this section, we also recall, for the sake of completeness, the classic Laplacian Eigenmaps algorithm. Section 3 is devoted to the presentation of our new proposed algorithm. Section 4 presents extensive experimental results obtained on a man-made object data set and on six face databases. Finally, section 5 contains our conclusions and guidelines for future work.

[1] By classical Laplacian Eigenmaps we refer to the algorithm introduced in [6].

2. Related work

During the last few years, a large number of approaches have been proposed for constructing and computing an embedded subspace by finding an explicit or non-explicit mapping that projects the original data to a new space of lower dimensionality [8, 9, 10]. These methods can be grouped in two families: linear and non-linear approaches.

2.1. Linear Approaches

The classical linear embedding methods (e.g., PCA, LDA, MDS, Maximum Margin Criterion (MMC)[11]) and Locally LDA [12] are demonstrated to be computationally efficient and suitable for practical applications, such as pattern classification and visual recognition. Recent proposed methods attempt to linearize some non-linear embedding techniques. This linearization is obtained by forcing the mapping to be explicit, i.e., performing the mapping by a projection matrix. For example, Locality Preserving Projection (LPP) [13, 14, 15] and Neighborhood Preserving Embedding (NPE) [16] can be seen as linearized versions of LE and LLE, respectively. The main advantage of the linearized embedding techniques is that the mapping is defined everywhere in the original space. However, since the embedding is approximated by a linear process, these methods ignore the geodesic structure of the true manifold. All these linear methods cannot reveal the perfect geometric structure of the non-linear manifold. Some researchers tried to remedy to the global nature of the linear methods PCA, LDA and LPP by proposing localized models [17]. In this work, localized PCA, LDA, or LPP models are built using the neighbors of a query sample. The authors have shown that the obtained localized linear models can outperform the global models for face recognition and coarse head pose problems. However, it is not clear how neighbors can be optimally selected. In [18], the authors have extended the LPP to the supervised case by adapting the entries of the similarity matrix according to the labels of the sample pair. In [19], the authors assessed the performance of the quotient and difference criteria used in LDA. They also proposed a unified criterion that combines Quotient-LDA and Difference-LDA criteria.

In addition to the above methods, some distance metric learning algorithms [20, 21] attempt to directly estimate an induced Mahalanobis distance over the samples. In essence, these methods provide a linear transform since the Euclidean distance in the embedded space is equal to the Mahalanobis distance in the original space. The proposed solutions for estimating the Mahalanobis matrix are not given in closed form but by iterative processes.

One interesting supervised linear method is given by [22]. This work proposed a linear discriminant method called Average Neighborhood Margin Maximization (ANMM). Since it estimates the linear transform, it can be solved in closed-form instead of the iterative methods . It associates to every sample a margin that is set to the difference between the average distance to heterogenous neighbors and the average distance to the homogeneous neighbors. The linear transform is then derived by maximizing the sum of the margins in the embedded space. A similar method based on similar and dissimilar samples was proposed in [23].

2.2. Non-Linear Approaches

The non-linear methods such as Locally Linear Embedding (LLE), Laplacian Eigenmaps, Isomap, Hessian LLE (hLLE) [24] focus on preserving the local structure of data. LLE formulates the manifold learning problem as a neighborhood-preserving embedding, which learns the global structure by exploiting the local symmetries of linear reconstructions. Isomap extends the classical Multidimensional Scaling (MDS) [25] by computing the pairwise distances in the geodesic space of the manifold. Essentially, Isomap attempts to preserve geodesic distances when data are embedded in the new low dimensional space. Based on the spectral decomposition of the graph Laplacian, Laplacian Eigenmaps actually try to find Laplacian eigenfunction on the manifold. Maximum Variance Unfolding (MVU) [26] is a global algorithm for nonlinear dimensionality reduction, in which all the data pairs, nearby and far, are considered. MVU attempts to 'unfold' a dataset by pulling the input patterns as far apart as possible subject to the constraints that distances and angles between neighboring points are strictly preserved.

The non-linear embedding methods have been successfully applied to some standard data sets and generated satisfying results in dimensionality reduction and manifold visualization. However, most of these approaches does not take into account the discriminant information that is usually available for many real world problems. Therefore, the application of these methods can be very satisfactory in terms of dimensionality reduction and visualization but can be fair for classification tasks. In [5], the authors propose a supervised version of Isomap. This version replaces pairwise Euclidean distances by a dissimilarity function that increases if the pair is heterogeneous and decreases otherwise. Since this algorithm is inherited from Isomap it suffers from the same disadvantage in the sense that outlier samples can give rise to an unwanted embedded space. In [27], the authors exploit label information to improve Laplacian Eigenmaps. The proposed method affects the computation of the affinity matrix entries in the sense that an homogeneous pair of neighbors will have large value and heterogeneous pairs of neighbors will have a small value. Although, the authors show some performance improvement, the proposed method has two drawbacks. First, there is no guarantee that the heterogenous samples will be pushed away from each other. Second, the method has at least three parameters to be tuned.

Some works extended the Locally Linear Embedding (LLE) technique to the supervised case [28, 29]. These extensions were made in the first stage of the LLE algorithm, i.e., the neighborhood graph construction. In these works, the K nearest neighbors of a given sample are looked for among the samples belonging to the same class.

As can be seen, all existing supervised non-linear dimensionality reduction techniques have used the labels either for adjusting the entries of a similarity matrix or for modifying the neighborhood graph. However, our proposed method exploits the large margin concept in order to increase the discrimination between homogenous and heterogenous samples.

2.2.1. Review of Laplacian Eigenmaps

To make the paper self-contained, this section will briefly present the Laplacian Eigenmaps (LE). Throughout the paper, capital bold letters denote matrices and small bold letters denote vectors.

Laplacian Eigenmaps is a recent non-linear dimensionality reduction techniques that aims to preserve the local structure of data [6]. Using the notion of the Laplacian of the graph, this non-supervised algorithm computes a low-dimensional representation of the data set by optimally preserving local neighborhood information in a certain sense. We assume that we have a set of N samples $\{\mathbf{y}_i\}_{i=1}^{N} \subset \mathbb{R}^D$. Define a neighborhood graph on these samples, such as a K-nearest-neighbor or ϵ-ball graph, or a full mesh, and weigh each edge $\mathbf{y}_i \sim \mathbf{y}_j$ by a symmetric affinity function $W_{ij} = K(\mathbf{y}_i; \mathbf{y}_j)$, typically Gaussian:

$$W_{ij} = \exp\left(-\frac{\|\mathbf{y}_i - \mathbf{y}_j\|^2}{\beta}\right) \tag{1}$$

where β is usually set to the average of squared distances between all pairs.

We seek latent points $\{\mathbf{x}_i\}_{i=1}^{N} \subset \mathbb{R}^L$ that minimize $\frac{1}{2}\sum_{i,j} \|\mathbf{x}_i - \mathbf{x}_j\|^2 W_{ij}$, which discourages placing far apart latent points that correspond to similar observed points. If $\mathbf{W} \equiv W_{ij}$ denotes the symmetric affinity matrix and $\mathbf{D}$ is the diagonal weight matrix, whose entries are column (or row, since $\mathbf{W}$ is symmetric) sums of $\mathbf{W}$, then the Laplacian matrix is given $\mathbf{L} = \mathbf{D} - \mathbf{W}$. It can be shown that the objective function can also be written as (A similar derivation is given in section 3.3):

$$\frac{1}{2}\sum_{i,j} \|\mathbf{x}_i - \mathbf{x}_j\|^2 W_{ij} = tr(\mathbf{Z}^T \mathbf{L} \mathbf{Z}) \tag{2}$$

where $\mathbf{Z} = [\mathbf{x}_1^T; \ldots; \mathbf{x}_N^T]$ is the $N \times L$ embedding matrix and $tr(.)$ denotes the trace of a matrix. The i^{th} row of the matrix $\mathbf{Z}$ provides the vector $\mathbf{x}_i$—the embedding coordinates of the sample $\mathbf{y}_i$.

The embedding matrix $\mathbf{Z}$ is the solution of the optimization problem:

$$\min_{\mathbf{Z}} tr(\mathbf{Z}^T \mathbf{L} \mathbf{Z}) \;\; s.t. \;\; \mathbf{Z}^T \mathbf{D} \mathbf{Z} = \mathbf{I}, \;\; \mathbf{Z}^T \mathbf{L} \mathbf{e} = \mathbf{0} \tag{3}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{e} = (1, \ldots, 1)^T$. The first constraint eliminates the trivial solution $\mathbf{Z} = \mathbf{0}$ (by setting an arbitrary scale) and the second constraint eliminates the trivial solution $\mathbf{e}$ (all samples are mapped to the same point). Standard methods show that the embedding matrix is provided by the matrix of eigenvectors corresponding to the smallest eigenvalues of the generalized eigenvector problem,

$$\mathbf{L}\,\mathbf{z} = \lambda\,\mathbf{D}\,\mathbf{z} \tag{4}$$

Let the column vectors $\mathbf{z}_0, \ldots, \mathbf{z}_{N-1}$ be the solutions of (4), ordered according to their eigenvalues, $\lambda_0 = 0 \leq \lambda_1 \leq \ldots \leq \lambda_{N-1}$. The eigenvector corresponding to eigenvalue 0 is left out and only the next eigenvectors for embedding are used. The embedding of the original samples is given by the row vectors of the matrix $\mathbf{Z}$, that is,

$$\mathbf{y}_i \longrightarrow \mathbf{x}_i = (z_1(i), \ldots, z_L(i))^T \tag{5}$$

where $L < N$ is the dimension of the new space. From equation (4), we can observe that the dimensionality of the subspace obtained by LE is limited by the number of samples N.

3. Supervised Laplacian Eigenmaps

While the LE may give good results for non-linear dimensionality reduction, it has not been widely used and assessed for classification tasks. Indeed, many experiments show that the recognition rate in the embedded space can be highly depending on the choice of the neighborhood size in the reconstructed graph [15, 30, 31]. Choosing the ideal size, K or ϵ, in advance can be a very difficult task. Moreover, the introduced mapping by LE does not exploit the discriminant information given by the labels of data. In this section, we present our Supervised LE algorithm which has two interesting properties: (i) it adaptively estimates the local neighborhood surrounding each sample based on data density and similarity, and (ii) the objective function simultaneously maximizes the local margin between heterogenous samples and pushes the homogeneous samples closer to each other. The main contributions of our work are as follows: (1) bypassing the use and selection of a predefined neighborhood for graph construction, and (2) exploiting the discriminant information in order to get the non-linear embedding (spectral projection).

3.1. Two graphs and adaptive neighborhood

In order to discover both geometrical and discriminant structure of the data manifold, we split the global graph in two components: the within-class graph G_w and between-class graph G_b. Let $l(\mathbf{y}_i)$ be the class label of $\mathbf{y}_i$. For each data point $\mathbf{y}_i$, we compute two subsets, $N_b(\mathbf{y}_i)$ and $N_w(\mathbf{y}_i)$. $N_w(\mathbf{y}_i)$ contains the neighbors sharing the same label with $\mathbf{y}_i$, while $N_b(\mathbf{y}_i)$ contains the neighbors having different labels. We stress the fact that unlike the classical LE, our algorithm adapts the size of both sets according to the local sample point $\mathbf{y}_i$ and its similarities with the rest of samples. To this end, each set is defined for each sample point $\mathbf{y}_i$ and is computed in two consecutive steps. First, the average similarity of the sample $\mathbf{y}_i$ is computed by the total of all similarities with the rest of the data set (Eq. (6)). Second, the sets $N_w(\mathbf{y}_i)$ and $N_b(\mathbf{y}_i)$ are computed using Eqs. (7) and (8), respectively.

$$AS(\mathbf{y}_i) = \frac{1}{N} \sum_{k=1}^{N} \exp\left(- \frac{\|\mathbf{y}_i - \mathbf{y}_k\|^2}{\beta} \right) \tag{6}$$

$$N_w(\mathbf{y}_i) = \{\mathbf{y}_j \mid l(\mathbf{y}_j) = l(\mathbf{y}_i), \exp\left(-\frac{\|\mathbf{y}_i - \mathbf{y}_j\|^2}{\beta}\right) > AS(\mathbf{y}_i)\} \tag{7}$$

$$N_b(\mathbf{y}_i) = \{\mathbf{y}_j \mid l(\mathbf{y}_j) \neq l(\mathbf{y}_i), \exp\left(-\frac{\|\mathbf{y}_i - \mathbf{y}_j\|^2}{\beta}\right) > AS(\mathbf{y}_i)\} \tag{8}$$

Equation (7) means that the set of within-class neighbors of the sample $\mathbf{y}_i$, $N_w(\mathbf{y}_i)$, is all data samples that have the same label of $\mathbf{y}_i$ and that have a similarity higher then the average similarity associated with $\mathbf{y}_i$. There is a similar interpretation for the set of between-class neighbors $N_b(\mathbf{y}_i)$. From Equations (7) and (8) it is clear that the neighborhood size is not the same for every data sample. This mechanism adapts the set of neighbors according to the local density and similarity between data samples in the original space. It is worth noting that for real data sets the mean similarity is always a positive value.

It is obvious that Eqs. (6), (7) and (8) computes a sample-based neighborhood for every sample. A simple geometrical interpretation of these equations is illustrated in Figure 1. In Figure 1.(a), the average similarity of the sample P_1 is relatively low. Thus, according to Eqs. (7) and (8), the neighborhood of the sample P_1 will be relatively large, i.e., the sample P_1 will have many neighbors (both homogeneous and heterogeneous). In Figure 1.(b), the average similarity of the sample P_1 is relatively high so its neighborhood will be small. From the above equations, one can conclude that an isolated sample will have a small mean similarity which increases the chance that other samples will consider it as a graph neighbor. This is less likely to happen with a fixed neighborhood size where the edges of the graph will be kept within the dense regions. Since the concepts of similarity and closeness of samples are tightly related, one can conclude, at first glance, that our introduced strategy for adaptive estimation of neighbors is equivalent to the use of an ε-ball neighborhood. It is worth noting that there are two main differences: (1) the use of an ε-ball neighborhood requires a user-defined value for the ball radius ε, and (2) the ball radius is constant for all data samples, whereas our strategy uses an adaptive threshold (Eq. (6)) that depends on the local sample. Thus, our strategy adapts the graph construction to all databases without parameter tuning.

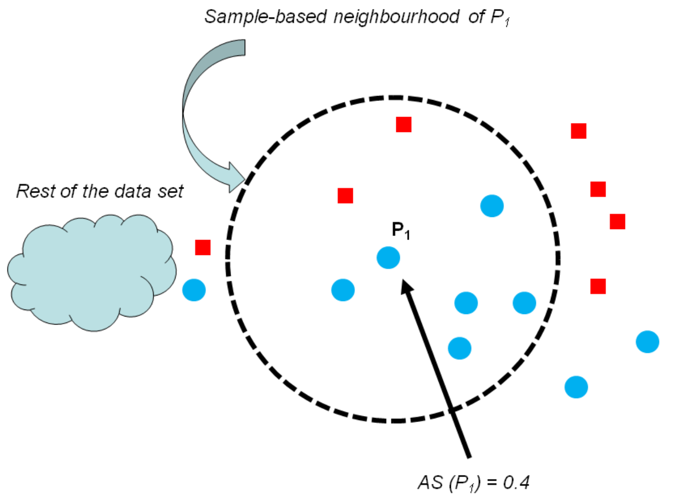
3.2. Two affinity matrices

Let $\mathbf{W}_w$ and $\mathbf{W}_b$ be the weight matrices of G_w and G_b, respectively. These matrices are defined as:

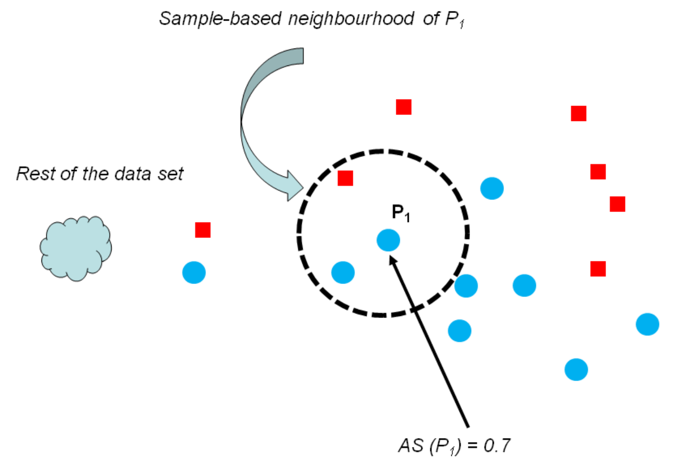
$$
\begin{aligned}
W_{w,ij} &= \begin{cases} \exp\left(-\frac{\|\mathbf{y}_i-\mathbf{y}_j\|^2}{\beta}\right) & if\ \mathbf{y}_j \in N_w(\mathbf{y}_i)\ or\ \mathbf{y}_i \in N_w(\mathbf{y}_j) \\ 0, otherwise & \end{cases} \\
W_{b,ij} &= \begin{cases} 1 & if\ \mathbf{y}_j \in N_b(\mathbf{y}_i)\ or\ \mathbf{y}_i \in N_b(\mathbf{y}_j) \\ 0, otherwise & \end{cases}
\end{aligned}
$$

If the same type of weighting is used (0-1 weighting or Gaussian kernel weighting), then it is easy to show that the global affinity matrix, $\mathbf{W}$, associated with the Laplacian Eigenmaps graph can be written as:

$$\mathbf{W} = \mathbf{W}_w + \mathbf{W}_b$$



(a)



(b)

Figure 1: Sample-based neighbor computation exploits a sample-based measure which relates the sample to the whole set of data. **(a)** The average similarity of P_1 is relatively low so the neighborhood of P_1 will be extended. (b) The average similarity of P_1 is relatively high so its neighborhood will be somehow small.

Figure 2 illustrates the principle of the proposed S-LE. It also shows how the graph weight matrices are computed using the sample-based neighborhood.

3.3. Optimal mapping

Each data sample $\mathbf{y}_i$ is mapped into a vector $\mathbf{x}_i$. The aim is to compute the embedded coordinates $\mathbf{x}_i$ for each data sample. The objective functions are:

$$\min \frac{1}{2} \sum_{i,j} \|\mathbf{x}_i - \mathbf{x}_j\|^2 \, W_{w,ij} \tag{9}$$

$$\max \frac{1}{2} \sum_{i,j} \|\mathbf{x}_i - \mathbf{x}_j\|^2 \, W_{b,ij} \tag{10}$$

Let the matrix $\mathbf{Z}$ denotes $[\mathbf{x}_1^T; \ldots; \mathbf{x}_N^T]$, it can be shown that the above objective functions can be written as:

$$\min tr(\mathbf{Z}^T \, \mathbf{L}_w \, \mathbf{Z}) \tag{11}$$

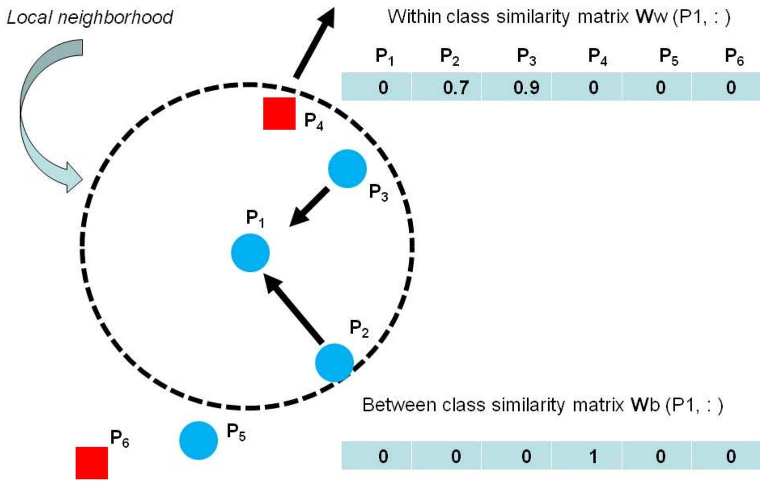
$$\max tr(\mathbf{Z}^T \, \mathbf{L}_b \, \mathbf{Z}) \tag{12}$$

where $\mathbf{L}_w = \mathbf{D}_w - \mathbf{W}_w$ and $\mathbf{L}_b = \mathbf{D}_b - \mathbf{W}_b$.

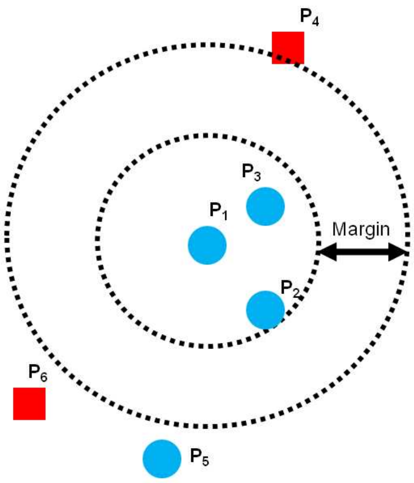
Using the scale constraint $\mathbf{Z}^T \, \mathbf{D}_w \, \mathbf{Z} = \mathbf{I}$ and the equation $\mathbf{L}_w = \mathbf{D}_w - \mathbf{W}_w$, the above two objective functions can be combined into one single objective function:

$$arg \max_{\mathbf{Z}} \{\gamma \, tr(\mathbf{Z}^T \, \mathbf{L}_b \, \mathbf{Z}) + (1-\gamma) \, tr(\mathbf{Z}^T \, \mathbf{W}_w \, \mathbf{Z})\} \quad s.t. \;\; \mathbf{Z}^T \, \mathbf{D}_w \, \mathbf{Z} = \mathbf{I}$$

By using the matrix $\mathbf{B} = \gamma \, \mathbf{L}_b + (1-\gamma) \, \mathbf{W}_w$, the problem becomes:



(a)



(b)

Figure 2: **(a) Before embedding:** in the original space, the sample P_1 has three neighbors. The samples with the same color and shape belong to the same class. The first row of the within class similarity matrix $\mathbf{W}_w$ and of the between class similarity matrix $\mathbf{W}_b$. **(b) After embedding:** the non-linear embedding obtained by the proposed S-LE.

$$arg \max_{\mathbf{Z}} tr(\mathbf{Z}^T\,\mathbf{B}\,\mathbf{Z}) \quad s.t. \;\; \mathbf{Z}^T\,\mathbf{D}_w\,\mathbf{Z} = \mathbf{I}$$

This gives a generalized eigenvalue problem having the following form:

$$\mathbf{B}\,\mathbf{z} = \lambda\,\mathbf{D}_w\,\mathbf{z} \tag{13}$$

Let the column vectors $\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_L$ be the generalized eigenvectors of (13) according to their eigenvalue: $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_L$. Then, the $N \times L$ embedding matrix $\mathbf{Z} = [\mathbf{x}_1^T; \ldots; \mathbf{x}_N^T]$ will be given by concatenating the obtained eigenvectors $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_L]$.

It is worth noting that the outlier labels have not a significant impact on the learned embedding since the method relies on penalizing local geometric distances in the mapped space. Indeed, Eq. (6) does not depend on the labels and thus it is invariant to the outlier labels. Regarding Eqs. (7) and (8) and assuming that a mislabeled sample is within the selected neighborhood, then the mechanism of pushing and pulling will be reversed for only this sample, while the rest of all samples will contribute to the correct desired behavior.

3.4. Theoretical analysis of the proposed approach

3.4.1. Difference between the classic LE and the proposed S-LE

The main difference between the classic LE and our proposed S-LE are as follows:

- The classic LE has one single objective, namely preserving the locality of samples. The proposed S-LE aims at a discriminant analysis via a non-linear embedding. The classic LE does not exploit the sample labels. This means that the only objective of the classic LE is to preserve locality of samples. On the other hand, our proposed S-LE exploits the labels of samples by collapsing homogeneous neighbors, and pulling away heterogeneous neighbors. In other words, the constructed graph is split in into two graphs: within class graph and between class graph.

- The classic LE works on a single graph that is built using an artificial graph. The most popular graph construction manner is based on the K nearest neighbor and ϵ-ball neighborhood criteria. K and ϵ are user-defined parameters that should be fixed in advance. It is well known that the choice of these parameters can affect the performance of the embedding. On the other hand, our graph does not need a user-defined parameter. Instead, the graph edges are set according to adaptive neighborhood that is only sample-based. Therefore, our proposed strategy for graph construction can automatically adapt the graph to all databases without parameter adjustment. Thus, looking for the best value for either K or ϵ is bypassed by the proposed approach.

3.4.2. Relation to Normalized Graph-cut formulation

There is some similarity between the use of the Laplacian formulations and normalized cut formulations [32, 33]. However, we stress the fact that there are two main differences between our proposed formulation and the normalized cut formulation. Indeed, the objectives and the Laplacian computation are different. (i) Our introduced method is fully supervised and addresses classification tasks. On the other hand, the normalized cut formulations address the clustering problems from unlabeled samples–it can be unsupervised or semi-supervised technique. (ii) Our proposed method computes two Laplacian matrices based on local similarity and labels, whereas in the Normalized cut formulation, the computation of the Laplacian matrix relies on the concept of similarity among unlabeled samples.

4. Experimental results

In this section, we report the experimental results obtained from the application of our proposed algorithm to the problem of visual pattern recognition. Extensive experiments in terms of classification accuracy have been carried out on a man-made object database as well as on some public face databases. All these databases are characterized by a large variation in object appearance.

4.1. COIL-20 database

The COIL-20[2] database (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. Some instances are shown in Figure 3.

We compared the proposed algorithm (S-LE) against the following ones: PCA, LDA, ANMM, KPCA, Isomap and the classical LE. After projecting the data on the embedded space, we split the data in several train/test sets. The following ratios have been used: 30%-70%, 50%-50% and 70%-30%. The test samples were classified based on Nearest Neighbor. Table 1 illustrates the best recognition rates obtained with every algorithm and for every training

[2] $http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php$

Figure 3: The 20 objects of COIL-20 data set.

percentage. It can be appreciated from table 1 that when the size of training set is relatively small, S-LE outperforms all the other methods.

Figure 4: Some samples in UMIST data set.

4.2. Face data sets

In this study, six public face data sets are considered. The details of these data sets are described in Table 2.

1. The ORL face data set[3]. There are 10 images for each of the 40 human

[3] *http://www.cl.cam.ac.uk/research/dtg/attarchive/*

	COIL-20 (30/70)	COIL-20 (50/50)	COIL-20 (70/30)
PCA	97.12%	99.30%	**100.00%**
LDA	91.27%	93.88%	93.75%
ANMM	90.1%	94.3%	96.4%
KPCA	95.78%	98.52%	99.56%
Isomap	88.80%	91.86%	93.21%
LE	90.05%	95.18%	97.73%
S-LE	**99.46%**	**99.75%**	99.72%

Table 1: Best recognition accuracy obtained with COIL20 data set. The classification accuracy when the training/test percentage was set to 30%-70%, 50%-50% and 70%-30%, respectively.

Figure 5: Some samples in Extended Yale data set.

Figure 6: Some samples in PF01 data set.

Figure 7: Some samples in PIE data set.

subjects, which were taken at different times, varying the lighting, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses). The images were taken with a tolerance for some tilting and rotation of the face up to 20^o.

2. The UMIST face data set[4]. The UMIST data set contains 575 gray images of 20 different people. The images depict variations in head pose. Figure 4 shows some face samples in the UMIST face Database.
3. The Yale face data set[5]. It contains 11 grayscale images for each of the 15 individuals. The images demonstrate variations in lighting condition (left-light, center-light, right-light), facial expression (normal, happy, sad, sleepy, surprised, and wink), and with/without glasses.
4. The extended Yale Face Database B[6]. It contains 16128 images of 28 human subjects under 9 poses and 64 illumination conditions. In our

$facedatabase.html$

[4] $http://www.shef.ac.uk/eee/research/vie/research/face.html$

[5] $http://see.xidian.edu.cn/vipsl/database\_Face.html$

[6] $http://vision.ucsd.edu/\sim leekc/ExtYaleDatabase/ExtYaleB.html$

study, a subset of 1800 images has been used. Figure 5 shows some face samples in the extended Yale Face Database B.

5. The PF01 face data set[7]. It contains the true-color face images of 103 people, 53 men and 50 women, representing 17 various images (1 normal face, 4 illumination variations, 8 pose variations, 4 expression variations) per person. All of the people in the database are Asians. There are three kinds of systematic variations, such as illumination, pose, and expression variations in the database. Some samples are shown in Figure 6.
6. The PIE face data set[8] contains 41,368 images of 68 people. Each person is imaged under 13 different poses, 43 different illumination conditions, and with 4 different expressions. In our study, we used a subset of the original data set, considering 29 images per person. Some samples are shown in Figure 7.

4.3. Data preparation

Figure 8 illustrates the main steps of the application of S-LE to the problem of face recognition and object recognition. The initial face data set is projected on the embedded face subspace using the S-LE algorithm, whose steps have been summarized by a 4-block diagram (according to section 4). A face image is recognized using the nearest neighbor (NN) classifier applied in this low dimensional space.

[7] *http://nova.postech.ac.kr/special/imdb/imdb.html*
[8] *http://www.ri.cmu.edu/projects/project_418.html*

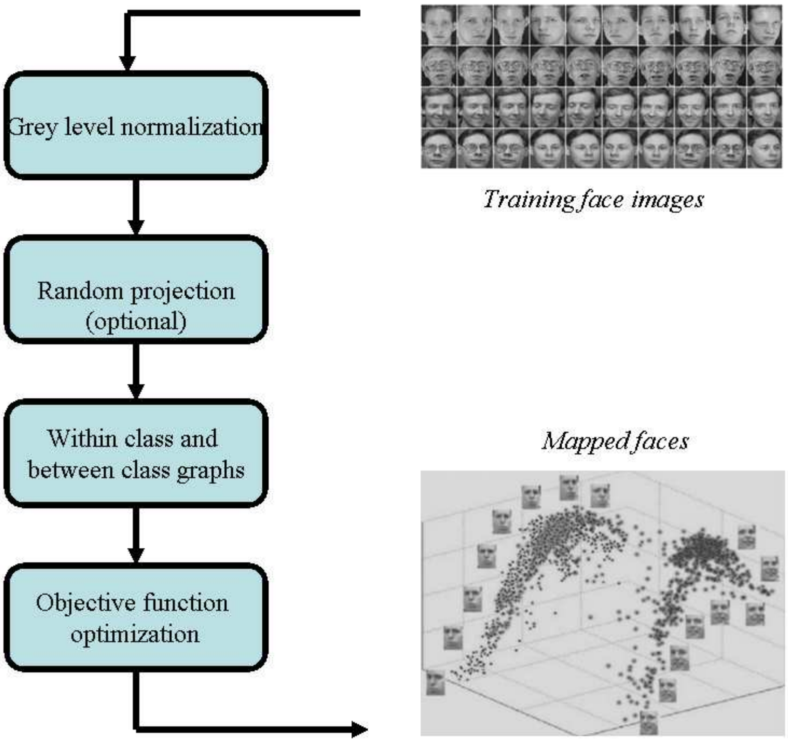


Figure 8: Supervised Laplacian Eigenmaps embedding for the face recognition problem.

Data set	# samples	# dimension	# classes
ORL	400	2576	40
UMIST	575	2576	20
YALE	165	2304	15
Extended Yale	1800	2016	28
PF01	1819	2304	107
PIE	1926	1024	68

Table 2: Details of benchmark face data sets.

To make the computation of the embedding more efficient, the dimensionality of the original data is reduced by applying random projections [34]. The main goal of random projections is to reduce the dimensionality of the original face data samples. It has a similar role to that of PCA yet with the obvious advantage that random projections do not need any training data.

The parameters of the proposed algorithm are: i) the heat Kernel parameter β, and ii) the parameter γ that balances the impact of within class and between class graphs. In our experiments, the heat Kernel parameter β is set to the average of squared distances between all pairs. This scheme was also used by the classical Laplacian Eigenmaps in our experiments. The parameter γ is estimated by cross-validation that is carried out on a part of the data set. Figure 9 illustrates the obtained average recognition rate as a function of the parameter γ when a part of the ORL data set is used as a validation set. Similar behaviors were obtained with other face data sets.

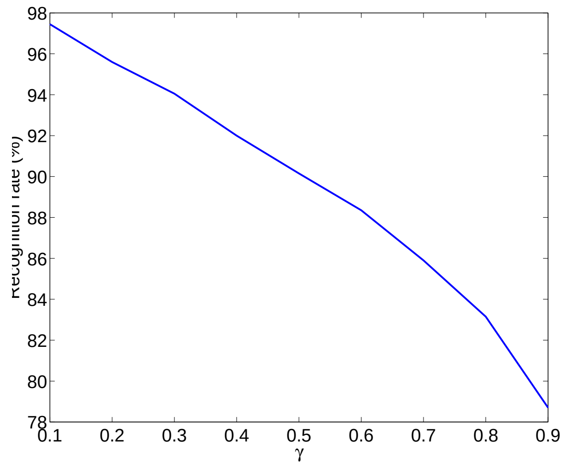
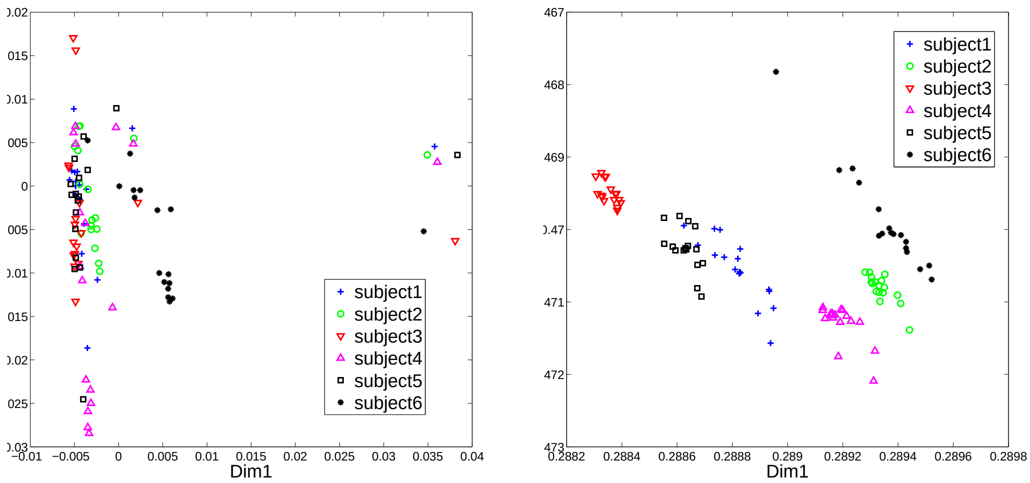


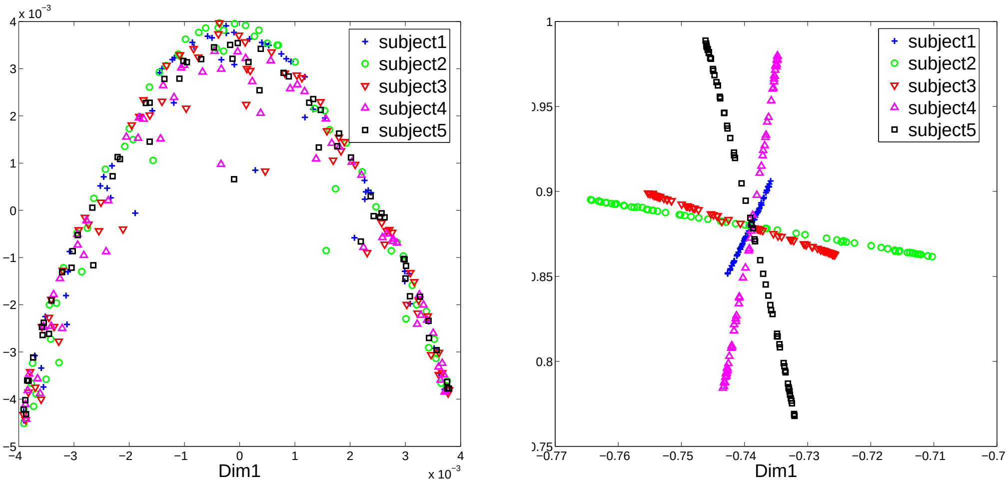
Figure 9: Average recognition rate as function of the blending parameter γ using a part of the ORL data set.



(a) Classical LE

(b) Proposed S-LE

Figure 10: Embedded faces of six persons of PF01 face data set.



(a) Classical LE

(b) Proposed S-LE

Figure 11: Embedded faces of five persons of Extended Yale face data set.

4.4. Visualization of the embedding process

Before presenting the quantitative evaluation of classification, it would be worthy to visualize the obtained embedded face data. To this end, we visualize some embedded samples using two methods: the classical LE and the proposed Supervised LE. Figure 10.(a) visualizes the embedding of faces associated with six persons of the PF01 data set obtained with the classical LE. In this plot, only the first two dimensions were used. Figure 10.(b) visualizes the embedding of the same six persons obtained with the proposed S-LE.

Figure 11.(a) visualizes the embedding of faces associated with five persons of the Extended Yale data set obtained with the classical LE. In this plot, only the first two dimensions were used. As can be seen, the intra and extra person variabilities are best presented in the embedded space obtained

with the proposed S-LE.

From a quantitative point of view, the differences between classical LE and S-LE could be estimated by adopting the Fisher criterion [35]. Fisher criterion is a measure used to quantify the degree of separation between classes. From the many possibilities to define the Fisher criterion, we chose the following one:

$$FC = tr(\mathbf{S}_w^{-1}\,\mathbf{S}_b) \tag{14}$$

where $\mathbf{S}_w$ and $\mathbf{S}_b$ represent the within- and between-class scatter matrices (as defined in the Linear Discriminant Analysis), computed from the samples projected on the embedded subspaces. Table 3 shows Fisher criterion obtained with the classical LE and S-LE for some face data sets when the dimension of the embedded space is set to 10. A higher value corresponds to a better separation between classes.

	UMIST	Extended Yale	PF01	PIE
LE	43.00	7.56	111.34	42.73
S-LE	101.5	25.68	140.66	116.92

Table 3: Fisher criterion computed in the embedded space for four face data sets when the dimension of the embedded space is fixed to 10.

4.5. Evaluation methodology

We have compared our method with six different methods, namely: PCA, LDA, ANMM, Kernel PCA (KPCA), Isomap, and LE. For methods relying on neighborhood graphs (Isomap, ANMM, and LE), five trials have been performed in order to choose the optimal neighborhood size. The final values correspond to those giving the best recognition rate in test sets. The implementation of the methods PCA, LDA, KPCA, and Isomap were retrieved from $http://www.zjucadcg.cn/dengcai/Data/data.html$. We have implemented the ANMM method and the classic LE.

For each face data set and for every method, we conducted three groups of experiments for which the percentage of training samples was set to 30%, 50% and 70% of the whole data set. The remaining data was used for testing. The partition of the data set was done randomly.

For a given embedding method, the recognition rate was computed for several dimensions belonging to $[1, L_{max}]$. For most of the tested methods L_{max} is equal to the number of samples used except for LDA and ANMM. For LDA, the maximum dimension is equal to the number of classes minus one. For ANMM the maximum dimension is variable since it is equal to the number of positive eigenvalues.[9]

Figures 12, 13, 14, 15, and 16 illustrates the average recognition rate associated with ORL, Yale, Extended Yale, PF01, and PIE data sets, respectively. The average recognition rate was computed (over ten folds) by PCA, KPCA, Isomap, ANMM, LE, and S-LE. The training/test percentage was set to 50%-50% for YALE and PF01 data sets, and to 30%-70% for ORL, Extended Yale, and PIE data sets. Since the maximum dimension for LDA is equal to the number of classes minus one, the corresponding curve was not plotted. Its rate was reported in Tables 4, 5 and 6. The maximum dimension depicted in

[9]This dimension is bounded by the the dimension of the input samples.

the plots was set to a fraction of L_{max}, in order to guarantee meaningful results. Moreover, we can observe that after a given dimension the recognition rate associated with the three methods PCA, KPCA, and Isomap becomes stable. However, the recognition rate associated with LE and S-LE methods decreases if the number of used eigenvectors becomes large—a general trend associated with many non-linear methods. This means that the last eigenvectors do not have any discriminant information, lacking completely of statistical significance.

The best (average) performance obtained by the embedding algorithms, based on 10 random splits, are shown in Tables 4, 5 and 6, respectively. Table 4 summarizes the results obtained with the first group of experiments (i.e., training/test percentage was set to 30%-70%). Table 5 summarizes the results obtained with the second group of experiments (i.e., training/test percentage was set to 50%-50%). And finally, Table 6 summarizes the results obtained with the third group of experiments (i.e., training/test percentage was set to 70%-30%). In [22], it is shown that the ANMM technique performs equally to or better than the following linear methods: Maximum Margin Criterion (MMC) [11], Marginal Fisher Analysis (MFA) [36], and Step non-parametric maximum margin criterion (SNMMC) [37]. Thus, the comparisons shown in these tables implicitly include these methods.

The number appearing in parenthesis corresponds to the optimal dimensionality of the embedded subspace (at which the maximum average recognition rate has been reported). We can observe that: i) the S-LE outperforms all other methods on all six face data sets, and ii) the difference in performance between methods is depending on the particularity of the data set used: for instance, in the case of UMIST data set the improvement is small; however, with PF01 data set, this improvement becomes very significant. This is due to the fact that in the UMIST data set we have a high number of samples per class, meanwhile in the case of PF01 data set we have only a fraction of it. Furthermore, the intra-class variation in the case of PF01 data set (due to light variation and changes in facial expression) is much higher than in UMIST's case.

Table 7 illustrates the average performance of all methods averaged over the six face data sets for three groups of experiments.

We have studied the performance of the classic LE when an adaptive neighborhood is used for constructing its graph. Table 8 illustrates a comparison between the classic LE adopting an adaptive neighborhood and the proposed S-LE. The training/test percentage was set to 70%-30%. The first row depicts the results obtained with the classic LE adopting an adaptive neighborhood. The second row depicts the results obtained with the proposed S-LE. As can be seen, for most of face data sets, the performance of LE with adaptive neighborhood become worse than that of LE with a predefined neighborhood (sixth row of Table 6). Possible explanations for this are the following facts: (i) the adaptive threshold, used for computing the LE local neighborhood, was set to simple statistics—the global mean of all similarities between the sample in question and the rest of samples, and (ii) only one kind of similarity functions was used, i.e., the Kernel heat function. In the future, for both LE and S-LE, we envision the use of more sophisticated statistics for computing the adaptive threshold. In addition, we envision the use of other types of similarity functions such as the polynomial and sigmoid functions.

We have also studied the performance of the proposed S-LE when other types of classifiers are used in the mapped space. Table 9 illustrates such a comparison when the training/test percentage was set to 30%-70%, 50%-50%, and 70%-30%. For each percentage, the first row depicts the results obtained with the NN classifier. The second row depicts the results obtained with the Support Vector Machines (SVM) classifier with Radial Basis Function. We can observe that for a small training percentage the NN classifier gave better results than the SVM classifier. However, when the training percentage is relatively high the SVM classifier gave better results.

In conclusion, the advantage of classification based on non-linear dimensionality techniques is that only a relative small number of dimensions are required, compared with their linear counterparts (as it can be appreciate from the tables). This is a very important result especially for the case when the data lie in a very high dimensionality space (like hyperspectral images, for instance) because it allows a powerful compression of the data without any relevant loss of intrinsic information. Furthermore, they achieve very good results even with a small number of training samples.

4.6. Algorithm complexity

Regarding the complexity of the algorithm, there are two main processes for obtaining the non-linear embedding. (1) Building the two graphs: the within class graph and the between class graph. (2) Solving a generalized eigenvalue decomposition which a complexity of $\mathcal{O}(N^3)$ where N denotes the number of samples. Table 10 illustrates the learning CPU time associated with several face data sets. The time[10] (in seconds) corresponds to the projection of the original data on the embedded space (learning phase). We performed the experiments using a non-optimized MATLAB code running on a PC equipped with a dual-core Intel processor at 2 Ghz and 2 Gb of RAM memory.

5. Conclusions and Future Work

We proposed a novel supervised non-linear dimensionality reduction technique, namely Supervised Laplacian Eigenmap (S-LE). Our algorithm benefits from two important properties: (i) it adaptively estimates the local neighborhood surrounding each sample based on data density and similarity, and (ii) the objective function simultaneously maximizes the local margin between heterogenous samples and pushes the homogeneous samples closer to each other.

For validation purposes, we applied our method to the face recognition problem. The experimental results obtained on six face data sets show that our approach outperforms many recent linear and non-linear dimensionality reduction techniques. The proposed method is based on optimizing a certain local margin and is therefore intuitively related to the NN classifier that was used in the current study. Future work will be concentrated on three directions. First, we will investigate the generalization of the proposed

[10] Although the computation time for S-LE is much higher than that of the classic LE, the former method has a much higher recognition rate. Seen from the point of view of off-line learning, this is not a critical aspect, since the learning phase is performed only once.

method to other classifiers, such as Sparse Representation classifiers (SRC) [38]. Second, we will investigate a variant of S-LE that will be parameter free. And finally, we will try to find for a given classification task the best set of obtained eigenvectors using the feature selection paradigm.

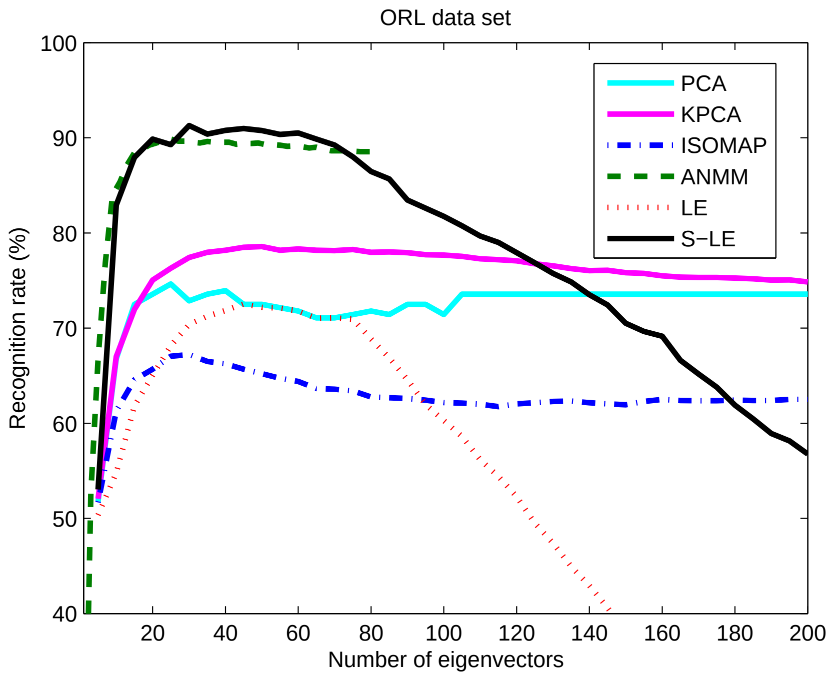


Figure 12: Average recognition rate as function of the number of eigenvectors obtained with ORL data set. The training/test percentage was set to 30%-70%

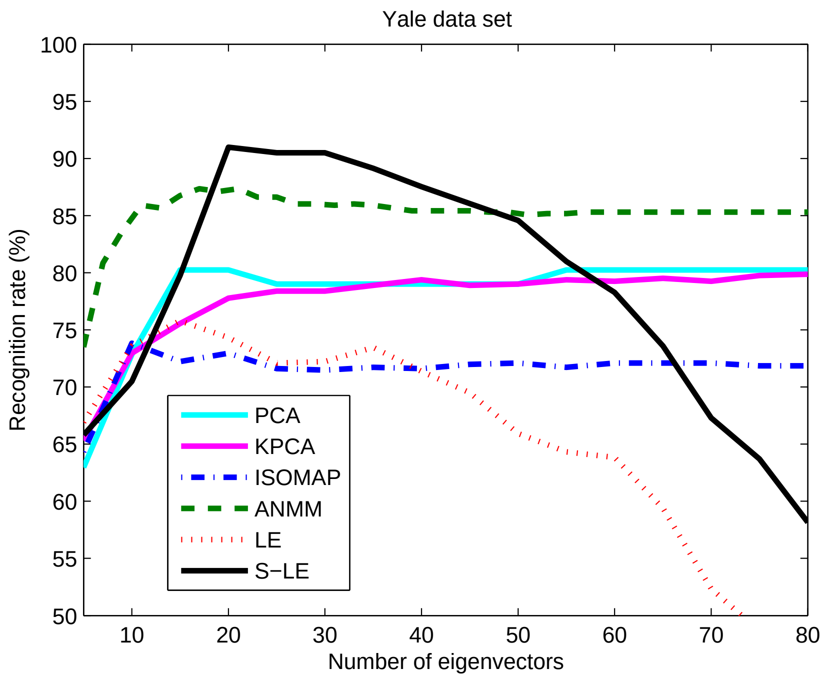


Figure 13: Average recognition rate as function of the number of eigenvectors obtained with YALE data set. The training/test percentage was set to 50%-50%

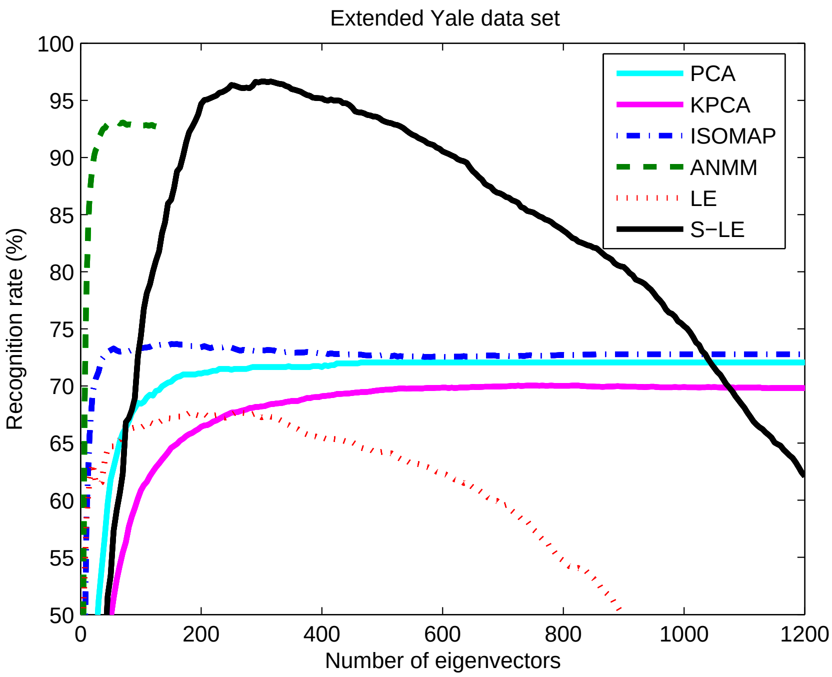


Figure 14: Average recognition rate as function of the number of eigenvectors obtained with Extended YALE data set. The training/test percentage was set to 30%-70%

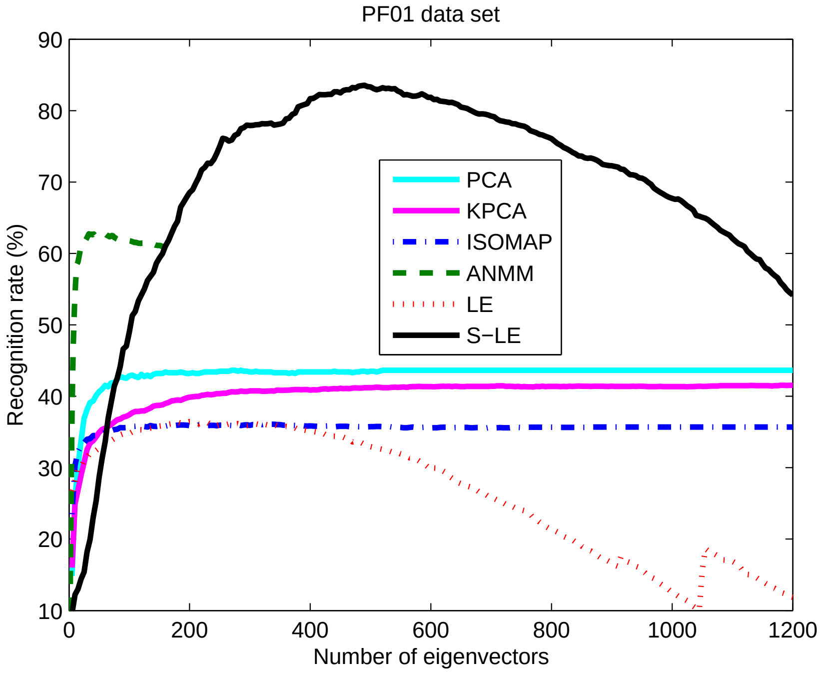


Figure 15: Average recognition rate as function of the number of eigenvectors obtained with PF01 data set. The training/test percentage was set to 50%-50%

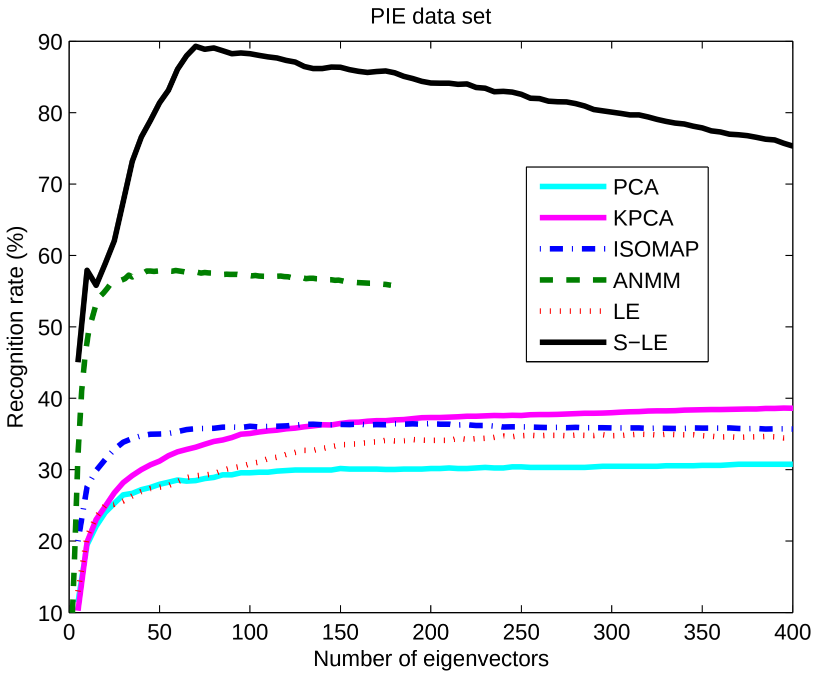


Figure 16: Average recognition rate as function of the number of eigenvectors obtained with PIE data set. The training/test percentage was set to 30%-70%

30%-70%	ORL	UMIST	Yale	Extended Yale	PF01	PIE
PCA	74.62% (25)	88.08% (45)	71.05% (10)	72.06% (465)	33.28% (385)	30.76% (370)
LDA	62.50% (25)	85.35% (10)	79.82% (10)	88.00% (10)	62.16% (55)	63.38% (65)
ANMM	89.8% (25)	96.4% (61)	82.3% (19)	93.0% (51)	54.2% (41)	57.9% (59)
KPCA	78.57% (50)	89.82% (85)	72.36% (60)	70.04% (725)	34.91% (940)	39.25% (1030)
Isomap	67.21% (30)	84.11% (25)	66.92% (10)	73.69% (125)	31.39% (115)	36.47% (200)
LE	72.53% (45)	77.88% (40)	71.76% (15)	67.69% (185)	32.71% (170)	34.97% (330)
S-LE	**91.28%** (30)	**99.25%** (15)	**86.40%** (20)	**96.65%** (300)	**80.73%** (405)	**89.28%** (70)

Table 4: Maximal average recognition obtained with six face data sets. The training/test percentage was set to 30%-70%.

50%-50%	ORL	UMIST	Yale	Extended Yale	PF01	PIE
PCA	92.50% (125)	94.44% (65)	80.24% (15)	81.73% (395)	43.62% (270)	39.25% (330)
LDA	78.00% (35)	90.27% (15)	88.88% (10)	95.94% (25)	80.40% (30)	60.33% (65)
ANMM	96.6% (53)	99% (23)	87.3% (17)	95.9% (75)	62.8% (53)	69.7% (59)
KPCA	89.25% (75)	95.79% (85)	79.87% (80)	79.40% (820)	41.53% (1180)	50.34% (1190)
Isomap	77.30% (25)	91.63% (45)	73.82% (10)	79.23% (165)	36.13% (330)	45.02% (210)
LE	82.35% (60)	86.52% (40)	75.80% (15)	74.00% (445)	36.44% (200)	42.09% (385)
S-LE	**97.45%** (30)	**99.68%** (15)	**90.98%** (20)	**99.92%** (45)	**83.54%** (490)	**93.56%** (85)

Table 5: Maximal average recognition obtained with six face data sets. The training/test percentage was set to 50%-50%.

70%-30%	ORL	UMIST	Yale	Extended Yale	PF01	PIE
PCA	97.50% (60)	99.42% (25)	85.71% (15)	88.15% (540)	48.16% (455)	46.53% (360)
LDA	83.33% (25)	97.10% (10)	97.95% (10)	97.55% (25)	89.56% (60)	68.51% (60)
ANMM	97.07% (57)	98.58% (51)	85.56% (21)	95.88% (76)	65.40% (55)	76.56% (63)
KPCA	95.25% (65)	98.49% (155)	83.46% (40)	83.79% (820)	46.22% (1150)	58.96% (1195)
Isomap	84.25% (25)	94.68% (20)	78.57% (10)	81.82% (48)	39.23% (350)	51.26% (180)
LE	87.5% (75)	90.69% (40)	78.95% (15)	75.99% (445)	39.10% (340)	46.36% (365)
S-LE	**99.08%** (15)	**100.00%** (15)	**100.00%** (10)	**99.98%** (40)	**92.28%** (405)	**96.14%** (120)

Table 6: Maximal average recognition accuracy obtained with six face data sets. The training/test percentage was set to 70%-30%.

training/test	30/70	50/50	70/30
PCA	61.64%	71.96%	77.57%
LDA	73.53%	82.30%	89.00%
ANMM	78.93%	85.21%	86.50%
KPCA	64.15%	72.69%	77.69%
Isomap	59.96%	67.18%	71.63%
LE	59.59%	66.2%	69.76%
S-LE	**90.60%**	**94.18%**	**97.91%**

Table 7: Performance of different methods averaged on six face data sets. The training/test percentage was set to 30%-70%, 50%-50%, and 70%-30%.

	ORL	UMIST	Yale	Extended Yale	PF01	PIE
LE (adaptive NB)	78.33%	96.01%	71.22%	66.60%	34.21%	36.50%
S-LE	99.08%	100.0%	100.0%	99.98%	92.28%	96.14%

Table 8: Recognition rates associated with the classic LE adopting an adaptive neighborhood and the proposed S-LE.

30/70	ORL	UMIST	Yale	Extended Yale	PF01	PIE
S-LE (NN)	91.28%	99.25%	86.4%	96.65%	80.73%	89.28%
S-LE (SVM)	56.00%	46.60%	38.70%	76.50%	37.50%	62.20%
50/50						
S-LE (NN)	97.45%	99.68%	90.98%	99.92%	83.54%	93.56%
S-LE (SVM)	91.50%	94.30%	82.40%	99.40%	87.20%	99.50 %
70/30						
S-LE (NN)	99.08%	100.0%	100.0%	99.98%	92.28%	96.14%
S-LE (SVM)	100.0%	99.5%	98.00%	99.60%	98.80%	99.80%

Table 9: Recognition rates associated with the proposed S-LE for the three groups of experiments 30%-70%, 50%-50%, and 70%-30%. For each training/test percentage, the first row depicts the results obtained with the NN classifier. The second row depicts the results obtained with the SVM classifier.

	ORL	UMIST	Yale	Extended Yale	PF01	PIE
LE	1.46	3.32	0.18	66.79	71.27	84.72
S-LE	5.81	14.05	0.73	252.48	284.42	323.96

Table 10: CPU times (in seconds) associated with the learning phase for the classic LE and the proposed S-LE.

References

[1] B. Schölkopf, A. Smola, K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Computation 10 (1998) 1299–1319.

[2] S. Roweis, L. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.

[3] L. K. Saul, S. T. Roweis, Y. Singer, Think globally, fit locally: Unsupervised learning of low dimensional manifolds, Journal of Machine Learning Research 4 (2003) 119–155.

[4] J. B. Tenenbaum, V. de Silva, J. C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.

[5] X. Geng, D. Zhan, Z. Zhou, Supervised nonlinear dimensionality reduction for visualization and classification, IEEE Transactions on systems, man, and cybernetics-part B: cybernetics 35 (2005) 1098–1107.

[6] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Computation 15 (6) (2003) 1373–1396.

[7] P. Jia, J. Yin, X. Huang, D. Hu., Incremental Laplacian Eigenmaps by preserving adjacent information between data points, Pattern Recognition Letters 30 (16) (2009) 1457–1463.

[8] L. Saul, K. Weinberger, F. Sha, J. Ham, D. Lee, Semisupervised Learning, MIT Press, Cambridge, MA, 2006, Ch. Spectral methods for dimensionality reduction.

[9] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, S. Lin, Graph embedding and extension: a general framework for dimensionality reduction, IEEE Trans. on Pattern Analysis and Machine Intelligence 29 (1) (2007) 40–51.

[10] T. Zhang, D. Tao, X. Li, J. Yang, Patch alignment for dimensionality reduction, IEEE Trans. on Knowledge and Data Engineering 21 (9) (2009) 1299–1313.

[11] H. Li, T. Jiang, K. Zhang, Efficient and robust feature extraction by maximum margin criterion, IEEE Trans. on Neural Networks 17 (1) (2006) 157–165.

[12] T. Kim, J. Kittler, Locally linear discriminant analysis for multimodally distributed classes for face recognition with a single model image, IEEE Trans. on Pattern Analysis and Machine Intelligence 27 (3) (2005) 318–327.

[13] X. He, P. Niyogi, Locality preserving projections, in: Conference on Advances in Neural Information Processing Systems, 2003.

[14] X. He, S. Yan, Y. Hu, P. Niyogi, H.-J. Zhang, Face recognition using laplacianfaces, IEEE Trans. Pattern Anal. Mach. Intelligence 27 (3) (2005) 328–340.

[15] L. Zhang, L. Qiao, S. Chen, Graph-optimized locality preserving projections, Pattern Recognition 43 (2010) 1993–2002.

[16] X. He, D. Cai, S. Yan, H.-J. Zhang, Neighborhood preserving embedding, in: IEEE International Conference on Computer Vision, 2005.

[17] Y. Fu, Z. Li, J. Yuan, Y. Wu, T. S. Huang, Locality versus globality: Query-driven localized linear models for facial image computing, IEEE Trans. on Circuits and Systems for Video Technololgy 18 (12) (2008) 1741–1752.

[18] W. Yu, X. Teng, C. Liu, Face recognition using discriminant locality preserving projections, Image and vision computing 24 (2006) 239–248.

[19] Y. Tao, J. Yang, Quotient vs. difference: Comparison between the two discriminant criteria, Neurocomputing 18 (12) (2010) 1808–1817.

[20] K. Q. Weinberger, L. K. Saul, Distance metric learning for large margin nearest neighbor classification, Journal of Machine Learning Research 10 (2009) 207–244.

[21] A. Globerson, S. Roweis, Metric learning by collapsing classes, in: Conference on Advances in Neural Information Processing Systems, 2006.

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

[23] B. Alipanahi, M. Biggs, A. Ghodsi, Distance metric learning vs. Fisher discriminant analysis, in: AAAI Conference on Artificial Intelligence, 2008.

[24] D. Donoho, C. Grimes, Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data, in: Proc. of the National Academy of Arts and Sciences, 2003.

[25] I. Borg, P. Groenen, Modern Multidimensional Scaling: theory and applications, Springer-Verlag New York, 2005.

[26] K. Q. Weinberger, L. K. Saul, Unsupervised learning of image manifolds by semidefinite programming, International Journal of Computer Vision 70 (1) (2006) 77–90.

[27] Q. Jiang, M. Jia, Supervised laplacian eigenmaps for machinery fault classification, in: World Congress on Computer Science and Information Engineering, 2009.

[28] O. Kouropteva, O. Okun, M. Pietikäinen, Supervised locally linear embedding algorithm for pattern recognition, in: Pattern Recognition and Image Analysis, Lecture Notes in Computer Science, Volume 2652/2003, 2003.

[29] M. Pillati, C. Viroli, Supervised locally linear embedding for classification: an application to gene expression data analysis, in: Proceedings of 29th Annual Conference of the German Classification Society, 2005.

[30] Y. Xu, A. Zhong, J. Yang, D. Zhang, LPP solution schemes for use with face recognition, Pattern Recognition 43 (2010) 4165–4176.

[31] J. Liu, Face recognition on Riemannian manifolds, Master's thesis, Autonomous University of Barcelona (2011).

[32] J. Shi, J. Malik, Normalized cuts and image segmentation., IEEE Trans. on Pattern Analysis and Machine Intelligence 22 (8) (2000) 888–905.

[33] S. X. Yu, J. Shi, Multiclass spectral clustering, in: IEEE International Conference on Computer Vision, 2003.

[34] N. Goel, G. Bebis, A. Nefian, Face recognition experiments with random projections, in: SPIE Conference on Biometric Technology for Human Identification, 2005.

[35] R. Duda, P. Hart, D. Stork, Pattern Classification, John Wiley and Sons, New York.

[36] S. Yan, D. Xu, B. Zhang, H. Zhang, Graph embedding: a general framework for dimensionality reduction, in: IEEE Conference on Computer Vision and Pattern Recognition, 2005.

[37] X. Qiu, L. Wu, Face recognition by stepwise nonparametric margin maximum criterion, in: IEEE International Conference on Computer Vision, 2005.

[38] J. Wright, A. Yang, A. Ganesh, S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Trans. on Pattern Analysis and Machine Intelligence 31 (2) (2009) 210–227.