J. Basic. Appl. Sci. Res., 3(4)483-491, 2013

© 2013, TextRoad Publication

ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

Supervised Nonlinear Dimensionality Reduction Based on Evolution Strategy

Mudasser Naseer[a], **Shi-Yin Qin**[b]

[a]Department of Computer Science, COMSATS Institute of Information Technology, Lahore, Pakistan
[b]School of Automation Science and Electrical Engineering, Beihang University, Beijing China

ABSTRACT

Most of the classifiers suffer from the curse of dimensionality during classification of high dimensional image and non-image data. In this paper, we introduce a new supervised nonlinear dimensionality reduction (S-NLDR) algorithm called supervised dimensionality reduction based on evolution strategy (SDRES) for both image and non-image data. The SDRES method uses the power of evolution strategy (ES) algorithm to find low dimensional embedding of high dimensional labeled data. The new algorithm makes the interclass dissimilarity larger than the intraclass dissimilarity while finding low dimensional embedding values. Simulation studies on some well-known benchmark datasets demonstrate that SDRES generally gives better results in dimensionality reduction and classification as compared to other famous S-NDLR methods such as WeightedIso, supervised S-Isomap, supervised locally linear embedding (SLLE), enhanced supervised locally linear embedding (ESLLE) and supervised local tangent space alignment (SLTSA).

KEYWORDS: nonlinear dimensionality reduction; supervised classification; evolution strategy, manifold learning.

1. INTRODUCTION

The problem of classification or supervised learning is about predicting the unknown class of a new observation, based on the class membership information about the available dataset. Typically speaking a set of patterns of features along with correct classification, known as training data, is available and the task is to classify a new set of patterns, known as test data. Many decision-making problems fall into the general category of classification [1]. In a number of practical applications digital images are used for classification of objects; most examples may deal with the classification of handwritten digits, motor vehicle number plates, human face images and so on. The higher dimensions of the image data and in some cases non-image data, deteriorate the performance of any classifier and needs a reduction in the dimension before the application of the classification method such as K-NN. Fortunately most of the high dimensional data is intrinsically low dimensional. Thus, the problem of classification of high dimensional data can be solved by first mapping the data into low dimensional subspace and then applying the classification method [2].

Recently a class of nonlinear dimensionality reduction (NLDR) methods, based on the concept of manifold learning, have captured the attention of researchers. For classification purposes, supervised versions of these manifold learning methods known as supervised nonlinear dimensionality reduction (S-NLDR) are presented by different authors. Commonly used examples are WeightedIso [3], Supervised S-Isomap [4], Supervised Locally Linear Embedding (SLLE) [2, 5], Modified Supervised Locally Linear Embedding (MSLLE) [6], Enhanced Supervised Locally Linear Embedding (ESLLE) [7], and Supervised LTSA[8-9].

In this paper, we introduce a new method for classification of high dimensional image and non-image labeled data. The new method uses the concept of Evolution Strategy (ES) to reduce dimensionality of the labeled data points and then uses estimation method discussed by Li [8] to find low dimensional embedded points for out-of-sample high dimensional data points. We call the new method supervised dimensionality reduction based on evolution strategy (SDRES). A comparative analysis of the performance of the new method with some well-known S-NLDR methods shows very promising results.

The rest of the paper is organized as follows: in Section 2 an overview of the related work is presented. In Section 3 a brief description of evolutionary algorithms and evolution strategy algorithm is given. The proposed method is described in detail in Section 4. Discussions on the experimental results are given in Section 6. Conclusion and future work are presented in Section 7.

Corresponding Author: Mudasser Naseer, Department of Computer Science, COMSATS Institute of Information Technology, Lahore, Pakistan. Mobile: +92-333-4439321 Email: mnaseer@ciitlahore.edu.pk; mudaser1@yahoo.com

2. RELATED WORK

In recent years, a number of nonlinear dimensionality reduction methods have been developed under the assumption that data points lie on an underlying manifold embedded in a high-dimensional space [7]. Among them, two representative algorithms are isometric mapping (Isomap) [10] and locally linear embedding (LLE) [11]. Both these methods are unsupervised dimensionality reduction methods and could not be directly used for the purpose of classification. Different authors develop supervised versions of these methods by modifying the computation of pairwise distances between data points so that the interclass dissimilarities become larger compared to intraclass dissimilarities. A brief description of some well known supervised versions of Isomap and LLE is given in the following subsections.

2.1. WeightedIso

The WeightedIso changes first step of the standard Isomap algorithm. It proceeds by first computing the *K* nearest neighbors of each data point *x* and denotes with K_{same} the set of nearest neighbors having the same class label as *x*. Then "move" each nearest neighbor in K_{same} closer to *x* by re-scaling their Euclidean distance by a constant factor $1/\lambda,\ (\lambda>1)$. Remaining steps of the algorithm remain same as of the unsupervised Isomap.

2.2. Supervised Isomap

In S-Isomap Euclidean distance *D* is replaced by D' , where

$$D'(x_i, x_j) = \begin{cases} \sqrt{1-e^{\frac{-D^2(x_i,x_j)}{\beta}}} & L_i = L_j \\ \sqrt{e^{\frac{D^2(x_i,x_j)}{\beta}}} - \alpha & L_i \neq L_j \end{cases} \tag{1}$$

The parameter β is used to prevent $D'(x_i, x_j)$ to increase too fast when $D(x_i, x_j)$ is relatively large. Thus, the value of β should depend on the density of the dataset. Usually, it is set to be the average Euclidean distance between all pairs of data points. The parameter α ($0\leq \alpha \leq 1$) gives a certain chance to the points in different classes to be "more similar" so that the dissimilarity in different classes may be smaller than that in the same class.

2.3. Supervised LLE

Among different versions of SLLE, we choose the one given in [5], due to its simplicity of application. The SLLE uses the idea of adding distance between samples in different classes as

$$D' = D + \alpha \max(D)\Delta \tag{2}$$

where *D* is the pairwise Euclidean distance matrix for the complete dataset without considering the class label information, and D' is the distance matrix integrating with the class label information. If the data points are from different classes, then $\Delta_{ij} = 1$ and $\Delta_{ij} = 0$ otherwise. Here $\alpha \in (0,1)$ controls the amount to which class information should be incorporated. When $\alpha = 0$, SLLE is equivalent to the original unsupervised LLE; when $\alpha = 1$, the result is the fully supervised LLE (called 1-SLLE). This supervised version of LLE behaves as a nonlinear Fisher mapping where α controls the nonlinearity.

2.4. Enhanced Supervised LLE

In ESLLE [7] the author uses the same logic as used by Geng et al. [4] in S-Isomap. In the ESLLE algorithm Euclidean distance *D* is replaced by D' as given in the following equation

$$D' = \begin{cases} \sqrt{1-e^{\frac{-D^2}{\beta}}} & L_i = L_j \\ \sqrt{e^{\frac{D^2}{\beta}}} - \alpha & L_i \neq L_j \end{cases} \tag{3}$$

The characteristics of parameters α and β are already explained in section 2.2 above.

2.5. Supervised LTSA

Supervised LTSA also use the idea of artificially increasing the shift distances between points belonging to different classes, but leaving them unchanged if samples are from the same class. The new pairwise distance matrix is given as $D' = D + \rho\Delta$, where the shift distance ρ is assigned a relatively very large value in comparison with the distance between any pairs of points, $\Delta_{ij} = 1$ if the data points are from different classes, and 0 otherwise.

We choose above mentioned S-NLDR methods due to their similar approach of using class information by increasing distance between data points of different classes. Supervised NLDR methods do not explicitly provide any mapping function for out-of-sample data points, it can be learnt by estimation method [8, 12-13] or by some nonlinear interpolation techniques, such as Generalized Regression Networks (GRN) [14]. To summarize, the general classification procedure has three steps as follows:

i. Map high dimensional data into a lower dimensional space using any S-NLDR method.

ii. Find mapping function for out-of-sample data points using estimation method or GRN.

iii. Map the given query data point in a low dimensional space using the mapping function and then predict its class label using a classifier.

3. EVOLUTION STRATEGY (ES) ALGORITHM

Evolutionary algorithms (EAs) are generic, population-based, meta-heuristic optimisation algorithms that use biology-inspired mechanisms like mutation, crossover, natural selection and survival of the fittest [15]. It is widely believed that the specific potential of EAs originates from their parallel search by means of entire populations [16]. Evolution Strategies (ESs), along with Genetic Algorithms (GAs), are one of the variants of EAs. The emphasis in ES techniques is not on the acquisition of a structure with high fitness, as in GA, but for a behavior that is rated well by the fitness function [17].

In our new supervised NLDR method we use ES algorithm proposed in [17-18], to find low dimensional embedding in case of class information. The authors of the ES algorithm introduce the concept of competing subpopulations in the process of crossover and then a dynamic time variant mutation (TVM) operator was used [17-18]. We use this particular ES algorithm in our method due to its fast convergence and more precise results [17]. A brief structure of ES algorithm is given below, for detail of the algorithm the readers are referred to see Hashem et al. 1997) and Watanabe and Hashem (2004).

3.1. The Evolution Strategy Algorithm

i) Initial Population

The initial population is generated by using uniform random numbers (URN). The individuals of the population are represented by $y_1, y_2, ...y_\mu$, where $y_1 = [u_1, u_2, ..., u_n]^T$ and $-100 < u_i < 100$. The μ denotes the number of individuals in the population.

ii) Competing Subpopulation-based Arithmetical Crossover

Divide the parent population consisting of μ individuals into equal sized l competing subpopulations. Define $y^t_{j,\max}$ as an elite individual that maximizes a cost function, at time t, within the j-th subpopulation and $\overline{y}^t_j$ as a mean strength of the j-th subpopulation except the $y^t_{j,\max}$, define the crossover operation for the competing subpopulation j to produce two offspring as follows :

$$\zeta^t_1 = \alpha y^t_{j,\max} + (1-\alpha)\overline{y}^t_j \tag{4}$$

$$\zeta^t_2 = (1-\alpha) y^t_{j,\max} + \alpha\overline{y}^t_j \tag{5}$$

where α is selected from URN[0, l]

iii) The Time-Variant Mutation

The TVM is defined for a child ***i*** , as

$$\zeta'_i = \zeta_i + \sigma(t).N_i(0,1) \qquad \forall_i \in \{1, 2, ..., n\} \tag{6}$$

where $N_i(0,1)$ is the Gaussian random value with zero-mean and unity variance, and $\sigma(t)$ is the time-variant mutation step generating function at the generation t, which is defined by

$$\sigma(t) = \left(1 - r^{\left(1-\frac{t}{T}\right)^{\gamma}} \right) \qquad (7)$$

The value of r is selected from the URN[0, 1], T is the maximal generation number and γ is a real-valued parameter greater than unity.

iv) Evaluation

After the mutation operation, each child is evaluated in its cost function (fitness) for a possible solution in each generation.

v) Alternation of Generation

The $\mu + \mu$ individuals are ordered in proportional to the amount of the cost function, and the best μ individuals are selected for the next generation.

4. SUPERVISED DIMENSIONALITY REDUCTION BASED ON EVOLUTION STRATEGY (SDRES)

For any classification algorithm, the goal is to map the data into a feature space in which members of different classes are clearly separated [4]. To achieve this goal we artificially increase the pairwise distances between samples belonging to different classes, but leaving them unchanged if samples are from the same class. After constructing modified pairwise distance matrix, we find pairwise geodesic distances between all the points to find actual distances between these points in curvilinear space. In order to find low dimension embedded points we use the ES algorithm to minimize SSE between pairwise Euclidean distances among low dimensional embedded points and the geodesic distances among corresponding high dimensional data points. Implementation steps of SDRES algorithm are;

Algorithm: Supervised Dimensionality Reduction based on Evolution Strategy (SDRES)

Input: D-dimensional N data points $\mathrm{X}_i \in \Re^D$ (where i = 1, 2,…, N), along with class labels.

Output: d-dimensional embedded points $\mathrm{Y}_i \in \Re^D$ (where i = 1, 2,…, N),

PHASE I: Compute pairwise geodesic distances between all data points

1. Compute all possible pairwise Euclidean distances $d(x_i, x_j)$ between high dimensional dmatrix withto form a N×N matrix D_E with 0-diagonals.
2. Compute new distance matrix D'_E by adding distance between data points from different classes as $D'_E = D_E + \alpha \max(D_E)\Delta$, where $\Delta_{ij} = 1$ if the data points are from different classes, and 0 otherwise. Here $\alpha \in (0,1)$ controls the amount to which class information should be incorporated.
3. Compute neighbourhood graph G from D'_E: Put an edge between nodes i and j if X_i and X_j are "close".
   There are two variations:
   (i) K-nearest neighbours. Nodes i and j are connected by an edge if i is among K nearest neighbours of j or j is among K nearest neighbours of i. Set edge lengths equal to $d(x_i, x_j)$.
   (ii) ε-neighbourhoods. Nodes i and j are connected by an edge if $d(x_i, x_j) < \varepsilon$.
4. Use Floyd Warshall algorithm to find the N×N matrix D_G of geodesic distances from G.

PHASE II: Find low dimensional optimal values of anchor points

5. Randomly select N_a high dimensional data points (called anchor points) and initialize the coordinates of corresponding low dimensional points using URN[a, b]. Place these low dimensional points in set-A.
6. Randomly select two points from set-A. Find optimal coordinates of these points by using the ES algorithm to minimize the difference between the Euclidean distance between these points and geodesic distance between corresponding high dimensional data points. Move these optimal points to set-B.
7. Randomly select a point from set-A and compute Euclidean distances between this point and all the points in the set-B.

8. Find optimal coordinates of the new point by using the ES algorithm to minimize SSE between Euclidean distances from Step-7 and geodesic distances between corresponding high dimensional data points, where $\text{SSE} = \sum_{j=1}^{N_B} \left(d\left(y_k, y_j\right) - d_G\left(x_k, x_j\right) \right)^2$. Move new optimal point to set-B.

Repeat steps 7-8 until set-A is empty. Now set-B becomes a low dimensional representation of high dimensional anchor points.

PHASE III: Find low dimensional optimal values of non-anchor points

9. Select a point ***r*** from non-anchor points and initialize its low dimensional coordinates using URN[a, b].

10. Compute all possible Euclidean distances between ***r*** and points in set-B.

11. Find optimal coordinates of the new point ***r*** by using the ES algorithm to minimize SSE between Euclidean distances from Step-10 and geodesic distances between corresponding high dimensional data points.

Repeat steps 9-11 for all non-anchor points.

The above SDRES algorithm involves following parameters,

i. *K* for *K*-array neighbourhood or *ε* for *ε*-balls

ii. The number of anchor points and

iii. Range [a, b] for initialization of low dimensional points

For computational purpose, we use *K*=8 and 10 for all the datasets. The number of anchor points should be at least twice the number of classes so that at least two points represent are selected from each class. A proportional allocation method can be used to allocate anchor points to different classes. The selection of anchor points from different classes should be done entirely at random. Range for initialization of low dimensional embedding values does not affect the results greatly. However, if we take a wider range say [-100, 100], the ES algorithm may take more time to converge. For our sample datasets we use a range of [-10, 10], for all the data points.

Other parameters of SDRES are related to implementation of ES algorithm. These are population size *μ*, sub-population size *m* and dependency parameter *γ*. We take *μ*=60, *m*=6 and *γ*=8, as suggested in [18].

Convergence/termination of the DRES algorithm depends on the convergence/termination of ES algorithm. We set the termination condition of ES algorithm as; if the improvement in the optimal value of low dimensional point is less than 10×10^{-7} or if the generation number reaches its maximum value of 500.

The complete procedure for classification process is summarized as below:

i. Map high dimensional data into lower dimensional subspace using SDRES method.

ii. Find low dimensional mapping for out-of-sample data point using estimation method (discussed in subsection 6.2).

iii. Find class membership for out-of-sample data point using *κ*-NN classifier.

4.1. Advantages of SDRES algorithm

(i) Most of the algorithms mentioned in section-2 use eigenvalue decomposition of a square matrix constructed from high dimensional data points. The DRES algorithm does not require this step, so the problem of an ill-condition matrix does not arise.

(ii) The DRES algorithm is memory efficient as it does not require storing all the points, rather it stores only anchor points both in high and low dimensional spaces.

4.2. Limitations of SDRES

The main drawback of our proposed algorithm is that it takes a long time during the training phase due to its incremental structure. This drawback makes it inefficient in the case of a very large training dataset. This problem can be overcome by selecting a small representative training dataset from each class.

5. PERFORMANCE EVALUATION CRITERIA

Commonly used performance evaluation criteria for classification is the error rate (ER). If unlimited cases for training and testing part are available, the ER can readily be obtained as the ER on the test cases. The simplest technique for estimating error rates, the holdout method, is a single train and test experiment [19]. The sample cases are broken into two groups of cases: a training group and a test group. The classifier is independently derived from the training cases, and the error estimate is the performance of the classifier on the test cases. A single random partition of training and test cases can be somewhat misleading. Random resampling can produce better estimates than a single train and test partition.

In this study, we use 10-fold cross validation resampling method to find error rate. The original dataset is randomly divided into ten equal-sized subsets. Then in each fold, one subset is used as a testing set and the union of the remaining ones is used as training set. After 10-folds, the average result is taken as the final ER. We use standard deviation (SD) of 10-fold results as a performance consistency indicator of any S-NLDR method. For Olivettie faces and Lübeck University face datasets, we use leave-one-out resampling method due to a small number of images available for each subject.

6. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we present results of several examples to illustrate the performance of our ES based SNLDR algorithm. Test datasets include non-image and high dimensional grayscale image datasets.

6.1 Datasets

In order to demonstrate the performance of our proposed method, experiments on several image and non-image benchmark datasets are carried out. Table-1 gives summary information about used datasets. Some sample images from different image datasets are shown in Figure-1 to Figure-7.

Table 1. Summary Information of datasets used

Sr. No.	Dataset	No. of classes	No. of instances	No. of dimensions	Resampling Method	Source
1.	Iris	3	150	4	10-fold	http://archive.ics.uci.edu/ml/datasets/Iris
2.	Wine	3	178	13	10-fold	http://archive.ics.uci.edu/ml/datasets/Wine
3.	ISOLET	6	500	617	10-fold	http://archive.ics.uci.edu/ml/datasets/ISOLET
4.	Optical recognition of handwritten digits	6	600	64	10-fold	http://archive.ics.uci.edu/ml/datasets/Optical+Recognition +of+Handwritten+Digits
5.	CMU faces	6	184	40×44=1760	10-fold	http://archive.ics.uci.edu/ml/datasets/CMU+Face+Images
6.	Yale-B	4	550	36×32=1152	10-fold	http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html
7.	UMIST faces	3	100	45×37=1665	10-fold	http://www.cs.toronto.edu/~roweis/data.html
8.	MNIST digits	5	500	28×28=784	10-fold	http://www.cs.toronto.edu/~roweis/data.html
9.	Olivettie faces	7	70	64×64=4096	Leave-one-out	http://www.cs.toronto.edu/~roweis/data.html
10.	Lübeck Univ. faces	5	65	36×48=1728	Leave-one-out	http://www.inb.uni-luebeck.de/lehre-de/ss08/ ni1/ueb3/faces.mat/view?set_language=en
11.	Student faces dataset	6	500	48×64=3072	10-fold	Data is captured in our lab using Logitech Quickcam Pro 5000 Webcam with a white background. Images are used without any cropping.

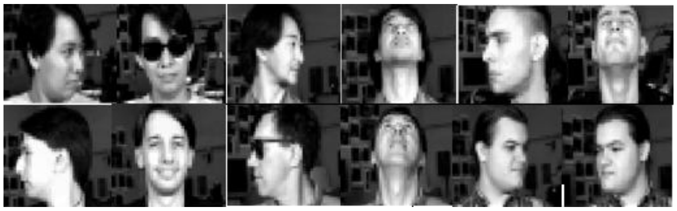

Figure 1. Sample Images from CMU face dataset

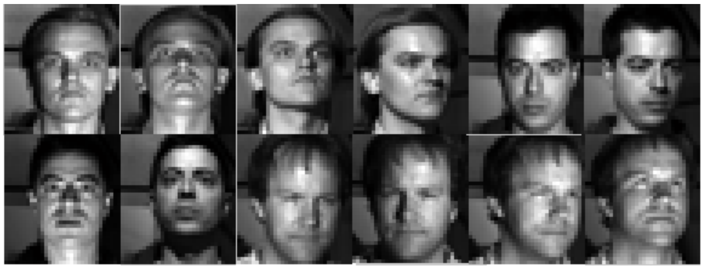

Figure 2. Sample Images from Yale-B face images dataset

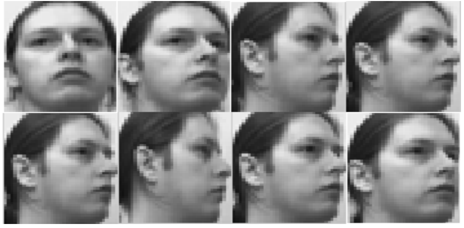

Figure 3. Sample Images from UMIST face images dataset

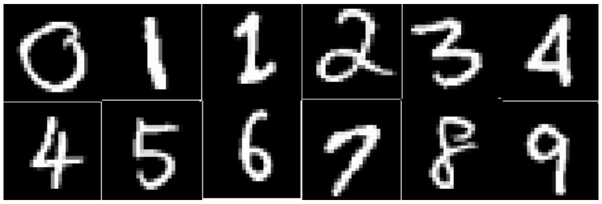

Figure 4. Sample Images from MNIST handwritten digits dataset

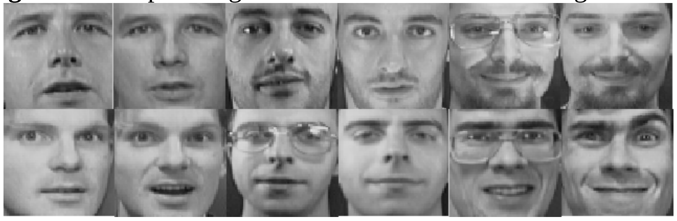

Figure 5. Sample Images from Olivettie faces dataset

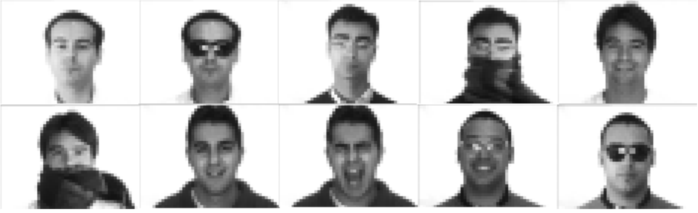

Figure 6. Sample Images from Lübeck Uni. face dataset

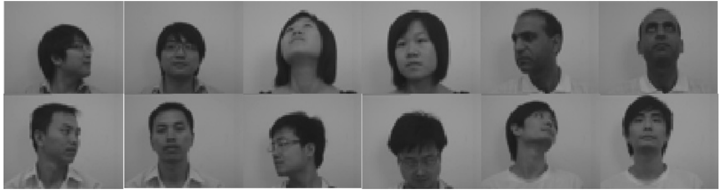

Figure 7. Sample images from student faces dataset

6.2 Experimental Setup

The number of neighboring points *K* is a common parameter for all the supervised NLDR methods except SDRES. We take two different values of *K*=8 and 10. Different researcher most commonly uses these values of *K*. Target dimension *d* is 2. The value of λ in WeightedIso is 10, α in S-Isomap is 0.5 and *β* is set to an average Euclidean distance of all pairwise distances, whereas the value of *ρ* in SLTSA is set to max(D). These parameter values are suggested by the respective authors. The value of α in SLLE and SDRES is set to 1, to make full use of class information. As an out-of-sample mapping function, we use Estimation method discussed in [8]. The authors construct a mapping function by assuming that new points come from those parts of the high-dimensional space that have already been explicitly mapped. The low dimensional representative point of high dimensional new point x_{N+1} is obtained as;

$$y_{N+1} = \overline{y}_j + L(x_{N+1} - \overline{x}_j)^+ \quad (7)$$

where x_j is nearest neighbour about x_{N+1}, y_j is corresponding low dimensional point, $\overline{x}_j$ and $\overline{y}_j$ are respectively the mean of *K* nearest neighbours of x_j and y_j . The transformation matrix *L* is obtained as $L = (y_j - \overline{y}_j)(x_j - \overline{x}_j)^+$, where (.)$^+$ is the Moor-Penrose generalized inverse of a matrix. Finally, to predict class labels for training and test data, the non-parametric *κ*-NN (*κ* = 9) classifier with the Euclidean metric is used due to its simplicity.

6.3 Results

We run all the algorithms on above mentioned datasets. For most of the datasets, results for *K*=8 are marginally better than *K*=10, therefore we present results only for *K*=8. Table 2 shows the percentage mean error rate (PMER) and SD for 10-fold or leave-one-out resampling method for different S-NLDR methods. We did not present results for S-Isomap because it produces disconnected geodesic distance graphs for all the datasets, except Olivettie faces. For every dataset, best value of PMER is represented by bold face.

Table 2. PMER and SD for Supervised NLDR Methods

Sr. No.	Datasets	WeightedIso		SLLE		ESSLE		SLTSA		SDRES	
		PMER	SD	PMER	SD	PMER	SD	PMER	SD	PMER	SD
1.	Iris	26.7	0.17	**4.0**	0.06	10.7	0.09	**4.0**	0.06	**4.0**	0.06
2.	Wine	25.6	0.09	**24.0**	0.09	**24.0**	0.09	**24.0**	0.09	**24.0**	0.09
3.	ISOLET	**38.6**	0.09	74.1	0.13	68.4	0.15	43.4	0.12	54.1	0.07
4.	Optical recognition of handwritten digits	**4.8**	0.05	33.9	0.11	32.4	0.08	5.0	0.05	**4.8**	0.03
5.	CMU faces	**28.1**	0.12	56.0	0.15	53.8	0.16	64.0	0.10	**28.0**	0.14
6.	Yale-B faces	14.6	0.04	35.9	0.10	39.1	0.08	13.6	0.06	**13.2**	0.05
7.	UMIST faces	66.7	0.11	71.5	0.13	75.0	0.18	77.0	0.12	**52.5**	0.14
8.	MNIST digits	18.0	0.06	43.9	0.08	41.4	0.09	19.4	0.06	**17.7**	0.05
9.	Olivettie faces	**35.8**	0.21	62.5	0.16	68.0	0.15	77.7	0.05	57.5	0.08
10.	Lübeck University faces	7.7	0.10	13.8	0.17	12.3	0.13	**1.5**	0.05	**1.5**	0.05
11.	Student faces dataset	60.4	0.1	Crashed		Crashed		**0.2**	0.01	**0.2**	0.01

An overview of Table 2 reveals that no single algorithm performs best for all the datasets. However, the performance of SDRES is significantly better than other S-NLDR methods. For non-image data i.e. Iris, Wine, and Optical recognition of handwritten digits the SDRES method gives best performance when compared with other S-NLDR methods. For image data, it gives the minimum error rate for six datasets. i.e. CMU faces, Yale-B faces, UMIST faces, MNIST faces, Lübeck University faces, and Student faces dataset. For remaining one dataset i.e. Olivettie faces the SDRES method gives the second best performance. Small values of SD for different datasets are an indication of consistence performance of the SDRES algorithm across 10-fold or leave-one-out samples.

7. CONCLUSIONS

A new ES based method for supervised NLDR mapping is proposed in this paper. The new method takes into account class membership to enhance discriminating ability of low dimensional embedding.

We compare the proposed method with other supervised mapping techniques such as WeightedIso, SLLE, ESLLE and SLTSA on a number of image and non-image datasets in order to gain insight into what methods are suitable for data classification. The SDRES method yields very promising classification results in our experiments and comes out to be the best method.

The only disadvantage we come across during experimentation with different datasets is that SDRES method takes a long time during the training phase, which is due to its incremental structure.

8. ACKNOWLEDGEMENTS

This work is supported by hi-tech research and development (863) program of China [Grant No. 2008AA12A200 and Grant No. 2006AA04Z207] and research fund for doctoral program of higher education of China [Grant No. 20060006018] as well as Programs of National Natural Science Foundation of China [Grant No. 60875072].

REFERENCES

[1] James, M., 1985. Classification Algorithms, John Wiley & Sons, New York.

[2] Ridder, D. d. and R. P. W. Duin, 2002. Locally linear embedding for classification, Pattern Recognition Group, Imaging Science & Technology Department, Faculty of Applied Science Delft University of Technology: Delft, The Netherlands.

[3] Vlachos, M., C. Domeniconi, D. Gunopulos, G. Kollios and N. Koudas, 2002. Nonlinear Dimensionality Reduction Techniques for Classification and Visualization. In the Eighth ACM SIGKDD international conference on Knowledge discovery and data mining Edmonton, Alberta, Canada, pp: 645 - 651.

[4] Geng, X., D.-C. Zhan and Z.-H. Zhou, 2005. Supervised Nonlinear Dimensionality Reduction for Visualization and Classification, IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics, 35(6): 1098-1107.

[5] Ridder, D. d., O. Kouropteva, O. Okun, M. Pietikainen and R. P. W. Duin, 2003. Supervised locally linear embedding. In the Artificial Neural Networks and Neural Information Processing, ICANN/ICONIP, Lecture Notes in Computer Science 2714, Istanbul, Turkey, pp: 333-341.

[6] Busa-Fekete, R. and A. Kocsor, 2005. Locally Linear Embedding and its Variants for Feature Extraction, in IEEE International Workshop on Soft Computing Applications, SOFA IEEE: Szeged-Hungary

[7] Zhang, S.-q., 2009. Enhanced supervised locally linear embedding, Pattern Recognition Letters, 30: 1208-1218.

[8] Li, H., L. Teng, W. Chen and I.-F. Shen, 2005. Supervised Learning on Local Tangent Space. In the Proceedings of 2nd International Symposium on Neural Networks, pp: 546-551.

[9] Li, H., W. Chen and I.-F. Shen, 2005. Supervised Local Tangent Space Alignment for Classification. In the Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, IJCAI, Edinburgh, Scotland, UK, pp: 1620-1621.

[10] Tenenbaum, J. B., V. d. Silva and J. C. Langford, 2000. A Global Geometric Framework for Nonlinear Dimensionality Reduction, Science, 290: 2319-2323.

[11] Roweis, S. T. and L. K. Saul, 2000. Nonlinear Dimensionality Reduction by Locally Linear Embedding, Science, 290: 2323-2326.

[12] Bengio, Y., J.-F. Paiement, P. Vincent, O. Delalleau, N. L. Roux, and M. Ouimet, 2003. Out-of-Sample Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering. In the Neural Information Processing Systems (NIPS), 16, Vancouver, Canada.

[13] Kouropteva, O., O. Okun, A. Hadid, M. Soriano, S. Marcos, and M. Pietikainen, 2002. Beyond Locally Linear Embedding Algorithm, University of Oulu: Finland.

[14] Ho, T. K., 1998. Nearest Neighbors in random subspaces. In the Proceeding of the joint IAPR International Workshops on Advances in Pattern Recognition, pp: 640-648.

[15] Weise, T., 2007. Global Optimization Algorithms: Theory and Application, Thomas Weise.

[16] Salomon, R., 1998. Evolutionary Algorithms and Gradient Search: Similarities and Differences, IEEE Transactions on Evolutionary Computation, 2(2): 45-55.

[17] Hashem, M. M. A., K. Watanabe and K. Izumi, 1997. Evolution Strategy : A New Time-Variant Mutation for Fine Local Tuning. In the Proceeding of the SICE International Sessions, tokushima, Japan, pp: 1099-1103.

[18] Watanabe, K. and M. M. A. Hashem, 2004. Evolutionary Computations: New Algorithms and their Applications to Evolutionary Robots, Studies in Fussiness and Soft Computing, Springer, Berlin Heidelberg.

[19] Fukunaga, K., 1990. Introduction to Statistical Pattern Recognition, 2nd Ed. ed, Computer Science and Scientific Computing, ed. W. Rhenboldt, Academic Press, New York.