IET Image Processing

Research Article

IET Journals
The Institution of Engineering and Technology

Supervised sparse neighbourhood preserving embedding

ISSN 1751-9659
Received on 15th April 2016
Revised 6th October 2016
Accepted on 27th November 2016
E-First on 4th January 2017
doi: 10.1049/iet-ipr.2016.0254
www.ietdl.org

Liqiang Qian[1], Li Zhang[1,2] ✉, Xing Bao[1], Fanzhang Li[1,2], Jiwen Yang[1]

[1]*School of Computer Science and Technology & Joint International Research Laboratory of Machine Learning and Neuromorphic Computing, Soochow University, Suzhou 215006, Jiangsu, People's Republic of China*

[2]*Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 210000, Jiangsu, People's Republic of China*

✉ *E-mail: zhangliml@suda.edu.cn*

Abstract: Both neighbourhood preserving embedding (NPE) and sparsity preserving projection (SPP) are unsupervised learning methods, where NPE can preserve the local neighbourhood information of a given dataset and SPP can preserve the sparsely reconstructive relationship of the dataset. However, it is not satisfactory when applying the two methods to classification tasks. First, a modified SPP is presented here. Then this study proposes a supervised sparse neighbourhood preserving embedded algorithm (SSNPE) based on NPE, the modified SPP and the label information of a given task. SSNPE inherits the merits of NPE and SPP, which can preserve not only the local neighbourhood information but also the sparsely reconstructive relationship. The connection between SSNPE and both NPE and SPP is discussed. Experimental results on the datasets of UCI, ORL and MNIST indicate that the proposed method is effective.

1 Introduction

Dimensionality reduction has attracted many attentions in computer vision, machine learning and pattern recognition [1, 2]. Since a lot of features may even pull down the performance of classifiers and result in the curse of dimensionality, it is necessary to adopt dimensionality reduction techniques to preprocess high-dimensional data. Generally, dimensionality reduction can reduce the data storage and the computational complexity using dimensionality reduction techniques for the task at hand.

Over the past few decades, many methods have been proposed for reducing dimensionality, most of which have been deployed to practical applications successfully, such as image data [3, 4] and gene expression data [5]. There are two kinds of dimensionality reduction methods: unsupervised and supervised. Presently, principle component analysis (PCA) [6, 7] is one of the most popular unsupervised methods. PCA represents the original data with a lower dimension by minimising the reconstruction error. Since PCA ignores the label information, it has little to do with the classification task. Unlike PCA, linear discriminant analysis (LDA) [8], a classical supervised method, can find an optimal projection by maximising the ratio of the between-class scatter to the within-class scatter. Due to utilising the class information available, LDA is more effective than PCA in classification applications. It is noteworthy to point out that both PCA and LDA share a common characteristic, that is, only the global linear Euclidean structure of data is taken into account and no attention is paid to the local structure of data. As a result, both LDA and PCA are unable to explore the essential structure of data if the data points are located in a non-linear manifold.

Manifold learning is a family of popular approaches for non-linear dimensionality reduction. In manifold learning, there is an assumption that data points are essentially sampled from a manifold with low dimension which is embedded in a high-dimensional ambient space. The goal of manifold learning is to uncover these parameters and find a low-dimensional representation for the high-dimensional data. Methods for manifold learning include isometric mapping [9], locally linear embedding (LLE) [10], Laplacian eigenmap (LE) [11], locality preserving projection (LPP) [12], sparsity preserving projecting (SPP) [13], and neighbourhood preserving embedding (NPE) [14]. LPP and NPE are linearised versions of LE and LLE, respectively. SPP has a similar procedure as NPE. Here, we mainly focus on NPE and SPP.

The goal of NPE is to preserve the local manifold structure. Given a set of samples in the ambient space, NPE first builds a weight matrix which describes the relationship between the data points, and then finds an optimal embedding such that the neighbourhood structure can be preserved in the subspace. Both the k nearest neighbour algorithm and the least squares method are used for constructing the weight matrix in NPE. SPP chooses its neighbourhood automatically instead of using the k nearest neighbour algorithm, preventing it from suffering from the difficulty of parameter selection as in the case of NPE [14]. Despite of such difference, SPP can actually be thought as a regularised extension of NPE through the modified l_1 regularisation problem. Though NPE and SPP have been applied in many domains, they suffer from deemphasising discriminant information, which make them not suitable for recognition tasks. Neighbourhood preserving discriminant embedding (NPDE) [15] was proposed for finding the subspace with abundant discrimination information by maximising the between-class distance, while minimising the within-class distance. Bao *et al.* proposed a novel supervised NPE (SNPE) which uses the label information to construct attraction vectors [16]. In SNPE, each embedded sample should be around the corresponding attraction vector. Thus, SNPE can achieve better classification performance compared with NPDE. As a new sparse subspace learning algorithm, DSNPE (discriminant sparse NPE) [17] incorporates the discriminant information into SPP. As a result, DSNPE can not only preserve the sparse reconstructive relationship but also employ the global discriminant structures sufficiently. The objective function of DSNPE contains the maximum margin criterion (MMC) [18]. When constructing the weight matrix, only examples with the same label are selected to obtain the sparse reconstructive relationship.

To improve the classification performance of SNPE, this paper proposes an effective supervised manifold learning method, called supervised sparse neighbourhood preserving embedding (SSNPE) which can be taken as a framework of some relative methods, such as NPE, SPP and SNPE. SSNPE not only takes the local neighbourhood information and local sparse reconstructive relationship of data into account when building the weight matrices but also constructs attractors using discrimination information. In SSNPE, there are two weight matrices, or sparse representation

weight matrix and neighbourhood weight matrix. The objective of SSNPE is simultaneously to minimise the reconstruction error in the subspace with low dimension and to attract the samples with different labels to the corresponding attractors. Experimental results show that SSNPE can meaningfully and stably represent data in a subspace with low dimension.

This paper is organised as follows: We briefly introduce the related works in Section 2, including NPE, NPDE, SNPE, SPP and DSNPE. In Section 3, SSNPE is discussed in details. In Section 4, the experimental results are reported to evaluate the performance of SSNPE, with conclusions presented in Section 5.

2 Related works

In this section, we review some related works about NPE and SPP.

2.1 Neighbourhood preserving embedding

Suppose we have a training sample matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n]$, where $\boldsymbol{x}_i \in R^m$ is the ith column of $\boldsymbol{X}$, m and n are the dimensionality and the number of training samples, respectively. As an unsupervised learning method, NPE aims to reduce the dimensionality of data, meanwhile, maintain the inherent local neighbour manifold structure. The optimal transformation matrix $\boldsymbol{A} \in R^{m \times d}$ in NPE is responsible to map the data with high dimension into a feature subspace with relatively low dimension.

Like LLE, NPE uses a local least squares approximation to evaluate the affinity weight matrix $\boldsymbol{W}$. The local approximation error for NPE is given by the following cost function:

$$\varphi(\boldsymbol{W}) = \sum_{i=1}^{n} \| \boldsymbol{x}_i - \sum_{j=1}^{n} W_{ij} \boldsymbol{x}_j \|^2 \tag{1}$$

under two constraints: (1) $W_{ij} = 0$, if the k neighbours of $\boldsymbol{x}_i$ do not include $\boldsymbol{x}_j$; otherwise, $W_{ij} \neq 0$; (2) $\sum_{j=1}^{n} W_{ij} = 1$, $j = 1, 2, \ldots, n$.

To obtain a good projection, a reasonable criterion is to minimise the following cost function:

$$\begin{aligned} \boldsymbol{A} &= \arg\min_{\boldsymbol{A}} \sum_{i=1}^{n} \| \boldsymbol{y}_i - \sum_{j=1}^{n} W_{ij} \boldsymbol{y}_j \|^2 \\ &= \arg\min_{\boldsymbol{A}} \operatorname{tr}(\boldsymbol{A}^{\mathrm{T}} \boldsymbol{X} \boldsymbol{M} \boldsymbol{X}^{\mathrm{T}} \boldsymbol{A}) \end{aligned} \tag{2}$$

which subjects to $\boldsymbol{Y} = \boldsymbol{A}^{\mathrm{T}} \boldsymbol{X}$ and $\boldsymbol{A}^{\mathrm{T}} \boldsymbol{X} \boldsymbol{X}^{\mathrm{T}} \boldsymbol{A} = \boldsymbol{I}$, where $\operatorname{tr}(\cdot)$ is the trace of matrix, $\boldsymbol{I}$ is the identity matrix, and $\boldsymbol{M} = (\boldsymbol{I} - \boldsymbol{W})^{\mathrm{T}} (\boldsymbol{I} - \boldsymbol{W})$. NPE needs to address the following generalised eigenvector problem after employing the Lagrange multiplier:

$$\boldsymbol{X} \boldsymbol{M} \boldsymbol{X}^{\mathrm{T}} \boldsymbol{A} = \lambda \boldsymbol{X} \boldsymbol{X}^{\mathrm{T}} \boldsymbol{A} \tag{3}$$

2.2 Neighbourhood preserving discriminant embedding

Since NPE is unable to employ the label information, Han *et al.* [15] proposed NPDE, which can keep the local neighbourhood structure on data manifold by minimising a local neighbourhood reconstruction error, and meanwhile emphasise the discriminate information of data by minimising the ratio of the within-class scatter to the between-class scatter in the subspace. Similarly, NPDE also involves the matrix decomposition problem.

Suppose we have the training example matrix $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_C]$ in an original subspace, where C represents the number of sample categories and $\boldsymbol{X}_c$ is the sample sub-matrix belonging to class c whose number is n_c. Then, the number of training examples can be denoted as $n = \sum_{c=1}^{C} n_c$. Suppose the projected matrix of the training examples in the subspace with low dimension is $\boldsymbol{Y} = [\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_C]$, then the objective function of NPDE can be denoted as

$$\min \frac{\sum_{c=1}^{C} \sum_{i=1}^{n_c} [\boldsymbol{y}_i^c - \sum_{j=1}^{n_c} W_{ij}^c \boldsymbol{y}_j^c]}{\sum_{c=1}^{C} n_c (\boldsymbol{u}_c - \boldsymbol{u})(\boldsymbol{u}_c - \boldsymbol{u})^{\mathrm{T}}} \tag{4}$$

where $\boldsymbol{y}_i^c$ represents the ith embedded vectors with label c, W_{ij}^c is the reconstruction weighting coefficient of training examples with label c, $\boldsymbol{u}_c$ indicates the average of embedded vectors with label c, and $\boldsymbol{u}$ represents the average of all embedded vectors.

2.3 Supervised neighborhood preserving embedding

Bao *et al.* presented SNPE as a supervised extension. SPNE constructs attraction vectors and makes the embedded points be around them. SPNE uses the label information to construct an attractor $\boldsymbol{h}_i \in R^C$ for a given sample $\boldsymbol{x}_i$. If this sample $\boldsymbol{x}_i$ belongs to class c, then the cth entry of $\boldsymbol{h}_i$ is set to be 1 and other entries are zero. That is to say, the examples with the same category share the same attractor. In other words, the number of attractors is the same as that of training examples, and the number of attractor types is the same as that of classes. SNPE assumes that each example in the subspace should be attracted to its attractor, namely

$$\min \sum_{i=1}^{n} \| \boldsymbol{y}_i - \boldsymbol{h}_i \|^2 \tag{5}$$

where $\boldsymbol{y}_i = \boldsymbol{A}^{\mathrm{T}} \boldsymbol{x}_i$ is the embedded point of $\boldsymbol{x}_i$ in the low-dimensional subspace, and $\boldsymbol{A} \in R^{m \times C}$ is the projection matrix. Note that the dimension of subspace can be determined by the number of classes, or C. Then, SPNE has the following optimisation problem:

$$\min_{\boldsymbol{A}} \frac{1}{2} \sum_{i=1}^{n} \| \boldsymbol{y}_i - \sum_{j=1}^{k} \boldsymbol{W}_{ij} \boldsymbol{y}_j \|^2 + \beta \sum_{i=1}^{n} \| \boldsymbol{y}_i - \boldsymbol{h}_i \|^2 \tag{6}$$

where $\beta > 0$ is a control parameter.

2.4 Sparse representation and SPP

Due to good classification performance and robustness properties, sparse representation [19, 20] is recently a hot topic in machine learning. Sparse reconstruction problem is one of important issues in compressed sensing. Actually, sparse reconstruction problem is how to find a sparse solution to an underdetermined system of equations. Sparse representation has a compact mathematical expression.

Given a training sample matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n]$ with $\boldsymbol{x}_i \in R^m$, sparse representation aims to represent some sample $\boldsymbol{x}_i$ using as few entries of $\boldsymbol{X}$ as possible. The sparse coefficient vector $\boldsymbol{s}_i \in R^n$ is obtained by optimising the following problem:

$$\begin{aligned} \min_{\boldsymbol{s}_i} \quad & \| \boldsymbol{s}_i \|_1 \\ \text{s.t} \quad & \| \boldsymbol{x}_i - \boldsymbol{X} \boldsymbol{s}_i \| < \varepsilon \\ & \boldsymbol{e}^{\mathrm{T}} \boldsymbol{s}_i = 1 \end{aligned} \tag{7}$$

where $\| \cdot \|_1$ is the l_1 norm, $\boldsymbol{e}$ denotes the vector of ones, and $\varepsilon > 0$ is an admissible error. Since the l_0 norm regularisation is so discontinuous, the optimisation to the objective function is very difficult. As an approximation of the l_0 norm regularisation, the l_1 norm regularisation can also induce sparseness and is segment-wise differentiable [21]. The l_1 norm regularisation is widely used to recover a compressible signal in compressed sensing [22, 23]. Some typical methods using the l_1 norm regularisation include orthogonal match pursuit (OMP) [24, 25], least absolute shrinkage and selection operator (LASSO) [26] and gradient projection for sparse reconstruction [27].

Inspired by sparse representation, Qiao *et al.* [14] proposed SPP. SPP seeks a sparse reconstructive weight $\boldsymbol{s}_i$ for each $\boldsymbol{x}_i$. Given the sparse representation weight matrix $\boldsymbol{S} = [\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_n]$, the objective function of SPP can be defined as

$$\min_{\boldsymbol{A}} \operatorname{tr}\left(\sum_{i=1}^{n} \| \boldsymbol{A}^{\mathrm{T}}\boldsymbol{x}_i - \boldsymbol{A}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{s}_i \|^2 \right) \tag{8}$$

which can be rewritten as

$$\begin{aligned} &\min_{\boldsymbol{A}} \quad \operatorname{tr}\left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{S}_{\alpha}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{A}\right) \\ &\text{s.t.} \quad \boldsymbol{A}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{A} = \boldsymbol{I} \end{aligned} \tag{9}$$

where $\boldsymbol{S}_{\alpha} = \boldsymbol{I} - \boldsymbol{S} - \boldsymbol{S}^{\mathrm{T}} + \boldsymbol{S}^{\mathrm{T}}\boldsymbol{S}$, and $\boldsymbol{X}\boldsymbol{S}_{\alpha}\boldsymbol{X}^{\mathrm{T}}$ is both symmetric and positive semi-definite.

2.5 Discriminant sparse NPE

DSNPE tries to introduce the discriminant information into SPP. It considers both MMC and sparsity criterion to map the high-dimensional data into a subspace with low dimension. MMC maximises the margin between classes after reducing dimension [18]. DSNPE considers two distinct sets of sparse reconstruction weights that are computed from the face data of the same and different persons.

Based on SPP above, DSNPE is equal to address the following multi-object optimisation problem:

$$\begin{aligned} &\begin{cases} \min\limits_{\boldsymbol{A}} & \operatorname{tr}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{S}_{\alpha}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{A}) \\ \max\limits_{\boldsymbol{A}} & \operatorname{tr}(\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{S}_{\mathrm{b}} - \boldsymbol{S}_{\mathrm{w}})\boldsymbol{A}) \end{cases} \\ &\text{s.t.} \quad \boldsymbol{A}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{A} = \boldsymbol{I} \end{aligned} \tag{10}$$

where the matrix $\boldsymbol{S}_{\mathrm{b}}$ and $\boldsymbol{S}_{\mathrm{w}}$ are the between-class scatter and the within-class scatter matrix, respectively.

3 Supervised sparse neighbourhood preserving embedded

In this section, we propose an alternative method called SSNPE to utilise the label information of the given data.

3.1 Modified SPP

Although SPP [14] chooses its neighbourhood automatically instead of using the k nearest neighbour algorithm, the number of neighbourhoods is determined by the admissible error ϵ. In addition, the global structure in SPP may result in the loss of local linear structure. Leng *et al.* [28] proposed an incremental LLE algorithm-based OMP. In this method, OMP is used to find the sparse representation coefficients of k nearest neighbours instead of all training samples.

To avoid the loss of local linear structure, we introduce OMP to solve the following optimisation problem:

$$\begin{aligned} &\min_{\boldsymbol{s}_i'} \quad \| \boldsymbol{s}_i' \|_0 \\ &\text{s.t.} \quad \boldsymbol{x}_i = \boldsymbol{X}_i\boldsymbol{s}_i' \end{aligned} \tag{11}$$

where $\boldsymbol{X}_i \in R^{m\times k}$ is the neighbour sample matrix of $\boldsymbol{x}_i$, and $\boldsymbol{s}_i'$ is the weight vector corresponding to these neighbours. Given a sparse degree p, $\boldsymbol{s}_i'$ would contain p non-zero elements. Thus, $p \le k$ in the modified SPP. We can expand $\boldsymbol{s}_i'$ into $\boldsymbol{s}_i$ with zero, where $\boldsymbol{s}_i$ is the sparse representation weight vector corresponding to all training samples.

3.2 Objective of SSNPE

SSNPE is an extension of SPNE by incorporating SPP into SNPE. Let the set of training examples be $\{\boldsymbol{x}_i, v_i\}_{i=1}^{n}$, where $\boldsymbol{x}_i \in R^m$ and $v_i \in \{1, 2, \ldots, C\}$. The training example matrix can be denoted as $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \in R^{m\times n}$. Let $\boldsymbol{y}_i = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{x}_i$ be the embedded point of $\boldsymbol{x}_i$, and $\boldsymbol{A} \in R^{m\times C}$ be the projection matrix. The objective function of SSNPE is

$$\begin{aligned} &\min_{\boldsymbol{A}} \frac{1}{2}\sum_{i=1}^{n} \| \boldsymbol{y}_i - \sum_{j=1}^{n} \left(\alpha S_{ij} + (1-\alpha)W_{ij}\right)\boldsymbol{y}_j \|^2 \\ &+ \beta \sum_{i=1}^{n} \| \boldsymbol{y}_i - \boldsymbol{h}_i \|^2 \end{aligned} \tag{12}$$

where $\alpha \in [0, 1]$ is a balance parameter, $\beta > 0$ is a hyper-parameter that controls the weight of the label information, $\boldsymbol{h}_i$ is the attractor for $\boldsymbol{x}_i$, $\boldsymbol{W}$ is the neighbourhood weight matrix which can be found by solving (1) and $\boldsymbol{S}$ is the sparse representation weight matrix which can be found by solving (11).

The first term in (12) can be taken as the combination of NPE and SPP and can be rewritten in matrix form:

$$\begin{aligned} &\frac{1}{2}\sum_{i=1}^{n} \| \boldsymbol{y}_i - \sum_{j=1}^{n} \left(\alpha S_{ij} + (1-\alpha)W_{ij}\right)\boldsymbol{y}_j \|^2 = \boldsymbol{Y}\boldsymbol{M}\boldsymbol{Y} \\ &= \boldsymbol{A}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{A} \end{aligned} \tag{13}$$

where $\boldsymbol{M} = (\boldsymbol{I} - (\alpha\boldsymbol{S} + (1-\alpha)\boldsymbol{W}))^{\mathrm{T}}(\boldsymbol{I} - (\alpha\boldsymbol{S} + (1-\alpha)\boldsymbol{W}))$ and $\boldsymbol{Y} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{X}$ denotes the examples in the subspace. When $\alpha = 1$, the optimisation problem (13) is the modified SPP. If $\alpha = 0$, (13) is the optimisation problem of NPE.

The second term in (12) is related to attractors and can be represented in matrix form:

$$\begin{aligned} &\beta\sum_{i=1}^{n} \| \boldsymbol{y}_i - \boldsymbol{h}_i \|^2 = \beta(\boldsymbol{Y} - \boldsymbol{H})(\boldsymbol{Y} - \boldsymbol{H})^{\mathrm{T}} = \beta\left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{X} - \boldsymbol{H}\right. \\ &\left.\right)\left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{X} - \boldsymbol{H}\right)^{\mathrm{T}} \end{aligned} \tag{14}$$

Substituting (13) and (14) into (12), we have

$$\min_{\boldsymbol{A}} \operatorname{tr}\left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{A} + \beta\left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{X} - \boldsymbol{H}\right)\left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{X} - \boldsymbol{H}\right)^{\mathrm{T}}\right) \tag{15}$$

3.3 Solution to SSNPE

We design the objective function (15) for SSNPE, now we should find a solution to (15). Similar to SNPE [16], we have the following theorem.

Theorem 1: Suppose that for the optimisation problem (15) there exists a symmetric semi-positive definite matrix $\boldsymbol{M} \in R^{n\times n}$, a real matrix $\boldsymbol{H} \in R^{C\times n}$, a full rank matrix $\boldsymbol{X} \in R^{m\times n}$ and a positive constant β such that

$$\boldsymbol{A} = \beta(\boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^{\mathrm{T}} + \beta\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})^{-\mathrm{T}}\boldsymbol{X}\boldsymbol{H}^{\mathrm{T}} \tag{16}$$

is the solution to (15).

The proof of Theorem 1 is given in Appendix. Theorem 1 translates the optimisation problem (15) into a much easier linear set of equations. SSNPE is summarised in Algorithm 1 in detail.

3.4 Comparison with relative works

Table 1 lists the method and the corresponding balance parameters α and β. According to (12), we can get six methods, including three supervised methods and three unsupervised method. In addition,

SSNPE can be thought as a framework of supervised methods. Obviously, SSNPE is an extension to SPP and NPE.

3.4.1 NPE and SPP: SSNPE is a supervised dimensionality reduction method. In (12), SSNPE would be an unsupervised method if $\beta = 0$. In this unsupervised case, we can see that SSNPE is a hybrid of NPE and SPP. As mentioned above, SPP has a similar objective function to NPE, but they construct the weight matrix in a completely different manner. The weight matrix $\boldsymbol{S}$ in SPP is called sparse representation weight matrix, and that $\boldsymbol{W}$ in NPE called neighbourhood weight matrix.

In SSNPE, the two weight matrices are linearly combined. Since we use a local way to find $\boldsymbol{S}$ as (11), the weight of useful neighbours would be enhanced in the way of $(\alpha S_{ij} + (1-\alpha)W_{ij})$. In the unsupervised case, SSNPE will degenerate into unsupervised NPE algorithm when $\alpha = 0$. Obviously, SSNPE will degenerate into the modified SPP algorithm when $\alpha = 1$. SSNPE inherits the merits of NPE and SPP, which can preserve not only the local neighbourhood information, but also the sparse locally reconstructive relationship of the data.

3.4.2 Connections to supervised NPE: Supervised versions for NPE and SPP have been proposed, or NPDE [15], DSNPE [17] and SNPE [16]. Here, we compare SSNPE with them.

NPDE tries to not only preserve the local neighbourhood structure but also focus on the discriminate information of data. NPDE is able to minimise the local neighbourhood reconstruction error, meanwhile, maintain points with minimum within-class scatter and maximum between-scatter in the subspace, which embodies the discrimination information.

DSNPE considers both MMC and sparsity criterion to map high-dimensional data into a subspace with low dimension. As a result, DSNPE is both robust as sparse representation and distinctive as MMC. Here, the discrimination information is obtained by MMC.

Since SSNPE is proposed based on SNPE, SSNPE inherits its characteristics. In both SSNPE and SNPE, the discrimination information is reflected by attractors. Both methods employ the label information to construct attractors and make the samples in the subspace drawn to these attractors. In addition, the dimensionality of subspace is determined by the class number. In other words, both methods can determine the dimensionality of subspace in advance. However, the local sparse reconstructive relationship of data in SSPNE makes the difference between them.

Algorithm 1: Supervised sparse neighbourhood preserving embedding

Input: Training samples $\{\boldsymbol{x}_i, v_i\}_{i=1}^n$, the neighbour number k, and the sparse degree p, the balance parameters α and β.

Output: Projection matrix $\boldsymbol{A}$.

Step 1: Construct the attractor matrix $\boldsymbol{H}$. For each $\boldsymbol{x}_i$, construct a vector $\boldsymbol{h}_i$ in which only one element takes one, and the others take zero. If $v_i = c$, then the cth element in $\boldsymbol{h}_i$ is 1 and other elements are zero. Let $\boldsymbol{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_n] \in R^{C \times n}$.

Step 2: Generate the neighbourhood weight matrix $\boldsymbol{W}$. First, find the k nearest neighbours for $\boldsymbol{x}_i$, and denote them by $\boldsymbol{x}_i^j$, $j = 1, \ldots, k$. Then, solve the following problem to get the matrix $\boldsymbol{W}' \in R^{n \times k}$. Namely,

$$\min_{\boldsymbol{W}'} \quad \sum_{i=1}^{n} \| \boldsymbol{x}_i - \sum_{j=1}^{k} W'_{ij} \boldsymbol{x}_i^j \|^2$$

$$\text{s.t.} \quad \sum_{j=1}^{k} W'_{ij} = 1$$

Expand $\boldsymbol{W}'$ to obtain the neighbourhood weight matrix $\boldsymbol{W}$.

Step 3: Generate the sparse representation weight matrix $\boldsymbol{S}$. Given the k nearest neighbour matrix $\boldsymbol{X}_i = [\boldsymbol{x}_i^1, \ldots, \boldsymbol{x}_i^k]$ for $\boldsymbol{x}_i$, solve the following problem to get the matrix $\boldsymbol{S}' \in R^{n \times k}$. Namely,

$$\min_{\boldsymbol{s}_i'} \quad \| \boldsymbol{s}_i' \|_0$$

$$\text{s.t.} \quad \boldsymbol{x}_i = \boldsymbol{X}_i \boldsymbol{s}_i', \boldsymbol{e}^{\mathrm{T}} \boldsymbol{s}_i' = 1$$

Expand $\boldsymbol{S}'$ to obtain the sparse representation weight matrix $\boldsymbol{S}$.

Step 4: Compute the projection matrix $\boldsymbol{A} = \beta(\boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^{\mathrm{T}} + \beta\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})^{-\mathrm{T}}\boldsymbol{X}\boldsymbol{H}^{\mathrm{T}}$, where $\boldsymbol{M} = (\boldsymbol{I} - (\alpha\boldsymbol{S} + (1-\alpha)\boldsymbol{W}))^{\mathrm{T}}(\boldsymbol{I} - (\alpha\boldsymbol{S} + (1-\alpha)\boldsymbol{W}))$, and $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n]$.

Table 1 Method and its balance parameters

Method	α	β
SSNPE	$0 \le \alpha \le 1$	$\beta \ne 0$
SNPE/SSNPE	$\alpha = 0$	$\beta \ne 0$
supervised MSPP/SSNPE	$\alpha = 1$	$\beta \ne 0$
NPE + MSPP	$0 < \alpha < 1$	$\beta = 0$
NPE	$\alpha = 0$	$\beta = 0$
MSPP	$\alpha = 1$	$\beta = 0$

4 Experiments

We experimentally validate the effectiveness of SSNPE according to the classification performance and carry out experiments on 13 UCI datasets, the ORL dataset and the MNIST dataset, where experiments on the Iris dataset validate the effect of parameters of SSNPE on its performance. The embedded samples are classified by a nearest neighbour (NN) classifier. All experiments are implemented by MATLAB.

4.1 Parameter analysis

In this section, we use the famous dataset, Iris to show the effect of parameters in SSNPE on its performance. The Iris dataset from the UCI machine-learning repository [29] contains three classes of 50 instances each, where each class refers to a type of Iris plant. Each instance has four features. All features are normalised into the interval [0, 1]. We randomly separate the Iris dataset into two subsets and repeat 10 times. One subset contains 2/3 samples for training and the other consists of the rest samples for test.

SSNPE involves two control parameters α and β, the neighbourhood parameter k, and the sparse degree p. These parameters have an effect on the classification performance theoretically.

First, we check the effect of the parameter β on the performance of SSNPE when setting the neighbourhood parameter $\alpha = 0.5$, $k = 10$ and $p = 5$. The control parameter β varies in the set $\{10^{-4}, 10^{-3}, \ldots, 10^{4}\}$. Experiments are independently repeated for 10 times with the average accuracy shown in Fig. 1*a*. As we can see, the accuracy remains increasing with β increasing. When $\beta \ge 1$, the curve keeps flat gradually. In addition, the label information must be overemphasised if β is too large. In this case, the local structure may be destroyed. To avoid this situation, we just set $\beta = 1$ in the experiments so as to make the tradeoff between the original manifold geometry and the label information of training examples.

Then we check the effect of the parameter α on the performance of SSNPE when setting $k = 10$ and $p = 5$. Let the control parameter α change in the set $\{0, 0.1, \ldots, 0.9, 1\}$. Experiments are independently repeated for 10 times with the average accuracy shown in Fig. 1*b*, where four curves are plotted for different β. When $\beta = 0.001$, the classification performance of SSNPE significantly varies. For $\beta \ge 1$, the curves are relative flat. Therefore, to make the tradeoff between sparse reconstruction weighting coefficient and neighbourhood reconstruction weighting coefficient, $\alpha = 0.5$ is an ideal choice in the experiment.

Finally, we observe the effect of the neighbourhood parameter k and the sparse degree p. Here, $\beta = 1$ and $\alpha = 0.5$. Since $p \le k$, we observe the relationship between them. Let $p = p_i$ where $p_i = \lceil k/i \rceil$, $i = 1, 2, 3, 4, 5$ and $\lceil \cdot \rceil$ rounds the element to the nearest integers towards infinity. The parameter k changes in the

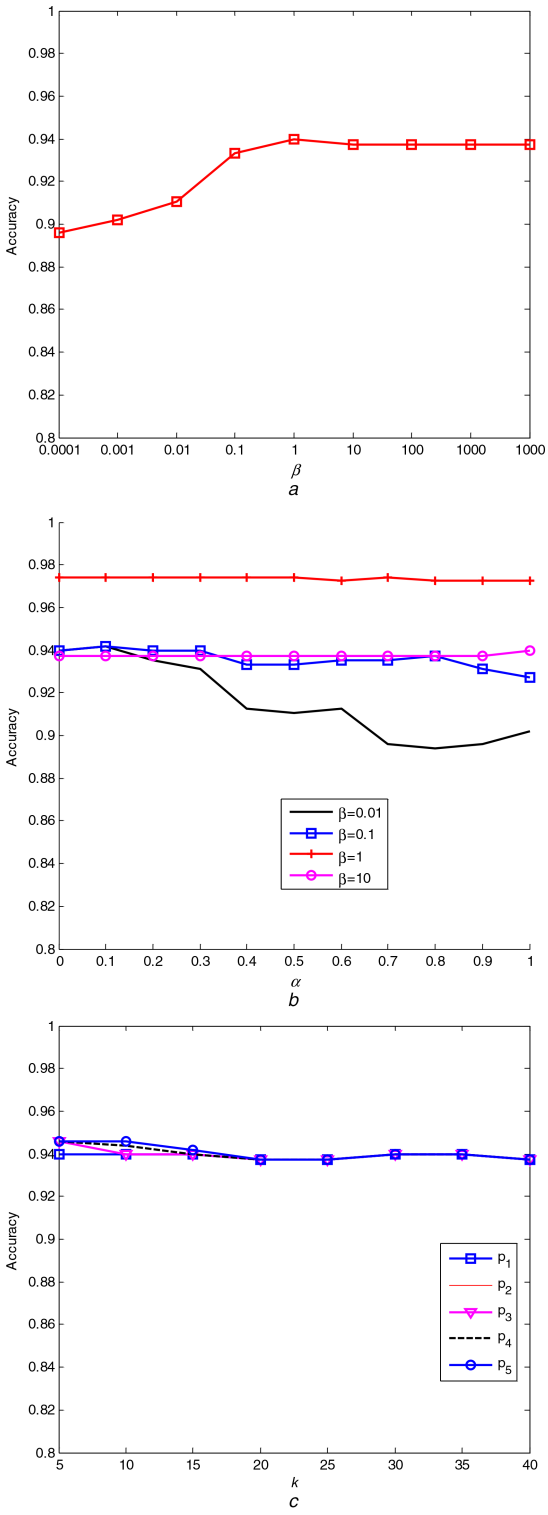

Fig. 1 *Effect of parameters on performance in SSNPE*
(a) Accuracy vs. control parameter β, ***(b)*** Accuracy vs. control parameter α, ***(c)*** Accuracy vs. neighbourhood size k

set $\{5, 10, 15, 20, 25, 30, 35, 40\}$. Fig. 1*c* shows the results under different neighbourhood sizes, where the results of $p_i = \lceil k/2 \rceil$ is exact identical with that of $p_i = \lceil k/3 \rceil$ in this experiment. When k is small, the sparse degree has an important effect on the classification performance. If k is large enough, five curves are overlapped. Obviously, we can make $p = \lceil k/5 \rceil$.

4.2 Experiments on UCI datasets

Thirteen UCI datasets are selected for validating the performance of SSNPE, including the Iris data set [29]. The descriptions on these data sets are summarised in Table 2. In addition, the features in each dataset are normalised to the interval [0, 1]. For each dataset, we run 10 trials where the training set contains 2/3 of the samples (randomly selected) of each class, and the test set contains the remaining 1/3.

The neighbourhood size k varies in the set $\{5, 10, \ldots, 35, 40\}$ for NPE, SPP, NPDE, DSNPE and SSNPE. Let the sparse degree be $p = \lceil k/5 \rceil$, $\beta = 1$ and $\alpha = 0.5$ in SSNPE. NN is used to classify embedded samples obtained by these feature extraction methods. In addition, we also compare the proposed method with SRC. Let the admissible error in SRC be 0.001 followed the setting in [21]. The reduced dimension for each data is its class number for all compared methods. PCA is taken as the dimension reduction method in SRC.

We report the best average results on 10 trials under different neighbourhood sizes in Table 3. The corresponding neighbourhood size k is shown in Table 4 for all compared methods except SRC. For comparison, we conduct the paired two-tailed t-test with the 0.05 significant level to determine whether there is a significant difference between SSNPE and other methods. A win-loss-tie (W-L-T) summarisation based on the mean and t-test is also attached at the bottom of Table 3, where a win/loss means that the method compared with SSNPE is better/worse than SSNPE on a dataset.

According to Table 3, we have the following conclusions. In the view of average accuracy, SSNPE achieves the best accuracy in ten out of 13 datasets, followed by SNPE. SSNPE is significantly better than both NPE and NPDE in nine out of 13 datasets, SRC in all 13 datasets, SPP in 12 out of 13 datasets, and DSNPE in eight out of 13 datasets. SRC gets the worst performance in these UCI datasets since SRC cannot well classify the data with the same direction distribution [21]. Thus, SSNPE has significant difference compared with four methods (NPE, SPP, NPDE and DSNPE) at least eight datasets. Moreover, SSNPE is significantly better than SNPE on three datasets, Iris, Musk and Wdbc.

4.3 ORL database

The ORL database [30] contains 400 greyscale images of 40 individuals each of which has 10 images with resolution of 92×112. These images were taken at different times, different lighting conditions, different facial expressions and facial accessories (glasses/no glasses). Fig. 2 gives some examples of the ORL database.

We randomly take n_c examples for each class as the training ones, and the rest samples are used as the test ones, where n_c takes value in the set $\{4, 5, 6, 7, 8\}$. Thus, the training sample number is 160, 200, 240, 280 and 320, respectively. For each giving n_c, we repeat 20 times randomly choice of data and present the average accuracy. In this experiment, to address the problem of singular matrices, PCA is applied to reduce dimensionality to 100 and remove the noise so as to avoid the sample matrix to be singular. The final dimensionality of subspace is 40 for all methods.

The neighbourhood size k is taken from the set $\{5, 10, 15, 20\}$ for NPE, SPP, NPDE, DSNPE and SSNPE. Random projection is taken as the dimension reduction method in SRC. Other parameter settings are the same as Section 4.2.

The curves of accuracy vs. training sample number under different neighbourhood sizes are given in Fig. 3. From the experimental results, we can see that the classification accuracy is increased when the training sample number increased for all compared methods. In addition, the variation on the neighbourhood size also has an effect on the classification performance. Generally, NPE, SPP, DSNPE, SNPE and SSNPE have better performance for $k = 10$ or $k = 15$ if the training sample number keeps the same. For NPDE, the best performance is achieved when $k = 20$. The best performance for seven methods is listed in Table 5. When the training sample is not >280, SSNPE outperforms other methods. NPDE is the best one when the training sample is relative large, say 280, and followed by SSNPE.

A part of feature faces obtained by the six feature extraction methods are shown in Fig. 4. Totally, there are 40 feature faces obtained by these methods. For saving space, we only pick up the first 10 feature faces corresponding to the first 10 columns of

projection matrices. The feature faces obtained by NPE are very vague. The supervised NPE methods are better than NPE. Among them, SSNPE is the best from the visual view.

4.4 MNIST database

The MNIST dataset is a database consisting of 0–9 handwritten digit images, which has two subsets for training and test, respectively [31]. The training set contains 60,000 images, and the test 10,000. The size of each digit image is 28×28. We randomly take n_c examples for each class as the training ones, where n_c takes value in the set $\{100, 200, \ldots, 1300\}$. Thus, the training sample number is from 1000 to 13,000, respectively. The whole test set is adopted to test the classification performance of the compared methods. For each giving n_c, we repeat 10 times randomly choice of data and report the average accuracy. To speed training speed, we first use PCA to reduce dimensionality to 100. The final dimensionality of subspace is 10 for all methods. Other parameter settings are the same as Section 4.2.

The curves of accuracy vs. training sample number under different neighbourhood sizes obtained by compared methods are shown in Fig. 4. Note that experimental results could not be obtained by SPP and DSNPE for out of memory when the training sample number is larger. Thus, their curves are incomplete in Fig. 5. In addition, since SPP is much worse than other methods, its curves under both $k = 15$ and $k = 20$ are not given in Figs. 5*c* and *d*. Similarly, the conclusion that the classification accuracy is increased when the training sample number increased for all compared methods holds true on the MNIST dataset. SSNPE outperforms the compared methods under different k.

The best performance selected from four k values for these six methods and from SRC is listed in Table 6. Since SPP can deal with 6000 training samples, we only give the training sample number from 1000 to 6000. The main reason that SRC achieves a bad performance is that the reduced dimension is too small. For SRC, 10 features are not enough. Thus, it is easy to see the superiority of SSNPE.

Table 2 Description of UCI datasets used in experiments

Dataset	#Feature	#Class	#Sample
balance	4	3	625
breast	9	2	699
heart	13	2	303
hepatitis	19	2	155
pima	8	2	768
liver	6	2	345
musk	166	2	476
iris	4	3	150
sonar	60	2	208
vote	16	2	435
Wdbc	30	2	569
Wine	13	3	178
Wpbc	33	2	198

4.5 Statistical comparison over multiple datasets

We conduct experiments on 13 UCI datasets, the ORL and the MNIST datasets and compare SSNPE with other six methods. For statistical comparison, we perform statistical tests on these datasets

Table 3 Average test accuracy and standard deviation for 13 UCI benchmark datasets

Dataset	SRC	NPE	SPP	NPDE	DSNPE	SNPE	SSNPE
balance	69.44 ± 7.13	66.79 ± 13.48	71.44 ± 15.37	49.38 ± 14.10	87.27 ± 1.34	87.61 ± 1.18	**87.66** ± 1.08
breast	34.48 ± 0.35	94.05 ± 1.31	86.90 ± 3.30	93.75 ± 1.32	93.79 ± 1.24	**95.17** ± 1.33	94.91 ± 1.11
heart	49.00 ± 5.66	73.30 ± 4.95	59.20 ± 5.07	71.50 ± 5.10	**74.80** ± 3.19	74.60 ± 3.31	**74.80** ± 2.70
hepatitis	52.35 ± 3.93	60.59 ± 5.66	58.82 ± 10.90	**63.33** ± 56.08	56.08 ± 8.88	53.73 ± 5.56	59.22 ± 6.84
pima	35.05 ± 0.20	68.86 ± 2.56	59.61 ± 3.03	64.47 ± 2.96	62.12 ± 3.49	70.59 ± 2.32	**70.90** ± 2.27
liver	53.07 ± 7.78	57.63 ± 4.53	52.37 ± 2.07	56.32 ± 5.62	56.05 ± 4.23	61.58 ± 5.06	**61.93** ± 4.88
musk	55.45 ± 1.87	64.62 ± 5.06	55.00 ± 5.73	65.76 ± 4.82	73.86 ± 5.24	75.89 ± 2.54	**80.89** ± 1.73
iris	66.46 ± 2.85	91.88 ± 3.02	82.92 ± 9.86	**95.42** ± 2.15	89.79 ± 4.86	93.75 ± 3.11	94.58 ± 3.29
sonar	47.83 ± 2.26	62.17 ± 7.62	56.96 ± 4.53	62.17 ± 6.01	68.55 ± 6.07	71.74 ± 5.93	**73.09** ± 7.46
vote	86.48 ± 2.00	81.45 ± 4.83	72.90 ± 4.80	82.83 ± 8.75	87.24 ± 2.98	**92.00** ± 1.87	**92.00** ± 2.06
Wdbc	45.93 ± 1.47	92.43 ± 1.09	65.77 ± 7.00	91.11 ± 2.16	89.26 ± 2.73	94.29 ± 1.80	**95.61** ± 1.48
Wine	61.72 ± 4.05	91.03 ± 2.67	56.90 ± 7.80	86.72 ± 4.67	96.21 ± 2.67	97.41 ± 1.68	**97.76** ± 1.83
Wpbc	64.77 ± 17.01	69.85 ± 6.08	69.23 ± 6.15	70.46 ± 3.46	71.85 ± 4.65	74.31 ± 3.48	**76.77** ± 4.78
W-L-T (mean)	0-13-0	1-12-0	0-13-0	2-11-0	0-12-1	1-11-1	—
W-L-T (*t*-test)	0-13-0	0-9-4	0-12-1	0-9-4	0-8-5	0-3-10	—

The bold values are the best accuracies among the compared methods.

Table 4 Neighbourhood size corresponding to the best average test accuracy for 13 UCI benchmark datasets

Dataset	NPE	SPP	NPDE	DSNPE	SNPE	SSNPE
balance	5	5	5	5	5	10
breast	40	5	40	10	10	35
heart	5	5	5	5	5	35
hepatitis	30	5	35	15	5	10
pima	35	10	10	5	5	15
liver	5	5	35	5	5	25
musk	35	10	10	5	5	25
iris	15	5	35	5	5	5
sonar	40	15	25	5	5	25
vote	25	10	15	5	5	20
Wdbc	10	20	10	5	5	10
Wine	10	10	5	5	5	5
Wpbc	5	10	5	5	5	20

Fig. 2 *Sample images of the ORL database*

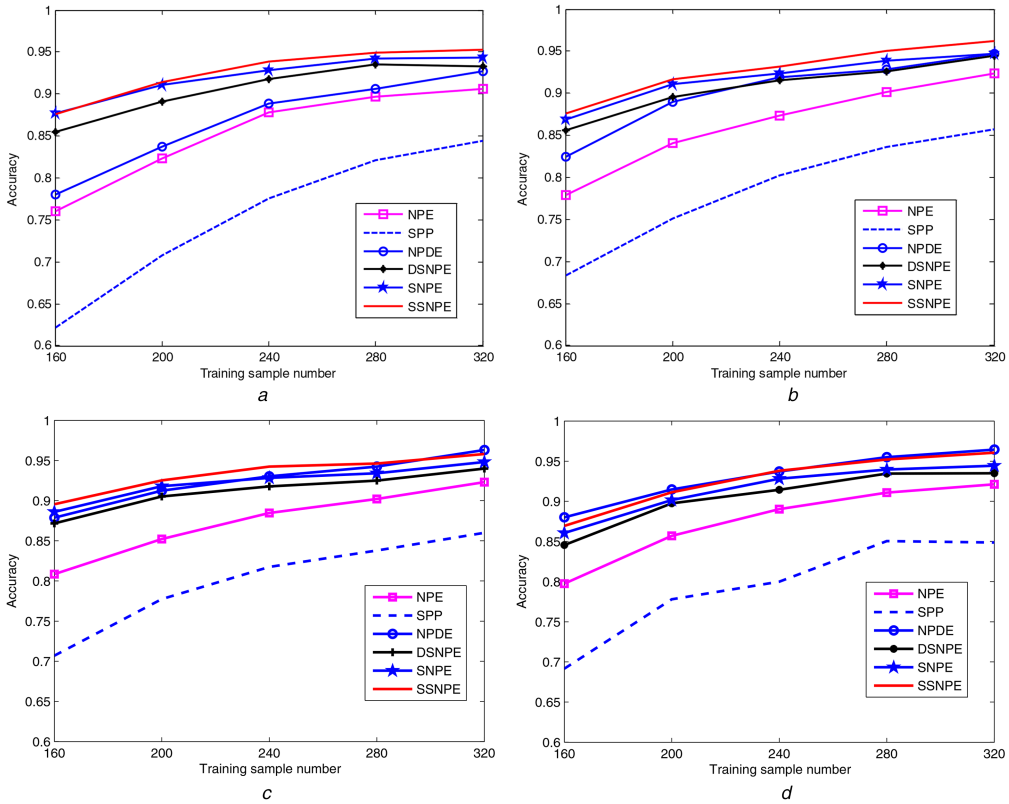

Fig. 3 *Accuracy vs. training sample number under different neighbourhood sizes on the ORL dataset*
(a) $k = 5$, ***(b)*** $k = 10$, ***(c)*** $k = 15$, ***(d)*** $k = 20$

for multiple methods. The Friedman test [32] with the corresponding post hoc tests is used to test whether all the methods are equivalent. The Friedman test has the null hypothesis that all the methods are equivalent [33]. Thus, the ranks of all methods should be equal to each other. If the test result rejects the null hypothesis, then we proceed to a post hoc test. The Bonferroni–Dunn test is adopted as post hoc tests when methods in our experiments are compared to SSNPE. The performance of pairwise methods is significantly different if their corresponding average ranks differ by at least the critical difference [33]:

$$CD = q_{\alpha}\sqrt{\frac{j(j+1)}{6T}}$$

where j is the number of methods, T is the number of datasets, the critical values q_{α} can be found in [34], and the subscript α is the threshold value. Generally, let $\alpha = 0.1$ [34]. Here, we have $j = 7$, $q_{0.10} = 2.128$ and $T = 23$, where the ORL dataset is taken as five datasets for five different training numbers, and the MNIST dataset is six. Thus, we get $CD = 1.3556$.

Table 7 lists the mean rank of all methods used in experiments. In our experiments, the p-value for Friedman's χ^2 statistic is

Table 5 Average accuracies obtained by seven methods on the ORL dataset

#Training	SRC	NPE	SPP	NPDE	DSNPE	SNPE	SSNPE
160	87.15 ± 2.05	80.85 ± 2.65	70.71 ± 2.20	88.00 ± 2.28	87.17 ± 2.36	88.60 ± 1.96	**89.56** ± 2.20
200	89.13 ± 3.85	85.70 ± 1.97	77.80 ± 2.75	91.47 ± 2.19	90.53 ± 2.10	91.80 ± 2.11	**92.53** ± 1.71
240	90.84 ± 3.11	89.03 ± 2.75	81.75 ± 2.94	93.72 ± 1.94	91.78 ± 2.08	92.84 ± 1.67	**94.25** ± 1.47
280	90.67 ± 3.69	91.08 ± 2.25	85.04 ± 1.92	**95.50** ± 1.56	93.54 ± 2.00	94.25 ± 1.29	95.21 ± 2.02
320	91.19 ± 4.28	92.31 ± 2.76	86.00 ± 4.21	**96.44** ± 1.83	94.50 ± 2.73	94.81 ± 2.85	96.19 ± 1.70

The bold values are the best accuracies among the compared methods.

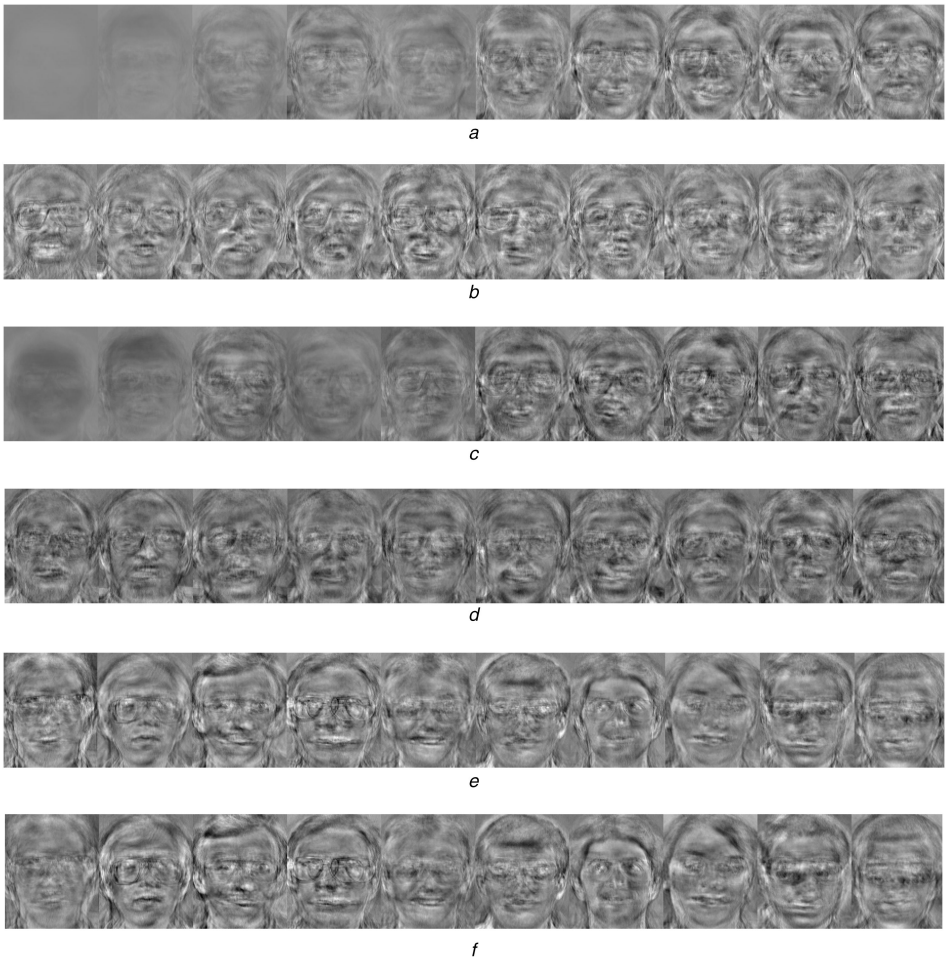

Fig. 4 *The first ten feature faces obtained by the feature extraction methods*
(a) NPE, ***(b)*** SPP, ***(c)*** NPDE, ***(d)*** DSNPE, ***(e)*** SNPE and ***(f)*** SSNPE

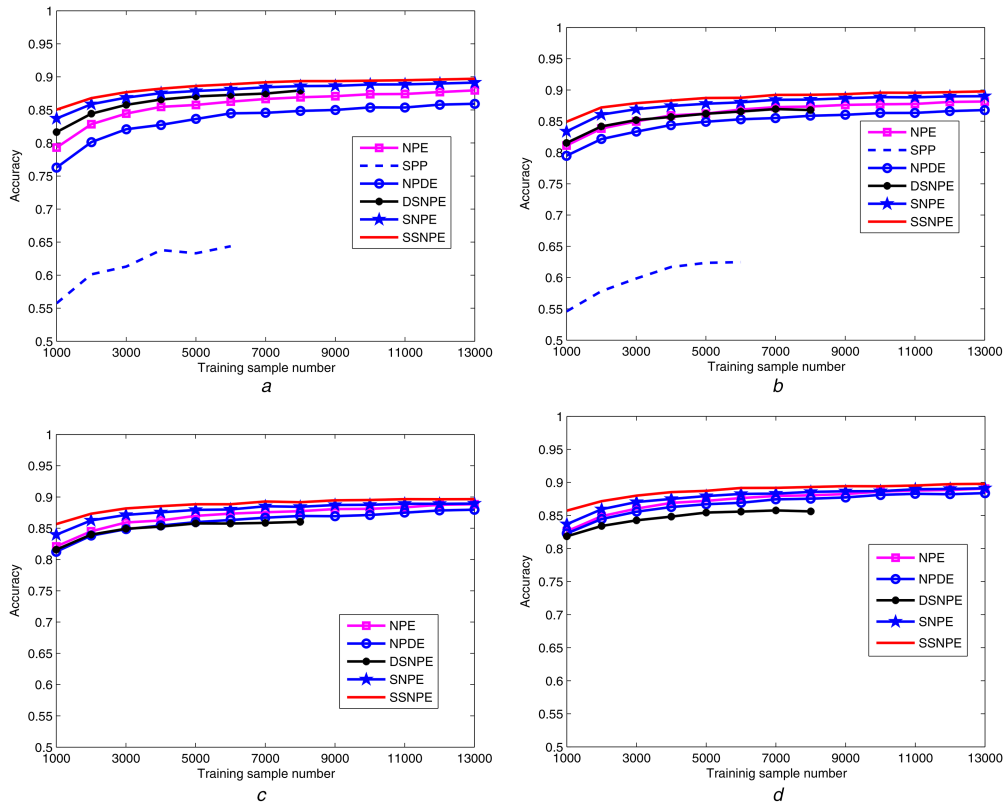

Fig. 5 *Accuracy vs. training sample number under different neighbourhood sizes on the MNIST dataset*
(a) $k = 5$, ***(b)*** $k = 10$, ***(c)*** $k = 15$, ***(d)*** $k = 20$

Table 6 Average accuracies obtained by six methods on the MNIST dataset

#Training	SRC	NPE	SPP	NPDE	DSNPE	SNPE	SSNPE
1000	60.47 ± 3.69	82.62 ± 0.93	55.74 ± 1.37	82.28 ± 0.85	81.86 ± 0.40	83.97 ± 0.38	**85.74** ± 0.39
2000	60.40 ± 5.74	84.89 ± 0.22	60.13 ± 2.16	84.47 ± 0.28	84.43 ± 0.64	86.25 ± 0.38	**87.33** ± 0.34
3000	57.16 ± 4.41	86.05 ± 0.55	61.30 ± 0.93	85.58 ± 0.36	85.76 ± 0.26	87.14 ± 0.26	**88.17** ± 0.28
4000	62.41 ± 6.92	86.92 ± 0.64	63.79 ± 1.18	86.28 ± 0.71	86.57 ± 0.26	87.54 ± 0.51	**88.54** ± 0.29
5000	58.71 ± 4.55	87.22 ± 0.33	63.79 ± 1.18	86.70 ± 0.47	87.03 ± 0.49	87.94 ± 0.35	**88.84** ± 0.38
6000	57.72 ± 6.03	87.63 ± 0.46	64.38 ± 1.85	86.94 ± 0.48	87.27 ± 0.42	88.27 ± 0.20	**89.17** ± 0.26

The bold values are the best accuracies among the compared methods.

Table 7 Statistical comparison of seven methods

	SRC	NPE	SPP	NPDE	DSNPE	SNPE	SSNPE
mean rank	6.2609	4.1522	6.1739	3.8043	3.9783	2.3261	1.3043
rank difference	4.9565	2.8478	4.8696	2.5000	2.6739	1.0217	0

4.82×10^{-19}, which is very small. Thus, we should reject the null hypothesis that all methods are equivalent for each method in Table 7. The rank difference (1.0217) between SSNPE and SNPE is smaller than the critical difference (1.3556), which means that we could not detect any significant difference between them. The rank differences between SSNPE and other methods including SRC, NPE, SPP, NPDE and DSNPE are greater than the critical difference. Thus, SSNPE is significantly better than these five methods in this current experimental setting.

5 Conclusion

In this paper, we develop a novel supervised dimension reduction technique named supervised sparse neighbourhood preserving embedded algorithm (SSNPE). The idea behind SSNPE is to simultaneously combine attractors constructed by discriminate information, sparse structure and neighbourhood structure. SSNPE requires two weight matrices, or the neighbourhood and the sparse representation weight ones. Given the two weight matrices, SSNPE can be cast into a set of linear equations. The effectiveness of SSNPE is verified by comparing with several related algorithm including NPE, SPP, NPDE, DSNPE and SNPE on some publicly available datasets, which further illustrates discriminate information is valuable to improving the performance of classification. Experimental results on 13 UCI datasets, ORL face image dataset and handwritten digit image datasets show SSNPE performs better than the compared methods in most cases. Statistical tests also indicate that SSNPE is significant better than SRC, NPE, SPP, NPDE and DSNPE.

To improve the classification performance, we will extend SSNPE into its non-linear version, including kernel version.

6 Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant no. 61373093, by the Natural Science Foundation of Jiangsu Province of China under Grant nos. BK20140008 and BK2012624, by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant no. 13KJA520001, and by the Soochow Scholar Project.

7 References

[1] Wang, H.: 'Structured sparse linear graph embedding', *Neural Netw.*, 2012, **27**, (3), pp. 38–44
[2] Kambhatla, N., Leen, T.K.: 'Fast nonlinear dimension reduction'. Proc. of IEEE Int. Conf. on Neural Networks, 1993, pp. 1213–1218
[3] Gui, J., Jia, W., Zhu, L., *et al.*: 'Locality preserving discriminant projections for face and palmprint recognition', *Neurocomputing*, 2010, **73**, (13–15), pp. 2696–2707
[4] Gui, J., Tao, D., Sun, Z., *et al.*: 'Group sparse multiview patch alignment framework with view consistency for image classification', *IEEE Trans. Image Process.*, 2014, **23**, (7), pp. 3126–3137
[5] Gui, J., Wang, S.L., Lei, Y.K.: 'Multi-step dimensionality reduction and semi-supervised graph-based tumor classification using gene expression data', *Artif. Intell. Med.*, 2010, **50**, (3), pp. 181–191
[6] Jolliffe, I.T.: '*Principal component analysis*' (Springer-Verlag, New York, 1986)
[7] Lu, B.W., Pandolfo, L.: 'Quasi-objective nonlinear principal component analysis', *Neural Netw.*, 2011, **24**, (2), pp. 159–170
[8] Duda, R.O., Hart, P.E., Stork, D.G.: '*Pattern classification*' (Wiley, New York, 2001)
[9] Tenenbaum, J., de Silva, V., Langford, J.: 'A global geometric framework for nonlinear dimensionality reduction', *Science*, 2000, **290**, (5500), pp. 2319–2323
[10] Roweis, S.T., Saul, L.K.: 'Nonlinear dimensionality reduction by locally linear embedding', *Science*, 2000, **290**, (5500), pp. 2323–2326
[11] Belkin, M., Niyogi, P.: 'Laplacian eigenmaps for dimensionality reduction and data representation', *Neural Comput.*, 2003, **15**, (6), pp. 1373–1396
[12] He, X.F., Partha, N.: 'Locality preserving projections'. Proc. of the 17th Annual Conf. on Neural Information Processing Systems, Vancouver, 2003, pp. 153–160
[13] Qiao, L., Chen, S., Tan, X.: 'Sparsity preserving projections with applications to face recognition', *Pattern Recogn.*, 2010, **43**, (1), pp. 331–341
[14] He, X.F., Cai, D., Yan, S.C., *et al.*: 'Neighborhood preserving embedding'. Proc. of IEEE Int. Conf. on Computer Vision, 2005, pp. 1208–1213
[15] Han, P.Y., Jin, A.T.B., Abas, F.S.: 'Neighborhood preserving discriminant embedding in face recognition', *J. Vis. Commun. Image Represent.*, 2009, **20**, (8), pp. 532–542
[16] Bao, X., Zhang, L., Wang, B., *et al.*: 'A supervised neighborhood preserving embedding for face recognition'. Proc. of IEEE Int. Joint Conf. on Neural Networks, 2014, pp. 278–284
[17] Gui, J., Sun, Z., Jia, W., *et al.*: 'Discriminant sparse neighborhood preserving embedding for face recognition', *Pattern Recogn.*, 2012, **45**, (8), pp. 2884–2893
[18] Liu, J., Chen, S.C., Tan, X.Y., *et al.*: 'Comments on efficient and robust feature extraction by maximum margin criterion', *IEEE Trans. Neural Netw.*, 2007, **18**, (6), pp. 1862–1864
[19] Wright, J., Yang, A.Y., Ganesh, A., *et al.*: 'Robust face recognition via sparse representation', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2009, **31**, (2), pp. 210–227
[20] Zhang, L., Zhou, W.D., Chang, P.C., *et al.*: 'Kernel sparse representation-based classifier', *IEEE Trans. Signal Process.*, 2012, **60**, (4), pp. 1684–1695
[21] Zhang, L., Zhou, W.D.: 'On the sparseness of 1-norm support vector machines', *Neural Netw.*, 2010, **23**, (3), pp. 373–385
[22] Deligiannakis, A., Kotidis, Y., Roussopoulos, N.: 'Compressing historical information in sensor networks'. Proc. of the 2004 ACM SIGMOD Int. Conf. on Management of Data, 2004, pp. 527–538
[23] Marcelloni, F., Vecchio, M.: 'A simple algorithm for data compression in wireless sensor networks', *Commun. Lett.*, 2008, **12**, (6), pp. 411–413
[24] Chen, S., Billings, S.A., Luo, W.: 'Orthogonal least squares methods and their application to non-linear system identification', *Int. J. Control*, 2007, **50**, (5), pp. 1873–1896
[25] Pati, Y.C., Rezaiifar, R., Krishnaprasad, P.S.: 'Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition'. Proc. of the Twenty-Seventh Annual Asilomar Conf. Signal Systems and Computers, November 1993, pp. 40–44
[26] Tibshirani, R.: 'Regression shrinkage and selection via the lasso', *J. R Stat. Soc. B*, 1996, **58**, (1), pp. 267–288
[27] Figueiredo, M.A.T., Nowak, R.D., Wright, S.J.: 'Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems', *IEEE J. Sel. Topics Signal Process.*, 2007, **1**, (4), pp. 586–598
[28] Leng, Y., Zhang, L., Yang, J.: 'Locally linear embedding algorithm based on OMP for incremental learning'. Proc. of IEEE Int. Joint Conf. on Neural Networks, 2014, pp. 3100–3107
[29] Bache, K., Lichman, M.: 'UCI machine learning repository', http://www.archive.ics.uci.edu/ml, accessed January 2013
[30] Samaria, F.S., Harter, A.: 'Parameterisation of a stochastic model for human face identification'. Proc. of the Second IEEE Workshop on Applications of Computer Vision, 1994, pp. 138–142. http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html, accessed April 2012
[31] Yann, L.: 'The MNIST database of handwritten digits', http://www.yann.lecun.com/exdb/mnist/, accessed June 2002
[32] Friedman, M.: 'The use of ranks to avoid the assumption of normality implicit in the analysis of variance', *J. Am. Stat. Assoc.*, 1937, **32**, (200), pp. 675–701

[33] Chen, H.H., Tiňo, P., Yao, X.: 'Predictive ensemble pruning by expectation propagation', *IEEE Trans. Knowl. Data Eng.*, 2009, **21**, (7), pp. 999–1013
[34] Demšar, J.: 'Statistical comparisons of classifiers over multiple data sets', *J. Mach. Learn. Res.*, 2006, **7**, (1), pp. 1–30

8 Appendix

8.1 Proof of Theorem 1

Proof: For convenience, we denote the objective function in (15) by $L(\boldsymbol{A}) = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{A} + \beta\left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{X} - \boldsymbol{H}\right)\left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{X} - \boldsymbol{H}\right)^{\mathrm{T}}$. To find the minimal solution to $\mathrm{tr}(L(\boldsymbol{A}))$, we can find the derivative of $L(\boldsymbol{A})$ and vanish them, or $(\partial L(\boldsymbol{A})/\partial \boldsymbol{A}) = 0$. Then we have

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^{\mathrm{T}} + \beta(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{X} - \boldsymbol{H})\boldsymbol{X}^{\mathrm{T}} = 0 \tag{17}$$

which can be re-expressed as

$$\left(\boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^{\mathrm{T}} + \beta\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\right)^{\mathrm{T}}\boldsymbol{A} = \beta\boldsymbol{X}\boldsymbol{H}^{\mathrm{T}} \tag{18}$$

Next, what we need to do is to show that $\boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^{\mathrm{T}} + \beta\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$ is invertible. Without loss of generality, this term can be represented as

$$\boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^{\mathrm{T}} + \beta\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} = \boldsymbol{X}(\boldsymbol{M} + \beta\boldsymbol{I})\boldsymbol{X}^{\mathrm{T}} \tag{19}$$

where $\boldsymbol{I} \in R^{n\times n}$ is the identity matrix.

Obviously, the matrix $\boldsymbol{X} \in R^{m\times n}$ is a full rank real matrix since $\boldsymbol{X}$ is either row full rank or column full rank. If $m < n$, $\boldsymbol{X}$ is a row full rank matrix; otherwise a column full rank matrix.

Now suppose that the matrix $\boldsymbol{X}$ is a row full rank matrix, and its rank is m. In this case, we prove that $\boldsymbol{X}(\boldsymbol{M} + \beta\boldsymbol{I})\boldsymbol{X}^{\mathrm{T}}$ is invertible. Since $\boldsymbol{M}$ is a symmetric semi-positive definite matrix, $\boldsymbol{M} + \beta\boldsymbol{I}$ is also real symmetric positive definite. Therefore, we can implement the square root decomposition on $\boldsymbol{M} + \beta\boldsymbol{I}$ as follows: $\boldsymbol{M} + \beta\boldsymbol{I} = \boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$, where $\boldsymbol{L} \in \boldsymbol{R}^{n\times n}$ is a positive definite full rank matrix. So $\boldsymbol{X}(\boldsymbol{M} + \beta\boldsymbol{I})\boldsymbol{X}^{\mathrm{T}}$ can be represented as $\boldsymbol{X}(\boldsymbol{M} + \beta\boldsymbol{I})\boldsymbol{X}^{\mathrm{T}} = \boldsymbol{D}\boldsymbol{D}^{\mathrm{T}}$, where $\boldsymbol{D} = \boldsymbol{X}\boldsymbol{L}$ is still a full rank matrix. We only verify $\boldsymbol{D}\boldsymbol{D}^{\mathrm{T}}$ is invertible as follows.

For an arbitrary vector $\boldsymbol{\gamma} \in R^{m}$, where $\boldsymbol{\gamma}$ is a non-zero vector, we can get $\boldsymbol{\gamma}^{\mathrm{T}}(\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D})\boldsymbol{\gamma} = (\boldsymbol{D}\boldsymbol{\gamma})^{\mathrm{T}}(\boldsymbol{D}\boldsymbol{\gamma})$. Let $\boldsymbol{D}\boldsymbol{\gamma} = [t_1, t_2, \ldots, t_m]^{\mathrm{T}}$, thus $\boldsymbol{\gamma}^{\mathrm{T}}(\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D})\boldsymbol{\gamma} = t_1^2 + t_2^2 + \cdots + t_m^2 \geq 0$. If $\boldsymbol{\gamma}^{\mathrm{T}}(\boldsymbol{D}^{\mathrm{T}}\boldsymbol{\gamma})\boldsymbol{D} = 0$, then $t_1 = t_2 = \cdots = t_m = 0$. In other word, we find the zero solution to the linear set of equations $\boldsymbol{D}\boldsymbol{\gamma} = \boldsymbol{0}$, or $\boldsymbol{\gamma} = \boldsymbol{0}$ where $\boldsymbol{0}$ is a vector of all zeros. Therefore, we get $\mathrm{rank}(\boldsymbol{D}) < m$ which contradicts the known or $\boldsymbol{\gamma}$ is a non-zero vector. Therefore, we can get $t_1^2 + t_2^2 + \cdots + t_m^2 > 0$. According to the above description, we know that $\boldsymbol{D}\boldsymbol{D}^{\mathrm{T}}$ is non-singular. In other words, $\boldsymbol{X}(\boldsymbol{M} + \beta\boldsymbol{I})\boldsymbol{X}^{\mathrm{T}}$ is invertible.

When the matrix $\boldsymbol{X}$ is a column full rank matrix, we can also have the same conclusion that $\boldsymbol{X}(\boldsymbol{M} + \beta\boldsymbol{I})\boldsymbol{X}^{\mathrm{T}}$ is invertible.

That is to say, the value of $L(\boldsymbol{A})$ is minimal when

$$\boldsymbol{A} = \beta\left(\boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^{\mathrm{T}} + \beta\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\right)^{-\mathrm{T}}\boldsymbol{X}\boldsymbol{H}^{\mathrm{T}} \tag{20}$$

This completes the proof. □