IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 29, 2019, accepted September 20, 2019, date of publication September 30, 2019, date of current version October 16, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2944427

Weighted Local Discriminant Preservation Projection Ensemble Algorithm With Embedded Micro-Noise

YUCHUAN LIU, XIAOHENG TAN, YONGMING LI, AND PIN WANG
School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 400044, China
Chongqing Key Laboratory of Space Information Network and Intelligent Information Fusion, Chongqing University, Chongqing 400044, China

Corresponding authors: Yongming Li (yongmingli@cqu.edu.cn) and Xiaoheng Tan (txh@cqu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61771080 and Grant 61571069, in part by the Graduate Research and Innovation Foundation of Chongqing, China, under Grant CYB18068 and Grant CYB19058, in part by the Southwest Hospital Science and Technology Innovation Program under Grant SWH2016LHYS-11, in part by the Basic and Advanced Research Project in Chongqing under Grant cstc2018jcyjAX0779, in part by the Open Project Program of the National Laboratory of Pattern Recognition (NLPR) under Grant 201800011, and in part by the Fundamental Research Funds for the Central Universities under Grant 2019CDQYTX019 and Grant 2019CDCGTX306.

ABSTRACT High-dimensional data often cause the ''curse of dimensionality'' in data processing. Dimensionality reduction can effectively solve the curse of dimensionality and has been widely used in high-dimensional data processing. However, the existing dimensionality reduction algorithms neglect the effect of noise injection, failing to account for the datasets of large variance within classes and not effectively considering the stability of dimensionality reduction. To solve the problems, this paper proposes a weighted local discriminant preservation projection algorithm based on an ensemble imbedded mechanism with micro-noise injection (n_w_LPPD). The proposed algorithm aims to overcome the problem of large variance within classes and introduces an ensemble projection matrix via Bayesian fusion mechanism with micro-noise to enhance the antijamming capability of the model. Ten public datasets were used to verify the proposed algorithm. The experimental results demonstrated that the proposed algorithm is significantly effective, especially for the case of small sample datasets with high intraclass variance. The classification accuracy is improved by at least 10% compared to the case without dimensionality reduction. Even compared with some representative dimensionality reduction algorithms, the proposed n_w_LPPD has significantly superior classification performance.

INDEX TERMS High-dimensional data, curse of dimensionality, ensemble projection matrix, Bayesian fusion, manifold learning, dimensionality reduction, small sample datasets.

I. INTRODUCTION

Most of the data generated in real life have high dimensionality. However, high-dimensional data often cause the curse of dimensionality in data processing. Manifold learning has provided effective ways to improve classification accuracy and generalization ability as well as reduce the complexity and runtime of the model, which helps to solve the curse of dimensionality for high-dimensional data and plays an important role in classification. Therefore, manifold learning has been widely applied to deal with high-dimensional data [1]–[4].

The associate editor coordinating the review of this manuscript and approving it for publication was Larbi Boubchir.

Manifold learning assumes that the features of the dataset are located or approximated on a low-dimensional manifold embedded in a high-dimensional observation space [5]. Therefore, the basic idea of manifold dimensionality reduction is to obtain a low-dimensional representation of the dataset by maintaining the global geometric properties of the intrinsic low-dimensional manifold. A typical representative algorithm of manifold dimensional reduction is the locality preserving projections (LPP) algorithm, which optimally preserves the neighborhood structure of the dataset [6] and has received extensive attention. However, the LPP algorithm still has some shortcomings. For example, this algorithm is sensitive to the number of neighborhood samples. In addition, LPP suffers from the small sample size problem, which means that

when the dimension size is larger than the number of samples, the data matrix becomes singular. Since inverting the data matrix is a necessary operation for LPP, it is imperative to overcome the small sample problem [7].

In order to overcome these shortcomings, improved LPP algorithms have been developed [8]–[27]. Yu *et al.* [8] proposed the enhanced LPP algorithm, which addresses the sensitivity of LPP to noise and outliers by introducing a robust path into the computational affinity matrix . Li *et al.* [9] proposed the bilateral LPP (BLPP) algorithm, which adds a filter term after the Euclidean distance function of the model to balance the weight of each edge of the sample. Some scholars have optimized LPP by introducing a kernel function [10], [11]. In addition, some special optimization methods have also been used to overcome the small sample size problem for the LPP [12]–[14]. The representative algorithm is optimal LPP (OLPP) [12], which converts the singular eigen-computation to eigenvalue decomposition problems without losing any discriminative information. The class-regularized LPP (CR-LPP) algorithm [13] proposed by Chao *et al.* considers global and local prior information to prevent the data matrix from becoming a singular matrix. This algorithm aims to maximize class independence while maintaining local feature similarity through dimensionality reduction. Zhang [14], propose a 2DNPP that extracts neighborhood preserving features by minimizing the Frobenius norm-based reconstruction error. Some scholars have used unlabeled data to assist in the discovery of manifold structures with labeled data to overcome the small sample problem [15]–[17]. Recently, many scholars have proposed improved dimensionality reduction methods that combine the LPP algorithm with other dimensionality reduction algorithms [18]–[21]. For example, the Fisher LPP (FLPP) algorithm [18] introduces a newly defined matrix that is used to create neighborhood graphs for different classes of samples. Discriminant LPP based on maximum margin criterion (DLPP/MMC) [19] connects DLPP-generated terms using criteria based on maximum margins to construct DLPP based on the MMCs. Discriminant information [20]–[25], regularization constraints [25], [26] and tensors [27] have also been used by some scholars to improve the LPP algorithm.

However, the improved LPP methods still overlook some issues: 1) Noise injection is rarely considered in the design of algorithms as a means of improvement. In fact, in a deep learning network, Hinton and Graves et al. have increased the generalization ability of the model by adding noise [28], [29]. Many regularization methods also solve the overfitting problem by adding noise to the training data [30]–[32]. Some scholars believe that adding noise into the process of training could instruct the model to learn feature representations that are robust to the effect of noise, thereby reducing the risk of overfitting and improving the generalization ability of the model [33]. 2) Most improved LPP-based algorithms focus on increasing the variance between classes rather than reducing the variance within classes. 3) There is a lack of stability for high-dimensional small samples. When there is a difference in data distribution between the training data and the test data, the results of a predictive learner can be degraded [34]. The problems above are widespread across datasets but are nonetheless neglected by LPP-related algorithms.

To solve these problems, this paper proposes a weighted local discriminant preservation projection ensemble algorithm with embedded micro-noise. The algorithm proposed in this paper uses the following main procedures. First, the training samples are randomly sampled, and micro-noise is added. Second, the noisy samples are mapped according to the designed objective function. Third, the first two operations are repeated to obtain multiple projection matrices. Finally, the ensemble projection matrix via Bayesian fusion is introduced to construct the final projection matrix.

The main contributions and innovations of this paper can be stated as follows:

1) Noise injection is introduced to improve the generalization ability of the model. Micro-noise injection into the process of training could instruct the model to learn feature representations that are robust to the effect of noise, thereby reducing the risk of overfitting and improving the generalization ability of the model.
2) The objective function is improved, so the dataset with the largest intraclass variance is prioritized, helping the proposed algorithm better handle datasets with large intraclass variances.
3) In order to further enhance the stability of the algorithm, an ensemble projection matrix via Bayesian fusion is introduced to construct the ensemble LPP projection matrix.

The remainder of this paper is organized as follows. In section II, the related works about the proposed method are described in this paper. Section III mainly introduces the design of the proposed algorithm. Section IV shows the experimental results and verifies the effectiveness of the proposed algorithm. Section V presents a discussion and conclusion.

II. RELATED WORKS

In this section, some representative dimensionality reduction methods are reviewed briefly. These methods comprise principal component analysis (PCA), linear discriminant analysis (LDA), LPP, locality preserving discriminant projections (LPDP), local discriminant preservation projection (LDPP), ensemble discriminative local metric learning (EDLML), globality-locality preserving projections (GLPP) and other feature selection methods such as P-value, correlation coefficient, relief, and the least absolute shrinkage and selection operator (LASSO). The main principles of some of these methods are written below. Before proceeding, we need to introduce some notation.

In this paper, the data matrix is denoted as $X = \{x_1^{(1)}, x_2^{(1)}, \cdots, x_{N_1}^{(1)}, x_1^{(2)}, x_2^{(2)}, \cdots, x_{N_2}^{(2)}, \cdots, x_1^{(C)}, x_2^{(C)}, \cdots, x_{N_C}^{(C)}\} \in R^{N \times K}$ and $x^{(i)} = \{x_1^{(i)}, x_2^{(i)}, \cdots, x_{N_i}^{(i)}\}$ denotes the

ith class samples, where $N = N_1 + N_2 + \cdots + N_C$ is the number of samples, C is the number of classes, and K is the dimensionality. The label vector of data is denoted as $y = [y_1, y_2, \cdots, y_N]^T \in R^N$. Let $U = (u_1, u_2, \cdots, u_m) \in R^{K\times k}$ represent the projection matrix that is used to map the original data from $R^{N\times K}$ to a new low-dimensional space $R^{N\times k}$.

A. PRINCIPAL COMPONENT ANALYSIS (PCA)

PCA [35] is one of the most commonly used techniques for dimension reduction in many applications, such as neuroscience, genomics, and finance. This method extracts latent principal factors that preserve most of the variation in the dataset. Let X have a mean of zero, and let its covariance matrix be $\sum$. With this formalism, the objective function of PCA can be expressed as

$$\begin{cases} \min\limits_{U} \ Tr(U^T X X^T U) \\ s.t.\ U^T U = I \end{cases} \Leftrightarrow \begin{cases} \min\limits_{U} Tr(U^T \Sigma U) \\ s.t.\ U^T U = I \end{cases} \tag{1}$$

PCA seeks the top k eigenvectors of $\sum$ as projection direction vectors and projects the original high-dimensional data onto the low-dimensional space spanned to achieve the goal of dimensionality reduction.

B. LINEAR DISCRIMINANT ANALYSIS (LDA)

The idea of LDA [36] can be summarized as follows: the variance within classes after projection is minimized, and the variance between classes is maximized. The target function can be expressed as

$$\max_{U} \frac{Tr(U^T S_B U)}{Tr(U^T S_W U)} \tag{2}$$

where S_B is the between-class scatter matrix, S_W is the within-class scatter matrix.

C. LOCALITY PRESERVING PROJECTIONS (LPP)

LPP [37] aims to optimally preserve the neighborhood structure of data, and the objective function of LPP minimizes the distance between those data points with neighborhood relations in the raw data space (i.e., locality preservation). Generally, the LPP model can be formulated as

$$\min_{U} \sum_{i,j=1}^{N} A_{ij} \left\| U^T x_i - U^T x_j \right\|_F^2 \tag{3}$$

where $A \in R^{N\times N}$ is the affinity matrix and $A_{ij} = 0$ if x_i and x_j are not adjacent.

D. LOCALITY PRESERVING DISCRIMINANT PROJECTIONS (LPDP)

The LPDP algorithm [20], proposed in 2009 by Gui et al., aims to optimally preserve the neighborhood structure of data, thereby enhancing global class discrimination after projection. The objective function of LPDP [38] is written as

$$\begin{aligned} &\min_{U} \ Tr(U^T (XLX^T - (S_B - \mu S_W))U) \\ &s.t.\ U^T X D X^T U = I \end{aligned} \tag{4}$$

where $L = D - A$ is a Laplacian matrix, $D_{ii} = \sum\limits_{j} A_{ij}$ is a diagonal matrix.

E. LOCAL DISCRIMINANT PRESERVATION PROJECTION (LDPP)

The LDPP algorithm [21] aims to capture neighborhood structure and local discrimination. The objective function of LDPP can be described as

$$\max_{U} \ Tr(U^T (S_{LB} - \lambda(\mu S_{LW} + \gamma X L X^T))U) \tag{5}$$

where S_{LB} is the local between-class scatter matrix and S_{LW} is the local within-class scatter matrix.

F. ENSEMBLE DISCRIMINATIVE LOCAL METRIC LEARNING (EDLML)

The EDLML algorithm [39] aims to learn a subspace to keep all the samples in the same class as close together as possible, while those from different classes are separated. The learned local metrics are then used to build an ensemble metric. The objective function of EDLML can be formulated as

$$\begin{aligned} &\arg\min_{U_i} \ Tr(U_i^T \sum_{j=1}^{k} A_{ij} u_{ij} (x_i - x_{ij})^T (x_i - x_{ij}) U_i) \\ &s.t.\ \arg\min_{u_{ij}} \sum_{i=1}^{k} \left\| x_i - \sum_{j=1}^{k} u_{ij} x_{ij} \right\|^2 \\ &\sum_{j=1}^{k} u_{ij} = 1, \quad u_{ij} \geq 0, \quad j = 1, 2, \ldots, k \end{aligned} \tag{6}$$

where u_{ij} represents the contribution of the jth sample to the ith reconstruction. A means the affinity matrix, and

$$A_{ij} = \begin{cases} 1, & \text{if } x_j \text{ is near } x_i \text{and they have the same label} \\ -1, & \text{if } x_j \text{ is near } x_i \text{and they have the different labels.} \end{cases}$$

In [39], EDLML has only an affinity matrix, as in the simple-minded method (Eq. (12)). We extended the affinity matrix of EDLML to obtain a heat-kernel mode (Eq. (13)) as follows:

$$A_{ij} = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{t}}, & \text{if } x_j \text{ is near } x_i \text{ and they have the same label} \\ -e^{-\frac{\|x_i - x_j\|^2}{t}}, & \text{if } x_j \text{ is near } x_i \text{ and they have the different labels} \end{cases}$$

G. GLOBALITY-LOCALITY PRESERVING PROJECTIONS (GLPP)

The GLPP algorithm [40] replaces the original graph Laplacian of LPP with a new graph Laplacian to present a new supervised dimensionality reduction algorithm. The objective

function of GLPP can be described as

$$\arg\min_{u} \Big(\sum_{i,j\in C} (u^T m_i - u^T m_j)^2 B_{ij} + \beta \Big(\sum_{c\in C} \sum_{i,j\in c} (u^T x_i - u^T x_j)^2 S_{ij} \Big) \Big) \quad (7)$$

where S and B are the adjacency weight matrices of the dynamic factor objective term and the static factor objective term, respectively. m_i means the center of the ith class.

III. THE PROPOSED METHOD

In this part, the proposed algorithm is introduced. The proposed algorithm in this paper addresses three points. First, it introduces random subspace sampling and adds Gaussian micro-noise to the training data. Second, a method of noise-embedded LPPD (improved Locality preserving projection discriminant model) is established based on the proposed objective function. Finally, ensemble projection matrix via Bayesian fusion is used to construct the final projection matrix.

In the first part, the number of sampled samples (n_s) and the number of subspaces (p) are used. The relationship between the signal and the Gaussian micro-noise must comply with Eq. (8):

$$A_b \le \frac{1}{M} \min\ distance(u^{(p)}, u^{(q)}) \quad (8)$$

where A_b represents the maximum amplitude of the Gaussian noise, $\min distance(u^{(p)}, u^{(q)})$ represents the minimum distance between the centers of the different classes, and M represents the ratio of $\min distance(u^{(p)}, u^{(q)})$ to A_b. Considering that the label of the sample cannot be changed due to the addition of noise, the amplitude of the noise should not be excessively large. According to statistical knowledge, the value range of M is between 50 and 1000.

Then, feature transformation is performed on the sampled subsets with micro-noise (named $X'^1_{train}, X'^2_{train}, \cdots, X'^p_{train}$). The algorithm proposed – noise-embedded locality preserving discriminant projections (n-LPPD) – takes into account the similarities between samples, removing some samples far away from the center of the class. Assuming that the number of samples for the cth class is k_{mc}, the total number of samples after sampling is $k_m = \sum_{c=1}^{C} k_{mc}$.

The algorithm is proposed to make the samples of the same class as close as possible after the mapping. It can be described as

$$\begin{aligned} &\min_{U^n} \sum_{c=1}^{C} \left\| U^{nT} x'^{(c)} - U^{nT} \overline{x_w'^{(c)}} \right\|_{x'^{(c)}\in X_{wc}} \\ &= \min_{U^n} \sum_{c=1}^{C} (U^{nT}(x'^{(c)} - \overline{x_w'^{(c)}})(x'^{(c)} - \overline{x_w'^{(c)}})^T U^n) \Big|_{x'^{(c)}\in X_{wc}} \\ &= \min_{U^n} U^{nT} S^*_{WL} U^n (n = 1, 2, \cdots, p) \end{aligned} \quad (9)$$

where $S^*_{WL} = \sum_{c=1}^{C} \sum_{x'^{(c)}\in X_{wc}} (x'^{(c)} - \overline{x_w'^{(c)}})(x'^{(c)} - \overline{x_w'^{(c)}})^T$ means the local within-class scatter matrix, $\overline{x_w'^{(c)}} = \frac{1}{k_{mc}} \sum_{i=1, x\in N_{kmc}(m)}^{k_{mc}} x_i^{(c)} + b_i^{(c)}$ is the center of the cth local class for S^*_{WL} computation with micro-noise, b is the micro-noise, $b_i^{(c)}$ denotes the ith micro-noise added to the cth class.

In the similar way, the samples of different classes are as far apart as possible after mapping can be described as

$$\begin{aligned} &\max_{U^n} \sum_{c=1}^{C} \left\| U^{nT} \overline{x_b'^{(c)}} - U^{nT} \overline{x_b'} \right\|_{x_b'^{(c)}, x_b' \in X_{bc}} \\ &= \max_{U^n} \sum_{c=1}^{C} (U^{nT}(\overline{x_b'^{(c)}} - \overline{x_b'})(\overline{x_b'^{(c)}} - \overline{x_b'})^T U^n) \Big|_{x_b'^{(c)}, x_b' \in X_{bc}} \\ &= \max_{U^n} U^{nT} S^*_{BL} U^n (n = 1, 2, \cdots, p) \end{aligned} \quad (10)$$

where $S^*_{BL} = \sum_{c=1}^{C} (\overline{x_b'^{(c)}} - \overline{x_b'})(\overline{x_b'^{(c)}} - \overline{x_b'})^T$ means the local between-class scatter matrix, $\overline{x_b'} = \frac{1}{k_m} \sum_{i=1, x\in N_{km}(m)}^{k_m} x_i + b_i$ is the center of local part for S^*_{BL} computation with micro-noise, $\overline{x_b'^{(c)}} = \frac{1}{N_{lc}} \sum_{i=1, x\in N_{km}(m)}^{N_{lc}} x_i^{(c)} + b_i^{(c)}$ is the center of the cth local class for S^*_{BL} computation with micro-noise, and N_{lc} is the number of the cth class in the local part.

Furthermore, locality preservation can be described as

$$\begin{aligned} &\sum_{c=1}^{C} \sum_{i=1}^{N_c} \sum_{j=1}^{N} A_{ij}^c \left\| U^{nT} x_i'^{(c)} - U^{nT} x_j' \right\|_F^2 \Big|_{x_i'^{(c)}, x_j' \in X'^n_{train}} \\ &= (2 \sum_{c=1}^{C} (U^{nT} (\sum_{i=1}^{N_c} x_i'^{(c)} Q_{ii}^c x_i'^{(c)T} \\ &\quad - \sum_{i=1}^{N_c} \sum_{j=1}^{N} x_i'^{(c)} A_{ij}^c x_j'^T) U^n) \Big|_{x_i'^{(c)}, x_j' \in X'^n_{train}} \\ &\Leftrightarrow \sum_{c=1}^{C} (U^{nT} X'^{(c)} (Q^c - A^c) X'^{nT}_{train} U^n) \Big|_{X'^{(c)} \subset X'^n_{train}} \\ &= \sum_{c=1}^{C} (U^{nT} X'^{(c)} (Q - A) X'^{nT} U^n) \Big|_{X'^{(c)} \subset X'^n_{train}} \\ &= U^{nT} X'^n_{train} L X'^{nT}_{train} U^n (n = 1, 2, \cdots, p) \end{aligned} \quad (11)$$

where $L = Q - A$ is a Laplacian matrix, $Q_{ii}^c = \sum_j A_{ij}^c$ is a diagonal matrix, $Q = \begin{pmatrix} Q^1 & & \\ & \ddots & \\ & & Q^C \end{pmatrix}$, and $A = \begin{pmatrix} A^1 & & \\ & \ddots & \\ & & A^C \end{pmatrix}$

is the affinity matrix, calculated in the following ways [41].

$$\text{Simple-minded: } A_{ij}^c = \begin{cases} 1, & if\ x_i^c \in N_k(x_j)\ \|\ x_j \in N_k(x_i^c) \\ 0, & otherwise \end{cases} \tag{12}$$

$$\text{Heat kernel:} A_{ij}^c = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{t}}, & if\ x_i^c \in N_k(x_j)\ \| \\ & x_j \in N_k(x_i^c) \\ 0, & otherwise. \end{cases} \tag{13}$$

where t is the kernel parameter.

The objective function of the proposed algorithm minimizing the local intra-class scattering matrix while maximizing the inter-class scattering matrix and preserving the locality of the sample. With Eq. (9-11) and the limit of noise, the proposed algorithm can be described as

$$\min_{U^n} Tr\left(\frac{U^{nT} S_{WL}^* U^n + U^{nT} X_{train}^{'n} L X_{train}^{'nT} U^n}{U^{nT} S_{BL}^* U^n}\right) \tag{14}$$

Equation (14) can be equivalent to the Lagrangian function as follows.

$$L(U^n, \lambda) = U^{nT}(S_{WL}^* + \lambda(\gamma X_{train}^{'n} L X_{train}^{'nT} - \mu S_{BL}^*))U^n \tag{15}$$

The derivative of U is taken, and optimal solutions are obtained.

$$\begin{aligned} &\left.\frac{\partial L(U^n, \lambda)}{\partial U^n}\right|_{x_i', x_j' \in X_{train}^{'n}} = 0 \\ &\Rightarrow \frac{\partial\{U^{nT}(S_{WL}^* + \lambda(\gamma X_{train}^{'n} L X_{train}^{'nT} - \mu S_{BL}^*))U^n\}}{\partial U^n} = 0 \\ &\Rightarrow 2S_{WL}^* U^n + 2\lambda(\gamma X_{train}^{'n} L X_{train}^{'nT} U^n - \mu S_{BL}^* U^n) = 0 \\ &\Leftrightarrow S_{WL}^* U^n = \lambda(\mu S_{BL}^* - \gamma X_{train}^{'n} L X_{train}^{'nT}) U^n \\ &\Rightarrow (\mu S_{BL}^* - \gamma X_{train}^{'n} L X_{train}^{'nT})^{-1} S_{WL}^* U^n = \lambda U^n \end{aligned} \tag{16}$$

From Eq. (16), the projection matrix U^n can be obtained easily. The vector $U_k^n = (u_1, u_2, \ldots, u_k)$ is comprised of the top k eigenvectors of U^n. Then, the original high-dimensional data can be projected onto the low-dimensional space spanned by columns U_k^n to achieve dimensionality reduction.

The vector U_k^n was used to map $X_{train}^{'n}$ and $X_{train}^{'n-}$($X_{train}^{'n-} = X_{train}^{'} - X_{train}^{'n}$, $X_{train}^{'}$ denotes the training set with micro-noise). The mapped data were named $Z_{train}^{'n}$ and $Z_{train}^{'n-}$, respectively. $Z_{train}^{'n}$ was used to train the classifier, and the trained classifier predicted the labels of $Z_{train}^{'n-}$. The prediction result of the classifier for $Z_{train}^{'n-}$ was recorded in a confusion matrix named $CM_n \in R^{C\times C}$ (C means the number of classes of samples). The element of the ith row and jth column in the matrix is denoted as $cm_n^{i,j}$, indicating the number of data whose actual label is L_i and whose predicted label is L_j. According to CM_n, we can easily obtain the posterior probability matrix (LM_n) of the unknown sample by Eq. (17).

$$lm_n^{i,j} = \frac{cm_n^{i,j}}{cm_n^{:,j}} \tag{17}$$

where $lm_n^{i,j}$ means the posterior probability that the unknown sample was predicted to be L_j and $cm_n^{:,j}$ means the number of samples classified as L_j.

In this paper, there are a total of p confusion matrices similar to CM_n; then, the confusion matrix after classifier integration is

$$LM = \sum_{j=1}^{C} \prod_{n=1}^{p} lm_n^{i,j} \tag{18}$$

The prediction result of $Z_{train}^{'n-}$ by the classifier was recorded as $L_{train}^{'n-}$. According to the $L_{train}^{'n-}$, the probability that each test sample belonged to each class was calculated by Eq. (18), and the prediction label with the highest posterior probability was used as the predicted output L_{pre} of the unknown sample. The weight α_n of the corresponding projection matrix U_k^n is calculated as shown below.

$$\begin{aligned} &\alpha_n = \Big(\sum_{i=1, j=n}^{N_L} j\Big)/(n * N_L), \quad (n = 1, 2, \cdots, p) \\ &s.t.\ j = \arg\max_n P(U_k^n \,|\, L_{pre}^i) \end{aligned} \tag{19}$$

where $P(U_k^n \,|\, L_{pre}^i) = LM_n^i$ denotes the posterior probability that the predictive label of the ith test sample is obtained by the U_k^n, N_L denotes the number of L_{pre}.

Based on the procedure above, the final projection matrix U_k^{final} is obtained by Eq. (20):

$$U_k^{final} = \sum_{n=1}^{p} \alpha_n U_k^n \tag{20}$$

Based on the description above, the process of the proposed algorithm (n_w_LPPD) is as follows.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. SUBJECTS/DATABASE

In this paper, some representative dimensionality reduction algorithms, including PCA, LDA, LPP, LPDP, and LDPP, are used for comparisons. In the classification process, the classifiers include support vector machine (SVM), extreme learning machine (ELM) and random forest (RF). The kernel functions of SVM are linear kernels and radial basis function (RBF) kernels. Some symbols are introduced in the algorithms, and the parameters represented by these symbols are listed along with their parameter settings in TABLE 1.

We tested the effectiveness of the algorithm on ten widely used public datasets. Brief information about each dataset is shown in TABLE 2. All experiments were carried out in the following experimental environment: the experimental operating system was 64-bit Windows 7, and the memory size was 128 GB. The programming tool was MATLAB, 2014a.

In addition to the parameters in TABLE 1 that needed to be set, we also needed to set the relevant parameters of some classifiers. An SVM, an ELM and an RF were involved in the classification operation. The kernel functions of the

TABLE 1. Symbols in the algorithm and their meanings.

Algorithm	Symbol	Meaning	Parameter settings
PCA	$\sum$	Covariance matrix	XX^T
	k	Dimension after dimension reduction	5,10,15,…
LDA	k	Dimension after dimension reduction	5,10,15,…
LPP	t	Kernel parameter for affinity matrix **A**	$10^{-4},10^{-3},\ldots,10^{3},10^{4}$
	k	Dimension after dimension reduction	5,10,15,…
LPDP	μ	Lagrange penalty factor for $U^T S_W U$	$10^{-4},10^{-3},\ldots,10^{3},10^{4}$
	t	Kernel parameter for affinity matrix **A**	$10^{-4},10^{-3},\ldots,10^{3},10^{4}$
	k	Dimension after dimension reduction	5,10,15,…
LDPP	μ	Lagrange penalty factor for $U^T S_{LW} U$	$10^{-4},10^{-3},\ldots,10^{3},10^{4}$
	γ	Lagrange penalty factor for $U^T XLX^T U$	$10^{-4},10^{-3},\ldots,10^{3},10^{4}$
	k_m	Number of nearest neighbors to local data center	$0.9*N_{train}$
	k_{mc}	Number of nearest neighbors to the class center	$0.9* N_{lc}$
	k	Dimension after dimension reduction	5,10,15,…
	t	Kernel parameter for affinity matrix **A**	$10^{-4},10^{-3},\ldots,10^{3},10^{4}$
	μ	Lagrange penalty factor for $U^{nT} S^*_{BL} U^n$	$10^{-4},10^{-3},\ldots,10^{3},10^{4}$
	γ	Lagrange penalty factor for $U^{nT} X'^n_{train} L X'^{nT}_{train} U^n$	$10^{-4},10^{-3},\ldots,10^{3},10^{4}$
	n_s	Number of sampling	$0.8*N_{train}$
n_w_LPPD	k_m	Number of nearest neighbors to local data center	$0.9* n_s$
	k_{mc}	Number of nearest neighbors to the class center	$0.9* N_{lc}$
	k	Dimension after dimension reduction	5,10,15,…
	t	Kernel parameter for affinity matrix **A**	$10^{-4},10^{-3},\ldots,10^{3},10^{4}$

Note: N_{train} means the number of training sets.

SVM are the RBF kernel and the linear kernel. The number of hidden neurons L of the ELM was set as 5000, and the number of random trees in the random forest was set to 300. The parameters M in Eq. (8) and p in Eq. (18) were set to 200 and 3, respectively. Other related parameters were set to the default values.

B. COMPARISON AND ANALYSIS OF THE ALGORITHMS

This part of the experiment consists of two parts. The first part compares the proposed algorithm with some representative feature extraction algorithms. The second part compares the proposed algorithm with some representative feature selection algorithms. Each experiment was repeated five times to eliminate the interference caused by the occasionality.

The first group of experiments discusses the classification performance of each feature extraction algorithm in an SVM with a linear kernel, an SVM with an RBF kernel, an ELM and a random forest classifier. The experimental results for these four types of classifiers are recorded in TABLE 3-6, respectively. The significant differences between the proposed algorithm and other comparison algorithms for the AD dataset are reported in TABLE 7. Then, the proposed algorithm is compared with the typical feature selection algorithm. The experimental results are recorded in TABLE 8.

In TABLE 3-6, N_DR means the dataset without dimensionality reduction, LPP(S) means the affinity matrix **A** calculated in simple-minded mode, LPP(H) means the affinity matrix **A** calculated in heat-kernel mode, and the same is true of the others.

As shown in TABLE 3, the proposed n_w_LPPD algorithm has the highest classification accuracy of all tested algorithms in most datasets in the case of SVM with a linear kernel. Especially for some datasets with small samples, such as AD and LSVT, compared to N_DR, the classification accuracy can be improved by more than 10%. Regarding AD and LSVT, the proposed algorithm is significantly better than other dimensionality reduction methods. One possible reason is that both the AD and LSVT datasets have large inter-invariance, and the proposed algorithm takes into account the case of samples with large variance matrices compared to other algorithms within the class. From TABLE 3, we can see that the classification accuracy of the simple-minded mode is lower than that of the heat-kernel mode. One possible reason is that the heat-kernel mode takes into account the ''closeness'' relationship between the samples, while the simple-minded mode does not consider this relationship and treats all samples as identical. Almost all dimensionality reduction algorithms are helpful for improving the classification of data.

TABLE 2. Basic information about datasets.

Database name	Instances		Standard deviation within the class	Attributes	Class	Relevant papers
amazon	1500	30 ⋮ 30	85.17~129.59	10000	50	Reference [42]
musk	476	269 207	$1.1034*10^3$ 973.5629	166	2	Reference [43]
Low Resolution Spectrometer Dataset (LRS)	531	158 62 86 225	$1.3809*10^4$ $1.5756*10^4$ $2.0741*10^4$ $1.3881*10^4$	100	4	Reference [42]
AD	90	30 30 30	$9.1273*10^3$ $1.1824*10^4$ $1.0871*10^4$	32	3	Reference [44]
Parkinson Speech Dataset (PD)	1040	520 520	236.2477 289.8749	26	2	Reference [45]
Statlog_Landsat_Satellite (Statlog)	6435	1533 703 1358 626 707 1508	67.7329 94.1301 45.6840 49.6459 74.9571 50.5412	36	6	Reference [42]
Pen-Based Recognition of Handwritten Digits (Pen-Digits)	10992	1143 1143 1144 1055 1144 1055 1056 1142 1055 1055	82.6034 92.1269 53.3727 47.3689 68.1537 124.5134 56.5827 75.1044 111.7903 88.5514	16	10	Reference [46]
LSVT Voice Rehabilitation Data Set (LSVT)	126	42 84	$1.3801*10^{10}$ $1.8408*10^{10}$	309	2	Reference [47]
Urban land cover (Urban)	675	36 116 106 122 59 112 61 34 29	$5.0449*10^3$ $1.5741*10^4$ $5.1165*10^3$ $1.2440*10^4$ $8.6568*10^3$ $1.2229*10^4$ $5.3454*10^3$ $1.1399*10^4$ $2.9726*10^3$	148	9	Reference [48]
Statlog_Vehicle Silhouettes (Vehicle)	846	199 217 218 212	76.3650 177.4222 205.4768 186.6789	18	4	Reference [42]

TABLE 3. Classification results using an SVM with a linear kernel.

	N_DR (%)	PCA (%)	LDA (%)	LPP (S) (%)	LPP (H) (%)	LPDP (S) (%)	LPDP (H) (%)	EDLML (S) (%)	EDLML (H) (%)	GLPP (S) (%)	GLPP (H) (%)	LDPP (S) (%)	LDPP (H) (%)	n_w_LPP D(S) (%)	n_w_LPP D(H) (%)
AD	53.33	49.33	52.67	54.00	60.00	58.00	63.33	62.00	64.66	48.00	58.00	60.67	64.00	**67.33**	**67.33**
PD	61.44	60.98	61.03	61.44	63.68	63.56	64.08	63.62	64.71	61.03	61.03	64.08	65.17	65.11	**65.40**
Statlog	86.32	86.67	85.73	86.92	87.06	86.86	87.22	86.88	87.17	87.24	**89.37**	86.99	87.20	87.30	87.38
Pen-Digits	97.77	97.49	97.19	97.65	97.73	97.78	97.85	97.76	97.79	**99.20**	98.97	97.92	97.96	97.80	97.89
LSVT	79.52	80.95	78.10	66.67	79.05	84.29	87.62	87.14	90.38	72.86	71.90	85.71	87.62	88.57	**93.33**
Urban	78.49	78.49	78.31	78.49	78.93	79.02	79.64	80.09	81.31	45.42	67.91	79.82	80.27	80.09	**81.60**
Vehicle	77.45	77.73	77.45	76.38	77.73	79.43	79.72	77.80	78.16	68.01	60.07	80.35	**80.57**	76.60	78.87
LRS	67.20	63.77	61.71	67.20	70.51	66.17	66.74	67.43	67.31	52.46	49.60	73.26	73.60	70.97	**74.06**
musk	77.09	77.09	75.32	77.22	79.59	81.27	82.03	81.27	83.16	82.15	85.06	82.28	82.78	84.43	**86.20**
amazon	46.80	58.80	58.65	10.60	57.90	59.50	60.00	59.00	59.20	18.40	23.85	60.10	**60.60**	60.10	60.25

TABLE 4. Classification results using an SVM with an RBF kernel.

	N_DR (%)	PCA (%)	LDA (%)	LPP (S) (%)	LPP (H) (%)	LPDP (S) (%)	LPDP (H) (%)	EDLML (S) (%)	EDLML (H) (%)	GLPP (S) (%)	GLPP (H) (%)	LDPP (S) (%)	LDPP (H) (%)	n_w_LPP D(S) (%)	n_w_LPP D(H) (%)
AD	52.67	48.67	50.67	52.00	56.67	54.67	56.00	56.66	66.00	48.00	58.00	58.67	60.67	65.33	**68.67**
PD	65.06	66.32	66.21	64.48	65.46	65.34	66.21	66.15	66.67	61.03	61.03	66.44	66.55	66.95	**67.24**
Statlog	89.16	**90.86**	88.61	90.09	90.14	88.84	89.27	90.00	90.08	87.24	89.37	89.34	89.78	90.17	90.30
Pen-Digits	99.09	**99.39**	99.31	99.09	99.16	99.08	99.15	99.10	99.13	99.20	98.97	99.15	99.19	99.13	99.16
LSVT	81.43	83.81	84.29	71.90	80.48	86.19	86.19	85.71	87.14	72.86	71.90	87.14	88.10	85.24	**88.57**
Urban	78.93	79.38	79.11	79.20	79.91	81.16	81.69	80.09	80.80	45.42	67.91	81.69	**82.84**	80.80	81.24
Vehicle	72.20	77.45	**78.23**	71.99	72.55	74.04	75.11	72.98	73.90	68.01	60.07	76.03	76.60	71.84	74.47
LRS	66.51	66.17	66.40	66.51	66.63	66.17	63.20	66.74	64.91	52.46	49.60	67.43	67.89	67.31	**68.69**
musk	84.94	87.47	87.09	85.32	87.82	85.95	84.43	89.24	89.49	82.15	85.06	87.09	85.82	87.97	**89.49**
amazon	25.56	54.20	54.05	32.45	46.90	48.85	**56.00**	47.85	48.35	18.40	23.85	48.90	55.05	48.30	51.70

TABLE 5. Classification results using an ELM.

	N_DR (%)	PCA (%)	LDA (%)	LPP (S) (%)	LPP (H) (%)	LPDP (S) (%)	LPDP (H) (%)	EDLML (S) (%)	EDLML (H) (%)	GLPP (S) (%)	GLPP (H) (%)	LDPP (S) (%)	LDPP (H) (%)	n_w_LPP D (S) (%)	n_w_LPP D (H) (%)
AD	52.67	54.00	49.33	52.00	57.33	54.67	62.67	60.67	62.67	48.00	58.00	61.33	66.00	60.00	**67.33**
PD	60.23	59.25	61.43	60.86	62.41	61.90	63.79	65.00	66.15	61.03	61.03	64.60	65.69	66.03	**68.33**
Statlog	85.34	88.62	88.29	88.11	88.32	89.27	89.48	88.87	89.39	87.24	89.37	89.30	**89.55**	88.70	89.18
Pen-Digits	99.01	99.08	98.97	99.05	99.12	99.10	99.26	99.20	99.26	99.20	98.97	99.25	99.33	99.31	**99.37**
LSVT	83.81	82.38	83.33	66.67	84.29	85.71	90.00	88.10	90.00	72.86	71.90	88.57	90.48	90.00	**91.90**
Urban	77.96	77.51	78.13	77.69	79.29	79.56	80.89	81.00	81.51	45.42	67.91	81.33	**81.60**	80.00	79.91
Vehicle	76.38	75.18	75.04	75.18	77.45	78.58	81.21	77.59	79.15	68.01	60.07	80.64	**82.13**	75.25	78.09
LRS	64.69	65.03	62.86	63.89	68.57	66.63	65.14	67.20	66.86	52.46	49.60	71.77	**73.26**	68.57	69.26
musk	84.30	87.22	86.58	79.87	86.39	84.56	84.81	87.97	**89.37**	82.15	85.06	86.58	86.71	87.85	88.86
amazon	37.12	56.30	53.15	23.85	53.55	57.30	58.90	56.45	57.35	18.40	23.85	57.40	59.50	57.30	**60.25**

On the AD and PD datasets, as shown in TABLE 4, the proposed algorithm achieves the best classification accuracy. Again, the proposed algorithm markedly improves the classification accuracy of small sample datasets such as AD and LSVT. On the Statlog and Pen-Digits datasets, the PCA algorithm achieved the best result. One possible reason is that the number of the Statlog and Pen-Digits datasets is sufficient, resulting in a small difference between the traditional dimensionality reduction algorithm and the manifold dimensionality reduction algorithm. The same can also be seen from the classification results. On the Statlog, Pen-Digits, LSVT, Urban, LRS, and amazon datasets, the proposed algorithm still outranks the other algorithms in classification accuracy.

On the AD, PD, Pen-Digits and amazon datasets, as shown in TABLE 5, the algorithm proposed in this paper achieved the highest classification accuracy of all tested algorithms. On the LSVT dataset, the proposed algorithm is second only to LDPP(H), and its classification result is only 0.48% lower than that of the latter. On the Statlog, LRS, musk and Urban datasets, the classification accuracy of the proposed algorithm is within the top three algorithms overall. From the table, the dimensionality reduction effects obtained by PCA and LDA are the lowest. One possible reason is that PCA and LDA are linear dimensionality reduction methods, which may perform well when variables and observations have a linear relationship.

TABLE 6. Classification results using an RF.

	N_DR (%)	PCA (%)	LDA (%)	LPP (S) (%)	LPP (H) (%)	LPDP (S) (%)	LPDP (H) (%)	EDLML (S) (%)	EDLML (H) (%)	GLPP (S) (%)	GLPP (H) (%)	LDPP (S) (%)	LDPP (H) (%)	n_w_LPP D(S) (%)	n_w_LPP D(H) (%)
AD	52.00	53.33	53.33	38.00	56.67	57.33	63.33	66.00	69.33	48.00	58.00	59.33	68.00	66.00	**69.33**
PD	65.75	66.09	65.69	64.71	67.70	66.38	67.76	67.59	**69.54**	61.03	61.03	68.62	69.20	68.05	68.45
Statlog	90.35	90.12	90.42	89.72	90.20	89.91	90.04	89.94	90.34	87.24	89.37	90.09	90.20	90.42	**90.83**
Pen-Digits	98.42	98.45	98.47	98.22	98.49	98.21	98.37	98.45	98.57	**99.20**	98.97	98.30	98.45	98.62	98.61
LSVT	75.24	77.14	77.62	55.71	78.57	85.71	87.14	82.86	89.05	72.86	71.90	88.10	90.95	**90.95**	89.52
Urban	83.29	83.29	83.91	69.78	84.27	84.00	84.89	79.91	**86.49**	45.42	67.91	85.69	85.96	77.60	80.00
Vehicle	73.05	73.33	72.77	72.70	75.89	78.65	80.28	73.40	75.74	68.01	60.07	79.93	**81.06**	72.77	73.55
LRS	65.49	64.69	64.69	64.91	66.40	65.37	61.37	66.74	62.4	52.46	49.60	69.60	69.70	68.11	**70.74**
musk	84.68	82.28	82.53	80.76	86.08	84.18	84.05	87.34	**88.61**	82.15	85.06	85.82	85.44	88.22	86.46
amazon	32.12	53.45	**54.15**	22.80	26.80	33.35	33.95	32.35	29.20	28.40	33.85	33.15	34.35	32.35	38.35

TABLE 7. Significant differences between n_w_LPPD and other algorithms (AD dataset and $\alpha \leq 0.05$).

Classifier	Algorithm	N_DR	LPP (S)	LPP (H)	LPDP (S)	LPDP (H)	LDPP (S)	LDPP (H)	PCA	LDA	EDLML (S)	EDLML (H)	GLPP (S)	GLPP (H)
SVM (linear)	n_w_LPPD(S)	**0.0032**	**0.0046**	**0.0341**	**0.0255**	**0.0341**	0.0533	0.2420	**0.0014**	**0.0249**	0.0515	0.4258	**0.0189**	0.1926
	n_w_LPPD(H)	**0.0032**	**0.0015**	0.1419	**0.0255**	0.2302	0.0533	0.0993	**0.0095**	**0.0063**	0.0515	0.0702	**0.0065**	0.1795
SVM (RBF)	n_w_LPPD(S)	**0.0115**	**0.0135**	**0.0329**	**0.0299**	**0.0249**	0.0751	0.1599	**0.0004**	**0.0129**	**0.0254**	0.6223	**0.0138**	0.2822
	n_w_LPPD(H)	**0.0039**	**0.0039**	**0.0061**	**0.0102**	**0.0115**	**0.0285**	**0.0418**	**0.0004**	**0.0053**	**0.0086**	0.3372	**0.0037**	0.1404
ELM	n_w_LPPD(S)	**0.0196**	**0.0240**	0.2420	0.0777	0.0993	0.4766	0.0533	0.1523	**0.0161**	0.8272	0.2940	**0.0021**	0.6455
	n_w_LPPD(H)	**0.0056**	**0.0036**	**0.0054**	**0.0066**	0.0800	**0.0367**	0.6483	**0.0189**	**0.0043**	0.1028	0.1600	**0.0003**	0.0800
RF	n_w_LPPD(S)	**0.0393**	**0.0002**	**0.0516**	**0.0186**	0.2943	0.1419	0.3046	**0.0450**	**0.0487**	0.9994	0.1890	**0.0197**	0.2995
	n_w_LPPD(H)	**0.0029**	**0.0002**	**0.0005**	**0.0213**	0.1045	0.0581	0.6483	**0.0039**	**0.0051**	0.3736	0.9998	**0.0077**	0.1147

TABLE 8. Comparison with the representative feature selection algorithms.

	AD	PD	Statlog	Pen-Digits	LSVT	Urban	Vehicle	LRS	musk	amazon
N_DR (%)	53.33	61.44	86.32	97.77	79.524	78.49	77.45	67.20	77.09	46.80
P_Value (%)	45.33	56.90	23.85	10.40	60.95	18.84	35.82	48.11	70.13	2.00
Correlation coefficient (%)	46.67	61.55	79.53	96.98	78.57	62.84	73.83	53.37	69.87	10.00
LASSO (%)	50.00	61.38	85.32	97.09	79.52	68.62	76.88	62.17	75.82	23.20
n_w_LPPD(S) (%)	**66.67**	**65.12**	**87.30**	**97.80**	**87.62**	**80.09**	76.60	**70.97**	**84.43**	**60.10**
n_w_LPPD(H) (%)	**66.67**	**65.40**	**87.38**	**97.89**	**90.95**	**81.60**	**78.87**	**73.60**	**86.20**	**60.25**

As shown in TABLE 6, the algorithm proposed in this paper outperforms the other algorithms on the AD, Statlog and LSVT datasets. On other datasets, the proposed algorithm achieves the same classification effectiveness as other algorithms. An interesting phenomenon is that n_w_LPPD(S) is the highest-performing algorithm on the LSVT dataset. A possible explanation for this phenomenon is that in the case of the random forest as a classifier, there is no significant difference between the two modes of the affinity matrix. This phenomenon can also be found in ELM and SVM.

It can be seen from TABLE 3-6 that the proposed algorithm achieves satisfactory classification results regardless of the classifier selected. This advantage is especially apparent on small sample datasets such as AD, PD, and LSVT. The possible reasons are as follows: 1) The added micro-noise can improve the generalization ability of the model, especially for small sample sizes. 2) The datasets have large intraclass variances, which are considered by the objective function. 3) Compared with the method of randomly dividing the training set in the traditional machine learning algorithm, multiple sample subsets obtained by sampling the training set multiple times may have an improved ability to characterize the distribution of the dataset. 4) The introduction of ensemble projection matrix via Bayesian fusion helps to construct the ensemble LPP projection matrix, thereby improving the classification stability of the model. At the same time, it should be noted that on some datasets, such as the Vehicle dataset, the algorithm is not very effective. One possible reason is that the added micro-noise does not match the training sets. Another possible reason is that these datasets have small intraclass variance, whereas the proposed algorithm is more effective on datasets with large intraclass variances.

Algorithm 1 ***The Proposed n_w_LPPD Algorithm***

Input: The training dataset $X_{train} \rightarrow \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$, the number of sampled samples n_s, the number of subspaces p, the regularization coefficients μ and γ, the local numbers k_m and k_{mc}, and the new subspace's dimensionality k.

For i = 1 to p do
  From $X_{train} \rightarrow \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$, randomly choose a training set of size n_s, and name this set X^n_{train};
  Generate noise according to Eq. (8), and add it to X^n_{train}, resulting in $X^{'n}_{train}$
End for
For i = 1 to p do
  Calculate the sample center $\overline{x^{'}}$ and the class center $\overline{x^{'(c)}}$ of $X^{'n}_{train}$;
  Calculate the $\overline{x^{'}_b}$, $\overline{x^{'(c)}_b}$ and $\overline{x^{'(c)}_w}$;
  Calculate the scatter matrix S^*_{WL} and S^*_{BL};
  Construct the affinity matrix A based on Eq. (12) or (13);
  Calculate the diagonal matrix Q ;
  Calculate the Laplacian matrix L ;
  Solve the projection matrix U^n with Eq. (16);
  Obtain the projection matrix U^n_k
End for
Calculate LM with Eq. (18);
Calculate α_n via Eq. (19);
Calculate the projection matrix U^{final}_k with Eq. (20).

Output: The projection matrix U^{final}_k

As shown in Table 7, in the AD dataset, no matter which classifier is used, there is a significant difference between the proposed algorithm (n_w_LPPD) and most of the dimensionality reduction algorithms used in this paper. These results directly indicate the superiority of the proposed algorithm.

In order to further verify the effectiveness of the proposed algorithm, some representative feature selection algorithms are compared with the proposed algorithm. The representative feature selection algorithms include P_value, correlation coefficient, and LASSO. When the correlation coefficient is used for feature selection, the number of features selected is consistent with the number of features of the algorithm proposed in this paper. When LASSO is conducted for feature selection, it uses ten-fold cross-validation to obtain the best λ (penalty factor) and then determines its dimension according to the degrees of freedom. The classifier used in the experiment is an SVM (linear kernel function). The results are recorded in TABLE 8.

As seen from TABLE 8, the proposed algorithm has the highest classification accuracy in most cases. This also directly proves the superiority of the proposed algorithm. In TABLE 8, the classification accuracy of the P_Value method is often very low. The probable reason is that when

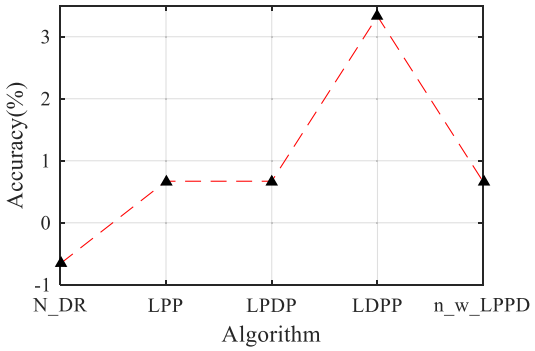

FIGURE 1. The impact of noise injection on the generalization capabilities of the model.

P_value is used for feature selection, the sample features between different categories are required to have significant differences, resulting in fewer features being selected. Especially in the case of multiclassification, few features are selected, resulting in little useful classification information and low accuracy.

C. VERIFYING THE VALIDITY OF THE THREE INNOVATIONS OF THE PROPOSED METHOD PRESENTED IN THIS PAPER.

In this section, the effectiveness of noise injection, LPPD (n_LPPD without noise embedded), and ensemble projection matrix via Bayesian fusion will be verified separately.

1) NOISE INJECTION

In this section, the impact of noise injection on the generalization capabilities of the model is explored. The experiment is performed on the AD dataset, the classifier is an SVM with a linear kernel, and the algorithm works in simple-minded mode. The experimental results are shown in FIGURE 1. The points on the line indicate the change in effect before and after noise injection. A value greater than zero indicates that the noise injection has a positive effect, and vice versa.

FIGURE 1 shows the performance changes of several reduction algorithms before and after noise injection. In Figure 1, in terms of N_DR, the classification accuracy after noise injection decreases slightly. However, when the feature extraction algorithm works, noise injection always plays a positive role in the case of simple-minded mode.

2) LPPD

In this section, the impact of LPPD on the classification of datasets with large intraclass variances is discussed. LPDP and LDPP are used for comparison because their objective functions have greater similarities. The experiment is performed on the AD and LSVT datasets, the classifier is SVM with linear kernel, and the algorithm works in heat kernel mode. Each experiment is repeated five times, and the experimental results are shown in FIGURE 2.

In FIGURE 2, regardless of whether the AD or LSVT dataset is used, LPPD achieves the highest classification

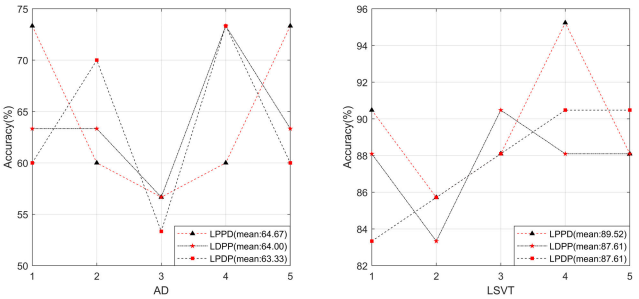

FIGURE 2. Experimental results of several algorithms on the AD and LSVT datasets.

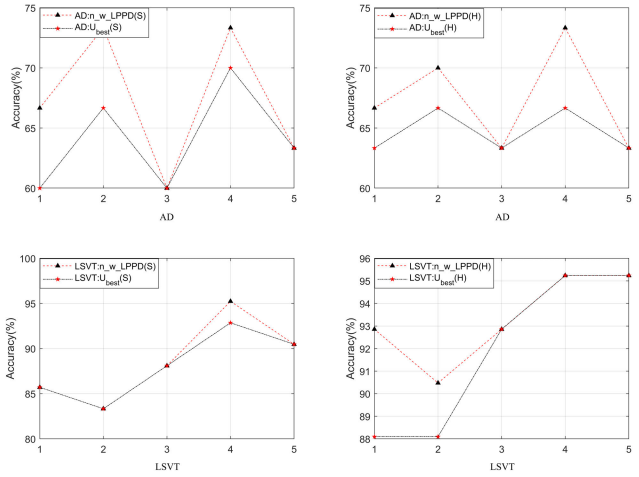

FIGURE 3. The effect of the Bayesian fusion algorithm on classification accuracy.

accuracy in most cases compared to LPDP and LDPP and has the highest average classification accuracy. These results demonstrate that the LPPD algorithm proposed in this paper is more suitable than the other two algorithms for processing such datasets with large intraclass variance.

3) ENSEMBLE PROJECTION MATRIX VIA BAYESIAN FUSION

In this section, the effect of ensemble projection matrix via Bayesian fusion on improving classification accuracy is analyzed. The experiment is performed on the AD and LSVT datasets, and the classifier is an SVM with a linear kernel. The experimental results are shown in FIGURE 3. Each experiment for each dataset is repeated five times, and each result was recorded as one point on the polyline. In FIGURE 3, U_{best}(S) represents the best classification result of the training set after U_{best} mapping in simple-minded mode, where $U_{best} = U_k^i, i = \arg\max_n Acc(U_k^n), n = 1, 2, 3$ and $Acc(U_k^n)$ means the accuracy of the data after U_k^n mapping. Similarly, U_{best}(H) means the best classification result of the training set after U_{best} mapping in heat kernel mode.

It can be seen from FIGURE 3 that the classification accuracy of the proposed algorithm using Bayesian fusion to construct the projection matrix is always not lower than U_{best},

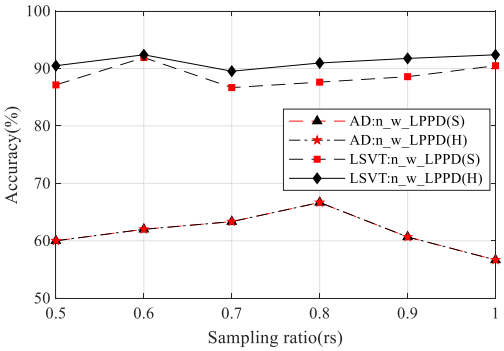

FIGURE 4. The effect of the subspace sampling ratio on the proposed algorithm.

whether in different datasets or using different modes. Thus, the use of Bayesian fusion to construct a projection matrix is more effective than the use of the single matrix. These results directly verify the effectiveness of the ensemble projection matrix via Bayesian fusion.

D. ANALYSIS OF THE INFLUENCE OF PARAMETERS ON THE PERFORMANCE OF THE PROPOSED ALGORITHM

The influence of some parameters on the proposed algorithm will be explored in this section. First, the effect of the subspace sampling ratio (r_s, defined as n_s/n_{train}, where n_s is the number of samples and n_{train} is the number of training datasets) on the proposed algorithm will be explored. Then, the influence of the dimension and the penalty coefficient on the proposed algorithm will be discussed. Finally, the impact of noise injection on the generalization capabilities of the model will be explored. All the above mentioned experiments in this part are performed on the AD and LSVT datasets, and the classifier is an SVM with a linear kernel.

FIGURE 4 shows the impact of the sampling ratio of the proposed algorithm on AD and LSVT datasets.

Figure 4 shows that the line graphs of the two methods on the AD dataset are completely coincident. The classification accuracy of AD starts to increase steadily with the increase in the sampling rate, but when it exceeds a limit ($r_s = 0.8$), the accuracy begins to decrease. Therefore, it can be concluded that the sampling rate has an impact on the proposed algorithm. Optimal performance is achieved when the sampling rate is equal to 0.8. On the LSVT dataset, the two methods showed the same trend and achieves optimal output performance at $r_s = 0.6$.

FIGURE 5 shows the situation where the accuracy of the proposed algorithm changes as the dimensions change.

It can be seen from FIGURE 5a that the classification accuracy of the n_w_LPPD(S) algorithm initially increases with dimensionality and then tends to be stable. The best performance was achieved when the dimensionality was 20. The classification accuracy of the n_w_LPPD(H) algorithm increases with dimensionality. However, after 20 dimensions, the growth rate decreases. As shown in FIGURE 5b,

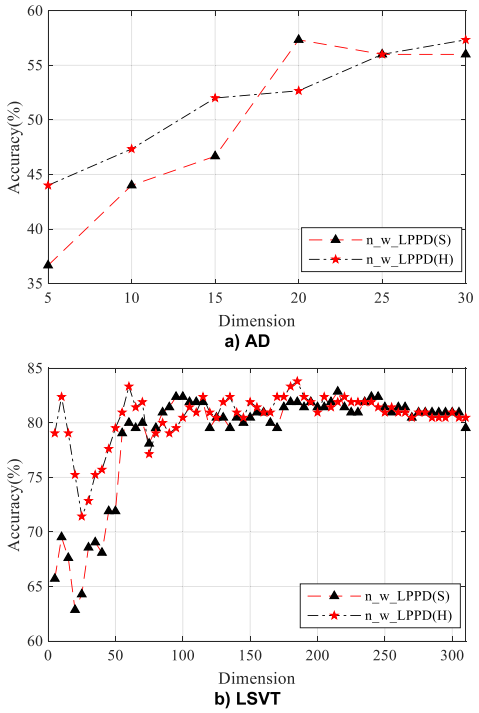

FIGURE 5. The effect of dimension on the proposed algorithm.

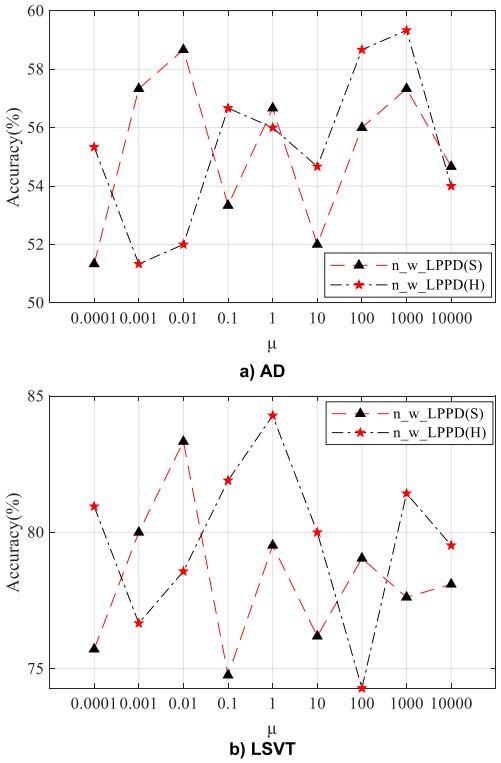

FIGURE 6. The effect of the penalty coefficient μ on the proposed algorithm.

the classification accuracy increases with dimensionality but gradually stabilizes after 95 dimensions.

FIGURE 6 shows the effect of μ on the proposed algorithm.

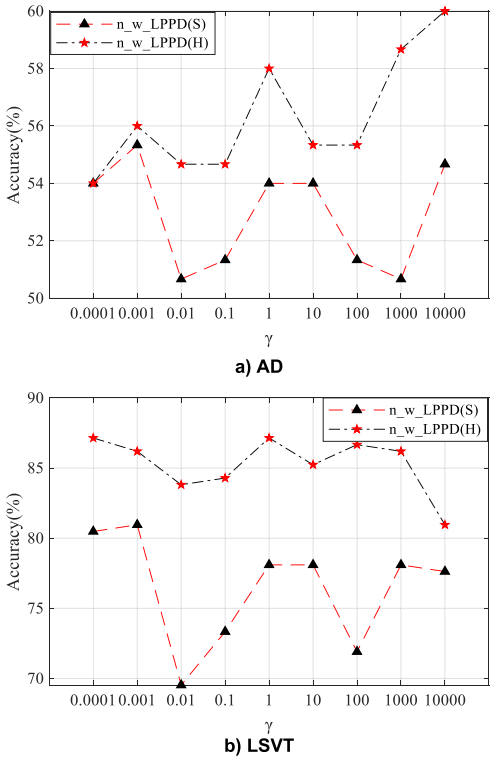

FIGURE 7. The effect of the penalty coefficient γ on the proposed algorithm.

In FIGURE 6a, the n_w_LPPD(S) algorithm fluctuates with the change in μ, but it can still be seen that this fluctuation shows a decreasing trend. The algorithm performs best when μ is 0.01. The n_w_LPPD(H) algorithm fluctuates with the change in μ but achieves the best performance when μ is equal to 1000. In FIGURE 6b, the accuracy of the n_w_LPPD(S) algorithm is similar to a damped oscillation curve, and it performs optimally when μ is equal to 0.01. The n_w_LPPD(H) algorithm changes with μ as a bell-shaped function, and its optimal output is at $\mu = 1$.

FIGURE 7 shows the effect of γ on the proposed algorithm.

In FIGURE 7a, the n_w_LPPD(S) algorithm fluctuates with changes in γ, but in general, the fluctuation is always within a range. The n_w_LPPD(H) algorithm fluctuates as γ changes, but there is an increasing trend overall. The two algorithms perform best when γ is equal to 0.001 and 10000, respectively. In FIGURE 7b, the n_w_LPPD(S) algorithm also has the same phenomenon as FIGURE 7a. However, the n_w_LPPD (H) algorithm has the opposite trend as FIGURE 7a shows.

FIGURE 8 shows the effect of the relevant parameters of the classifier on the experimental results. Since the AD dataset is not sensitive to changes in the parameters of the classifier, this section merely discusses the effect of the parameters on the LSVT dataset.

In FIGURE 8, as far as the ELM classifier is concerned, as the number of hidden layers changes, the accuracy of

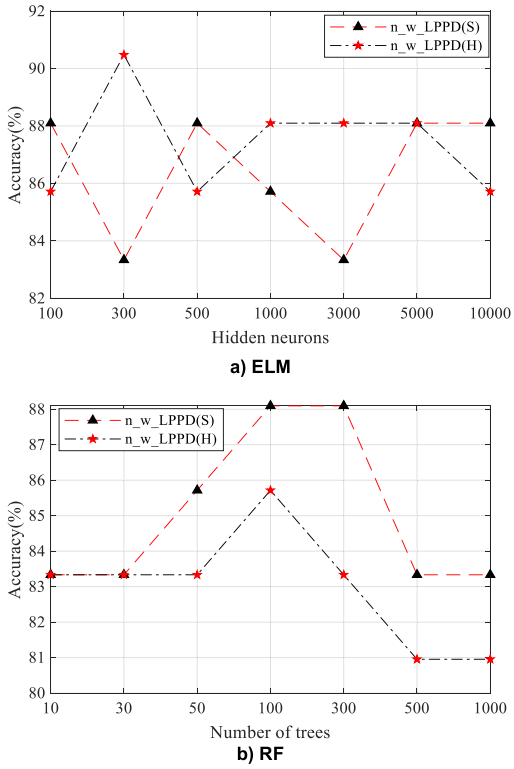

FIGURE 8. The effect of the classifier parameters on the experimental results.

n_w_LPPD(S) fluctuates with the number of hidden neurons. The accuracy of n_w_LPPD(H) first increases and then decreases as the number of hidden neurons increases, ultimately becoming stable. In terms of RF, both methods have the same trend: as the number of trees increases, the accuracy first increases, then decreases, ultimately becoming stable.

V. DISCUSSION AND CONCLUSION

An improved LPP algorithm, n_w_LPPD, was proposed in this paper. The major innovations are as follows: 1) Micro-noise is added to the training sets to improve the generalization ability of the model. 2) The objective function of the n_w_LPPD gives additional attention to large variance within classes, thereby achieving obvious advantages over other dimensionality reduction algorithms when faced with such datasets. 3) Ensemble projection matrix via Bayesian fusion helps to improve the classification stability of the model. The advantages above contribute to the increased accuracy and stability of the proposed algorithm.

Various public datasets were used to verify the performance of the proposed method. The experimental results showed that the proposed algorithm is effective. Noise injection always plays a positive role in the process of feature dimension reduction. The LPPD algorithm proposed in this paper is best suited for processing datasets with large intraclass variance. The proposed ensemble projection matrix via Bayesian fusion mechanism can construct a feature projection matrix to effectively improve classification accuracy. In most cases, regardless of the selected classifier, the algorithm proposed in this paper is significantly better than the other tested algorithms in terms of classification accuracy. Especially for the datasets with small sample sizes, such as the AD and LSVT datasets, the classification accuracy is improved by at least 10% compared to N_DR; even when compared with other dimensionality reduction algorithms, our algorithm achieves significantly superior classification performance.

The proposed algorithm has achieved certain improvements in classification accuracy, but in light of some studies that integrate locality into discriminant analysis [49], [50], there is still work to do in the future. The introduction of Bayesian fusion increases the runtime of the proposed algorithm. Therefore, in future research, it is necessary to further improve the dimensionality reduction efficiency of the proposed algorithm.

ACKNOWLEDGMENT

The authors thank the editor and reviewers for their valuable comments and suggestions.

REFERENCES

[1] W. Wang, Y. Yan, F. Nie, S. Yan, and N. Sebe, "Flexible manifold learning with optimal graph for image and video representation," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2664–2675, Jun. 2018.
[2] H. Huang, G. Shi, H. He, Y. Duan, and F. Luo, "Dimensionality reduction of hyperspectral imagery based on spatial-spectral manifold learning," *IEEE Trans. Cybern.*, to be published.
[3] X. Wang and J. S. Marron, "A scale-based approach to finding effective dimensionality in manifold learning," *Electron. J. Statist.*, vol. 2, no. 3, pp. 127–148, 2008.
[4] L. Tran, D. Banerjee, J. Wang, A. J. Kumar, F. McKenzie, Y. Li, and J. Li, "High-dimensional MRI data analysis using a large-scale manifold learning approach," *Mach. Vis. Appl.*, vol. 24, no. 5, pp. 995–1014, Jul. 2013.
[5] N. Passalis and A. Tefas, "Dimensionality reduction using similarity-induced embeddings," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3429–3441, Aug. 2018.
[6] X. F. He, "Locality preserving projections," Ph.D. dissertation, Dept. College Comput. Sci., Univ. Chicago, Chicago, IL, USA, 2005.
[7] Y. Zhu, Z. Wang, D. Gao, and D. Li, "GMFLLM: A general manifold framework unifying three classic models for dimensionality reduction," *Eng. Appl. Artif. Intell.*, vol. 65, pp. 421–432, Oct. 2017.
[8] G. Yu, H. Peng, J. Wei, and Q. Ma, "Enhanced locality preserving projections using robust path based similarity," *Neurocomputing*, vol. 74, no. 4, pp. 598–605, Jan. 2011.
[9] X. Li, J. Pan, Y. He, and C. Liu, "Bilateral filtering inspired locality preserving projections for hyperspectral images," *Neurocomputing*, vol. 164, pp. 300–306, Sep. 2015.
[10] J. Cheng, Q. Liu, H. Lu, and Y.-W. Chen, "Supervised kernel locality preserving projections for face recognition," *Neurocomputing*, vol. 67, pp. 443–449, Aug. 2005.
[11] G. Feng, D. Hu, D. Zhang, and Z. Zhou, "An alternative formulation of kernel LPP with application to image recognition," *Neurocomputing*, vol. 69, nos. 13–15, pp. 1733–1738, 2006.
[12] Y. Chen, X.-H. Xu, and J.-H. Lai, "Optimal locality preserving projection for face recognition," *Neurocomputing*, vol. 74, no. 18, pp. 3941–3945, Nov. 2011.
[13] W.-L. Chao, J.-J. Ding, and J.-Z. Liu, "Facial expression recognition based on improved local binary pattern and class-regularized locality preserving projection," *Signal Process.*, vol. 117, pp. 1–10, Dec. 2015.
[14] Z. Zhang, F. Li, M. Zhao, L. Zhang, and S. Yan, "Robust neighborhood preserving projection by nuclear/L2,1-norm regularization for image feature extraction," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1607–1622, Apr. 2017.

[15] Y. Zhang, Z. Zhang, S. Li, J. Qin, G. Liu, M. Wang, and S. Yan, "Unsupervised nonnegative adaptive feature extraction for data representation," *IEEE Trans. Knowl. Data Eng.*, to be published.

[16] Z. Yan, Z. Zhao, J. Qin, Z. Li, L. Bing, and L. Fanzhang, "Semi-supervised local multi-manifold Isomap by linear embedding for feature extraction," *Pattern Recognit.*, vol. 76, pp. 662–678, Apr. 2018.

[17] Z. Zhang, Y. Zhang, G. Liu, J. Tang, S. Yan, and M. Wang, "Joint label prediction based semi-supervised adaptive concept factorization for robust data representation," *IEEE Trans. Knowl. Data Eng.*, to be published.

[18] M. Laadjel, S. Al-Maadeed, and A. Bouridane, "Combining Fisher locality preserving projections and passband DCT for efficient palmprint recognition," *Neurocomputing*, vol. 152, pp. 179–189, Mar. 2015.

[19] G.-F. Lu, Z. Lin, and Z. Jin, "Face recognition using discriminant locality preserving projections based on maximum margin criterion," *Pattern Recognit.*, vol. 43, no. 10, pp. 3572–3579, Oct. 2010.

[20] J. Gui, C. Wang, and L. Zhu, "Locality preserving discriminant projections," in *Proc. Int. Conf. Emerg. Intell. Comput. Technol. Appl.*, 2009, pp. 566–572.

[21] L. Zhang, X. Wang, G.-B. Huang, T. Liu, and X. Tan, "Taste recognition in E-tongue using local discriminant preservation projection," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 947–960, Mar. 2019.

[22] Q. Yu, R. Wang, B. N. Li, X. Yang, and M. Yao, "Robust locality preserving projections with cosine-based dissimilarity for linear dimensionality reduction," *IEEE Access*, vol. 5, pp. 2676–2684, Oct. 2016.

[23] J. Liang, C. Chen, Y. Yi, X. Xu, and M. Ding, "Bilateral two-dimensional neighborhood preserving discriminant embedding for face recognition," *IEEE Access*, vol. 5, pp. 17201–17212, 2017.

[24] S.-I. Choi, S.-S. Lee, S. T. Choi, and W.-Y. Shin, "Face recognition using composite features based on discriminant analysis," *IEEE Access*, vol. 6, pp. 13663–13670, Mar. 2018.

[25] F. Zhong, J. Zhang, and D. Li, "Discriminant locality preserving projections based on L1-norm maximization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 11, pp. 2065–2074, Nov. 2014.

[26] Q. Wang, Q. Gao, D. Xie, X. Gao, and Y. Wang, "Robust DLPP with nongreedy ℓ_1 -norm minimization and maximization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 3, pp. 738–743, Mar. 2018.

[27] Y.-J. Deng, H.-C. Li, L. Pan, L.-Y. Shao, Q. Du, and W. J. Emery, "Modified tensor locality preserving projection for dimensionality reduction of hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 2, pp. 277–281, Feb. 2018.

[28] G. E. Hinton, "To recognize shapes, first learn to generate images," *Progr. Brain Res.*, vol. 165, no. 6, pp. 535–547, 2007.

[29] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 855–868, May 2009.

[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2012, vol. 25, no. 2, pp. 1097–1105.

[32] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," Aug. 2017, *arXiv:1708.04896*. [Online]. Available: https://arxiv.org/abs/1708.04896

[33] Z. You, J. Ye, K. Li, Z. Xu, and P. Wang, "Adversarial noise layer: Regularize neural network by adding noise," May 2018, *arXiv:1805.08000*. [Online]. Available: https://arxiv.org/abs/1805.08000

[34] K. Weiss, T. M. Khoshgoftaar, and D. D. Wang, "A survey of transfer learning," *J. Big Data*, vol. 3, no. 1, p. 9, 2016.

[35] R. Vidal, Y. Ma, and S. S. Sastry, "Principal component analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1945–1959, 2016.

[36] P. N. Belhumeur, J. P. Hespanha, and D. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.

[37] X. He and P. Niyogi, "Locality preserving projections," in *Advances in Neural Information Processing System*, vol. 16. Cambridge, MA, USA: MIT Press, 2003, pp. 100–115.

[38] J. Gui, W. Jia, L. Zhu, S.-L. Wang, and D. Huang, "Locality preserving discriminant projections for face and palmprint recognition," *Neurocomputing*, vol. 73, no. 13, pp. 2696–2707, 2010.

[39] Y. Dong, B. Du, L. Zhang, and L. Zhang, "Dimensionality reduction and classification of hyperspectral images using ensemble discriminative local metric learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 5, pp. 2509–2524, May 2017.

[40] S. Huang, A. Elgammal, J. Lu, and D. Yang, "Cross-speed gait recognition using speed-invariant gait templates and globality–locality preserving projections," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 10, pp. 2071–2083, Oct. 2015.

[41] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, vol. 14, no. 6, pp. 585–591.

[42] D. Dua and E. K. Taniskidou. (2017). UCI machine learning repository. University of California, School of Information and Computer Science, Irvine, CA, USA. [Online]. Available: http://archive.ics.uci.edu/ml

[43] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artif. Intell.*, vol. 89, nos. 1–2, pp. 31–71, 1997.

[44] X. Tan, Y. Liu, Y. Li, P. Wang, X. Zeng, F. Yan, and X. Li, "Localized instance fusion of MRI data of Alzheimer's disease for classification based on instance transfer ensemble learning," *Biomed. Eng. Online*, vol. 17, no. 1, p. 49, Dec. 2018.

[45] B. E. Sakar, M. E. Isenkul, C. O. Sakar, A. Sertbas, F. Gurgen, S. Delil, H. Apaydin, and O. Kursun, "Collection and analysis of a Parkinson speech dataset with multiple types of sound recordings," *IEEE J. Biomed. Health Inform.*, vol. 17, no. 4, pp. 828–834, Jul. 2013.

[46] F. Hoti and L. Holmström, "A semiparametric density estimation approach to pattern classification," *Pattern Recognit.*, vol. 37, no. 3, pp. 409–419, Mar. 2004.

[47] A. Tsanas, M. A. Little, C. Fox, and L. O. Ramig, "Objective automatic assessment of rehabilitative speech treatment in Parkinson's disease," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 1, pp. 181–190, Jan. 2014.

[48] B. Johnson and Z. Xie, "Classifying a high resolution image of an urban area using super-object information," *ISPRS J. Photogramm. Remote Sens.*, vol. 83, pp. 40–49, Sep. 2013.

[49] Q. Ye, L. Fu, Z. Zhang, H. Zhao, and M. Naiem, "Lp- and Ls-Norm distance based robust linear discriminant analysis," *Neural Netw.*, vol. 105, pp. 393–404, Sep. 2018.

[50] Z. Zhang, M. Zhao, and T. W. S. Chow, "Constrained large margin local projection algorithms and extensions for multimodal dimensionality reduction," *Pattern Recognit.*, vol. 45, no. 12, pp. 4466–4493, Dec. 2012.

YUCHUAN LIU received the bachelor's degree in communication engineering from the Southwest University of Science and Technology, China, in 2015. He is currently pursuing the Ph.D. degree with Chongqing University, Chongqing, China. His research interests include dimensionality reduction, pattern recognition, and machine learning.

XIAOHENG TAN received the B.E. and Ph.D. degrees in electrical engineering from Chongqing University, Chongqing, China, in 1998 and 2003, respectively. From 2008 to 2009, he was a Visiting Scholar with The University of Queensland, Brisbane, QLD, Australia. He is currently a Professor with the School of Microelectronics and Communication Engineering, Chongqing University. His current research interests include modern communications technologies and systems, communications signal processing, pattern recognition, and machine learning.

YONGMING LI received the M.S. and Ph.D. degrees in circuits and systems from Chongqing University, China, in 2003 and 2007, respectively. He was a Visiting Scholar with Pennsylvania State University, USA, from 2008 to 2009. He held a postdoctoral position with Carnegie Mellon University, USA, from February 2009 to December 2009. He is currently a Professor with the School of Microelectronics and Communication Engineering, Chongqing University. His current research interests include signal processing, pattern recognition, data mining, and machine learning. He serves as an Editorial Board Member for IEEE Access.

PIN WANG received the M.S. degree from Chongqing University, China, in 2003, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2008, where she held a postdoctoral position, from 2008 to 2009. From 2009 to 2010, she was funded by the National Energy Laboratory Postdoctoral Fund to Conduct Postdoctoral Research at the University of Pittsburgh. She is currently an Associate Professor with Chongqing University. Her current research interests include hyperspectral imaging and detection, intelligent information processing, and big data analysis decision.