Local Adaptive Projection Framework for Feature Selection of Labeled and Unlabeled Data

Xiaojun Chen, *Member, IEEE,* Guowen Yuan, Wenting Wang, Feiping Nie, Xiaojun Chang, and Joshua Zhexue Huang

***Abstract*—Most feature selection methods first compute a similarity matrix by assigning a fixed value to pairs of objects in the whole data or to pairs of objects in a class or by computing the similarity between two objects from the original data. The similarity matrix is fixed as a constant in the subsequent feature selection process. However, the similarities computed from the original data may be unreliable, because they are affected by noise features. Moreover, the local structure within classes cannot be recovered if the similarities between the pairs of objects in a class are equal. In this paper, we propose a novel local adaptive projection (LAP) framework. Instead of computing fixed similarities before performing feature selection, LAP simultaneously learns an adaptive similarity matrix S and a projection matrix W with an iterative method. In each iteration, S is computed from the projected distance with the learned W and W is computed with the learned S. Therefore, LAP can learn better projection matrix W by weakening the effect of noise features with the adaptive similarity matrix. A supervised feature selection with LAP (SLAP) method and an unsupervised feature selection with LAP (ULAP) method are proposed. Experimental results on eight data sets show the superiority of SLAP compared with seven supervised feature selection methods and the superiority of ULAP compared with five unsupervised feature selection methods.**

***Index Terms*—Local structure learning, sparse feature selection, supervised feature selection, unsupervised feature selection.**

I. Introduction

FEATURE selection is very important for high-dimensional data analysis, because it can help to remove irrelevant features without performance deterioration [1]. In contrast to other dimension-reduction techniques such as principal component analysis (PCA), feature selection techniques do not change the original features but merely select a subset of features that are most useful. Feature selection techniques preserve the original semantics of the dimensions, thereby offering the advantage of interpretability by a domain expert. Feature selection can be conducted in supervised, unsupervised, or semisupervised manner depending on whether the label information is available. In the supervised feature selection, feature relevance can be evaluated according to the correlations of features computed according to the class labels, e.g., Fisher score [2], Relief-F [3], [4], robust feature selection (RFS) [5], discriminative least squares regression [6], global redundancy minimization [7], stratified feature selection [8], and robust discriminant regression [9]. In the unsupervised feature selection, without label information, feature relevance can be evaluated by feature dependence or similarity, e.g., Laplacian Score [10], robust spectral feature selection [11], structured optimal graph feature selection (SOGFS) [12], robust joint graph sparse coding [13], and adaptive neighbors feature selection [14]. In the semisupervised feature selection, both labeled and unlabeled data are used, e.g., sSelect [15], semisupervised feature selection via spline regression [16], and rescaled linear square regression [17].

In most feature selection methods, a similarity matrix is initially constructed and fixed for the subsequent feature selection process. For supervised feature selection, we often assign a fixed value to all pairs of objects or the pairs of objects in a class. However, the inner class structure within such a similarity matrix is lost, and we may obtain undesired results on the *multimodal* data [18]. For unsupervised feature selection, we often compute the similarities from the original data. However, such similarities are affected by noise features and may mislead the feature selection methods into recovering wrong local structure.

To address these problems, we propose a new feature selection framework, called local adaptive projection (LAP). Instead of computing a fixed similarity matrix before performing feature selection, LAP learns an adaptive similarity matrix **S** and a projection matrix **W** simultaneously with an iterative method. In each iteration, **S** is computed from the projected distance with the learned **W**, and **W** is computed with the learned **S**. Therefore, LAP can better rank the features by weakening the affection of noise features with the adaptive similarity matrix. A supervised feature selection with LAP (SLAP) method and an unsupervised feature selection with LAP (ULAP) method are proposed. Experimental results on eight data sets show the superiority of SLAP in comparison with seven supervised feature selection methods and the superiority of ULAP in comparison with five unsupervised feature selection methods.

The rest of this paper is organized as follows. Section II presents the notation and definitions used in this paper, and Section III provides a brief survey of related work. We present

Manuscript received September 22, 2017; revised January 29, 2018 and April 18, 2018; accepted April 22, 2018. Date of publication May 18, 2018; date of current version November 16, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61305059, Grant 61773268, and Grant U1636202 and in part by Tencent Rhinoceros Birds—Scientific Research Foundation for Young Teachers of Shenzhen University. *(Corresponding author: Feiping Nie.)*

X. Chen, G. Yuan, W. Wang, and J. Z. Huang are with the College of Computer Science and Software, Shenzhen University, Shenzhen 518060, China (e-mail: xjchen@szu.edu.cn; 243864952@qq.com; ucahwwa@ucl.ac.uk; zx.huang@szu.edu.cn).

F. Nie is with the School of Computer Science and Center for OPTical IMagery Analysis and Learning, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: feipingnie@gmail.com).

X. Chang is with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: cxj273@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2018.2830186

the LAP framework in Section IV. In Section V, an SLAP method and a ULAP method are proposed. The experimental results are presented to verify the effectiveness of both SLAP and ULAP in Sections VI and VII. Conclusions and future work are given in Section VIII.

II. Notation and Definitions

We summarize the notation and definitions used in this paper. Matrices are written as boldface uppercase letters. Vectors are written as boldface lowercase letters. For matrix $\mathbf{M} = (m_{ij})$, its ith row is denoted by $\mathbf{m}^i$, and its jth column is denoted by $\mathbf{m}_j$. The trace of $\mathbf{M}$ is denoted by $Tr(\mathbf{M})$. The $\ell_{2,1}$-norm of the matrix $\mathbf{M} \in \mathbb{R}^{n\times m}$ is defined as $\|\mathbf{M}\|_{2,1} = \sum_{i=1}^{n}\sqrt{\sum_{j=1}^{m} m_{ij}^2}$.

III. Related Work

Local discriminant analysis (LDA) is a popular dimension-reduction method that aims to find a linear projection matrix that maps data in high-dimensional space into low-dimensional space, in which the between-class scatter is maximized while the within-class scatter is minimized [19]. Let $\mathbf{X} \in \mathbb{R}^{d\times n}$ be a data set with n objects $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, where $\mathbf{x}_i \in \mathbb{R}^{d\times 1}$. $\mathbf{X}$ is associated with c classes $\mathcal{O} = \{\mathbf{O}_1, \ldots, \mathbf{O}_c\}$ in which $\mathbf{O}_l$ consists of all objects in the lth class. Let μ_l be the mean vector of the lth class and μ be the overall mean vector of the original data. LDA is a supervised subspace learning method that aims to find a projection $\mathbf{W} \in \mathbb{R}^{n\times d}$, such that the between-class scatter is maximized while the within-class scatter is minimized, i.e.,

$$\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} Tr((\mathbf{W}^T\mathbf{S}_w\mathbf{W})^{-1}(\mathbf{W}^T\mathbf{S}_b\mathbf{W})) \tag{1}$$

where $\mathbf{S}_w$ is the within-class scatter matrix defined as

$$\mathbf{S}_w = \sum_{l=1}^{c}\sum_{\mathbf{x}_i\in\mathbf{O}_l}(\mathbf{x}_i - \mu_l)(\mathbf{x}_i - \mu_l)^T \tag{2}$$

and $\mathbf{S}_b$ is the between-class scatter matrix defined as

$$\mathbf{S}_b = \sum_{l=1}^{c}(\mu_l - \mu)(\mu_l - \mu)^T. \tag{3}$$

Following LDA, many incremental works have been reported, e.g., uncorrelated LDA and orthogonal LDA [20], local LDA [21], semisupervised LDA [22], and sparse LDA [23].

To handle *multimodal* data well, it is important to preserve the local structure of the data. The unsupervised dimension-reduction method locality-preserving projection (LPP) finds the local structure by keeping nearby data pairs in the original space close in the embedding space [24]. LPP finds a projection matrix $\mathbf{W} \in \mathbb{R}^{n\times d}$ by solving the following problem:

$$\min_{\mathbf{W}^T\mathbf{X}\mathbf{D}_A\mathbf{X}^T\mathbf{W}=\mathbf{I}} \frac{1}{2}\sum_{i,j=1}^{n} A_{ij}\|\mathbf{W}^T(\mathbf{x}_i - \mathbf{x}_j)\|^2 \tag{4}$$

where $\mathbf{A} = \{a_{ij}\}_{i,j=1}^{n}$ is a sparse affinity matrix computed from the original data, and $\mathbf{D}_A \in \mathbb{R}^{d\times d}$ is a diagonal matrix in which the jth diagonal element $d_{jj} = \sum_{i=1}^{n} a_{ji}$. a_{ij} is defined as

$$a_{ij} = \begin{cases} e^{\frac{-\|\mathbf{x}_i-\mathbf{x}_j\|^2}{t}} & \text{if } \mathbf{x}_i \in \mathcal{N}_k(j) \text{ or } \mathbf{x}_j \in \mathcal{N}_k(i) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

or

$$a_{ij} = \begin{cases} 1 & \text{if } \|\mathbf{x}_i - \mathbf{x}_j\|^2 < \varepsilon \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where $\mathcal{N}_k(j)$ is the k-nearest neighbors of $\mathbf{x}_j$, and ε and t are parameters.

He *et al.* [10] further improved LPP for feature selection and proposed a Laplacian score method that evaluates the features according to their locality-preserving power. The Laplacian score of the rth feature is defined as

$$L_r = \frac{\widetilde{\mathbf{f}}_r^T\mathbf{L}_A\widetilde{\mathbf{f}}_r}{\widetilde{\mathbf{f}}_r^T\mathbf{D}_A\widetilde{\mathbf{f}}_r} \tag{7}$$

where $\mathbf{L}_A = \mathbf{D}_A - \mathbf{A}$ is the graph Laplacian. $\widetilde{\mathbf{f}}_r = \mathbf{f}_r - (\mathbf{f}_r^T\mathbf{D}_A\mathbf{1})/(\mathbf{1}^T\mathbf{D}_A\mathbf{1})\mathbf{1}$, where $\mathbf{f}_r$ is the rth feature.

Motivated by both LPP and LDA, Sugiyama *et al.* [21] proposed a local Fisher discriminant analysis (LFDA). The LFDA preserves the local structure by maximizing the local between-class separability and minimizing the local within-class scatter. The objective function of the LFDA is defined as

$$\max_{\mathbf{W}} Tr((\mathbf{W}^T\overline{\mathbf{S}}^w\mathbf{W})^{-1}\mathbf{W}^T\overline{\mathbf{S}}^b\mathbf{W}) \tag{8}$$

where $\overline{\mathbf{S}}^w$ is the local within-class scatter matrix defined as

$$\overline{\mathbf{S}}^w = \frac{1}{2}\sum_{i,j=1}^{n}\overline{a}_{ij}^w(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \tag{9}$$

where

$$\overline{a}_{ij}^w = \begin{cases} \frac{a_{ij}}{|\mathbf{O}_l|} & \text{if } \mathbf{x}_i, \mathbf{x}_j \in \mathbf{O}_l \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

$\overline{\mathbf{S}}^b$ is the local between-class scatter matrix defined as

$$\overline{\mathbf{S}}^b = \frac{1}{2}\sum_{i,j=1}^{n}\overline{a}_{ij}^b(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \tag{11}$$

where

$$\overline{a}_{ij}^b = \begin{cases} a_{ij}\left(\frac{1}{n} - \frac{1}{|\mathbf{O}_l|}\right) & \text{if } \mathbf{x}_i, \mathbf{x}_j \in \mathbf{O}_l \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

Here. a_{ij} in (10) and (12) has the same meanings as the sparse affinity matrix in LPP.

Gu *et al.* [25] proposed a joint feature selection and subspace learning (FSSL) method that minimizes the graph-preserving criterion and uses the $\ell_{2,1}$ norm of the projection matrix for regularization. FSSL finds a projection matrix $\mathbf{W} \in \mathbb{R}^{n\times d}$ by solving the problem

$$\min_{\mathbf{W}^T\mathbf{X}\mathbf{D}_A\mathbf{X}^T\mathbf{W}=\mathbf{I}} [\|\mathbf{W}\|_{2,1} + \gamma Tr(\mathbf{W}^T\mathbf{X}\mathbf{L}_A\mathbf{X}^T\mathbf{W})] \tag{13}$$

where the affinity matrix $\mathbf{A}$ can be computed in the same way as the graph embedding framework [26]: PCA style, in which $a_{ij} = (1/n)$ for $i \neq j$; LDA style, in which $a_{ij} = (\delta_{\mathbf{O}_i,\mathbf{O}_j}/n_{\mathbf{O}_i})$, where $n_{\mathbf{O}_i}$ is the number of objects in the class that the ith object belongs to and the binary value $\delta_{\mathbf{O}_i,\mathbf{O}_j} = 1$ indicates that the ith and jth objects are in the same class; or LPP style, in which $a_{ij} = e^{(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/t)}$ if $\mathbf{x}_i \in \mathcal{N}_k(\mathbf{x}_j)$ or $\mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_i)$, or 0 otherwise.

In summary, the above feature selection methods first construct a fixed similarity matrix for the subsequent feature selection process. In general, there are two ways to construct a similarity matrix.

1) *Assigning a Fixed Value to All Pairs of Objects or the Pairs of Objects in the Same Class:* For example, the similarity between any two objects is set as $1/n$ in FSSL-PCA (FSSL with PCA-type adjacency matrix), where n is the number of objects. In FSSL-LDA (FSSL with LDA-type adjacency matrix), the similarity between two objects in a class $\mathbf{O}_l$ is set to $(1/|\mathbf{O}_l|)$, where $|\mathbf{O}_l|$ is the number of objects in $\mathbf{O}_l$. However, with the above similarity matrices, the local structure within classes cannot be recovered by the feature selection methods, which tends to produce undesired results.
2) *Computing the Similarities From the Original Data:* For example, FSSL-LPP (FSSL with LPP-type adjacency matrix) uses the heat kernel to compute the similarity between two objects. Spectral feature selection uses an radial basis function kernel to compute the similarity between objects without class label information [27]. However, such similarities are affected by noise features and may mislead the feature selection methods into recovering the wrong local structures.

In this paper, we propose a new feature selection framework that simultaneously performs feature selection and adaptive local structure learning.

IV. Local Adaptive Projection Framework

Let $\mathbf{X} \in \mathbb{R}^{d\times n}$ be a data set with n objects $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, and let $\{F_1, \ldots, F_d\}$ be d features of $\mathbf{X}$. According to the theory of manifold learning, there always exists a low-dimensional manifold that can express the structure of high-dimensional data. In this paper, we want to find a linear combination of original features to best approximate the low-dimensional manifold. Let $\mathbf{W} \in R^{d\times m}$ be the projection matrix, where m is the projection dimension, and we can learn $\mathbf{W}$ by solving the following objective function:

$$\min_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} \sum_{i,j=1}^{n} \|\mathbf{W}^T(\mathbf{x}_i - \mathbf{x}_j)\|_2. \tag{14}$$

The term in problem (14) is the sum of pairwise projected distances. Instead of the commonly used Frobenius-norm-based loss function that is sensitive to outliers, we use the $\ell_{2,1}$-norm-based loss function to remove outliers. The constraint $\mathbf{W}^T\mathbf{W} = \mathbf{I}$ is used to make the feature space distinctive after reduction.

To select only a few features, we add an $\ell_{2,1}$-norm-based regularization term to form the following problem:

$$\min_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} \textstyle\sum_{i,j=1}^{n} \|\mathbf{W}^T(\mathbf{x}_i - \mathbf{x}_j)\|_2 + \gamma\,\|\mathbf{W}\|_{2,1}. \tag{15}$$

The sparsity regularization makes $\mathbf{W}$ row sparse and thus suitable for selecting valuable features. γ is the regularization parameter. With the learned $\mathbf{W}$, the importance of d features can be ranked in the descending order of $\{\|\mathbf{w}^1\|_2, \ldots, \|\mathbf{w}^d\|_2\}$.

Clearly, $\|\mathbf{W}^T(\mathbf{x}_i - \mathbf{x}_j)\|_2$ and $\|\mathbf{W}\|_{2,1}$ can be zero in theory; however, they will make (15) nondifferentiable. To avoid this condition, we introduce a sufficiently small constant ϵ, e.g., 10^{-10}, to avoid a zero denominator and rewrite $\|\mathbf{W}^T(\mathbf{x}_i - \mathbf{x}_j)\|_2$ as $\sqrt{\|\mathbf{W}^T(\mathbf{x}_i - \mathbf{x}_j)\|_2^2 + \epsilon}$ and $\|\mathbf{W}\|_{2,1}$ as $\sum_{l=1}^{d}\sqrt{\mathbf{w}^l(\mathbf{w}^l)^T + \epsilon}$. Therefore, we have

$$\min_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} \sum_{i,j=1}^{n} \sqrt{\|\mathbf{W}^T(\mathbf{x}_i - \mathbf{x}_j)\|_2^2 + \epsilon} + \gamma \sum_{l=1}^{d} \sqrt{\mathbf{w}^l(\mathbf{w}^l)^T + \epsilon}. \tag{16}$$

The Lagrangian function of problem (16) is

$$\mathcal{L}(\mathbf{W}, \Lambda) = \sum_{i,j=1}^{n} \sqrt{\|\mathbf{W}^T(\mathbf{x}_i - \mathbf{x}_j)\|_2^2 + \epsilon} + \gamma \sum_{l=1}^{d} \sqrt{\mathbf{w}^l(\mathbf{w}^l)^T + \epsilon} + Tr(\Lambda(\mathbf{W}^T\mathbf{W} - \mathbf{I})) \tag{17}$$

where $\Lambda \in \mathbb{R}^{m\times m}$ is the Lagrangian multiplier. Taking the derivative of $\mathcal{L}(\mathbf{W}, \Lambda)$ with respect to $\mathbf{W}$ and setting the derivative to zero gives

$$\frac{\partial\mathcal{L}(\mathbf{W}, \Lambda)}{\partial\mathbf{W}} = \sum_{i,j=1}^{n} s_{ij} \frac{\partial\|\mathbf{W}^T(\mathbf{x}_i - \mathbf{x}_j)\|^2}{\partial\mathbf{W}} + 2\gamma\,\mathbf{Q}\mathbf{W} + \mathbf{W}\Lambda = 0 \tag{18}$$

where $\mathbf{Q} \in \mathbb{R}^{d\times d}$ is a diagonal matrix with the lth diagonal element defined as

$$q_{ll} = \frac{1}{2\sqrt{\mathbf{w}^l(\mathbf{w}^l)^T + \epsilon}} \tag{19}$$

and $\mathbf{S} \in \mathbb{R}^{n\times n}$ is defined as

$$s_{ij} = \frac{1}{2\sqrt{\|\mathbf{W}^T(\mathbf{x}_i - \mathbf{x}_j)\|_2^2 + \epsilon}}. \tag{20}$$

Equation (18) is difficult to solve because $\mathbf{Q}$ and $\mathbf{S}$ depend on $\mathbf{W}$. In this paper, we propose an iterative method to solve $\mathbf{W}$. We first fix $\mathbf{Q}$ and $\mathbf{S}$ to solve $\mathbf{W}$ by solving the following problem:

$$\min_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} \Big[Tr(\mathbf{W}^T\mathbf{X}\mathbf{L}_S\mathbf{X}^T\mathbf{W}) + \gamma\,Tr(\mathbf{W}^T\mathbf{Q}\mathbf{W})\Big]. \tag{21}$$

Problem (21) can be solved directly to obtain the optimal solution to $\mathbf{W}$ as the m eigenvectors of $\mathbf{X}\mathbf{L}_S\mathbf{X}^T + \gamma\,\mathbf{Q}$ corresponding to the m smallest eigenvalues, where $\mathbf{L}_S = \mathbf{D}_s - \mathbf{S}$ is the Laplacian matrix of $\mathbf{S}$ and $\mathbf{D}_s \in \mathbb{R}^{n\times n}$ is a diagonal matrix with the ith diagonal element as $\sum_{j=1}^{n} s_{ij}$.

Then, with the new $\mathbf{W}$, we update $\mathbf{Q}$ and $\mathbf{S}$ according to (19) and (20). The above algorithm, denoted as LAP, is described in Algorithm 1. In the new method, $\mathbf{W}$, $\mathbf{Q}$, and $\mathbf{S}$ are alternatively updated until convergence of the algorithm.

Theorem 1: The iteration process of Algorithm 1 will monotonically decreases the objective function of problem (16) in each iteration.

The proof of Theorem 1 can be found in the Appendix.

Thus, the alternating optimization process will converge as the number of iterations goes to infinity. Let $\hat{\mathbf{W}}$ be the converged solution. It can be verified that the derivative of (21) with respect to $\mathbf{W}$ is exactly (17), so the converged $\hat{\mathbf{W}}$ will satisfy (18), which is the Karush–Kuhn–Tucker condition of problem (16). Therefore, Algorithm 1 can converge to a local solution of problem (16).

Algorithm 1 LAP: Algorithm to Solve Problem (16)

```
Iutput: Data matrix $\mathbf{X} \in \mathbb{R}^{d \times n}$, initial $\mathbf{Q}$, $\mathbf{S}$, regularization parameter $\gamma$.
Set $t = 0$.
repeat
    Update $\mathbf{W}_{t+1}$ as the $m$ eigenvectors of $(\mathbf{X}\mathbf{L}_{S_t}\mathbf{X}^T + \gamma\,\mathbf{Q}_t)$ corresponding to its $m$ smallest eigenvalues.
    Update the diagonal matrix $\mathbf{Q}_{t+1}$, where the $l$-th diagonal element is $\frac{1}{2\sqrt{\mathbf{w}^l_{t+1}(\mathbf{w}^l_{t+1})^T+\epsilon}}$.
    Update the matrix $\mathbf{S}_{t+1}$, where $s_{ij}$ is defined in Eq. (23).
    Set $t = t + 1$.
until Converges
Output: $\mathbf{W}$.
```

According to the definition of s_{ij} in (20), we can consider s_{ij} as the local similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. Instead of assigning a fixed similarity to two objects, the new method assigns the learned similarity to pairs of objects, which is inversely proportional to the projected distance $\|\mathbf{W}^T(\mathbf{x}_i - \mathbf{x}_j)\|_2^2$. Compared with the existing methods introduced in Section III, the new LAP framework has three advantages: 1) instead of using the fixed affinity matrix, LAP learns local similarities and performs feature selection simultaneously; 2) LAP imposes the $\ell_{2,1}$ norm on both the pairwise projected distances and projection matrix $\mathbf{W}$ to simultaneously preserve the local structure and select only a few features; and 3) LAP makes the learned $\mathbf{W}$ both distinctive and row sparse.

V. Supervised and Unsupervised Feature Selection With Local Adaptive Projection

In this section, we propose two feature selection methods based on the LAP framework, i.e., SLAP and ULAP. Effective optimization algorithms with proved convergence are proposed to optimize the two methods.

1) Supervised Feature Selection With Local Adaptive Projection: Assume that $\mathbf{X}$ is associated with c classes $\mathcal{O} = \{\mathbf{o}_1, \ldots, \mathbf{o}_c\}$ in which $\mathbf{o}_l$ consists of all objects in the lth class. In the unsupervised feature selection task, k-nearest neighbor similarities are commonly used to preserve the local structure. Since the class labels are already known, it is natural to consider only the k-nearest neighbors in the same class. Therefore, we revise problem (16) to form the following objective function for SLAP:

$$\min_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} \sum_{i,j=1}^{n} v_{ij}\,\|\mathbf{W}^T(\mathbf{x}_i - \mathbf{x}_j)\|_2 + \gamma\,\|\mathbf{W}\|_{2,1}. \tag{22}$$

where $v_{ij} = 1$ if $\mathbf{x}_j \in \mathcal{M}_k(\mathbf{x}_i)$ or $\mathbf{x}_i \in \mathcal{M}_k(\mathbf{x}_j)$ and $v_{ij} = 0$ otherwise. $\mathcal{M}_k(\mathbf{x}_i)$ consists of at most k-nearest neighborhoods of $\mathbf{x}_i$ that are in the same class as $\mathbf{x}_i$. The distance between two objects $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined as $\|\mathbf{x}_i - \mathbf{x}_j\|_2$.

According to Theorem 1, it can be verified that problem (22) can be solved by solving (21), but the definition of $\mathbf{S}$ becomes

$$s_{ij} = \begin{cases} \dfrac{1}{2\sqrt{\|\mathbf{W}^T(\mathbf{x}_i - \mathbf{x}_j)\|_2^2 + \epsilon}} & \text{if } \mathbf{x}_j \in \mathcal{M}_k(\mathbf{x}_i) \text{ or } \mathbf{x}_i \in \mathcal{M}_k(\mathbf{x}_j) \\ 0 & \text{otherwise.} \end{cases} \tag{23}$$

From (23), we can see that $\mathbf{S}$ consumes the class label information by setting $s_{ij} = 0$ for objects in different classes. Based on Algorithm 1, we propose an iterative algorithm to obtain the optimal solution of $\mathbf{W}$, which is described in Algorithm 2. In each iteration, $\mathbf{W}$ is calculated with the current $\mathbf{Q}$ and $\mathbf{S}$; then, $\mathbf{Q}$ and $\mathbf{S}$ are updated based on the currently calculated $\mathbf{W}$. The iteration procedure is repeated until the algorithm converges. With the learned $\mathbf{W}$ by SLAP, the importance of d features can be ranked in the descending order of $\{\|\mathbf{w}^1\|_2, \ldots, \|\mathbf{w}^d\|_2\}$. We can select the first r features as the ultimate result. For simplicity, we denote our feature selection method as SLAP in the following. The convergence of SLAP can be verified according to Theorem 1.

Algorithm 2 SLAP: Algorithm to Solve Problem (22)

```
Iutput: Data matrix $\mathbf{X} \in \mathbb{R}^{d \times n}$, labels $\mathbf{Y} \in \mathbb{R}^{n \times c}$, projection dimension $m$, the number of nearest neighbors $k$, regularization parameter $\gamma$, and the number of selected features $r$.
Set $t = 0$.
For each object $\mathbf{x}_i \in \mathbf{X}$, form $\mathcal{M}_k(\mathbf{x}_i)$.
Initialize $\mathbf{Q}_0 = \mathbf{I}$, $\mathbf{S}_0$ such that $s_{ij} = 1$ if $\mathbf{x}_j \in \mathcal{M}_k(\mathbf{x}_i)$ or $\mathbf{x}_i \in \mathcal{M}_k(\mathbf{x}_j)$, and $s_{ij} = 0$ otherwise.
repeat
    Update $\mathbf{W}_{t+1}$ as the $m$ eigenvectors of $(\mathbf{X}\mathbf{L}_{S_t}\mathbf{X}^T + \gamma\,\mathbf{Q}_t)$ corresponding to its $m$ smallest eigenvalues.
    Update the diagonal matrix $\mathbf{Q}_{t+1}$, where the $l$-th diagonal element is $\frac{1}{2\sqrt{\mathbf{w}^l_{t+1}(\mathbf{w}^l_{t+1})^T+\epsilon}}$.
    Update the matrix $\mathbf{S}_{t+1}$, where $s_{ij}$ is defined in Eq. (23).
    Set $t = t + 1$.
until Converges
Output: Calculate all $\{\|\mathbf{w}^1\|_2, \ldots, \|\mathbf{w}^d\|_2\}$, sort them in the descending order and select the top $r$ ranked features as the ultimate result.
```

2) Unsupervised Feature Selection With Local Adaptive Projection: If the class labels are unknown, it is useful to explore local structure, especially when the number of objects n is very large. Let $\mathcal{N}_k(\mathbf{x}_i)$ denote the set of k-nearest neighbors of $\mathbf{x}_i$, where the distance between two objects $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined as $\|\mathbf{x}_i - \mathbf{x}_j\|_2$. We change problem (16) to form the following objective function for ULAP:

$$\min_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} \sum_{i,j=1}^{n} \upsilon_{ij} \|\mathbf{W}^T(\mathbf{x}_i - \mathbf{x}_j)\|_2 + \gamma \|\mathbf{W}\|_{2,1} \tag{24}$$

where $\upsilon_{ij} = 1$ if $\mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_i)$ and $\mathbf{x}_i \in \mathcal{N}_k(\mathbf{x}_j)$, or $\upsilon_{ij} = 0$ otherwise.

According to Theorem 1, it can be verified that problem (24) can be solved by solving (21), but the definition of $\mathbf{S}$ becomes

$$s_{ij} = \begin{cases} \dfrac{1}{2\sqrt{\|\mathbf{W}^T(\mathbf{x}_i - \mathbf{x}_j)\|_2^2 + \epsilon}} & \text{if } \mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_i) \text{ or } \mathbf{x}_i \in \mathcal{N}_k(\mathbf{x}_j) \\ 0 & \text{otherwise.} \end{cases} \tag{25}$$

Based on Algorithm 1, we propose an iterative algorithm to optimize (24), which is described in Algorithm 3. In each iteration, $\mathbf{W}$ is calculated with the current $\mathbf{Q}$ and $\mathbf{S}$, and then, $\mathbf{Q}$ and $\mathbf{S}$ are updated based on the currently calculated $\mathbf{W}$. The iteration procedure is repeated until the algorithm converges. With the learned $\mathbf{W}$ by ULAP, the importance of d features can be ranked in the descending order of $\{\|\mathbf{w}^1\|_2, \ldots, \|\mathbf{w}^d\|_2\}$. We can select the first r features as the ultimate result. For simplicity, we denote our feature selection method as ULAP in the following context. The convergence of ULAP can be verified according to Theorem 1.

Algorithm 3 ULAP: Algorithm to Solve Problem (24)

```
Input: Data matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$, the projection dimension $m$,
    the regularization parameter $\gamma$, the number of neighbors
    $k$, and the number of selected features $r$.
Set $t = 0$.
For each object $\mathbf{x}_i \in \mathbf{X}$, form $\mathcal{N}_k(\mathbf{x}_i)$.
Initialize $\mathbf{Q}_0 = \mathbf{I}$, $\mathbf{S}_0$ such that $s_{ij} = 1$ if $\mathbf{x}_i \in$
    $\mathcal{N}_k(\mathbf{x}_j)$ or $\mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_i)$, and $s_{ij} = 0$ otherwise.
repeat
    Update $\mathbf{W}_{t+1}$ as the $m$ eigenvectors of $(\mathbf{X}\mathbf{L}_{S_t}\mathbf{X}^T + \gamma\mathbf{Q}_t)$
        corresponding to its $m$ smallest eigenvalues.
    Update the diagonal matrix $\mathbf{Q}_{t+1}$, where the $l$-th diag-
        onal element is $\frac{1}{2\sqrt{\mathbf{w}_{t+1}^l(\mathbf{w}_{t+1}^l)^T+\epsilon}}$.
    Update the matrix $\mathbf{S}_{t+1}$ according to Eq. (25).
    Set $t = t + 1$.
until Converges
Output: Calculate all $\{\|\mathbf{w}^1\|_2, \ldots, \|\mathbf{w}^d\|_2\}$, sort them
    in the descending order and select the top $r$ ranked
    features as the ultimate result.
```

VI. Experimental Results and Analysis of SLAP

In this section, we show the performance of the proposed SLAP method on both synthetic and real-world data sets.

A. Experiments on Synthetic Data Sets

We generated a synthetic data set D_1 to test the projection ability of the proposed SLAP for feature selection. The data set consists of 12 dimensions, where the data in the first two dimensions are distributed in three Gaussian shapes while the data in the other dimensions are uniformly distributed noise features. Fig. 1(a) shows the data set in the first two dimensions, in which two small Gaussian clusters are contained in one class. In this experiment, our goal is to find a good direction of projection that can be used for feature selection. We compared SLAP with four methods, including LDA [19], linear discriminant feature selection (LDFS) [28], Hilbert–Schmidt feature selection (HSFS) [28], and FSSL-LDA (FSSL with LDA distance). In this experiment, the projected dimension was set as 1 and the number of the nearest neighborhoods k was set to 5. The regularization parameters in FSSL-LDA and SLAP were set to 1 for a fair comparison. The direction results of projection are displayed in Fig. 1(b), which shows that if we consider separating only the red class from the blue class, the direction of projection revealed by LDA is good. However, if we want to separate the two small classes contained within the red class, SLAP achieves the best direction of projection. The direction of projection revealed by FSSL-LDA is missing, because the first two dimensions are assigned as zero [see Fig. 1(c)].

To check whether SLAP is robust to noise features, we show the feature ranking values of all five methods in Fig. 1(c). This figure shows that SLAP has produced the best feature ranking values, where the first dimension is 1 and the other 11 dimensions are 0. Due to the introduction of the $\ell_{2,1}$ norm for both the sum of the within-class distances and the regularization term, SLAP can select the first dimension and ignore the effect of 10 noise dimensions. Besides SLAP, LDFS and HSFS are the only two methods that identify the importance of the first dimension. The seventh dimension is successfully affected three methods, including LDA, FSSL-LDA, and FSSL-LDA even scored the first dimension 0.

We also show the learned $\mathbf{S}$ in Fig. 2. In Fig. 2(a), we observe only a two-class structure in the initial $\mathbf{S}$. Surprisingly, in Fig. 2(b), we can clearly observe a three-class structure, which shows that SLAP successfully recovers hidden clusters in a class. This result shows that SLAP can better rank the features that help to solve the *multimodality* problem.

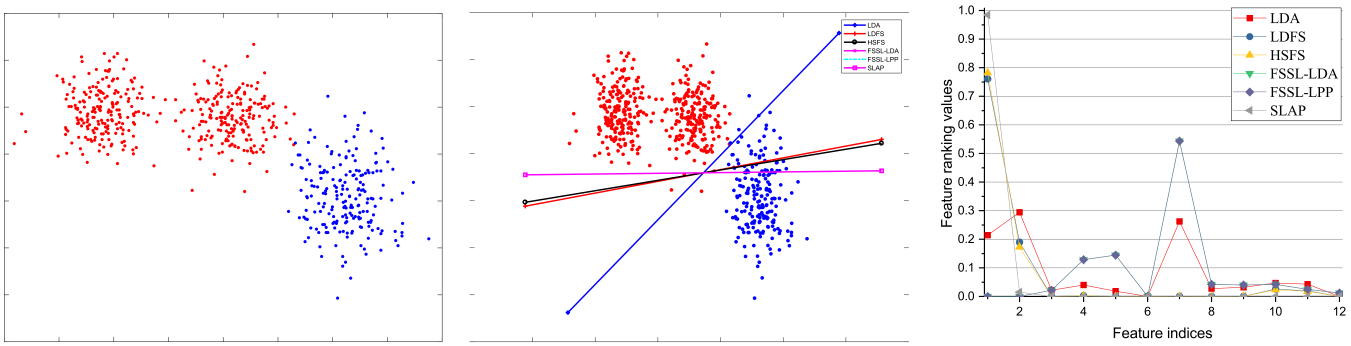
B. Experiments on Benchmark Data Sets

In this section, we present the experimental results and analysis for eight benchmark data sets.

1) Benchmark Data Sets: We selected eight benchmark data sets from the Xiaojun Chen's website.[1] Table I summarizes the characteristics of these eight data sets. The column "#Projected dimensions" represents the number of projection dimensions, which is the number of features to which the original figures are projected.

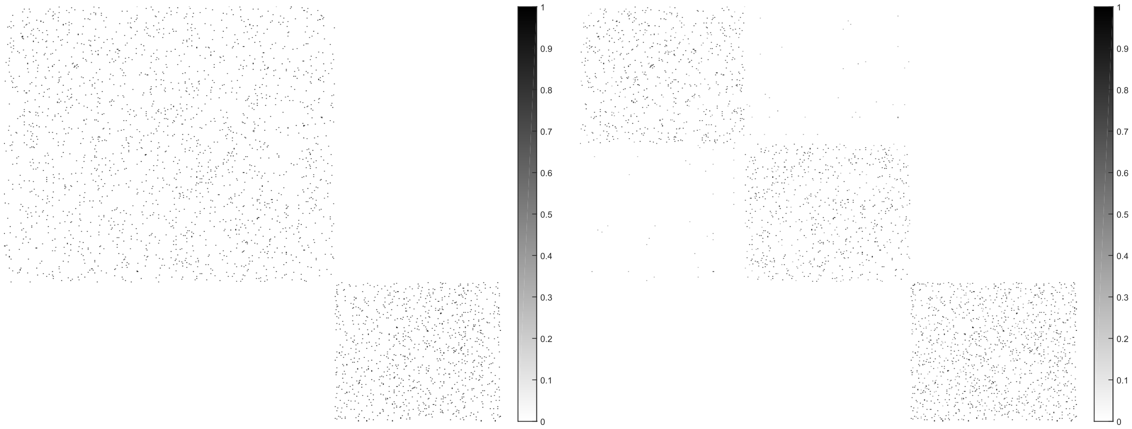
2) Comparison Results and Analysis: To validate the effectiveness of SLAP, we compared it with seven state-of-the-art supervised feature selection methods, namely,

[1] http://www.escience.cn/people/chenxiaojun/index.html



(a) (b) (c)

Fig. 1. Projection direction results of five methods on D (in Fig. 7, the direction of projection revealed by FSSL-LDA is missing, since the first two dimensions are assigned as zero). (a) Original data set D with two clusters in the first two dimensions. (b) Directions of projection in the first two dimensions revealed by five methods on D. (c) Feature ranking values computed by five methods on D.



(a) (b)

Fig. 2. (a) Initial **S** and (b) learned **S** by SLAP on D, in which values greater than 1 are set to 1 to clearly illustrate the structure in **S**.

TABLE I

CHARACTERISTICS OF EIGHT BENCHMARK DATA SETS

Name	#Objecyts	#Features	#Classes	#Projected dimensions
Wine	178	13	3	[2-13]
Musk	476	166	2	[10,20,...,100]
HV	606	100	2	[5,10,...,50]
Leukemia	38	3051	2	[50,...,500]
Breast2	77	4869	2	[50,...,500]
Vehicle	846	18	4	[2-18]
Ecoli	336	343	8	[20,40,...,200]
Segment	2310	19	7	[2-19]

Relief-F [3], [29], RFS [5], HSICLasso [30], feature selection via concave minimization [31], Fisher score [2], LDFS [28], HSFS [28], and FSSL-LDA [25]. We also used all features to run support vector machine (SVM) as a baseline. We set the parameters of all methods using the same strategy to make the experiments fair, i.e., $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}\}$. For LDFS, HSFS, FSSL-LDA, and SLAP, the projected dimensions for the different data sets are shown in Table I. The neighborhood parameters in Laplacian score, FSSL-LDA, LDFS, and SLAP were set to $\{2, 4, \ldots, 10\}$ for the Leukemia and Breast2 data sets and to $\{5, 10, \ldots, 50\}$ for the other six data sets in Table I.

For each data set, we ran each of the eight supervised feature selection methods to select different numbers of features. For the selected features, we performed 10-fold LibSVM, and the average accuracies were computed. Finally, the average accuracies versus the numbers of selected features of eight methods on eight data sets are reported in Fig. 3. To check whether SLAP is significantly outperformed all the other methods, we performed t-tests to test whether the result of our method is different from that of the other methods (the results are considered to be different if the p-value of the t-test is less than 0.05). We say that SLAP significantly outperforms all other methods if the average value of SLAP is higher than that of the other methods and the results of SLAP are different from those of all the other methods based on t-test. For each data set, the average accuracies on all selected features are summarized in Table II. Overall, our proposed SLAP method is significantly outperformed all other methods on four data sets, namely, Wine, Musk, Hill-Valley, and Vehicle. Specifically, SLAP has achieved >9% average improvement on the Vehicle data set compared with the second-best method FSSL-LDA. SLAP has even achieved a nearly 74% average improvement compared with the baseline. On the Wine data set, SLAP has achieved a nearly 4% average improvement compared with the second-best method FSSL-LDA and a 110% improvement

[1] http://www.escience.cn/people/chenxiaojun/index.html

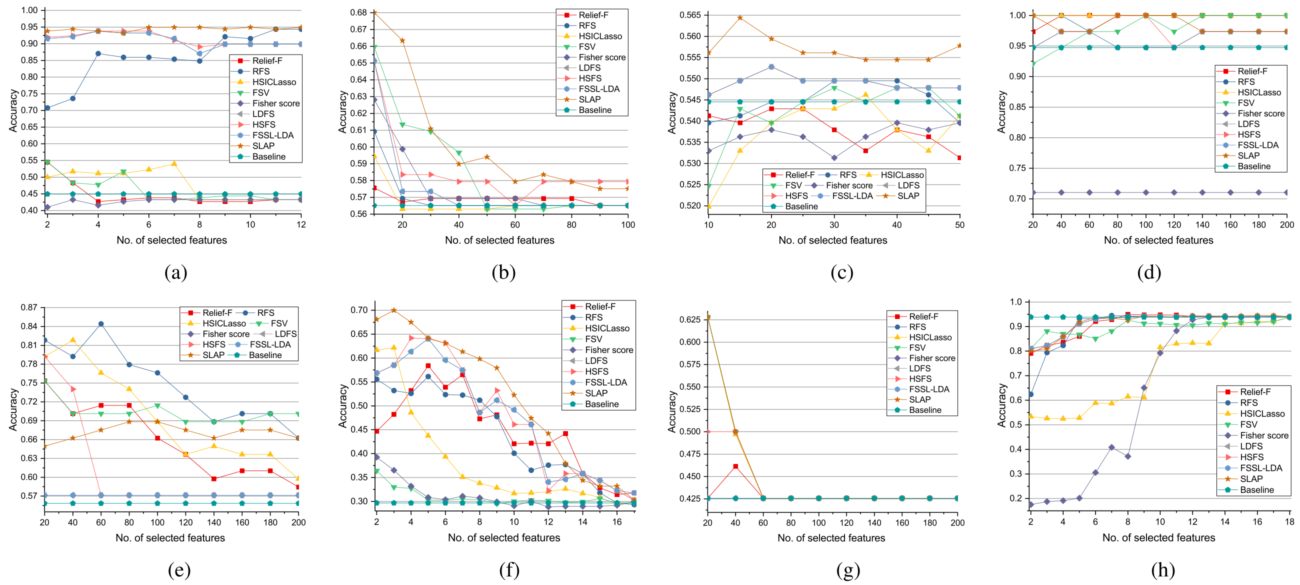


Fig. 3. Comparison of accuracy results by nine feature selection methods on eight benchmark data sets. (a) Results on the Wine data set. (b) Results on the Musk data set. (c) Results on the HV data set. (d) Results on the Leukemia data set. (e) Results on the Breast2 data set. (f) Results on the Vehicle data set. (g) Results on the Ecoli data set. (h) Results on the Segment data set.

TABLE II

Average Accuracies ± Standard Deviations of the Results of Nine Supervised Feature Selection Methods on Eight Benchmark Data Sets, in Which the Best Result on Each Data Set Is Highlighted in Bold and "*" Is Placed After the Average Result of SLAP If SLAP Significantly Outperformed the Other Eight Methods

Name	Wine	Musk	Hill-Valley	Leukemia	Breast2	Vehicle	Ecoli	Segment
Relief-F	0.45±0.04	0.569±0.003	0.538±0.004	0.997±0.008	0.658±0.059	0.445±0.085	0.429±0.011	0.913±0.051
RFS	0.852±0.076	0.570± 0.014	0.543±0.006	0.997±0.008	**0.748±0.061**	0.437±0.098	0.453±0.066	0.906±0.085
HSIClasso	0.483±0.045	0.567±0.010	0.536±0.008	**1.00±0.001**	0.696±0.077	0.380± 0.107	0.453±0.065	0.731±0.168
FSV	0.465±0.040	0.586±0.033	0.541±0.009	0.979±0.027	0.704±0.019	0.308±0.018	0.453±0.066	0.896±0.032
Fisher score	0.428±0.008	0.576±0.021	0.536±0.003	0.711±0.001	0.571±0.001	0.310±0.030	0.426±0.001	0.631± 0.331
LDFS	0.912±0.021	0.575±0.027	0.548±0.003	0.963±0.014	0.571±0.001	0.471±0.119	0.426±0.001	0.918±0.044
HSFS	0.914±0.020	0.586±0.026	0.548±0.003	0.974±0.014	0.610±0.001	0.474±0.118	0.440±0.001	0.919±0.042
FSSL-LDA	0.912±0.021	0.575±0.027	0.548±0.003	0.963±0.014	0.571±0.001	0.472±0.118	0.426±0.001	0.919±0.044
SLAP	**0.944±0.006***	**0.603±0.038***	**0.556±0.005***	0.984± 0.014	0.671±0.012	**0.516± 0.142***	**0.453±0.066**	0.919±0.047
Baseline	0.449	0.565	0.545	0.947	0.558	0.297	0.426	**0.939**

compared with the baseline. SLAP has also achieved a good performance on the remaining data sets. Besides SLAP, RFS, and HISClasso, each one has produced the best result in a single case. In addition, SLAP has outperformed FSSL-LDA on almost all data sets. This indicates that the learned implicit adaptive local structure learning improves the performance of supervised feature selection.

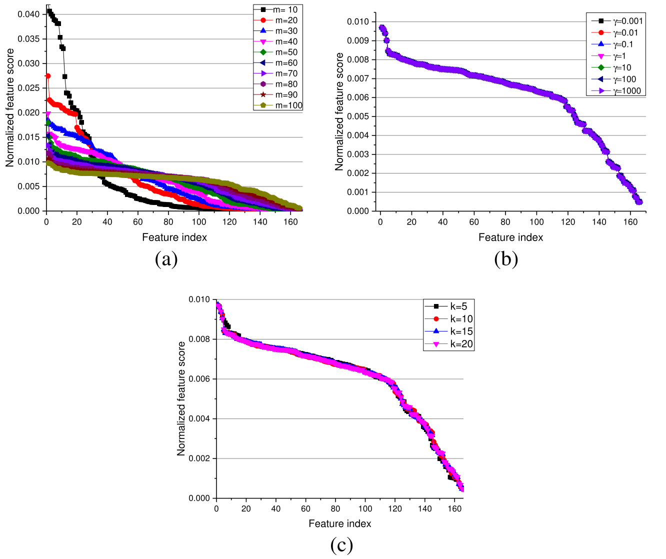
3) Parameter Sensitivity and Convergence Study: We select the Musk data set to illustrate the relationship between the projection matrix $\mathbf{W}$ and the two parameters γ and m in SLAP. For each γ, we selected the learned projection matrix $\mathbf{W}$ with $m = 100$ and $k = 20$ on the Musk data set. For each $\mathbf{W}$, we computed the normalized

$$\left\{\frac{\|\mathbf{w}^1\|_2}{\sum_{j=1}^d \|\mathbf{w}^j\|_2}, \ldots, \frac{\|\mathbf{w}^d\|_2}{\sum_{j=1}^d \|\mathbf{w}^j\|_2}\right\}$$

and sorted them in the descending order of normalized feature score **SW**. The results are shown in Fig. 4(b). From this figure, we can see that **SW** does not change substantially with the increase in γ. For each m, we select the learned projection matrix $\mathbf{W}$ with $\gamma = 1$ and $k = 20$ on the Musk data set and draw the relationships between the feature score **SW** and m in Fig. 4(a). From this figure, we can see that **SW** is highly affected by m. With the decrease in m, **SW** will contain more values that are close to zero. Compared with γ, m affects the learned **SW** more. Therefore, we can set smaller m in order to force SLAP to select fewer important features. We further analyze the change of **SW** with both parameters m and γ.

The average accuracies versus m, γ, and k on the Musk data set are reported in Fig. 5. We can see that m affects the results of SLAP more than γ and k. With the increase in m, the accuracies first drop and then increase. However, the highest accuracy is achieved with the smallest number of selected features. This indicates that we have to carefully set m even when we want to select only a small number of features. In real-world applications, we can perform hierarchy grid search to select a proper m, γ, and k for better results.

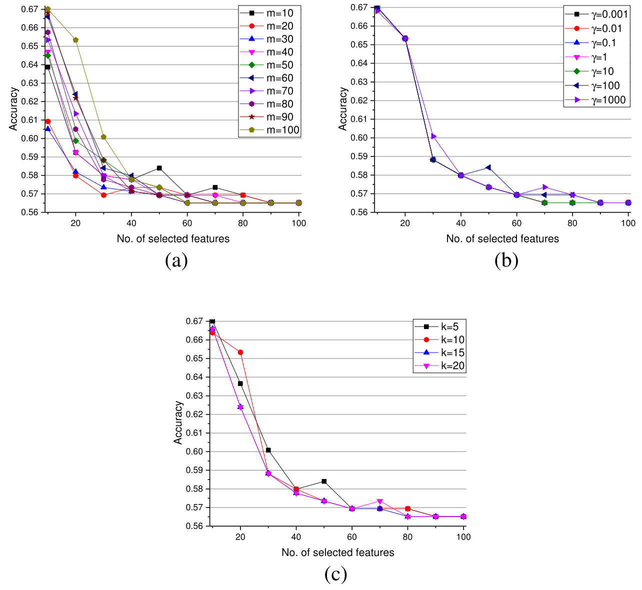
We have proved the convergence of SLAP in Section IV. Now, we experimentally study the speed of its convergence.



(a) (b)

(c)

Fig. 4. Feature score **SW** versus (a) m, (b) γ, and (c) k in SLAP on the Musk data set.



(a) (b)

(c)

Fig. 5. Average accuracies versus (a) m, (b) γ, and (c) k in SLAP on the Musk data set.

The convergence curve of the objective function values on the Musk data set is shown in Fig. 6. We can see that Algorithm 2 converges rapidly, which ensures the speed of the whole proposed approach.

VII. Experimental Results and Analysis on ULAP

In this section, we demonstrate the performance of the proposed unsupervised feature selection method ULAP on both synthetic and real-world data sets.

A. Experiments on Synthetic Data Sets

We generated a synthetic data set D_2 to test the projection ability of the proposed ULAP for unsupervised feature selection. The data set consists of 22 dimensions, where the data in the first two dimensions are distributed in two Gaussian shapes while the data in the other dimensions are noise features that are uniformly distributed in $[-1, 1]$. Fig. 7(a) shows the data set in the first two dimensions. From this figure, we can see that the horizontal dimension is the best dimension to distinguish the two clusters.

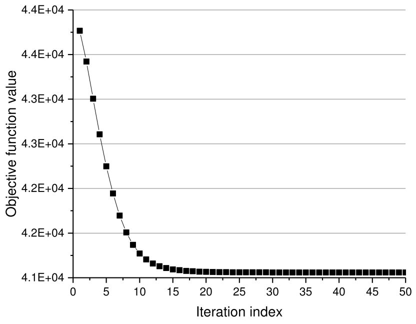


Fig. 6. Convergence curve of SLAP on the Musk data set.

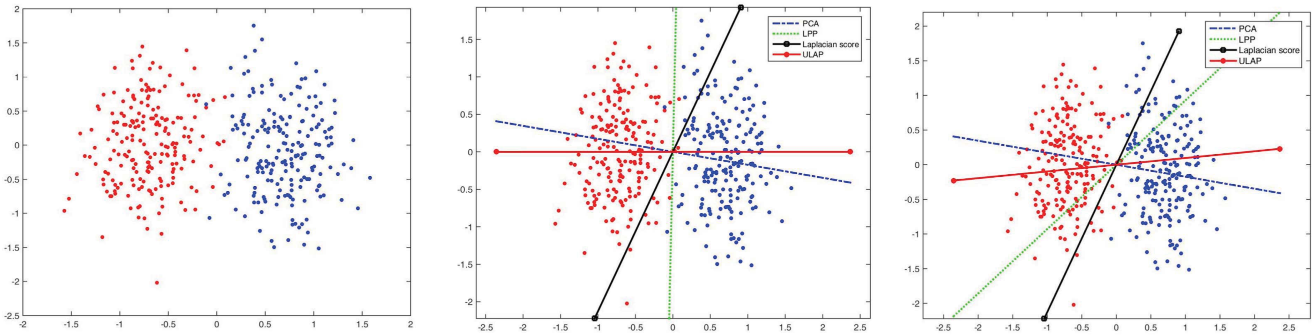
We compared the ULAP on D_2 with three methods, namely, PCA [32], LPPs [24], and Laplacian score [10]. In this experiment, the projected dimensions were set to 1 and 2. The direction results of projection are displayed in Fig. 7(b) and (c). These figures show that ULAP finds the best directions of projection in both experiments. Due to the introduction of the $\ell_{2,1}$ norm for both the sum of the pairwise distances and the regularization term, ULAP has selected the first dimension and ignored the second dimension when $m = 1$ and $m = 2$. Although PCA finds a good direction of projection when $m = 1$, its direction of projection is affected by the second dimension when $m = 2$.

B. Experiments on Benchmark Data Sets

In this section, we demonstrate the effectiveness of the proposed ULAP method on eight benchmark data sets.

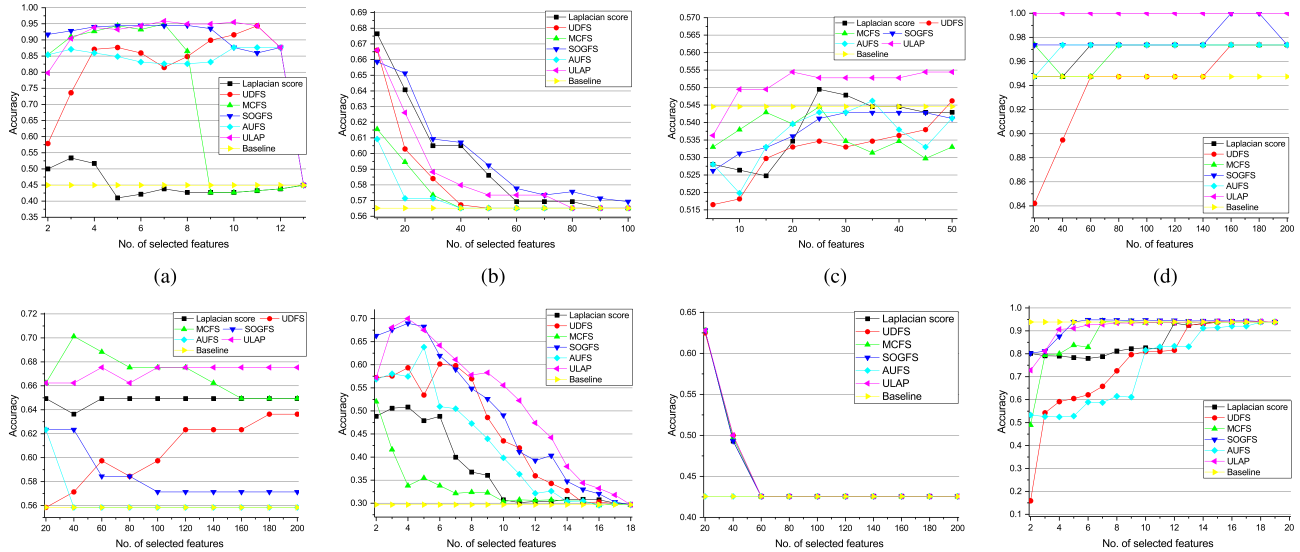
1) Comparison Results and Analysis: We evaluated ULAP on the eight benchmark data sets listed in Table I. To validate the effectiveness of ULAP, we compared it with five state-of-the-art unsupervised feature selection methods, namely, Laplacian score [10], unsupervised discriminative feature selection (UDFS) [33], multi-cluster feature selection (MCFS) [34], SOGFS [12], and adaptive unsupervised feature selection (AUFS) [35]. We also used all features to run SVM as the baseline. We set the parameters of all methods using the same strategy to make the experiments fair, i.e., $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}\}$. For SOGFS, AUFS, and ULAP, the projected dimensions for different data sets are shown in Table I. The neighborhood parameters, such as Laplacian score, UDFS, MCFS, SOGFS, and ULAP, were set to $\{2, 4, \ldots, 10\}$ for the Leukemia and Breast2 data sets and to $\{5, 10, \ldots, 50\}$ for the other six data sets in Table I.

For each data set, we ran each of the six unsupervised feature selection methods to select different numbers of features. For the selected features, we performed 10-fold LibSVM, and the average accuracies were computed. Finally, the average accuracies versus the number of selected features of six methods on eight data sets are reported in Fig. 8. For each data set, the average accuracies on all selected features are summarized in Table III in the same way as



(a) (b) (c)

Fig. 7. Comparison of the directions of projection in the first two dimensions on D_2. (a) Original data. (b) $m = 1$. (c) $m = 2$.



(a) (b) (c) (d)

(e) (f) (g) (h)

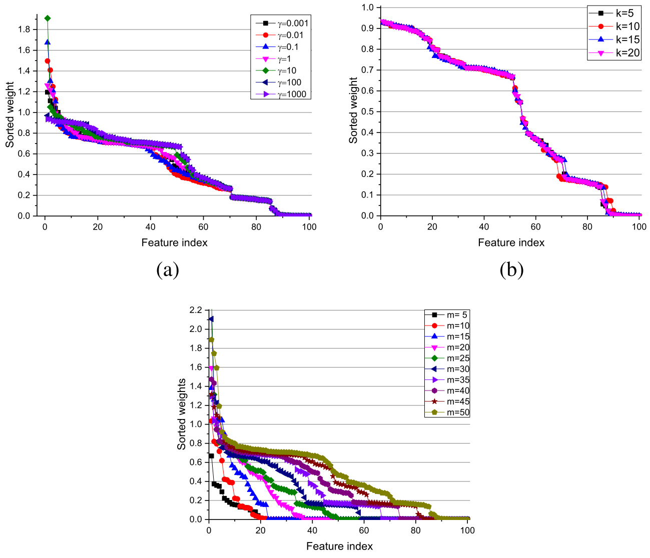
Fig. 8. Comparison of the average accuracy of six unsupervised feature selection methods on eight benchmark data sets. (a) Results on the Wine data set. (b) Results on the Musk data set. (c) Results on the Hill-Valley data set. (d) Results on the Leukemia data set. (e) Results on the Breast2 data set. (f) Results on the Vehicle data set. (g) Results on the Ecoli data set. (h) Results on the Segment data set.

TABLE III

Average Accuracies ± Standard Deviations of the Results of Six Supervised Feature Selection Methods on Eight Benchmark Data Sets, in Which the Best Result on Each Data Set Is Highlighted in Bold and "*" Is Placed After the Average Result of ULAP if ULAP Is Significantly Outperformed All Other Five Methods

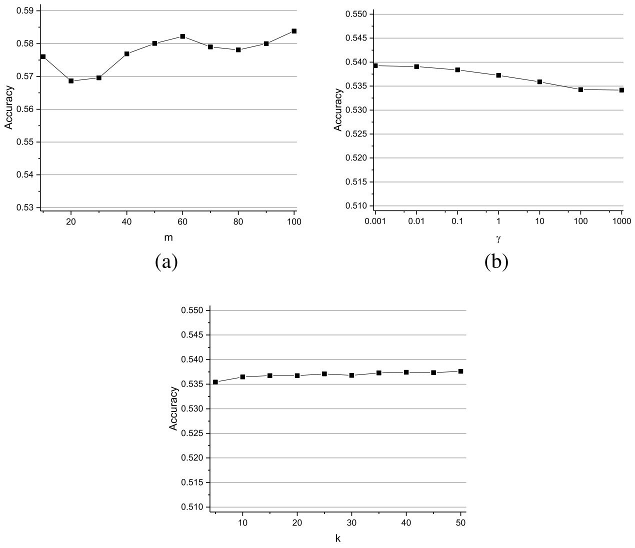
Name	Wine	Musk	Hill-Valley	Leukemia	Breast2	Vehicle	Ecoli	Segment
Laplacian score	0.452±0.039	0.579±0.036	0.539±0.009	0.974±0.011	0.648±0.004	0.373±0.083	0.453±0.063	0.861±0.068
UDFS	0.806±0.142	0.568±0.031	0.532±0.008	0.957±0.039	0.605±0.026	0.448±0.120	0.453±0.061	0.760± 0.200
MCFS	0.713±0.237	0.566±0.016	0.536±0.005	0.970±0.011	0.669±0.017	0.333±0.055	0.453±0.062	0.886± 0.109
SOGFS	0.880±0.134	**0.585±0.031**	0.538±0.006	0.980±0.011	0.584±0.020	0.488±0.142	0.453±0.063	0.925±0.045
AUFS	0.819±0.113	0.566±0.013	0.536±0.008	0.974±0.008	0.565±0.019	0.423±0.117	0.426±0.000	0.743± 0.165
ULAP	**0.883±0.130**	0.573±0.031	**0.551±0.005***	**1.000±0.000***	**0.671±0.006***	**0.512±0.133***	**0.453±0.062**	0.915±0.054
Baseline	0.449	0.565	0.545	0.947	0.558	0.297	0.426	**0.939**

in Table II. Overall, our proposed ULAP method is significantly outperformed all other methods on four data sets, i.e., the Hill-Valley, Leukemia, Breast2, and Vehicle data sets. ULAP has achieved a nearly 5% average improvement on the Vehicle data set compared with the second best method SOGFS. It even produced a nearly 70% average improvement compared to SVM with all features on the Vehicle data set. On the Leukemia data set, ULAP has achieved 100% accuracy with only 20 features. ULAP has also achieved a good performance on the remaining data sets. Besides ULAP, SOGFS has achieved the best result in only one case. This indicates that the results of ULAP are significantly better than



(a) (b) (c)

Fig. 9. Feature score **SW** versus γ, k, and m for ULAP on the Hill-Valley data set. (a) **SW** versus γ ($m = 200$ and $k = 10$). (b) **SW** versus k ($\gamma = 1$ and $k = 10$). (c) **SW** versus m ($\gamma = 1$ and $k = 10$).



(a) (b) (c)

Fig. 10. Average accuracy versus (a) m, (b) γ, and (c) k by ULAP on the Hill-Valley data set.

those of all other methods. Among the five methods excluding ULAP, the four embedded feature selection methods, such as SOGFS, UDFS, MCFS, and AUFS, have produced better results than the other two methods. This verifies the superiority of the embedded unsupervised feature selection methods.

2) Parameter Sensitivity and Convergence Study: We used the Hill-Valley data set to investigate the relationship between the projection matrix **W** and three parameters γ, m, and k in ULAP, and the results are shown in Fig. 9. These figures show that **SW** does not change substantially with increasing γ and k. **SW** is strongly affected by m. As m decreases, **SW** contain fewer nonzero values. Therefore, we can set smaller m to force ULAP to select fewer important features.

For each m, γ, and k, we computed the average accuracies of ULAP on the Hill-Valley data set and present the results in Fig. 10. From these figures, we can see that γ and k do not substantially affect the accuracy. The accuracy increases with increasing m when m is small and then becomes nearly stable on both data sets.

To study the convergence of ULAP and its speed, we draw the convergence curves of ULAP on the Hill-Valley data set in Fig. 11. From this figure, we can see that Algorithm 3 converges rapidly.

VIII. Conclusion

In this paper, we have proposed a novel feature selection framework, LAP, which simultaneously performs feature selection and adaptive local structure learning. The new method learns a projection matrix **W** and an adaptive similarity matrix **S** by minimizing the $\ell_{2,1}$ norm of the sum of pairwise distances and regularization term of **W**. An iterative optimization algorithm with the proved convergence is proposed to optimize the new model. In each iteration, **S** is computed from the projected distance with the learned **W**, and **W** is computed with the learned **S**. Therefore, LAP can better rank the features by weakening the effect of noise features with the adaptive similarity matrix. SLAP and ULAP methods are proposed based on the LAP framework. Empirical studies are conducted on eight benchmark data sets, and the results demonstrate the superiority of both SLAP and ULAP.

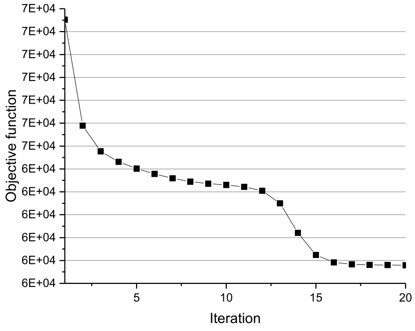


Fig. 11. Convergence curves of ULAP on the Hill-Valley data set.

In the future work, we will improve the LAP framework to address large-scale data.

Appendix
Proof of Theorem 1

Proof: Suppose the updated **W** by solving problem (21) is $\widetilde{\mathbf{W}}$. It is easy to see that

$$\widetilde{\mathbf{W}} = \arg_{\mathbf{W}} \min_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} \left[Tr(\mathbf{W}^T\mathbf{X}\mathbf{L}_S\mathbf{X}^T\mathbf{W}) + \gamma Tr(\mathbf{W}^T\mathbf{Q}\mathbf{W}) \right] \tag{26}$$

which indicates that

$$\begin{aligned} &Tr(\widetilde{\mathbf{W}}^T\mathbf{X}\mathbf{L}_S\mathbf{X}^T\widetilde{\mathbf{W}}) + \gamma Tr(\widetilde{\mathbf{W}}^T\mathbf{Q}\widetilde{\mathbf{W}}) \\ &\quad \le Tr(\mathbf{W}^T\mathbf{X}\mathbf{L}_S\mathbf{X}^T\mathbf{W}) + \gamma Tr(\mathbf{W}^T\mathbf{Q}\mathbf{W}). \end{aligned} \tag{27}$$

We add $\sum_{i,j=1}^{n} \frac{\epsilon}{2\left(\sqrt{\|\mathbf{W}^T(\mathbf{x}_i-\mathbf{x}_j)\|_2^2+\epsilon}\right)}$ and $\gamma \sum_{l=1}^{d} \frac{\epsilon}{\sqrt{\|\mathbf{w}^l\|_2^2+\epsilon}}$ to both sides of (27), and substitute the definition of $\mathbf{Q}$ in (19). Then, (27) can be rewritten as

$$\sum_{i,j=1}^{n} \frac{\|\widetilde{\mathbf{W}}^T(\mathbf{x}_i-\mathbf{x}_j)\|_2^2+\epsilon}{2\sqrt{\|\mathbf{W}^T(\mathbf{x}_i-\mathbf{x}_j)\|_2^2+\epsilon}} + \gamma \sum_{l=1}^{d} \frac{\|\widetilde{\mathbf{w}}^l\|_2^2+\epsilon}{\sqrt{\|\mathbf{w}^l\|_2^2+\epsilon}}$$
$$\leq \sum_{i,j=1}^{n} \frac{\|\mathbf{W}^T(\mathbf{x}_i-\mathbf{x}_j)\|_2^2+\epsilon}{2\sqrt{\|\mathbf{W}^T(\mathbf{x}_i-\mathbf{x}_j)\|_2^2+\epsilon}} + \gamma \sum_{l=1}^{d} \frac{\|\mathbf{w}^l\|_2^2+\epsilon}{\sqrt{\|\mathbf{w}^l\|_2^2+\epsilon}}. \tag{28}$$

According to [5, Lemma 1], we can verify that the following inequality holds for two positive vectors $\mathbf{a} \in \mathbb{R}^d$ and $\mathbf{b} \in \mathbb{R}^d$:

$$\sum_{j=1}^{d} \sqrt{a_j} - \sum_{j=1}^{d} \frac{a_j}{2\sqrt{b_j}} \leq \sum_{j=1}^{d} \sqrt{b_j} - \sum_{j=1}^{d} \frac{b_j}{2\sqrt{b_j}}. \tag{29}$$

Based on (29), we have

$$\sum_{i,j=1}^{n} \sqrt{\|\widetilde{\mathbf{W}}^T(\mathbf{x}_i-\mathbf{x}_j)\|_2^2+\epsilon} - \sum_{i,j=1}^{n} \frac{\|\widetilde{\mathbf{W}}^T(\mathbf{x}_i-\mathbf{x}_j)\|_2^2+\epsilon}{2\sqrt{\|\mathbf{W}^T(\mathbf{x}_i-\mathbf{x}_j)\|_2^2+\epsilon}}$$
$$\leq \sum_{i,j=1}^{n} \sqrt{\|\mathbf{W}^T(\mathbf{x}_i-\mathbf{x}_j)\|_2^2+\epsilon}$$
$$- \sum_{i,j=1}^{n} \frac{\|\mathbf{W}^T(\mathbf{x}_i-\mathbf{x}_j)\|_2^2+\epsilon}{2\sqrt{\|\mathbf{W}^T(\mathbf{x}_i-\mathbf{x}_j)\|_2^2+\epsilon}} \tag{30}$$

and

$$\gamma \sum_{l=1}^{d} \sqrt{\|\widetilde{\mathbf{w}}^l\|_2^2+\epsilon} - \gamma \sum_{l=1}^{d} \frac{\|\widetilde{\mathbf{w}}^l\|_2^2+\epsilon}{\sqrt{\|\mathbf{w}^l\|_2^2+\epsilon}}$$
$$\leq \gamma \sum_{l=1}^{d} \sqrt{\|\mathbf{w}^l\|_2^2+\epsilon} - \gamma \sum_{l=1}^{d} \frac{\|\mathbf{w}^l\|_2^2+\epsilon}{\sqrt{\|\mathbf{w}^l\|_2^2+\epsilon}}. \tag{31}$$

Summing over (28), (30), and (31), we arrive at

$$\sum_{i,j=1}^{n} \sqrt{\|\widetilde{\mathbf{W}}^T(\mathbf{x}_i-\mathbf{x}_j)\|_2^2+\epsilon} + \gamma \sum_{l=1}^{d} \sqrt{\|\widetilde{\mathbf{w}}^l\|_2^2+\epsilon}$$
$$\leq \sum_{i,j=1}^{n} \sqrt{\|\mathbf{W}^T(\mathbf{x}_i-\mathbf{x}_j)\|_2^2+\epsilon} + \gamma \sum_{l=1}^{d} \sqrt{\|\mathbf{w}^l\|_2^2+\epsilon}. \tag{32}$$

Therefore, the iteration process of Algorithm 1 will monotonically decrease the objective function of problem (16) in each iteration. □

REFERENCES

[1] S. H. Huang, "Supervised feature selection: A tutorial," *Artif. Intell. Res.*, vol. 4, no. 2, p. 22, 2015.

[2] O. D. Richard, P. E. Hart, and D. G. Stork, *Pattern Classification*. Hoboken, NJ, USA: Wiley, 2010.

[3] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Proc. 9th Int. Workshop Mach. Learn.*, 1992, pp. 249–256.

[4] I. Kononenko, "Estimating attributes: Analysis and extensions of RELIEF," in *Proc. ECML*, 1994, pp. 171–182.

[5] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint l_2, 1-norms minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1813–1821.

[6] S. Xiang, F. Nie, G. Meng, C. Pan, and C. Zhang, "Discriminative least squares regression for multiclass classification and feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 11, pp. 1738–1754, 2012.

[7] D. Wang, F. Nie, and H. Huang, "Feature selection via global redundancy minimization," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 10, pp. 2743–2755, Oct. 2015.

[8] R. Chen, N. Sun, X. Chen, M. Yang, and Q. Wu, "Supervised feature selection with a stratified feature weighting method," *IEEE Access*, vol. 6, p. 15087–15098, 2018.

[9] Z. Lai, D. Mo, W. K. Wong, Y. Xu, D. Miao, and D. Zhang, "Robust discriminant regression for feature extraction," *IEEE Trans. Cybern.*, to be published.

[10] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 507–514.

[11] L. Shi, L. Du, and Y.-D. Shen, "Robust spectral learning for unsupervised feature selection," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2014, pp. 977–982.

[12] F. Nie, W. Zhu, and X. Li, "Unsupervised feature selection with structured graph optimization," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 1302–1308.

[13] X. Zhu, X. Li, S. Zhang, C. Ju, and X. Wu, "Robust joint graph sparse coding for unsupervised spectral feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 6, pp. 1263–1275, Jun. 2017.

[14] M. Luo, F. Nie, X. Chang, Y. Yang, A. G. Hauptmann, and Q. Zheng, "Adaptive unsupervised feature selection with structure regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 944–956, Apr. 2017.

[15] Z. Zhao and H. Liu, "Semi-supervised feature selection via spectral analysis," in *Proc. SIAM Int. Conf. Data Mining*, 2007, pp. 641–646.

[16] Y. Han, Y. Yang, Y. Yan, Z. Ma, N. Sebe, and X. Zhou, "Semisupervised feature selection via spline regression for video semantic recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 2, pp. 252–264, Feb. 2015.

[17] X. Chen, G. Yuan, F. Nie, and J. Z. Huang, "Semi-supervised feature selection via rescaled linear regression," in *Proc. 26th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2017, pp. 1525–1531.

[18] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. San Diego, CA, USA: Academic, 1990.

[19] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, 2nd ed. New York, NY, USA: Springer, 2009.

[20] J. Ye, "Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems," *J. Mach. Learn. Res.*, vol. 6, pp. 483–502, Dec. 2005.

[21] M. Sugiyama, "Local Fisher discriminant analysis for supervised dimensionality reduction," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, New York, NY, USA, 2006, pp. 905–912.

[22] D. Cai, X. He, and J. Han, "Semi-supervised discriminant analysis," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–7.

[23] D. Cai, X. He, and J. Han, "Spectral regression: A unified approach for sparse subspace learning," in *Proc. 7th IEEE Int. Conf. Data Mining*, Oct. 2007, pp. 73–82.

[24] X. He and P. Niyogi, "Locality preserving projections," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 153–160.

[25] Q. Gu, Z. Li, and J. Han, "Joint feature selection and subspace learning," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2011, vol. 22. no. 1, pp. 1294–1299.

[26] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.

[27] Z. Zhao and H. Liu, "Spectral feature selection for supervised and unsupervised learning," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 1151–1157.

[28] M. Masaeli, G. Fung, and J. G. Dy, "From transformation-based dimensionality reduction to feature selection," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 751–758.

[29] H. Liu and H. Motoda, *Computational Methods of Feature Selection* (Data Mining and Knowledge Discovery Series). London, U.K.: Chapman & Hall, 2007.

[30] M. Yamada, W. Jitkrittum, L. Sigal, E. P. Xing, and M. Sugiyama, "High-dimensional feature selection by feature-wise kernelized lasso," *Neural Comput.*, vol. 26, no. 1, pp. 185–207, 2014.

[31] P. S. Bradley and O. L. Mangasarian, "Feature selection via concave minimization and support vector machines," in *Proc. 15th Int. Conf. Mach. Learn. (ICML)*, San Francisco, CA, USA, 1998, pp. 82–90.

[32] I. Jolliffe, *Principal Component Analysis*. Hoboken, NJ, USA: Wiley, 2002.

[33] Y. Yang, H. T. Shen, Z. Ma, Z. Huang, and X. Zhou, "$\ell_{2,1}$-Norm regularized discriminative feature selection for unsupervised learning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2011, pp. 1589–1594.

[34] D. Cai, C. Zhang, and X. He, "Unsupervised feature selection for multi-cluster data," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2010, pp. 333–342.

[35] M. Qian and C. Zhai, "Joint adaptive loss and l_2/l_0-norm minimization for unsupervised feature selection," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2015, pp. 1–8.

Xiaojun Chen (M'16) received the Ph.D. degree from the Harbin Institute of Technology, Harbin, China, in 2011.

He is currently an Assistant Professor with the College of Computer Science and Software, Shenzhen University, Shenzhen, China. His current research interests include subspace clustering, topic model, feature selection, and massive data mining.

Guowen Yuan is currently pursuing the M.A degree with the College of Computer Science and Software, Shenzhen University, Shenzhen, China.

His current research interests include clustering and feature selection.

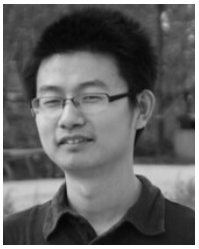
Wenting Wang received the Ph.D. degree from the Department of Mathematics, University College London, London, U.K.

She is currently a Post-Doctoral Researcher with the College of Computer Science and Software, Shenzhen University, Shenzhen, China. Her current research interests include deep learning and topology analysis of networks.

Feiping Nie received the Ph.D. degree in computer science from Tsinghua University, Beijing, China, in 2009.

He has published over 100 papers in the following top journals and conferences: the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the INTERNATIONAL JOURNAL OF COMPUTER VISION, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS/IEEE TRANSACTIONS ON NEURAL NETWORKS, the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, *ACM Transactions on Knowledge Discovery from Data*, *Bioinformatics*, the International Conference on Machine Learning, the Annual Conference on Neural Information Processing Systems, the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, the International Joint Conference on Artificial Intelligence, the AAAI Conference on Artificial Intelligence, the International Conference on Computer Vision, the IEEE Conference on Computer Vision and Pattern Recognition, and the ACM Multimedia. His current research interests include machine learning and its applications, such as pattern recognition, data mining, computer vision, image processing, and information retrieval.

Dr. Nie is currently serving as an associate editor or a PC member for several prestigious journals and conferences in the related fields.



Xiaojun Chang received the Ph.D. degree in computer science from the Center for Quantum Computation and Intelligent Systems, University of Technology Sydney, Ultimo, NSW, Australia, in 2016.

He currently holds a post-doctoral position at the Language Technology Institute, Carnegie Mellon University, Pittsburgh, PA, USA. His current research interests include machine learning, data mining, and computer vision.

Joshua Zhexue Huang received the Ph.D. degree from the Royal Institute of Technology, Stockholm, Sweden.

He is currently a Professor with the College of Computer Science and Software, Shenzhen University, Shenzhen, China, a Professor and a Chief Scientist with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Beijing, China, and an Honorary Professor with the Department of Mathematics, The University of Hong Kong, Hong Kong. His current research interests include data mining, machine learning, and clustering algorithms.