Top-*k* Feature Selection Framework Using Robust 0-1 Integer Programming

Xiaoqin Zhang, *Member, IEEE*, Mingyu Fan, Di Wang, Peng Zhou, and Dacheng Tao, *Fellow, IEEE*

***Abstract*— Feature selection (FS), which identifies the relevant features in a data set to facilitate subsequent data analysis, is a fundamental problem in machine learning and has been widely studied in recent years. Most FS methods rank the features in order of their scores based on a specific criterion and then select the *k* top-ranked features, where *k* is the number of desired features. However, these features are usually not the top-*k* features and may present a suboptimal choice. To address this issue, we propose a novel FS framework in this article to select the exact top-*k* features in the unsupervised, semisupervised, and supervised scenarios. The new framework utilizes the $\ell_{0,2}$-norm as the matrix sparsity constraint rather than its relaxations, such as the $\ell_{1,2}$-norm. Since the $\ell_{0,2}$-norm constrained problem is difficult to solve, we transform the discrete $\ell_{0,2}$-norm-based constraint into an equivalent 0-1 integer constraint and replace the 0-1 integer constraint with two continuous constraints. The obtained top-*k* FS framework with two continuous constraints is theoretically equivalent to the $\ell_{0,2}$-norm constrained problem and can be optimized by the alternating direction method of multipliers (ADMM). Unsupervised and semisupervised FS methods are developed based on the proposed framework, and extensive experiments on real-world data sets are conducted to demonstrate the effectiveness of the proposed FS framework.**

***Index Terms*— 0-1 integer programming, feature selection (FS), $\ell_{0,2}$-norm, nonconvex optimization.**

I. Introduction

IN MANY applications, such as image classification [1]–[3] and video recognition [4], [5], data are commonly represented by high-dimensional feature vectors. High-dimensional data are not suitable for learning algorithms because of “the curse of dimensionality” [6]. Furthermore, high-dimensional data also contain many noisy and redundant features that may degrade the performance of the learning algorithms. Feature selection (FS) has been used to address these problems, identify the most informative features, and improve the performance of the learning algorithms. As a result, FS has become a hot research topic in machine learning and pattern recognition [7].

Recent studies of sparse representation and compressed sensing have shown that sparsity is an intrinsic property of real-world data [8]. FS, which aims to find a sparse selection of input attributes to represent the data, can be regarded as an application of the sparse representation theory. Many FS methods apply sparsity-inducing regularization terms or constraints, such as the ℓ_0-norm, ℓ_1-norm, $\ell_{0,2}$-norm, and $\ell_{1,2}$-norm for feature evaluation and selection [9]. From a sparsity perspective, the ℓ_0-norm and $\ell_{0,2}$-norm, which counts the number of nonzero elements and columns of a matrix, respectively, are more desirable for FS because they provide the sparsest solution, i.e., each feature should be associated with either a zero score or a large score. However, ℓ_0-norm and $\ell_{0,2}$-norm regularized/constrained optimization problems have been proved to be nondeterministic polynomial-time hardness (NP-hard) [10] and, thus, are very difficult to solve. Some theoretical results have shown that the ℓ_0 ($\ell_{0,2}$)-norm is usually relaxed to the ℓ_1 ($\ell_{1,2}$)-norm in order to obtain a solvable optimization problem. However, to guarantee that the ℓ_1-norm yields the same result as the ℓ_0-norm, strong incoherence conditions have to be imposed [11]. In practice, these conditions may not be satisfied, so solving with ℓ_1 minimizer may fail to provide the desired sparse solution. Beside, some studies considered the ℓ_p-norm ($p \in (0, 1)$) [12] or $\ell_{p,r}$-norm ($p \in (0, 1),\ r \in [1, \infty)$) [13], [14] for FS, but all these nonconvex surrogates still cannot obtain the perfect sparse solution.

In this article, we propose a novel framework for exact top-*k* FS using robust 0-1 integer programming. The framework deals with the feature evaluation problem using the $\ell_{0,2}$-norm constraint directly without any relaxation. As mentioned earlier, the optimization of the $\ell_{0,2}$-norm constrained problem is difficult to solve because it is NP-hard. To address this problem, the $\ell_{0,2}$-norm constraint is equivalently transformed into a 0-1 integer constraint. Subsequently, the discrete 0-1 integer constraint is replaced with two continuous constraints. The obtained top-*k* FS framework with two continuous constraints is theoretically equivalent to the $\ell_{0,2}$-norm constrained feature

Manuscript received December 15, 2019; revised April 28, 2020; accepted July 9, 2020. Date of publication July 31, 2020; date of current version July 7, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61772373, Grant 61922064, Grant 61772374, and Grant 61806003, in part by the Zhejiang Provincial Natural Science Foundation under Grant LR17F030001, in part by the Project of Science and Technology Plans of Wenzhou City under Grant C20170008 and Grant ZG2017016, and in part by the Australian Research Council under Project FL-170100117. *(Xiaoqin Zhang and Mingyu Fan are co-first authors.) (Corresponding author: Peng Zhou.)*

Xiaoqin Zhang and Mingyu Fan are with the College of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou 325035, China (e-mail: zhangxiaoqinnan@gmail.com; fanmingyu@wzu.edu.cn).

Di Wang is with the Center of Intelligent Decision-Making and Machine Learning, School of Management, Xi’an Jiaotong University, Xi’an 710049, China (e-mail: wang.di@xjtu.edu.cn).

Peng Zhou is with the School of Computer Science and Technology, Anhui University, Hefei 230601, China, and also with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China (e-mail: zhoupeng@ahu.edu.cn).

Dacheng Tao is with the UBTECH Sydney Artificial Intelligence Centre and the School of Computer Science, Faculty of Engineering, The University of Sydney, Darlington, NSW 2008, Australia (e-mail: dacheng.tao@sydney.edu.au).

This article has supplementary downloadable material available at https://ieeexplore.ieee.org, provided by the authors.

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

Digital Object Identifier 10.1109/TNNLS.2020.3009209

evaluation problem and can be optimized by the alternating direction method of multipliers (ADMM) [15]. This study is an extension of our early study [16] that focuses on the supervised FS problem and can be regarded as a special example of this framework. Different from [16], in this article, we apply the framework to unsupervised and semisupervised scenarios by using the subspace clustering model [17], leading to a top-k unsupervised FS and a top-k semisupervised FS, respectively. The main contributions of this article are summarized as follows.

1) A novel exact top-k FS framework for unsupervised, semisupervised, and supervised scenarios is proposed in the formulation of a ℓ_0- or $\ell_{0,2}$-norm constrained problem. Then, an unsupervised method and a semisupervised method for top-k FS are proposed in the framework.
2) It is demonstrated that the $\ell_{0,2}$-norm constraint in the framework is equivalent to the 0-1 integer constraint. The discrete 0-1 integer constraint is further replaced with two continuous constraints. The obtained top-k FS framework with two continuous constraints is effectively optimized by the ADMM method.
3) Two novel top-k FS methods are proposed by considering the subspace clustering model, in which the two subtasks, i.e., clustering (or semisupervised learning) and FS, boost each other's results. Both algorithms ensure the selection of the exact top-k features. Extensive experimental results demonstrate the effectiveness and superiority of the proposed methods.

This article is organized as follows. Section II presents the related work. Section III introduces some notations and the proposed top-k FS framework. Section IV presents the unsupervised and semisupervised top-k FS method based on the proposed FS framework. Section V describes the experimental results on benchmark data sets, and Section VI concludes this article.

II. Related Work

FS can be roughly classified into three categories according to the data labels, i.e., supervised, unsupervised, and semisupervised FSs.

Supervised FS uses the class labels to identify discriminative features for recognition and classification [18] and evaluates the features by computing the correlation between the features and labels. Gu *et al.* [19] computed the generalized Fisher score of each feature. Nie *et al.* [20] proposed a robust FS method with $\ell_{1,2}$ regression. An efficient incremental FS method for high-dimensional data based on the Fisher score was introduced in [21]. In [22], an attention-based supervised feature evaluation mechanism was proposed. Peng and Liu [23] proposed a sparsity regularized discriminative FS model that used a novel smooth and robust hinge loss. In order to screen out irrelevant and noisy features, an $\ell_{1,2}$-norm with an exclusive lasso was developed for flexible FS [24]. In [25], a new perspective called the worst case was proposed to evaluate the features. Supervised FS methods usually provide good results because the label information is used for evaluating the features.

Due to the absence of class labels, unsupervised FS methods select features that preserve the intrinsic structure of the data. Yang *et al.* [26] used local total scatter and between-class scatter matrices to select features. Du and Shen [27] developed an adaptive graph for FS by preserving the global and local structures. By representing each feature as a linear combination of its relevant features, a regularized self-expressive unsupervised FS method was proposed [28]. Nie *et al.* [29] proposed a method for adaptively learning of the local structure based on the FS results. Luo *et al.* [30] created an adaptive graph with structure regularization. Zheng *et al.* [31] developed a method for learning a low rank structure for FS. Li *et al.* [32] proposed a generalized uncorrelated regression with an adaptive graph for FS. Zhou *et al.* [33] provided an FS method to select the features to reveal the balanced structure of data. In an unsupervised spectral FS method [34], features were selected that preserved the local and global structures of the features as well as the samples. Graphs representing the feature space and sample space were used in an unsupervised FS method [35], which incorporated nonnegative low-dimensional embedding and sparse feature evaluation. In [36], multiple graphs were used to select features. In [37], unsupervised FS was performed using column subset selection and an iterative Pareto optimization method.

For data with insufficient class labels, semisupervised methods have been used to propagate the label information and select features by considering both the label information and the intrinsic structure of the data. Han *et al.* [38] utilized the manifold structure of the data for semisupervised FS. Chang *et al.* [39] proposed a convex formulation of semisupervised FS for multilabel data. Chang and Yang [40] proposed a semisupervised FS method for multitask learning. Luo *et al.* [41] presented a semisupervised FS method with an adaptive neighbor assignment strategy. Pearson's correlation coefficients were computed for the labeled and unlabeled data to measure the feature-to-feature and feature-to-label information using max-relevance and min-redundancy criteria [42]. The semisupervised multiview FS method [43] grouped high-dimensional input features into several subsets and then encouraged the features in each subset to be sparse. A joint semisupervised FS and classification method that adaptively selected the features and simultaneously trained a classifier with the selected features was presented in [44].

The framework proposed in this study is able to give a unified treatment for supervised, unsupervised, and semisupervised FS approaches.

FS can be regarded as an example of the sparsity representation theory [9]. The ℓ_0-norm counts the number of nonzero entries of a vector or a matrix and is an ideal choice for the top-k FS problem. However, since the ℓ_0-norm constrained problem is difficult to solve, many FS methods have replaced the ℓ_0-norm with its convex surrogate, the ℓ_1-norm, and have shown promising results. An ℓ_1-norm regularized regression FS method, Lasso, was proposed in [45]. Destrero *et al.* [46] utilized the Lagrangian form of Lasso for FS in a face recognition task. In [47], the elastic net regularization was proposed to handle features with strong correlations. Yang *et al.* [48] proposed a continuous method for the sparse combinatorial optimization problem, which significantly improved the feature matching performance especially when the objective function is highly nonconvex. The group Lasso was introduced to integrate the feature structure and then evaluate the importance of features to remove the feature redundancy, where the structures included the disjoint groups [49], the overlapping groups [50], and so on. For multiclass data, Nie *et al.* [20] proposed the use of the $\ell_{1,2}$-norm instead of the ℓ_1-norm as the regularization term and obtained promising results. Subsequently, many FS methods were proposed for $\ell_{1,2}$-norm optimization problems [27], [38], [51]. Most recently, the matrix norm has been extended to $\ell_{p,r}$, $(p \in (0,1],\ r \in (1,\infty])$ [13], [18] for sparser solutions. Some studies [52], [53] have utilized the $\ell_{0,2}$-norm-based constraint for top-k FS to optimize the discrete problem using an augmented Lagrangian method. Compared with these studies, the FS framework proposed in this study first transforms the $\ell_{0,2}$-norm constraint into a 0-1 integer constraint and then replaces the 0-1 integer constraint with two continuous constraints.

III. Top-k Feature Selection Framework

A. Notations and Definitions

Throughout this article, we use boldface uppercase and lowercase letters to denote matrices and vectors, respectively. For a matrix $\mathbf{A}$, A_{ij} denotes its (i, j) th element. The ℓ_p-norm of a vector $\mathbf{v} \in \mathbb{R}^D$ is defined as $\|\mathbf{v}\|_p = (\sum_{i=1}^{D} |v_i|^p)^{(1/p)}$, where v_i denotes the ith entry in $\mathbf{v}$. The ℓ_0-norm of $\mathbf{v}$ is defined as $\|\mathbf{v}\|_0 = (\sum_{i=1}^{D} |v_i|^0)$, i.e., the counts of nonzero entries in $\mathbf{v}$. The $\ell_{p,r}$-norm of a matrix $\mathbf{A} \in \mathbb{R}^{C\times D}$ is defined as

$$\|\mathbf{A}\|_{p,r} = \left(\sum_{j=1}^{D}\left(\sum_{i=1}^{C}|A_{ij}|^r\right)^{\frac{p}{r}}\right)^{\frac{1}{p}}. \tag{1}$$

Consequently, the $\ell_{0,2}$-norm of $\mathbf{A}$ is defined as $\|\mathbf{A}\|_{0,2} = \sum_{j=1}^{D} \|\mathbf{A}_{.j}\|_2^0$, which counts the number of nonzero columns in $\mathbf{A}$. Here, $\mathbf{A}_{.j}$ denotes the jth column of $\mathbf{A}$. $\operatorname{diag}(\mathbf{v})$ ($\mathbf{v} \in \mathbb{R}^D$) is a diagonal matrix whose diagonal elements are the entries of vector $\mathbf{v}$, and $\operatorname{diag}(\boldsymbol{\Theta})$ ($\boldsymbol{\Theta} \in \mathbb{R}^{D\times D}$) is a D-dimensional vector consisting of the diagonal elements of the matrix $\boldsymbol{\Theta}$.

B. Proposed Framework

The input data matrix is

$$\mathbf{X} = \begin{pmatrix}\mathbf{x}_1, & \cdots, & \mathbf{x}_N\end{pmatrix} = \begin{pmatrix}\mathbf{f}_1^T \\ \vdots \\ \mathbf{f}_D^T\end{pmatrix} \in \mathbb{R}^{D\times N}$$

where $\mathbf{x}_i \in \mathbb{R}^D$ $(1 \le i \le N)$ denotes the ith data sample and $\mathbf{f}_j \in \mathbb{R}^N$ $(1 \le j \le D)$ denotes the jth feature vector. The objective of FS is to select the k most informative features from D features for use in subsequent learning algorithms. Considering a projective matrix $\mathbf{A} \in \mathbb{R}^{C\times D}$, $\mathbf{AX}$ admits the expansion $\mathbf{AX} = \sum_{j=1}^{D} \mathbf{A}_{.j}\mathbf{f}_j^T$. As can be seen, $\mathbf{A}_{.j}$ measures the importance of $\mathbf{f}_j$. The larger the value of $\|\mathbf{A}_{.j}\|_2$, the higher the contribution of $\mathbf{f}_j$ to $\mathbf{AX}$ is. If $\mathbf{A}_{.j} \approx \mathbf{0}$, then $\mathbf{f}_j$ is an invalid feature. Since we need to select the exact top-k features, the number of nonzero columns of $\mathbf{A}$ should be exactly k, that is, $\|\mathbf{A}\|_{0,2} = k$, where k is the number of selected features.

Our general top-k FS framework is defined as follows:

$$\min_{\mathbf{A}} f(\mathbf{AX}), \quad \text{s.t. } \|\mathbf{A}\|_{0,2} = k \tag{2}$$

where $f(\cdot)$ is the objective function. For supervised and semisupervised FSs, $f(\mathbf{AX}) = loss(\mathbf{AX}, \mathbf{Y}_l)$, where $loss$ is a loss function that measures the correlation between $\mathbf{AX}$ and $\mathbf{Y}_l$, $\mathbf{Y}_l$ is the label information of all data samples (in the supervised case) or partial data samples (in the semisupervised case), and $0 \le l \le N$ denotes the number of labeled data samples. For unsupervised FS, $f(\mathbf{AX})$ is usually designed to maintain the intrinsic data structure, such as the manifold structure or sparse representation. Regularization of $\mathbf{A}$ should be considered to avoid a trivial solution

$$\min_{\mathbf{A}} f(\mathbf{AX}) + \gamma\|\mathbf{A}\|_F^2, \quad \text{s.t. } \|\mathbf{A}\|_{0,2} = k \tag{3}$$

where $\|\cdot\|_F$ is the Frobenius norm and γ is a nonnegative hyperparameter. Here, the matrix $\mathbf{A}$ has two functions: first, it is columnwise k-sparse for selecting features; second, it uses projection mapping to fuse the selected features. We use $\mathbf{A}\operatorname{diag}(\mathbf{v})$ to separate the two functions and replace $\mathbf{A}$, where $\mathbf{v} \in \{0,1\}^D$ is a feature indicator vector, i.e., $v_j = 1$ means that the jth feature is selected; otherwise, it means that the jth feature is unselected. The top-k FS framework using 0-1 integer programming is defined as follows:

$$\begin{aligned}&\min_{\mathbf{A},\mathbf{v}} f(\mathbf{A}\operatorname{diag}(\mathbf{v})\mathbf{X}) + \gamma\|\mathbf{A}\|_F^2\\ &\text{s.t. } \mathbf{1}_D^T\mathbf{v} = k, \quad \text{and } \mathbf{v} \in \{0,1\}^D\end{aligned} \tag{4}$$

where $\mathbf{1}_D$ is a D-dimensional vector whose elements are all 1s. Different from (3), the matrix $\mathbf{A}$ in (4) works only as a projection matrix to fuse the features, and the FS is conducted by the vector $\mathbf{v}$.

In (4), the optimization of the variable $\mathbf{v}$ is a 0-1 integer programming problem, which is generally challenging to solve. We resort to an effective method, namely, the ℓ_2-box method [54], to address this problem. The binary set $\{0,1\}^D$ is equivalently replaced by the intersection of a solid cube and a shifted ℓ_2-sphere; the result is presented in the following proposition.

Proposition 1 [54]: Let $\mathbf{1}_D \in \mathbb{R}^D$ be the vector whose entries are all 1s; therefore, we obtain

$$\mathbf{v} \in \{0,1\}^D \Leftrightarrow \{\mathbf{v} : \mathbf{v} \in [0,1]^D\} \bigcap \left\{\mathbf{v} : \left\|\mathbf{v} - \frac{\mathbf{1}_D}{2}\right\|_2^2 = \frac{D}{4}\right\}.$$

Fig. 1 shows an illustrative example of Proposition 1 in 2-D space. The discrete 2-D binary set is equivalent to the intersection of two continuous sets. Based on this proposition, the proposed top-k FS framework with two continuous constraints is defined as

$$\begin{aligned}&\min_{\mathbf{A},\mathbf{v}} f(\mathbf{A}\operatorname{diag}(\mathbf{v})\mathbf{X}) + \gamma\|\mathbf{A}\|_F^2\\ &\text{s.t. } \mathbf{1}^T\mathbf{v} = k, \quad \mathbf{v} = \mathbf{v}_1, \quad \text{and } \mathbf{v} = \mathbf{v}_2,\\ &\qquad \mathbf{v}_1 \in S_b \text{ and } \mathbf{v}_2 \in S_p\end{aligned} \tag{5}$$

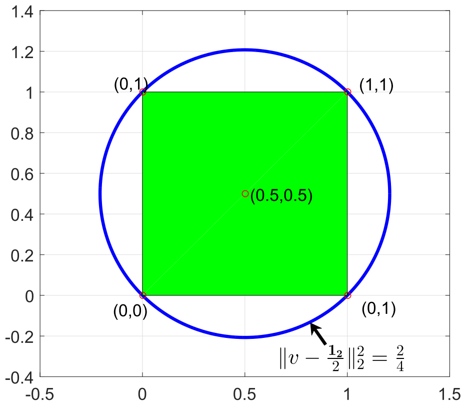


Fig. 1. Illustrative example of the equivalence between the binary constraint and the continuous constraints in 2-D space.

Algorithm 1 General ADMM for Top-k FS Formulation (5)

```
Input: Data $\mathbf{X}$, label $\mathbf{Y}_l$ (optional), $\rho_1 > 1$, $\rho_2 > 1$, and $\rho_3 > 1$, initialize the parameters and variables $\mathbf{A}^{(0)}, \mathbf{v}^{(0)} = \mathbf{1}_D, \mathbf{v}_1^{(0)} = \mathbf{0}_D, \mathbf{v}_2^{(0)} = \mathbf{0}_D$.
Output: Projective matrix $\mathbf{A}$ and vector $\mathbf{v}$.
Construct centered data matrix $\bar{\mathbf{X}} = [\mathbf{x}_1 - \bar{\mathbf{x}}, \ldots, \mathbf{x}_N - \bar{\mathbf{x}}]$, where $\bar{\mathbf{x}}$ is the mean of $\mathbf{x}_1, \ldots, \mathbf{x}_N$.
while not converge do
    Compute $\mathbf{A}^{(t+1)}$ by solving the optimization in Eq. (7).
    Compute $\mathbf{v}^{(t+1)}$ by solving the optimization in Eq. (8).
    Compute $\mathbf{v}_1^{(t+1)}$ and $\mathbf{v}_2^{(t+1)}$ by using the projection operators shown in Eq. (10) and Eq. (12).
    Update $\boldsymbol{\eta}_1^{(t+1)}$, $\boldsymbol{\eta}_2^{(t+1)}$, $\eta_3^{(t+1)}$ by Eq. (14).
end while
```

where the two continuous sets are $S_b = \{\mathbf{v} : \mathbf{v} \in [0,1]^D\}$ and $S_p = \{\mathbf{v} : \|\mathbf{v} - (\mathbf{1}_D/2)\|_2^2 = (D/4)\}$. It is evident that the framework formulations in (2)–(5) are essentially equivalent. The optimization in (5) is generally nonconvex due to the ℓ_2-sphere constraint ($\mathbf{v} = \mathbf{v}_2$ and $\mathbf{v}_2 \in S_p$) and possibly the nature of the objective function $f(\mathbf{A}\text{diag}(\mathbf{v})\mathbf{X})$.

C. Optimization

Although the ADMM has been used successful for convex (especially nonsmooth) optimization problems, there has been growing interests in using the ADMM in nonconvex optimization problems. Inspired by the study [54], a solution for the FS framework using ADMM is proposed. Using the positive parameters ρ_1, ρ_2, and ρ_3, the augmented Lagrangian function for (5) is defined as

$$\begin{aligned} L(\mathbf{A}, \mathbf{v}, \mathbf{v_1}, \mathbf{v_2}, \eta_1, \eta_2, \eta_3) &= f(\mathbf{A}\text{diag}(\mathbf{v})\mathbf{X}) + \gamma \|\mathbf{A}\|_F^2 \\ &\quad + \boldsymbol{\eta}_1^T(\mathbf{v} - \mathbf{v}_1) + \boldsymbol{\eta}_2^T(\mathbf{v} - \mathbf{v}_2) + \eta_3(\mathbf{1}^T\mathbf{v} - k) \\ &\quad + \frac{\rho_1}{2}\|\mathbf{v} - \mathbf{v}_1\|_2^2 + \frac{\rho_2}{2}\|\mathbf{v} - \mathbf{v}_2\|_2^2 + \frac{\rho_3}{2}(\mathbf{1}^T\mathbf{v} - k)^2 \end{aligned} \tag{6}$$

where $\boldsymbol{\eta}_1 \in \mathbb{R}^D$, $\boldsymbol{\eta}_2 \in \mathbb{R}^D$, and $\boldsymbol{\eta}_3 \in \mathbb{R}$ are the Lagrange multipliers for the three equality constraints. The ADMM approach then iteratively optimizes the variables individually. Denote by $(\mathbf{A}^{(t)}, \mathbf{v}^{(t)}, \mathbf{v}_1^{(t)}, \mathbf{v}_2^{(t)})$ the variables at iterative t and by $(\boldsymbol{\eta}_1^{(t)}, \boldsymbol{\eta}_2^{(t)}, \eta_3^{(t)})$ the Lagrange multipliers at iteration t; the update steps of the ADMM implementation are summarized in Algorithm 1 and are discussed in detail.

Step 1 (Update the Projection Matrix $\mathbf{A}^{(t+1)}$*):* When the other variables are fixed, the Lagrangian function (6) with respect to $\mathbf{A}$ is formulated as

$$\mathbf{A}^{(t+1)} = \arg\min_{\mathbf{A}} \{ f(\mathbf{A}\text{diag}(\mathbf{v}^{(t)})\mathbf{X}) + \gamma \|\mathbf{A}\|_F^2 \}. \tag{7}$$

It is evident that the optimization strategy is task specific and highly dependent on the nature of $f(\cdot)$. When $f(\cdot)$ is convex, the problem admits a global solution and can usually be optimized efficiently.

Step 2 (Update the FS Indicator Vector $\mathbf{v}$*):* The subproblem with respect to variable $\mathbf{v}$ when the other variables are fixed is defined as

$$\begin{aligned} \mathbf{v}^{(t+1)} = \arg\min_{\mathbf{v}} \Bigg\{ & f(\mathbf{A}^{(t+1)}\text{diag}(\mathbf{v})\mathbf{X}) + \frac{\rho_1 + \rho_2}{2} \\ & \times \left\| \mathbf{v} - \frac{\rho_1 \mathbf{v}_1^{(t)} + \rho_2 \mathbf{v}_2^{(t)} - \boldsymbol{\eta}_1^{(t)} - \boldsymbol{\eta}_2^{(t)}}{\rho_1 + \rho_2} \right\|_2^2 \\ & + \frac{\rho_3}{2}\left(\mathbf{1}^T\mathbf{v} - k + \frac{\eta_3^{(t)}}{\rho_3}\right)^2 \Bigg\}. \end{aligned} \tag{8}$$

This solution strategy is also dependent on the choice of $f(\cdot)$. We will provide a detailed description in Section IV on the optimization of $\mathbf{A}$ and $\mathbf{v}$ when $f(\cdot)$ is quadratic.

Step 3 (Update $\mathbf{v}_1$ *and* $\mathbf{v}_2$ *Using Projective Operators):* The $\mathbf{v}_1$ subproblem in (6) is formulated as

$$\mathbf{v}_1^{(t+1)} = \arg\min_{\mathbf{v}_1 \in S_b} \left\| \mathbf{v}_1 - \mathbf{v}^{(t+1)} - \frac{\boldsymbol{\eta}_1^{(t)}}{\rho_1} \right\|_2^2. \tag{9}$$

Equation (9) is a convex problem with a box constraint, which admits the following closed-form solution:

$$\mathbf{v}_1^{(t+1)} = P_{S_b}\left(\mathbf{v}^{(t+1)} + \frac{\boldsymbol{\eta}_1^{(t)}}{\rho_1} \right) \tag{10}$$

where P_{S_b} is the projection onto the $[0,1]^D$ box space, which can be presented as the elementwise operator $P_{S_b}(\mathbf{a}) = \min(\mathbf{1}_D, \max(\mathbf{0}_D, \mathbf{a}))$ for any $\mathbf{a} \in \mathbb{R}^D$, where $\mathbf{0}_D$ is a D-dimensional vector whose elements are all 0s.

The problem with respect to $\mathbf{v}_2$ is defined as

$$\mathbf{v}_2^{(t+1)} = \arg\min_{\mathbf{v}_2 \in S_p} \left\| \mathbf{v}_2 - \mathbf{v}^{(t+1)} - \frac{\boldsymbol{\eta}_2^{(t)}}{\rho_2} \right\|_2^2. \tag{11}$$

This problem has a convex objective function with a nonconvex constraint on the domain. The optimal solution is obtained by the Euclidean projection onto the D-dimensional sphere S_p, that is

$$\mathbf{v}_2^{(t+1)} = P_{S_p}\left(\mathbf{v}^{(t+1)} + \frac{\boldsymbol{\eta}_2^{(t)}}{\rho_2} \right) \tag{12}$$

where

$$P_{S_p}(\mathbf{a}) = \frac{\mathbf{1}_D}{2} + \frac{\sqrt{D}}{2}\frac{\mathbf{a}-\frac{\mathbf{1}_D}{2}}{\|\mathbf{a}-\frac{\mathbf{1}_D}{2}\|_2}, \quad \text{for any } \mathbf{a} \in \mathbb{R}^D. \tag{13}$$

Step 4 (Update the Lagrange Multipliers $\boldsymbol{\eta}_1^{(t+1)}$, $\boldsymbol{\eta}_2^{(t+1)}$, and $\eta_3^{(t+1)}$): The Lagrange multipliers are updated using a conventional gradient ascent method for the variables

$$\begin{aligned}\boldsymbol{\eta}_1^{(t+1)} &= \boldsymbol{\eta}_1^{(t+1)} + \rho_1\big(\mathbf{v}^{(t+1)} - \mathbf{v}_1^{(t+1)}\big)\\ \boldsymbol{\eta}_2^{(t+1)} &= \boldsymbol{\eta}_2^{(t+1)} + \rho_2\big(\mathbf{v}^{(t+1)} - \mathbf{v}_2^{(t+1)}\big)\\ \eta_3^{(t+1)} &= \eta_3^{(t+1)} + \rho_3\big(\mathbf{1}_D^T\big(\mathbf{v}^{(t+1)} - k\big)\big).\end{aligned} \tag{14}$$

IV. Top-*k* Unsupervised and Semisupervised Feature Selection Methods

Without sufficient label information, the crucial issue for unsupervised and semisupervised FS methods is data structure learning. The relevance of the utilized data structure and the input data features determine the importance of the features. Many methods for capturing the data structure using the weighted affinity graph have been proposed, such as the local linear reconstruction coefficient-based method [55], the local similarity-based method [56], and the sparsity-reconstruction coefficient-based method [57]. Meanwhile, other methods for capturing the data structure have focused on the prediction of labels using unsupervised clustering or semisupervised classification, such as the label propagation method [58] and the k-means method [59]. Following the structure regularized FS study [57], we use the weighted affinity graph to determine the soft data structure because the weights consist of real numbers, and for label prediction, we use unsupervised/semisupervised learning to determine the hard data structure because the integers are used to describe the categories.

A. Learning of Soft and Hard Data Structure

1) Learning of Soft Data Structure: Many studies have used an affinity graph method for the input data, such as the low-rank graph [60], the constrained Laplacian graph [61], and the manifold graph [55]. Inspired by the success of sparse representation in unsupervised [17] and semisupervised learning [62], we adopt the sparse coding-based graph for soft data structure learning. The general self-expressive sparse coding model is defined as

$$\begin{aligned}&\min_{\mathbf{Z},\mathbf{E}} \ |\mathbf{Z}\|_1 + \lambda_E\|\mathbf{E}\|_1\\ &\text{s.t. } \mathbf{X} = \mathbf{X}\mathbf{Z} + \mathbf{E}, \quad \text{diag}(\mathbf{Z}) = \mathbf{0}\end{aligned} \tag{15}$$

where $\mathbf{Z} \in \mathbb{R}^{N\times N}$ consists of the self-expressive coefficients, $\mathbf{E} \in \mathbf{R}^{D\times N}$ denotes the matrix of data noise, and λ_E is a tradeoff parameter. $\|\cdot\|_1$ is the ℓ_1-norm of a matrix, which is used to ensure the sparsity of the matrix. The basic assumption in (15) is that the data points are located in an area consisting of the union of the subspaces, and each point can be sparsely expressed by a linear combination of the other data points. Once the coding $\mathbf{Z}$ has been determined, we can compute the affinity matrix $\mathbf{W} = |\mathbf{Z}| + |\mathbf{Z}^T|$ as the soft data structure.

2) Learning of Hard Data Structure: Once the weight matrix $\mathbf{W}$ of the affinity graph has been learned, we can apply some advanced clustering techniques, such as spectral clustering [63] and autonomous data partitioning [64], in the unsupervised scenario to obtain the estimated labels of the data. In this article, we use the commonly used spectral clustering to obtain the partition. The objective function of spectral clustering is defined as

$$\begin{aligned}\mathbf{Y} &= \mathop{\arg\min}_{\mathbf{Y}\in\mathcal{Y}} \sum_{i,j=1}^{N} W_{i,j}\|\mathbf{y}_i - \mathbf{y}_j\|_2^2\\ &= \mathop{\arg\min}_{\mathbf{Y}\in\mathcal{Y}} \text{tr}(\mathbf{Y}\mathbf{L}\mathbf{Y}^T)\end{aligned} \tag{16}$$

where $\text{tr}(\cdot)$ is the trace operator, $\mathcal{Y}$ is the one-hot coding space, and $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]$ is the label matrix. The spectral clustering method [63] finds a suboptimal solution of (16), which first computes the smallest eigenvectors (except for the one with eigenvalue 0) of the graph Laplacian matrix $\mathbf{L}$ and then implements k-means clustering of the eigenvectors to obtain the clustering results.

Given the partly labeled data $\mathbf{X}$ and $\mathbf{Y}_l$, a semisupervised classification [65] can be utilized to estimate the label of the unlabeled data. The formulation is as follows:

$$\begin{aligned}\mathbf{Y}_u &= \mathop{\arg\min}_{\mathbf{Y}_u\in\mathcal{Y}} \sum_{i,j=1}^{N} W_{i,j}\|\mathbf{y}_i - \mathbf{y}_j\|_2^2\\ &= \mathop{\arg\min}_{\mathbf{Y}_u\in\mathcal{Y}} \text{tr}([\mathbf{Y}_l, \mathbf{Y}_u]\mathbf{L}[\mathbf{Y}_l, \mathbf{Y}_u]^T)\end{aligned} \tag{17}$$

where $\mathbf{Y}_u$ denotes the estimated label of the unlabeled data, $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the graph Laplacian matrix, and $\mathbf{D}$ is a diagonal matrix with diagonal elements $D_{ii} = \sum_{j=1}^{N} W_{ij}$.

Both the given $\mathbf{Y}_l$ and the estimated $\mathbf{Y}_u$ of the data represent the hard data structure $\mathbf{Y}$ in the semisupervised scenario. The clustering results $\mathbf{Y}$ are the hard data structure in the unsupervised scenario.

B. Data Structure for Regularized Discriminant Feature Selection

To preserve the soft structure $\mathbf{W}$ and hard data structure $\mathbf{Y}$, we can evaluate the features by minimizing the difference between the feature selected data $\mathbf{AX}$ and the structures. More formally, we can minimize the following formula:

$$\begin{aligned}\min_{\mathbf{A},\mathbf{Y},\mathbf{W}} \sum_{i,j=1}^{N} W_{ij}\left(\frac{\lambda_\Theta}{2}\|\mathbf{A}\mathbf{x}_i - \mathbf{A}\mathbf{x}_j\|_2^2 + \frac{1-\lambda_\Theta}{2}\|\mathbf{y}_i - \mathbf{y}_j\|_2^2\right)&\\ = \min_{\mathbf{A},\mathbf{Y},\mathbf{W}} \|\mathbf{W}\odot\boldsymbol{\Theta}\|_1&\end{aligned} \tag{18}$$

where $\mathbf{A}$ is the projection matrix, $\odot$ is the Hadamard product, and

$$\Theta_{ij} = \frac{\lambda_\Theta}{2}\|\mathbf{A}\mathbf{x}_i - \mathbf{A}\mathbf{x}_j\|_2^2 + \frac{1-\lambda_\Theta}{2}\|\mathbf{y}_i - \mathbf{y}_j\|_2^2 \tag{19}$$

with $\lambda_\Theta \in (0, 1)$ is a balancing parameter. Since $\mathbf{W} = |\mathbf{Z}| + |\mathbf{Z}^T|$, (18) can be equivalently rewritten as $\min_{\mathbf{A},\mathbf{Y},\mathbf{Z}} \|\mathbf{Z}\odot\boldsymbol{\Theta}\|_1$.

To preserve the hard structure $\mathbf{Y}$, we also use linear discriminant analysis (LDA) [66], which is a popular supervised feature extraction method. It seeks the directions in which the data points of the same classes are close and the data points from different classes are far away from each other. Given the label of the data, the objective function for feature selecting LDA is

$$\min_{\mathbf{A}} \frac{\mathrm{tr}(\mathbf{A}\mathbf{S}_w\mathbf{A}^T)}{\mathrm{tr}(\mathbf{A}\mathbf{S}_b\mathbf{A}^T)}. \tag{20}$$

$\mathbf{S}_w$ and $\mathbf{S}_b$ are the within-class scatter and between-class scatter matrices, respectively. In this analysis, the constraint $\|\mathbf{A}\|_{0,2} = k$ applies to the top-k FS.

C. Unsupervised and Semisupervised FS Algorithms

We combine the equations for learning the soft data structure [see (15)], hard data structure [see (16) or (17)], and the regularized discriminant FS terms [see (18) and (20)] to obtain the overall objective function

$$\begin{aligned}\min_{\mathbf{Z},\mathbf{E},\mathbf{Y},\mathbf{A}} \;& \|\mathbf{Z}\|_1 + \lambda_E\|\mathbf{E}\|_1 + \lambda_Z\|\mathbf{Z}\odot\mathbf{\Theta}\|_1 \\ & + \frac{\mathrm{tr}(\mathbf{A}\mathbf{S}_w\mathbf{A}^T)}{\mathrm{tr}(\mathbf{A}\mathbf{S}_b\mathbf{A}^T)} + \lambda_A\|\mathbf{A}\|_F^2 \\ \text{s.t. } & \mathbf{X} = \mathbf{X}\mathbf{Z} + \mathbf{E}, \quad \mathrm{diag}(\mathbf{Z}) = \mathbf{0}, \text{ and } \|\mathbf{A}\|_{0,2} = k \end{aligned} \tag{21}$$

where the Frobenius norm on $\mathbf{A}$ is used to avoid the overfitting, $\mathbf{\Theta}$ is defined in (19), and λ_E, λ_Z, and λ_A are nonnegative parameters. Instead of directly solving the $\ell_{0,2}$-norm constrained problem, as we discussed in Section III-B, we introduce a feature indicator vector $\mathbf{v} \in \{0,1\}^D$, where $v_i = 1$ indicates that the ith feature should be selected. We transform the objective function into an equivalent ℓ_0-norm constrained problem as follows:

$$\begin{aligned}\min_{\mathbf{Z},\mathbf{E},\mathbf{Y},\mathbf{A},\mathbf{v}} \;& \|\mathbf{Z}\|_1 + \lambda_E\|\mathbf{E}\|_1 + \lambda_Z\|\mathbf{Z}\odot\mathbf{\Theta}\|_1 \\ & + \frac{\mathrm{tr}(\mathbf{A}\mathrm{diag}(\mathbf{v})\mathbf{S}_w\mathrm{diag}(\mathbf{v})\mathbf{A}^T)}{\mathrm{tr}(\mathbf{A}\mathrm{diag}(\mathbf{v})\mathbf{S}_b\mathrm{diag}(\mathbf{v})\mathbf{A}^T)} + \lambda_A\|\mathbf{A}\|_F^2 \\ \text{s.t. } & \mathbf{X} = \mathbf{X}\mathbf{Z} + \mathbf{E}, \quad \mathrm{diag}(\mathbf{Z}) = \mathbf{0}, \; \mathbf{1}_D^T\mathbf{v} = k \\ & \text{and } \mathbf{v} \in \{0,1\}^D \end{aligned} \tag{22}$$

where $\mathbf{1}_D^T\mathbf{v} = k$, $\mathbf{v} \in \{0,1\}^D$ means $\|\mathbf{v}\|_0 = k$, and there are k 1s in $\mathbf{v}$, i.e., we select the exact k features. Here

$$\begin{aligned}\Theta_{ij} = & \frac{\lambda_\Theta}{2}\|\mathbf{Adiag}(\mathbf{v})\mathbf{x}_i - \mathbf{Adiag}(\mathbf{v})\mathbf{x}_j\|_2^2 \\ & + \frac{1-\lambda_\Theta}{2}\|\mathbf{y}_i - \mathbf{y}_j\|_2^2. \end{aligned} \tag{23}$$

Once let $\hat{\mathbf{A}} = \mathbf{A}\mathrm{diag}(\mathbf{v})$, we can see that (21) and (22) are essentially equivalent.

The binary constraint $\mathbf{v} \in \{0,1\}^D$ can be replaced with an equivalent set of continuous constraint, i.e., the intersection of a box and a shifted ℓ_2-sphere. Based on the discussion in Section III, we can rewrite (22) as follows:

$$\begin{aligned}\min_{\mathbf{Z},\mathbf{E},\mathbf{Y},\mathbf{A},\mathbf{v},\mathbf{v}_1,\mathbf{v}_2} \;& \|\mathbf{Z}\|_1 + \lambda_E\|\mathbf{E}\|_1 + \lambda_Z\|\mathbf{Z}\odot\mathbf{\Theta}\|_1 \\ & + \frac{\mathrm{tr}(\mathbf{A}\mathrm{diag}(\mathbf{v})\mathbf{S}_w\mathrm{diag}(\mathbf{v})\mathbf{A}^T)}{\mathrm{tr}(\mathbf{A}\mathrm{diag}(\mathbf{v})\mathbf{S}_b\mathrm{diag}(\mathbf{v})\mathbf{A}^T)} + \lambda_A\|\mathbf{A}\|_F^2 \\ \text{s.t. } & \mathbf{X} = \mathbf{X}\mathbf{Z} + \mathbf{E}, \quad \mathrm{diag}(\mathbf{Z}) = \mathbf{0}, \; \mathbf{1}_D^T\mathbf{v} = k \\ & \mathbf{v} = \mathbf{v}_1, \quad \mathbf{v} = \mathbf{v}_2, \; \mathbf{v}_1 \in S_b, \; \mathbf{v}_2 \in S_p \end{aligned} \tag{24}$$

```
Algorithm 2 ADMM for Solving (29)
Input: Data matrix $\mathbf{X}$, label matrix $\mathbf{Y}$, Laplacian matrix $\mathbf{L}$,
    parameters $\lambda_\theta$, $\lambda_Z$ and $\lambda_A$.
Output: Projection matrix $\mathbf{A}$ and vector $\mathbf{v}$.
Initialize $\mathbf{v}_1 = \mathbf{v}_2 = \mathbf{0}_D$, $\rho = 1$, and $\mu = 1.05$.
Construct centered data matrix $\bar{\mathbf{X}}$.
while not converge do
    Step 1: Compute $\mathbf{A}$ by Eq. (31).
    Step 2: Compute $\mathbf{v}$ by Eq. (32).
    Step 3: Compute $\mathbf{v}_1$ and $\mathbf{v}_2$ by Eq. (33).
    Step 4: Update $\mathbf{y}_1$, $\mathbf{y}_2$, $y_3$ and $\rho$ by Eq. (34).
end while
```

```
Algorithm 3 Top-$k$ Unsupervised and Semisupervised FSs
Input: Data matrix $\mathbf{X}$, $\mathbf{Y}_l$(semi-supervised scenario) parameters $k$, the numbers of iteration $T$, $\lambda_E$, $\lambda_Z$, $\lambda_\theta$ and $\lambda_A$.
Output: Feature selection vector $\mathbf{v}$.
Initialize $\mathbf{A}$ as a random matrix and $\mathbf{v} = \mathbf{1}_D$.
for $i = 1, 2, \ldots, T$ do
    Sub-problem 1: Compute $\mathbf{E}$ and $\mathbf{Z}$ by the Algorithm 1 in [57].
    Sub-problem 2: Implement spectral clustering on $\mathbf{W}$ to obtain the clustering labels in the unsupervised scenario, or implement semi-supervised classification with $\mathbf{W}$ and $\mathbf{Y}_l$ in the semi-supervised learning scenario.
    Sub-problem 3: Compute $\mathbf{A}$, $\mathbf{v}$, $\mathbf{v}_1$, $\mathbf{v}_2$ by the Algorithm 2.
end for
```

TABLE I

DESCRIPTION OF THE DATA SETS

	# of Instances	# of Features	# of Classes
Banc	520	2576	52
Coil20	1440	1024	20
Isolet	1560	617	26
MSRA	1799	256	12
ORL	400	1024	40
TOX	171	5748	4
WarpPIE	210	2420	10
Xm2v	2950	2576	295
Yale	165	1024	15
YaleB	2414	1024	38

where $\mathbf{\Theta}$ is defined in (23), the two sets S_b and S_p are defined as $S_b = \{\mathbf{v} : \mathbf{v} \in [0,1]^D\}$ and $S_p = \{\mathbf{v} : \|\mathbf{v} - (\mathbf{1}_D/2)\|_2^2 = (D/4)\}$. In (24), the two continuous constraints in Proposition 1 are separated by the two additional variables $\mathbf{v}_1$ and $\mathbf{v}_2$. If no data labels are provided, the problem is an unsupervised FS problem; if partial data labels are given, the problem is a semisupervised FS problem.

However, this problem cannot be optimized directly to obtain a global solution for the following reasons: 1) the objective function is nonconvex with respect to the variables $\mathbf{A}$ and $\mathbf{Y}$; 2) the domains of the variables $\mathbf{Y}$ and $\mathbf{v}_2$ are nonconvex; and 3) the variables are interrelated and have no

TABLE II

1-NN Results on All the Data Sets for Unsupervised Methods

Methods	Banc	Coil20	Isolet	MSRA	ORL	TOX	WarpPIE	Xm2v	Yale	YaleB
AllFea	0.773	0.943	0.883	0.999	0.938	0.652	1.000	0.527	0.750	0.880
PCA	0.268 ±0.049	0.791 ±0.178	0.480 ±0.150	0.943 ±0.120	0.675 ±0.177	0.433 ±0.047	0.917 ±0.153	0.158 ±0.075	0.532 ±0.091	0.604 ±0.155
Lap	0.280 ±0.126	0.659 ±0.126	0.562 ±0.121	0.872 ±0.145	0.634 ±0.127	0.496 ±0.056	0.730 ±0.124	0.063 ±0.026	0.428 ±0.077	0.417 ±0.093
MCFS	0.431 ±0.116	0.816 ±0.093	0.542 ±0.092	0.961 ±0.067	0.747 ±0.140	0.616 ±0.057	0.948 ±0.083	0.188 ±0.069	0.564 ±0.131	0.577 ±0.147
UDFS	0.530 ±0.160	0.526 ±0.366	0.353 ±0.060	0.917 ±0.073	0.665 ±0.152	0.542 ±0.089	0.847 ±0.091	0.095 ±0.035	0.437 ±0.051	0.445 ±0.124
SPFS	0.314 ±0.125	0.586 ±0.143	0.237 ±0.060	0.922 ±0.113	0.598 ±0.167	0.492 ±0.082	0.818 ±0.148	0.130 ±0.063	0.260 ±0.068	0.400 ±0.101
JELSR	0.077 ±0.026	0.764 ±0.083	0.529 ±0.167	0.869 ±0.130	0.480 ±0.150	0.538 ±0.100	0.858 ±0.139	0.080 ±0.032	0.275 ±0.047	0.431 ±0.136
RUFS	0.238 ±0.089	**0.858** **±0.119**	0.562 ±0.185	0.894 ±0.118	0.605 ±0.132	0.527 ±0.108	0.718 ±0.243	0.110 ±0.054	0.343 ±0.056	0.546 ±0.162
TRACK	0.339 ±0.100	0.786 ±0.118	0.365 ±0.146	0.869 ±0.869	0.733 ±0.185	0.558 ±0.100	0.691 ±0.155	0.125 ±0.063	0.305 ±0.040	0.481 ±0.124
SEFS21	0.279 ±0.083	0.788 ±0.112	0.354 ±0.130	0.847 ±0.126	0.734 ±0.158	0.542 ±0.079	0.949 ±0.076	0.093 ±0.041	0.420 ±0.106	0.558 ±0.167
SCUFS	0.343 ±0.089	0.156 ±0.045	0.345 ±0.105	0.933 ±0.112	0.727 ±0.202	0.544 ±0.068	0.886 ±0.139	0.091 ±0.035	0.471 ±0.117	0.373 ±0.074
URAFS	0.166 ±0.056	0.815 ±0.142	0.371 ±0.114	0.860 ±0.086	0.494 ±0.138	0.471 ±0.045	0.836 ±0.134	0.108 ±0.052	0.272 ±0.045	0.554 ±0.196
Ours	**0.607** **±0.214**	0.833 ±0.093	**0.635** **±0.163**	**0.972** **±0.053**	**0.843** **±0.147**	**0.663** **±0.080**	**0.982** **±0.040**	**0.222** **±0.091**	**0.645** **±0.088**	**0.718** **±0.183**

closed-form solutions. Therefore, we decompose this problem into several interrelated subproblems. Each subproblem is convex and can be solved.

D. Optimization

In this section, an effective solution to (24) is proposed by decomposing the problem into several subproblems. The subproblems are optimized sequentially and iteratively, as will be described in the following. Because $\mathbf{A}$ and $\mathbf{v}$ occur in the numerator, denominator, and the summed terms, it is difficult to optimize the problem directly. In this study, we use the spectral regression method [67] to transform (20) into an equivalent regression form for an easier solution.

Let $\bar{\mathbf{X}} = [\mathbf{x}_1 - \bar{\mathbf{x}}, \ldots, \mathbf{x}_N - \bar{\mathbf{x}}]$ be the centered data matrix. The between-class scatter matrix $\mathbf{S}_b$ is rewritten as

$$\mathbf{S}_b = \bar{\mathbf{X}}\bar{\mathbf{W}}\bar{\mathbf{X}}^T \tag{25}$$

where $\bar{W}_{ij} = 1/N$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same cluster and 0 otherwise. The following theorem is obtained:

Theorem 1 [57]: Let $\mathbf{T} \in \mathbb{R}^{C\times N}$ be a matrix in which each row vector is an eigenvector of the eigenproblem $\bar{\mathbf{W}}\mathbf{t} = \lambda\mathbf{t}$. If there exists a matrix $\mathbf{A} \in \mathbb{R}^{C\times D}$ where $\mathbf{A}\bar{\mathbf{X}} = \mathbf{T}$, then each row vector of $\mathbf{T}$ is an eigenvector of the generalized eigenproblem $\bar{\mathbf{X}}\bar{\mathbf{W}}\bar{\mathbf{X}}^T\alpha = \lambda\bar{\mathbf{X}}\bar{\mathbf{X}}^T\alpha$ (i.e., the eigenproblem for LDA) with the same eigenvalue λ.

Theorem 1 indicates that under mild conditions, the LDA term in (20) is equivalent to the regression formulation $\min_{\mathbf{A}} \|\mathbf{A}\bar{\mathbf{X}} - \mathbf{T}\|_F^2$, where the row vectors in $\mathbf{T}$ are eigenvectors of $\bar{\mathbf{W}}$. One advantage of this transformation is that we do not need to really solve the eigenproblem to obtain $\mathbf{T}$. The $C+1$ eigenvectors of $\bar{\mathbf{W}}$ are given as $\{\mathbf{1}\} \cup \{\mathbf{w}_c\}_{c=1}^C \subset \{0, 1\}^N$, where the jth entry of $\mathbf{w}_c$ is 1 if and only if $\mathbf{x}_j$ is in class c. Subsequently, we obtain the C useful orthogonal eigenvectors $\{\mathbf{t}_c\}_{c=1}^C$ by implementing the Gram–Schmidt orthogonalization algorithm on $\{\mathbf{1}\} \cup \{\mathbf{w}_c\}_{c=1}^C$. We rewrite (24) as

$$\begin{aligned}\min_{\mathbf{Z},\mathbf{E},\mathbf{Y},\mathbf{A},\mathbf{v},\mathbf{v}_1,\mathbf{v}_2} \; & \|\mathbf{Z}\|_1 + \lambda_E\|\mathbf{E}\|_1 + \lambda_Z\|\mathbf{Z}\odot\mathbf{\Theta}\|_1 \\ & + \|\mathbf{A}\mathrm{diag}(\mathbf{v})\bar{\mathbf{X}} - \mathbf{T}\|_F^2 + \lambda_A\|\mathbf{A}\|_F^2 \\ \text{s.t. } & \mathbf{X} = \mathbf{X}\mathbf{Z} + \mathbf{E}, \quad \mathrm{diag}(\mathbf{Z}) = \mathbf{0}, \; \mathbf{1}_D^T\mathbf{v} = k \\ & \mathbf{v} = \mathbf{v}_1, \quad \mathbf{v} = \mathbf{v}_2, \; \mathbf{v}_1 \in S_b, \; \mathbf{v}_2 \in S_p. \end{aligned} \tag{26}$$

We initialize variables $\mathbf{A}$ and $\mathbf{Y}_u$ as zero matrices, and $\mathbf{v} = \mathbf{1}_D$.

Subproblem 1 (Given $\mathbf{A}$ *and* $\mathbf{v}$*, Optimize the Variables* $\mathbf{Z}$ *and* $\mathbf{E}$*):* The problem defined in (26) with respect to $\mathbf{Z}$ and $\mathbf{E}$ is transformed into the following structured sparse problem:

$$\begin{aligned}\min_{\mathbf{Z},\mathbf{E}} \; & \|\mathbf{Z}\|_1 + \lambda_E\|\mathbf{E}\|_1 + \lambda_Z\|\mathbf{Z}\odot\mathbf{\Theta}\|_1 \\ \text{s.t. } & \mathbf{X} = \mathbf{X}\mathbf{Z} + \mathbf{E}, \quad \mathrm{diag}(\mathbf{Z}) = \mathbf{0}. \end{aligned} \tag{27}$$

To optimize (27), we introduce an auxiliary variable $\mathbf{Q}$ and optimize the following equivalent problem:

$$\begin{aligned}\min_{\mathbf{Z},\mathbf{E},\mathbf{Q}} \; & \|\mathbf{Z}\|_1 + \lambda_E\|\mathbf{E}\|_1 + \lambda_Z\|\mathbf{Z}\odot\mathbf{\Theta}\|_1 \\ \text{s.t. } & \mathbf{X} = \mathbf{X}\mathbf{Q} + \mathbf{E}, \quad \mathbf{Q} = \mathbf{Z} - \mathrm{diag}(\mathbf{Z}) \end{aligned} \tag{28}$$

TABLE III
ARI Results on All the Data Sets for Unsupervised Methods

Methods	Banc	Coil20	Isolet	MSRA	ORL	TOX	WarpPIE	Xm2v	Yale	YaleB
AllFea	0.143	0.549	0.543	0.363	0.338	0.092	0.095	0.036	0.192	0.007
PCA	0.061 ±0.006	0.382 ±0.075	0.277 ±0.106	0.252 ±0.047	0.197 ±0.049	0.016 ±0.014	0.365 ±0.065	0.017 ±0.006	0.132 ±0.014	0.025 ±0.006
Lap	0.053 ±0.016	0.287 ±0.074	0.331 ±0.057	0.208 ±0.051	0.182 ±0.025	0.079 ±0.009	0.035 ±0.002	0.009 ±0.003	0.130 ±0.025	0.017 ±0.011
MCFS	0.095 ±0.021	0.476 ±0.049	0.347 ±0.059	0.348 ±0.043	0.242 ±0.046	0.080 ±0.006	0.277 ±0.043	0.022 ±0.005	0.127 ±0.032	0.046 ±0.006
UDFS	0.129 ±0.028	0.179 ±0.140	0.125 ±0.026	0.222 ±0.035	0.196 ±0.041	0.065 ±0.026	0.148 ±0.026	0.013 ±0.004	0.134 ±0.016	0.012 ±0.003
SPFS	0.061 ±0.016	0.173 ±0.045	0.044 ±0.006	0.231 ±0.059	0.151 ±0.033	0.084 ±0.025	0.074 ±0.021	0.019 ±0.006	0.049 ±0.016	0.015 ±0.010
JELSR	0.021 ±0.005	0.408 ±0.034	0.273 ±0.076	0.215 ±0.039	0.131 ±0.038	0.079 ±0.008	0.119 ±0.020	0.013 ±0.003	0.074 ±0.021	0.027 ±0.002
RUFS	0.053 ±0.012	0.393 ±0.084	0.275 ±0.108	0.207 ±0.036	0.159 ±0.033	0.050 ±0.016	0.059 ±0.019	0.012 ±0.003	0.067 ±0.005	0.034 ±0.005
TRACK	0.066 ±0.008	0.348 ±0.062	0.130 ±0.052	0.214 ±0.013	0.236 ±0.033	0.071 ±0.012	0.032 ±0.029	0.013 ±0.005	0.074 ±0.025	0.018 ±0.011
SEFS21	0.063 ±0.012	0.328 ±0.056	0.163 ±0.056	0.188 ±0.035	0.235 ±0.051	0.070 ±0.025	0.169 ±0.050	0.010 ±0.003	0.088 ±0.014	0.031 ±0.005
SCUFS	0.069 ±0.019	0.035 ±0.012	0.122 ±0.045	0.333 ±0.073	0.244 ±0.055	0.073 ±0.022	0.106 ±0.009	0.015 ±0.005	0.091 ±0.022	0.012 ±0.003
URAFS	0.043 ±0.008	0.399 ±0.081	0.185 ±0.061	0.217 ±0.035	0.153 ±0.038	0.066 ±0.010	0.243 ±0.042	0.014 ±0.004	0.072 ±0.007	0.028 ±0.005
Ours	**0.152** **±0.047**	**0.487** **±0.060**	**0.377** **± 0.102**	**0.378** **±0.031**	**0.356** **±0.068**	**0.131** **±0.056**	**0.381** **±0.087**	**0.026** **±0.007**	**0.215** **±0.022**	**0.140** **±0.038**

which can be effectively solved, as described in [57] and presented in the supplemental material. Subsequently, the soft data structure $\mathbf{W} = |\mathbf{Z}| + |\mathbf{Z}^T|$ is also updated.

Subproblem 2 (Unsupervised or Semisupervised Learning Based on $\mathbf{W}$*):* In the unsupervised scenario, spectral clustering [63] is implemented on $\mathbf{W}$ to obtain the clustering result $\mathbf{Y}$. In the semisupervised scenario, semisupervised classification is implemented to obtain $\mathbf{Y}_u$ whose details are also presented in the supplemental material. The combination of $\mathbf{Y}_l$ and $\mathbf{Y}_u$ represents the hard data structure $\mathbf{Y}$.

Subproblem 3 (Optimize the Variables $\mathbf{A}$*,* $\mathbf{v}$*,* $\mathbf{v}_1$*, and* $\mathbf{v}_2$ *When the Other Variables Are Fixed):* After obtaining the soft and hard data structures, we rewrite the subproblem with respect to $\mathbf{A}$, $\mathbf{v}$, $\mathbf{v}_1$, and $\mathbf{v}_2$ as follows:

$$\begin{aligned}\min_{\mathbf{A},\mathbf{v},\mathbf{v}_1,\mathbf{v}_2}\; & \|\mathbf{A}\mathrm{diag}(\mathbf{v})\bar{\mathbf{X}} - \mathbf{T}\|_F^2 + \lambda_A\|\mathbf{A}\|_F^2 \\ & +\ \lambda_Z \mathrm{tr}(\mathbf{A}\mathrm{diag}(\mathbf{v})\mathbf{X}\mathbf{L}\mathbf{X}^T\mathrm{diag}(\mathbf{v})\mathbf{A}^T) \\ \text{s.t. } & \mathbf{1}_D^T\mathbf{v} = k, \quad \mathbf{v} = \mathbf{v}_1, \ \mathbf{v} = \mathbf{v}_2 \\ & \mathbf{v}_1 \in S_b, \quad \mathbf{v}_2 \in S_p \end{aligned} \tag{29}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the graph Laplacian matrix and $\mathbf{D}$ is a diagonal matrix with the diagonal elements $D_{ii} = \sum_{j=1}^N W_{ij}$.

Then, we optimize (29) for $\mathbf{A}$, $\mathbf{v}$, $\mathbf{v}_1$, and $\mathbf{v}_2$ using the ADMM. By introducing the Lagrange multipliers $\mathbf{y}_1$, $\mathbf{y}_2$, and y_3, we obtain the following augmented Lagrange function of (29):

$$\begin{aligned}\mathcal{L}_2 = & \|\mathbf{A}\mathrm{diag}(\mathbf{v})\bar{\mathbf{X}} - \mathbf{T}\|_F^2 + \lambda_A\|\mathbf{A}\|_F^2 \\ & + \lambda_Z\mathrm{tr}(\mathbf{A}\mathrm{diag}(\mathbf{v})\mathbf{X}\mathbf{L}\mathbf{X}^T\mathrm{diag}(\mathbf{v})\mathbf{A}^T) + \mathbf{y}_1^T(\mathbf{v} - \mathbf{v}_1) \\ & + \mathbf{y}_2^T(\mathbf{v} - \mathbf{v}_2) + y_3(\mathbf{1}_D^T\mathbf{v} - k) \\ & + \frac{\rho}{2}\left(\|\mathbf{v} - \mathbf{v}_1\|_2^2 + \|\mathbf{v} - \mathbf{v}_2\|_2^2 + \left(\mathbf{1}_D^T\mathbf{v} - k\right)^2\right)\end{aligned} \tag{30}$$

where $\rho > 0$ is an adaptive parameter.

Step 1: We set the partial derivative of (30) with respect to $\mathbf{A}$ to zero and obtain its closed-form solution

$$\mathbf{A} = \mathbf{T}\bar{\mathbf{X}}^T\mathrm{diag}(\mathbf{v})(\mathrm{diag}(\mathbf{v})(\bar{\mathbf{X}}\bar{\mathbf{X}}^T + \lambda_Z\mathbf{X}\mathbf{L}\mathbf{X}^T)\mathrm{diag}(\mathbf{v}) + \lambda_A\mathbf{I})^{-1} \tag{31}$$

where $\mathbf{I}$ denotes the identity matrix.

Step 2: When optimizing $\mathbf{v}$, since it is an unconstraint quadratic optimization problem, we can also set the partial derivative of $\mathcal{L}_2$ with respect to $\mathbf{v}$ to zero and obtain its closed-form solution

$$\begin{aligned}\mathbf{v} = & \left(2(\mathbf{A}^T\mathbf{A}) \odot (\bar{\mathbf{X}}\bar{\mathbf{X}} + \lambda_Z\mathbf{X}\mathbf{L}\mathbf{X}^T) + \rho\left(2\mathbf{I} + \mathbf{1}_D\mathbf{1}_D^T\right)\right)^{-1} \\ & \cdot (2\mathrm{diag}(\bar{\mathbf{X}}\mathbf{T}^T\mathbf{A}) - \mathbf{y}_1 - \mathbf{y}_2 + \rho(\mathbf{v}_1 + \mathbf{v}_2) \\ & + (y_3 - \rho k)\mathbf{1}_D).\end{aligned} \tag{32}$$

TABLE IV

NMI RESULTS ON ALL THE DATA SETS FOR UNSUPERVISED METHODS

Methods	Banc	Coil20	Isolet	MSRA	ORL	TOX	WarpPIE	Xm2v	Yale	YaleB
AllFea	0.591	0.746	0.772	0.601	0.732	0.134	0.324	0.615	0.456	0.105
PCA	0.499 ±0.007	0.602 ±0.071	0.524 ±0.110	0.463 ±0.061	0.614 ±0.046	0.048 ±0.018	0.630 ±0.087	0.596 ±0.005	0.434 ±0.015	0.188 ±0.027
Lap	0.488 ±0.020	0.550 ±0.066	0.608 ±0.049	0.432 ±0.064	0.600 ±0.021	0.114 ±0.011	0.197 ±0.004	0.577 ±0.003	0.405 ±0.021	0.160 ±0.036
MCFS	0.544 ±0.020	0.669 ±0.046	0.607 ±0.049	0.584 ±0.056	0.647 ±0.037	0.114 ±0.009	0.550 ±0.053	0.604 ±0.002	0.406 ±0.036	0.237 ±0.017
UDFS	0.579 ±0.027	0.448 ±0.129	0.386 ±0.033	0.391 ±0.044	0.611 ±0.039	0.096 ±0.030	0.364 ±0.034	0.594 ±0.003	0.406 ±0.011	0.153 ±0.010
SPFS	0.504 ±0.021	0.413 ±0.082	0.317 ±0.023	0.409 ±0.083	0.584 ±0.039	0.143 ±0.037	0.255 ±0.034	0.593 ±0.005	0.339 ±0.015	0.154 ±0.036
JELSR	0.430 ±0.008	0.612 ±0.032	0.539 ±0.088	0.422 ±0.059	0.556 ±0.042	0.116 ±0.017	0.352 ±0.041	0.579 ±0.005	0.355 ±0.024	0.187 ±0.008
RUFS	0.493 ±0.020	0.623 ±0.075	0.554 ±0.121	0.392 ±0.050	0.597 ±0.033	0.103 ±0.017	0.256 ±0.045	0.592 ±0.001	0.349 ±0.014	0.223 ±0.014
TRACK	0.521 ±0.015	0.624 ±0.050	0.393 ±0.115	0.407 ±0.042	0.654 ±0.056	0.111 ±0.015	0.166 ±0.038	0.596 ±0.006	0.360 ±0.007	0.190 ±0.050
SEFS21	0.507 ±0.011	0.563 ±0.062	0.410 ±0.081	0.384 ±0.061	0.646 ±0.043	0.110 ±0.025	0.399 ±0.075	0.591 ±0.002	0.372 ±0.016	0.226 ±0.018
SCUFS	0.522 ±0.016	0.340 ±0.031	0.367 ±0.063	0.549 ±0.099	0.640 ±0.057	0.133 ±0.056	0.346 ±0.026	0.595 ±0.003	0.400 ±0.022	0.153 ±0.027
URAFS	0.475 ±0.013	0.605 ±0.092	0.416 ±0.067	0.411 ±0.021	0.557 ±0.033	0.099 ±0.014	0.397 ±0.041	0.590 ±0.001	0.351 ±0.022	0.213 ±0.024
Ours	**0.593** **±0.042**	**0.683** **±0.057**	**0.634** **±0.095**	**0.622** **±0.031**	**0.730** **±0.050**	**0.180** **±0.066**	**0.637** **±0.044**	**0.610** **±0.005**	**0.480** **±0.026**	**0.374** **±0.057**

Step 3: We update $\mathbf{v}_1$ and $\mathbf{v}_2$ by projecting $\mathbf{v}$ into S_b and S_p. They are updated as follows:

$$\begin{cases} \mathbf{v}_1 = P_{S_b}\left(\mathbf{v} + \dfrac{\mathbf{y}_1}{\rho}\right) \\ \mathbf{v}_2 = P_{S_p}\left(\mathbf{v} + \dfrac{\mathbf{y}_2}{\rho}\right) \end{cases} \tag{33}$$

where the definitions of the projection operators P_{S_b} and P_{S_p} are given in Section III-C.

Step 4: We update the Lagrange multipliers and step size ρ as follows:

$$\begin{cases} \mathbf{y}_1 = \mathbf{y}_1 + \rho(\mathbf{v} - \mathbf{v}_1) \\ \mathbf{y}_2 = \mathbf{y}_2 + \rho(\mathbf{v} - \mathbf{v}_2) \\ y_3 = y_3 + \rho\left(\mathbf{1}_D^T\mathbf{v} - k\right) \\ \rho = \alpha\rho \end{cases} \tag{34}$$

where $\alpha > 1$ is a given parameter.

The updates for the details of the ADMM implementation for (29) are summarized in Algorithm 2.

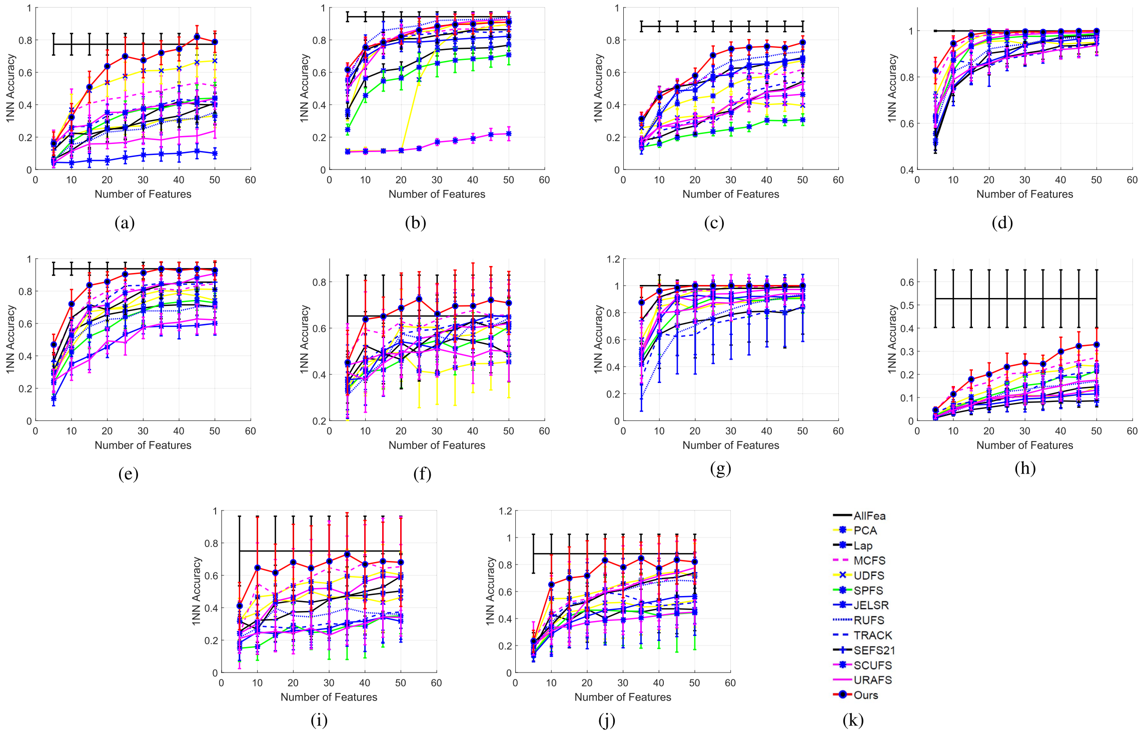
To sum up, the whole algorithms for the proposed top-k unsupervised and semisupervised FSs are summarized in Algorithm 3.

E. Time Complexity Analysis

In Algorithm 3, the three subproblems are optimized iteratively for T times. Subproblem 1 is a structure regularized sparse representation problem, and it can be solved by an ADMM procedure. The most expensive step in this ADMM procedure is computing the inverse of an $n \times n$ matrix that costs $O(N^3)$ time. Therefore, the total complexity of this procedure is $O(K_1\ N^3)$, where K_1 is the number of iterations in the corresponding ADMM procedure. For Subproblem 2, spectral clustering consumes $O(N^2c)$ computational time, where c is the number of clusters and the semisupervised learning algorithm consumes $O((N-l)^2)$ computational time where l is the number of labeled instances. For Subproblem 3, we assume that the steps 1-4 are repeated K_2 times until convergence. The major computations of Subproblem 3 come from steps 1 and 2 in (31) and (32), where the inversions of matrices of size $D \times D$ are involved. Therefore, the major computational time of Subproblem 3 scales to $O(K_2\ D^3)$. In summary, the computation time of the proposed algorithm scales to $O(T(K_1N^3 + K_2\ D^3))$. Note that the time complexity is comparable with some other subspace-based FS methods, such as in [68]. In the future, we will further study how to reduce the computation complexity of the proposed method.

V. EXPERIMENTS

In this section, experiments are conducted to compare the results of the proposed methods with those of several state-of-the-art unsupervised and semisupervised FS methods on benchmark data sets.

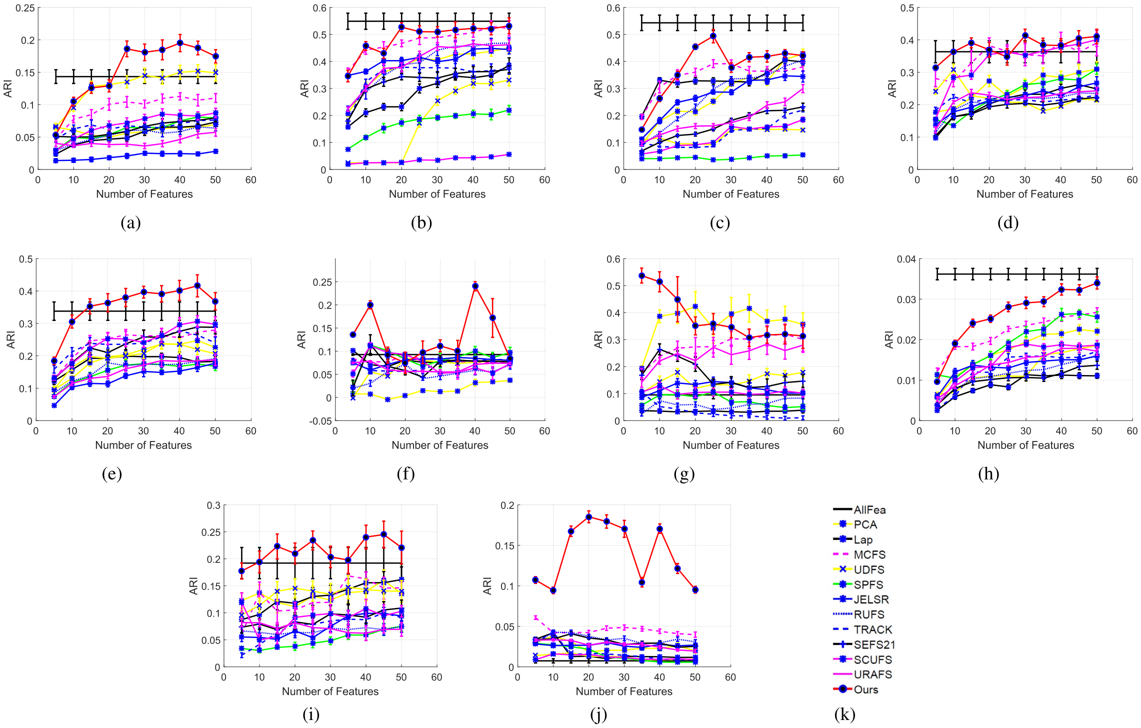


(a) (b) (c) (d)

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

(i) (j) (k)

Fig. 2. 1-NN results of unsupervised methods with different numbers of features. (a) Banc. (b) Coil20. (c) Isolet. (d) MSRA. (e) ORL. (f) TOX. (g) WarpPIE. (h) Xm2v. (i) Yale. (j) YaleB. (k) Legends.



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

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

(i) (j) (k)

Fig. 3. ARI results of unsupervised methods with different numbers of features. (a) Banc. (b) Coil20. (c) Isolet. (d) MSRA. (e) ORL. (f) TOX. (g) WarpPIE. (h) Xm2v. (i) Yale. (j) YaleB. (k) Legends.

A. Data Sets

We use ten data sets, including Banc,[1] Coil20,[2] Isolet,[2] ORL,[2] TOX,[1] WarpPIE,[1] Xm2v,[1] Yale,[2] and YaleB.[2] The details of the data sets are summarized in Table I.

B. Experiments on Unsupervised Learning

In this section, we present the setup and results of the unsupervised learning experiments.

1) Experimental Setup: We compare the unsupervised version of the proposed method with the following unsupervised FS methods.

[1] http://www.escience.cn/people/fpnie/papers.html

[2] http://www.cad.zju.edu.cn/home/dengcai/Data/data.html

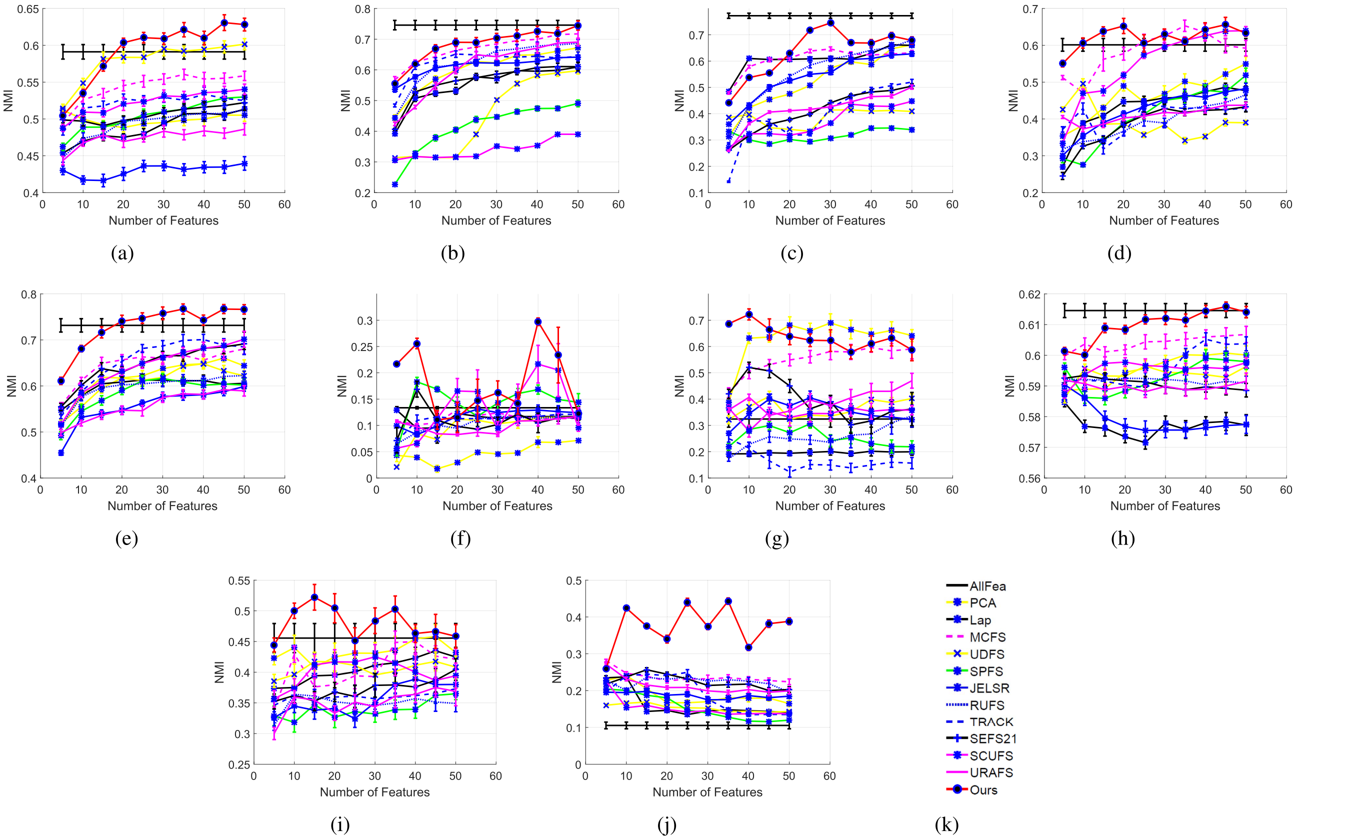


Fig. 4. NMI results of unsupervised methods with different numbers of features. (a) Banc. (b) Coil20. (c) Isolet. (d) MSRA. (e) ORL. (f) TOX. (g) WarpPIE. (h) Xm2v. (i) Yale. (j) YaleB. (k) Legends.

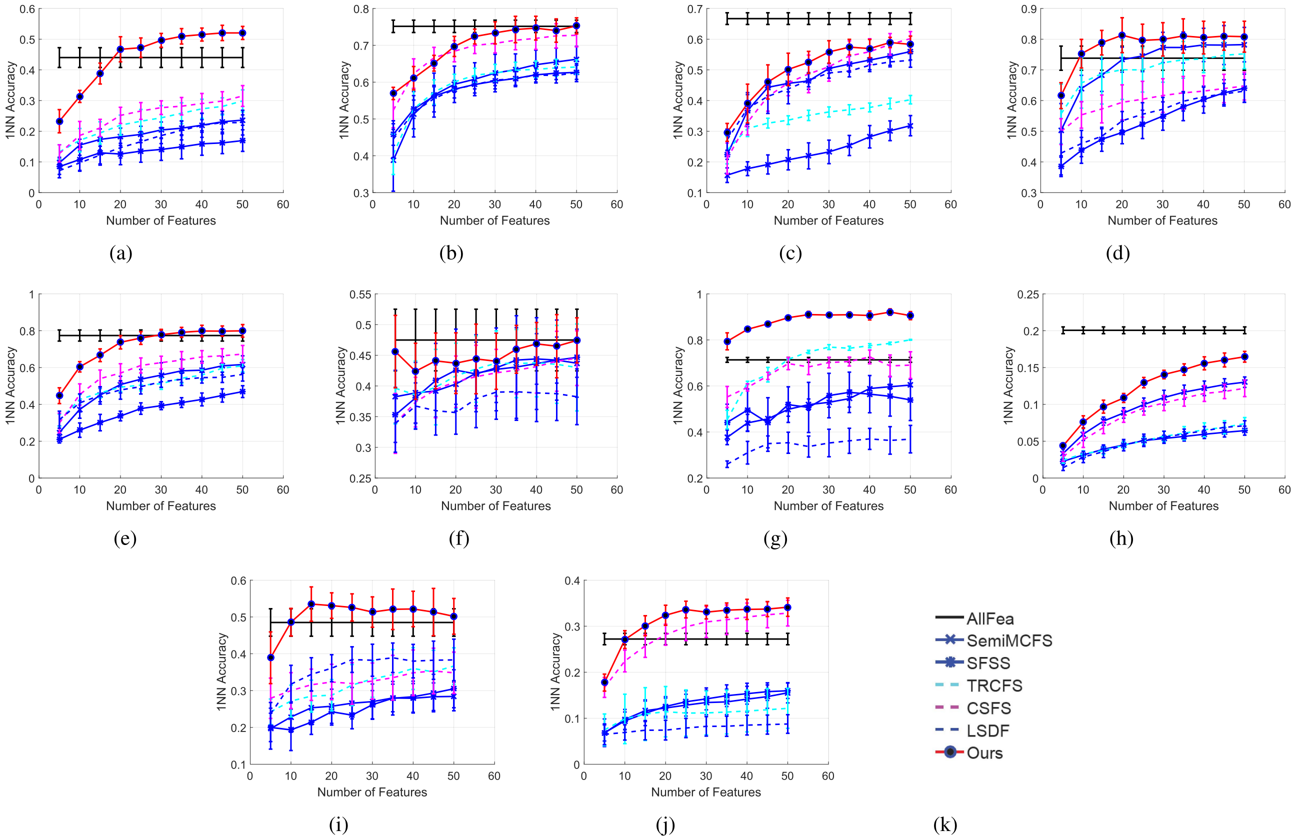
TABLE V

1-NN Results on All the Data Sets for Semisupervised Methods

Methods	Banc	Coil20	Isolet	MSRA	ORL	TOX	WarpPIE	Xm2v	Yale	YaleB
AllFea	0.440	0.752	0.667	0.738	0.774	0.475	0.713	0.201	0.485	0.272
SemiMCFS	0.190 ±0.018	0.589 ±0.044	0.234 ±0.025	0.719 ±0.045	0.507 ±0.040	0.416 ±0.049	0.515 ±0.017	0.096 ±0.006	0.260 ±0.037	0.125 ±0.023
SFSS	0.136 ±0.032	0.580 ±0.017	0.461 ±0.020	0.531 ±0.039	0.363 ±0.025	0.419 ±0.045	0.519 ±0.091	0.048 ±0.004	0.251 ±0.032	0.130 ±0.021
TRCFS	0.231 ±0.029	0.590 ±0.023	0.347 ±0.014	0.699 ±0.061	0.502 ±0.020	0.420 ±0.049	0.707 ±0.003	0.052 ±0.006	0.315 ±0.042	0.109 ±0.047
CSFS	0.250 ±0.020	0.679 ±0.025	0.470 ±0.034	0.598 ±0.043	0.574 ±0.037	0.409 ±0.039	0.668 ±0.022	0.089 ±0.008	0.325 ±0.045	0.282 ±0.022
LSDF	0.167 ±0.023	0.577 ±0.016	0.452 ±0.017	0.550 ±0.030	0.486 ±0.030	0.374 ±0.038	0.343 ±0.047	0.049 ±0.005	0.356 ±0.033	0.078 ±0.020
Ours	**0.443** **±0.019**	**0.697** **±0.020**	**0.505** **±0.022**	**0.780** **±0.045**	**0.718** **±0.022**	**0.451** **±0.035**	**0.887** **±0.007**	**0.122** **±0.003**	**0.504** **±0.030**	**0.309** **±0.014**

1) *AllFea:* It uses all features for learning.
2) *PCA [69]:* It is the principal component analysis.
3) *Lap [56]:* It sorts all features by their Laplacian score.
4) *MCFS [70]:* It is an FS method for multiclass data.
5) *UDFS [26]:* It applies the $\ell_{1,2}$-norm to select features.
6) *SPFS [71]:* It selects features to preserve the similarity.
7) *JELSR [72]:* It jointly selects features and learns the embeddings.
8) *RUFS [73]:* It is a robust FS method.
9) *TRACK [74]:* It is a k-means-based FS method.
10) *SEFS21 [14]:* It selects features for data reconstruction.
11) *SCUFS [68]:* It selects features by applying subspace clustering.
12) *URAFS [32]:* It is a generalized uncorrelated regression with an adaptive graph for FS.

After FS, we evaluate the performance in classification and clustering. For the classification experiments, we choose the 1-nearest neighbor (1-NN) classifier because



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

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

(i) (j) (k)

Fig. 5. 1-NN results of semisupervised methods with different numbers of features. (a) Banc. (b) Coil20. (c) Isolet. (d) MSRA. (e) ORL. (f) TOX. (g) WarpPIE. (h) Xm2v. (i) Yale. (j) YaleB. (k) Legends.

it is parameter-free and the results are easily reproducible. The average classification accuracy is determined by tenfold cross validation, i.e., we use 90% instances as a training set and the remaining 10% as a testing set, and we repeat it ten times. For the clustering experiments, we use k-means clustering and evaluate the performance using the adjusted rand index (ARI) [75] and normalized mutual information (NMI) [75]. Since the optimal number of selected features is unknown in advance, we report the average results and standard deviation for different numbers of selected features (in the range $\{5, 10, \ldots, 50\}$). Moreover, we also report the average results and standard deviation over the range of the selected features. We use the fixed values $\lambda_E = 10$, $\lambda_Z = 0.05$, and $\lambda_A = 0.01$ to ensure reproducibility. For the compared methods, we tune the parameters as suggested in the cited literature.

2) Experimental Results: Tables II–IV show the 1-NN, ARI, and NMI results, respectively. The results indicate that, for most data sets, the proposed method outperforms the state-of-the-art unsupervised FS methods for all evaluation indices. This demonstrates the effectiveness and superiority of the proposed method.

The details of 1-NN, ARI, and NMI results on each data set with different numbers of features are shown in Figs. 2–4. As can be seen, in most cases and for most data sets, the proposed method outperforms the other methods. On some data sets, such as ORL and Yale, our method performs better for all numbers of features. Moreover, our method often performs as well as or better than AllFea. The results demonstrate that the proposed method not only significantly reduces the number of features used for learning but also often ensures or even improves learning performance.

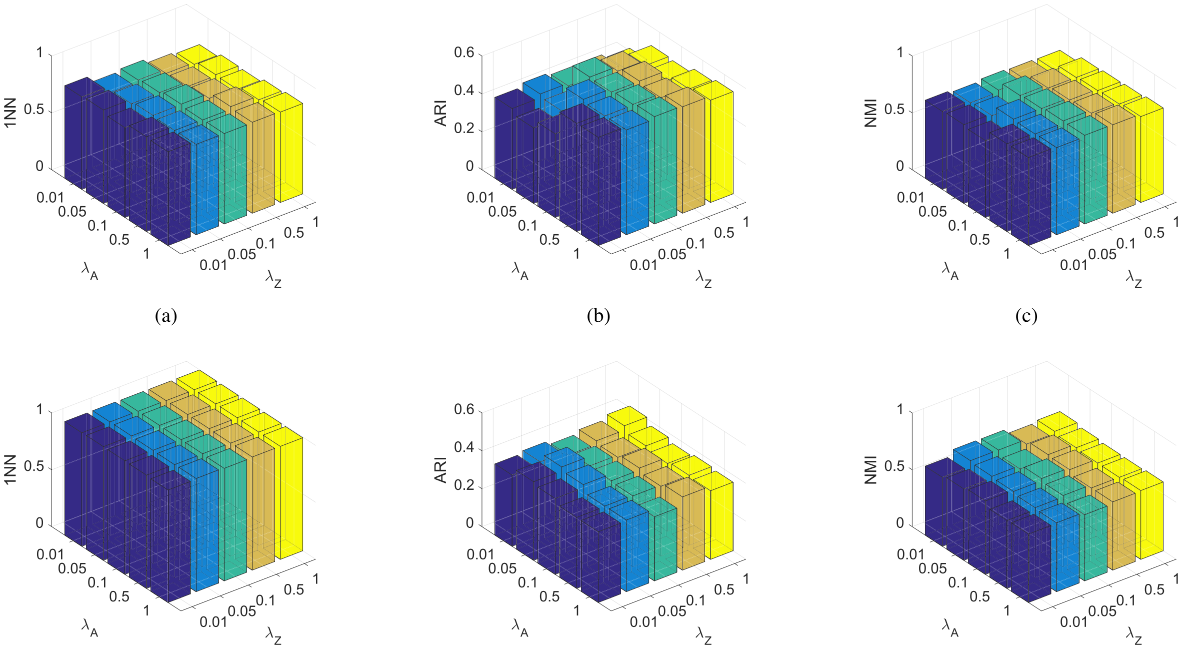
C. Experiments on Semisupervised Learning

In this section, we evaluate the performances of the semisupervised version of our method and several state-of-the-art semisupervised FS methods.

1) Experimental Setup: We compare the semisupervised version of our method with the following semisupervised FS methods.

1) *AllFea:* It uses all features for learning.
2) *SemiMCFS [70]:* It is a semisupervised version of MCFS.
3) *SFSS [76]:* It jointly selects features and learns the manifold.
4) *TRCFS [77]:* It is an FS method with a noise insensitive trace ratio criterion.
5) *CSFS [39]:* It formulates the FS in a convex formulation.
6) *LSDF [78]:* It is a locally sensitive semisupervised FS method.

In each data set and each class, we randomly select three instances as labeled data, and the remaining are unlabeled data. We repeat this ten times and report the average and



(a) (b) (c)

(d) (e) (f)

Fig. 6. Performance variation of our method on Isolet and MSRA with respect to different values of λ_A and λ_Z (number of features = 50). (a) 1-NN on Isolet. (b) ARI on Isolet. (c) NMI on Isolet. (d) 1-NN on MSRA. (e) ARI on MSRA. (f) NMI on MSRA.

standard deviation. After FS, we use the 1-NN classifier to classify the unlabeled data. Similar to the setting of the unsupervised learning experiment, we report the results for different numbers of selected features (in the range $\{5, 10, \ldots, 50\}$). We fix $\lambda_E = 10$, $\lambda_Z = 0.05$, $\lambda_A = 0.01$, and $\lambda_\Theta = 0.95$. For the other methods, we tune the parameters as suggested in the cited literature.

2) Experimental Results: Table V shows the 1-NN accuracy results of the proposed method and the other semisupervised methods. The proposed method outperforms the other semisupervised FS methods. Similar to the experiments on unsupervised learning, we also show the 1-NN results of each data set for different numbers of features in Fig. 5. For many data sets, such as WarpPIE and Yale, our method outperforms AllFea by only selecting a few features, which demonstrates the effectiveness of the proposed method. In addition, our method performs better than the other methods in most cases on all data sets.

D. Parameter Study

The two key parameters λ_Z and λ_E are used in the proposed framework. We tune these parameters in the range $\{0.01, 0.05, 0.1, 0.5, 1\}$ and show the results of the unsupervised version on the Isolet and MSRA data sets in Fig. 6. The results for the other data sets and those of the semisupervised learning version are similar. It is demonstrated that the performance of the proposed method is stable across a wide range of parameters.

VI. Conclusion

In this article, we proposed a novel exact top-*k* FS framework and applied it to the unsupervised and semisupervised learning scenarios. We directly used the ℓ_0-norm to select features and transformed the original integer programming problem into an optimization problem with two continuous constraints. We captured the soft and hard structures of the data and selected the features that preserved both structures. The extensive experiments showed that the proposed methods outperformed state-of-the-art methods for both unsupervised and semisupervised tasks on benchmark data sets, which demonstrated the effectiveness and superiority of the proposed framework.

References

[1] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: An evaluation of recent feature encoding methods," in *Proc. BMVC*, 2011, pp. 1–12.

[2] X. Chang, F. Nie, S. Wang, Y. Yang, X. Zhou, and C. Zhang, "Compound rank-*k* projections for bilinear analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 7, pp. 1502–1513, Jul. 2016.

[3] X. Zhang, D. Wang, Z. Zhou, and Y. Ma, "Robust low-rank tensor recovery with rectification and alignment," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jul. 16, 2019, doi: 10.1109/TPAMI.2019.2929043.

[4] Z. Lan, M. Lin, X. Li, A. G. Hauptmann, and B. Raj, "Beyond Gaussian pyramid: Multi-skip feature stacking for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 204–212.

[5] X. Chang, Z. Ma, Y. Yang, Z. Zeng, and A. G. Hauptmann, "Bi-level semantic representation analysis for multimedia event detection," *IEEE Trans. Cybern.*, vol. 47, no. 5, pp. 1180–1197, May 2017.

[6] C. M. Bishop, *Pattern Recognition and Machine Learning* (Information Science and Statistics). New York, NY, USA: Springer-Verlag, 2006.

[7] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Jan. 2003.

[8] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. New York, NY, USA: Springer, 2010.

[9] J. Gui, Z. Sun, S. Ji, D. Tao, and T. Tan, "Feature selection based on structured sparsity: A comprehensive study," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 7, pp. 1490–1507, Jul. 2017.

[10] E. Amaldi and V. Kann, "On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems," *Theor. Comput. Sci.*, vol. 209, nos. 1–2, pp. 237–260, Dec. 1998.

[11] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution," *Commun. Pure Appl. Math.*, vol. 59, no. 6, pp. 797–829, 2006.

[12] Y.-F. Ye, Y.-H. Shao, N.-Y. Deng, C.-N. Li, and X.-Y. Hua, "Robust ℓ_p-norm least squares support vector regression with feature selection," *Appl. Math. Comput.*, vol. 305, pp. 32–52, 2017.

[13] L. Wang, S. Chen, and Y. Wang, "A unified algorithm for mixed $\ell_{2,p}$-minimizations and its application in feature selection," *Comput. Optim. Appl.*, vol. 58, no. 2, pp. 409–421, Jun. 2014.

[14] P. Zhu, W. Zhu, W. Wang, W. Zuo, and Q. Hu, "Non-convex regularized self-representation for unsupervised feature selection," *Image Vis. Comput.*, vol. 60, pp. 22–29, Apr. 2017.

[15] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[16] M. Fan, X. Chang, X. Zhang, D. Wang, and L. Du, "Top-k supervise feature selection via admm for integer programming," in *Proc. IJCAI*, 2017, pp. 1646–1653.

[17] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.

[18] H. Tao, C. Hou, F. Nie, Y. Jiao, and D. Yi, "Effective discriminative feature selection with nontrivial solution," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 4, pp. 796–808, Apr. 2016.

[19] Q. Gu, Z. Li, and J. Han, "Generalized Fisher score for feature selection," in *Proc. 27th Conf. Uncertainty Artif. Intell.*, 2011, pp. 266–273.

[20] 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.

[21] N. Gu, M. Fan, L. Du, and D. Ren, "Efficient sequential feature selection based on adaptive eigenspace model," *Neurocomputing*, vol. 161, pp. 199–209, Aug. 2015.

[22] N. Gui, D. Ge, and Z. Hu, "AFS: An attention-based mechanism for supervised feature selection," in *Proc. AAAI*, 2019, pp. 3705–3713.

[23] H. Peng and C.-L. Liu, "Discriminative feature selection via employing smooth and robust hinge loss," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 3, pp. 788–802, Mar. 2019.

[24] D. Ming and C. Ding, "Robust flexible feature selection via exclusive L21 regularization," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 3158–3164.

[25] S. Liao, Q. Gao, F. Nie, Y. Liu, and X. Zhang, "Worst-case discriminative feature selection," in *Proc. 28th Int. Joint Conf. Artif. Intell. (IJCAI)*, Jul. 2019, pp. 2973–2979.

[26] Y. Yang, H. T. Shen, Z. Ma, Z. Huang, and X. Zhou, "ℓ_{21}-norm regularized discriminative feature selection for unsupervised learning," in *Proc. IJCAI*, 2011, pp. 1589–1594.

[27] L. Du and Y.-D. Shen, "Unsupervised feature selection with adaptive structure learning," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2015, pp. 209–218.

[28] P. Zhu, W. Zuo, L. Zhang, Q. Hu, and S. C. K. Shiu, "Unsupervised feature selection by regularized self-representation," *Pattern Recognit.*, vol. 48, no. 2, pp. 438–446, Feb. 2015.

[29] F. Nie *et al.*, "Unsupervised feature selection with structured graph optimization," in *Proc. AAAI*, 2016, pp. 1302–1308.

[30] 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. 2018.

[31] W. Zheng, C. Xu, J. Yang, J. Gao, and F. Zhu, "Low-rank structure preserving for unsupervised feature selection," *Neurocomputing*, vol. 314, pp. 360–370, Nov. 2018.

[32] X. Li, H. Zhang, R. Zhang, Y. Liu, and F. Nie, "Generalized uncorrelated regression with adaptive graph for unsupervised feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1587–1595, May 2019.

[33] P. Zhou, J. Chen, M. Fan, L. Du, Y.-D. Shen, and X. Li, "Unsupervised feature selection for balanced clustering," *Knowl.-Based Syst.*, vol. 193, Apr. 2020, Art. no. 105417.

[34] X. Zhu, S. Zhang, R. Hu, Y. Zhu, and J. Song, "Local and global structure preservation for robust unsupervised spectral feature selection," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 3, pp. 517–529, Mar. 2018.

[35] R. Shang, W. Wang, R. Stolkin, and L. Jiao, "Non-negative spectral learning and sparse regression-based dual-graph regularized feature selection," *IEEE Trans. Cybern.*, vol. 48, no. 2, pp. 793–806, Feb. 2018.

[36] P. Zhou, L. Du, X. Li, Y.-D. Shen, and Y. Qian, "Unsupervised feature selection with adaptive multiple graph learning," *Pattern Recognit.*, vol. 105, Sep. 2020, Art. no. 107375.

[37] C. Feng, C. Qian, and K. Tang, "Unsupervised feature selection by Pareto optimization," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 3534–3541.

[38] 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.

[39] X. Chang, F. Nie, Y. Yang, and H. Huang, "A convex formulation for semi-supervised multi-label feature selection," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014.

[40] X. Chang and Y. Yang, "Semisupervised feature analysis by mining correlations among multiple tasks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2294–2305, Oct. 2017.

[41] M. Luo, X. Chang, L. Nie, Y. Yang, A. G. Hauptmann, and Q. Zheng, "An adaptive semisupervised feature analysis for video semantic recognition," *IEEE Trans. Cybern.*, vol. 48, no. 2, pp. 648–660, Feb. 2018.

[42] J. Xu, B. Tang, H. He, and H. Man, "Semisupervised feature selection based on relevance and redundancy criteria," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 9, pp. 1974–1984, Sep. 2017.

[43] X. Chen, W. Liu, F. Su, and G. Zhou, "Semisupervised multiview feature selection for VHR remote sensing images with label learning and automatic view generation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 6, pp. 2876–2888, Jun. 2017.

[44] B. Jiang, X. Wu, K. Yu, and H. Chen, "Joint semi-supervised feature selection and classification through Bayesian approach," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 3983–3990.

[45] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc. B, Methodol.*, vol. 58, no. 1, pp. 267–288, Jan. 1996.

[46] A. Destrero, C. De Mol, F. Odone, and A. Verri, "A sparsity-enforcing method for learning face features," *IEEE Trans. Image Process.*, vol. 18, no. 1, pp. 188–201, Jan. 2009.

[47] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Roy. Stat. Soc. B, Stat. Methodol.*, vol. 67, no. 2, pp. 301–320, Apr. 2005.

[48] X. Yang, Z.-Y. Liu, and H. Qiao, "A continuation method for graph matching based feature correspondence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 1809–1822, Aug. 2019.

[49] S. Zhang, J. Huang, H. Li, and D. N. Metaxas, "Automatic image annotation and retrieval using group sparsity," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 3, pp. 838–849, Jun. 2012.

[50] R. Jenatton, J.-Y. Audibert, and F. Bach, "Structured variable selection with sparsity-inducing norms," *J. Mach. Learn. Res.*, vol. 12, pp. 2777–2824, Feb. 2011.

[51] 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, Nov. 2012.

[52] X. Cai, F. Nie, and H. Huang, "Exact top-*k* feature selection via ℓ_2, 0-norm constraint," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 1241–1246.

[53] T. Pang, F. Nie, J. Han, and X. Li, "Efficient feature selection via $\ell_{2,0}$ 2,0-norm constrained sparse regression," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 5, pp. 880–893, May 2019.

[54] B. Wu and B. Ghanem, "P-box ADMM: A versatile framework for integer programming," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1695–1708, Jul. 2019.

[55] C. Yao, Y.-F. Liu, B. Jiang, J. Han, and J. Han, "LLE score: A new filter-based unsupervised feature selection method based on nonlinear manifold embedding and its application to image recognition," *IEEE Trans. Image Process.*, vol. 26, no. 11, pp. 5257–5269, Nov. 2017.

[56] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 507–514.

[57] M. Fan, X. Chang, and D. Tao, "Structure regularized unsupervised discriminant feature analysis," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1870–1876.

[58] Y. Liu, F. Nie, J. Wu, and L. Chen, "Semi-supervised feature selection based on label propagation and subset selection," in *Proc. Int. Conf. Comput. Inf. Appl.*, Dec. 2010, pp. 293–296.

[59] Y.-M. Xu, C.-D. Wang, and J.-H. Lai, "Weighted multi-view clustering with feature selection," *Pattern Recognit.*, vol. 53, pp. 25–35, May 2016.

[60] L. Zhuang *et al.*, "Constructing a nonnegative low-rank and sparse graph with data-adaptive features," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3717–3728, Nov. 2015.

[61] F. Nie, X. Wang, M. I. Jordan, and H. Huang, "The constrained Laplacian rank algorithm for graph-based clustering," in *Proc. 13th AAAI Conf. Artif. Intell. (AAAI)*, 2016, pp. 1969–1976.

[62] M. Fan, N. Gu, H. Qiao, and B. Zhang, "Sparse regularization for semi-supervised classification," *Pattern Recognit.*, vol. 44, no. 8, pp. 1777–1784, Aug. 2011.

[63] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 849–856.

[64] X. Gu, P. P. Angelov, and J. C. Príncipe, "A method for autonomous data partitioning," *Inf. Sci.*, vols. 460–461, pp. 65–82, Sep. 2018.

[65] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. 20th Int. Conf. Mach. Learn. (ICML)*, 2003, pp. 912–919.

[66] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. Amsterdam, The Netherlands: Elsevier, 2013.

[67] D. Cai, X. He, and J. Han, "SRDA: An efficient algorithm for large-scale discriminant analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 1, pp. 1–12, Jan. 2008.

[68] P. Zhu, W. Zhu, Q. Hu, C. Zhang, and W. Zuo, "Subspace clustering guided unsupervised feature selection," *Pattern Recognit.*, vol. 66, pp. 364–374, Jun. 2017.

[69] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *J. Educ. Psychol.*, vol. 24, no. 6, p. 417, 1933.

[70] 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*, 2010, pp. 333–342.

[71] Z. Zhao, L. Wang, H. Liu, and J. Ye, "On similarity preserving feature selection," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 3, pp. 619–632, Mar. 2013.

[72] C. Hou, F. Nie, X. Li, D. Yi, and Y. Wu, "Joint embedding learning and sparse regression: A framework for unsupervised feature selection," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 793–804, Jun. 2014.

[73] M. Qian and C. Zhai, "Robust unsupervised feature selection," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 1621–1627.

[74] D. Wang, F. Nie, and H. Huang, "Unsupervised feature selection via unified trace ratio formulation and K-means clustering (TRACK)," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Nancy, France: Springer, 2014, pp. 306–321.

[75] S. Romano, J. Bailey, V. Nguyen, and K. Verspoor, "Standardized mutual information for clustering comparisons: One step further in adjustment for chance," in *Proc. 31st Int. Conf. Mach. Learn.*, Beijing, China, vol. 32, Jun. 2014, no. 2, pp. 1143–1151.

[76] Z. Ma, F. Nie, Y. Yang, J. R. R. Uijlings, N. Sebe, and A. G. Hauptmann, "Discriminating joint feature analysis for multimedia data understanding," *IEEE Trans. Multimedia*, vol. 14, no. 6, pp. 1662–1672, Dec. 2012.

[77] Y. Liu, F. Nie, J. Wu, and L. Chen, "Efficient semi-supervised feature selection with noise insensitive trace ratio criterion," *Neurocomputing*, vol. 105, pp. 12–18, Apr. 2013.

[78] J. Zhao, K. Lu, and X. He, "Locality sensitive semi-supervised feature selection," *Neurocomputing*, vol. 71, nos. 10–12, pp. 1842–1849, Jun. 2008.

Xiaoqin Zhang (Member, IEEE) received the B.Sc. degree in electronic information science and technology from Central South University, Changsha, China, in 2005, and the Ph.D. degree in pattern recognition and intelligent system from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2010.

He is currently a Professor with the College of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou, China. He has authored or coauthored more than 100 articles in international and national journals and international conferences. His research interests include pattern recognition, computer vision, and machine learning.

Mingyu Fan received the B.Sc. degree in information and computing science from the Central University for Nationalities, Beijing, China, in 2006, and the Ph.D. degree in applied mathematics from the Academy of Mathematics and System Science, Chinese Academy of Sciences, Beijing, in 2011.

He is currently a Professor with the College of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou, China. His current research interests include machine learning, artificial intelligence, and their applications in industrials.

Di Wang received the B.Sc. degree in information and computing science from Shandong University, Jinan, China, in 2007, and the Ph.D. degree in applied mathematics from the Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China, in 2012.

He is currently an Associate Professor with the Center of Intelligent Decision-Making and Machine Learning, School of Management, Xi'an Jiaotong University, Xi'an, China. His current research interests include machine learning and image denoising.

Peng Zhou received the B.E. degree in computer science and technology from the University of Science and Technology of China, Hefei, China, in 2011, and the Ph.D. degree in computer science from the Institute of Software, University of Chinese Academy of Sciences, Beijing, China, in 2017.

He is currently a Lecturer with the School of Computer Science and Technology, Anhui University, Hefei. His research interests include machine learning, data mining, and artificial intelligence.

Dacheng Tao (Fellow, IEEE) is currently a Professor of computer science and an ARC Laureate Fellow with the School of Computer Science and the Faculty of Engineering and the Inaugural Director of the UBTECH Sydney Artificial Intelligence Centre, The University of Sydney, Sydney, NSW, Australia. His research results in artificial intelligence have expounded in one monograph and more than 200 publications at prestigious journals and prominent conferences, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), the *International Journal of Computer Vision* (IJCV), the *Journal of Machine Learning Research* (JMLR), the *Artificial Intelligence* (AIJ), the AAAI Conference on Artificial Intelligence (AAAI), the International Joint Conference on Artificial Intelligence (IJCAI), Conference on Neural Information Processing Systems (NeurIPS), the International Conference on Machine Learning (ICML), the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), the International Conference on Computer Vision (ICCV), the European Conference on Computer Vision (ECCV), the IEEE International Conference on Data Mining (ICDM), and the ACM Knowledge Discovery and Data Mining (KDD), with several best paper awards.

Dr. Tao is also a Fellow of the American Association for the Advancement of Science (AAAS), the Association for Computing Machinery (ACM), and the Australian Academy of Science. He received the 2018 IEEE ICDM Research Contributions Award and the 2015 Australian Museum Scopus-Eureka Prize.