Half-Quadratic Minimization for Unsupervised Feature Selection on Incomplete Data

Heng Tao Shen, *Senior Member, IEEE*, Yonghua Zhu, Wei Zheng, and Xiaofeng Zhu, *Senior Member, IEEE*

***Abstract*— Unsupervised feature selection (UFS) is a popular technique of reducing the dimensions of high-dimensional data. Previous UFS methods were often designed with the assumption that the whole information in the data set is observed. However, incomplete data sets that contain unobserved information can be often found in real applications, especially in industry. Thus, these existing UFS methods have a limitation on conducting feature selection on incomplete data. On the other hand, most existing UFS methods did not consider the sample importance for feature selection, i.e., different samples have various importance. As a result, the constructed UFS models easily suffer from the influence of outliers. This article investigates a new UFS method for conducting UFS on incomplete data sets to investigate the abovementioned issues. Specifically, the proposed method deals with unobserved information by using an indicator matrix to filter it out the process of feature selection and reduces the influence of outliers by employing the half-quadratic minimization technique to automatically assigning outliers with small or even zero weights and important samples with large weights. This article further designs an alternative optimization strategy to optimize the proposed objective function as well as theoretically and experimentally prove the convergence of the proposed optimization strategy. Experimental results on both real and synthetic incomplete data sets verified the effectiveness of the proposed method compared with previous methods, in terms of clustering performance on the low-dimensional space of the high-dimensional data.**

***Index Terms*— Feature selection, half-quadratic minimization, incomplete data, robust statistics, sparse learning.**

I. Introduction

WHILE high-dimensional representation is widely applied in real applications to build high-quality machine learning models for data analysis, it still suffers some issues, such as high computation time, storage cost, and the issue of curse of dimensionality [1], [2]. Moreover, the data set with high-dimensional representation often contains redundant features and outliers to possibly degrade the performance of data analysis. To address these issues, dimensionality reduction was designed to reduce the dimensions of high-dimensional data by removing unimportant features [3], [4].

Unsupervised feature selection (UFS) is one of the dimensionality reduction techniques to handle high-dimensional data since UFS has interpretability and obtaining labels of the samples is usually difficult in real applications. By considering the way of the model construction [5], [6], previous UFS methods can be partitioned into three groups, i.e., filter model, wrapper model, and embedded model. The filter model is the most simple and efficient, whereas the embedded model usually achieves the best effectiveness [1]. Hence, this article is focusing on investigating a new embedded model of UFS methods to deal with the issue of high-dimensional data under the setting of unsupervised learning.

Feature selection first assumes that different features have different importance for data analysis, i.e., feature importance for short, and then keeps important features and removes unimportant features to reduce the dimensions of high-dimensional data. Actually, samples also have such characteristics. Specifically, different samples produce different influences for data analysis, i.e., sample importance for short. For example, outliers may have smaller importance compared with other samples because outliers usually result in larger residual than other samples [3], [7]. Moreover, both feature importance and sample importance are necessary for data analysis [8]. Otherwise, the performance of data analysis will be influenced if only considering each of them [9], [10]. However, most previous UFS methods do not take sample importance into account for feature selection [1], [11].

The incomplete data set with unobserved information in some samples is very normal in practical applications. In particular, an industrial data set may miss 90% of the whole information [12], [13]. On the other hand, most techniques of machine learning are designed to conduct data analysis on complete data sets where the whole information is observed. Hence, these techniques cannot be directly applied in incomplete data sets. Recently, a number of solutions have been proposed to deal with unobserved data by outputting the complete data set with different strategies so that existing machine learning techniques can be employed [12], [14], [15].

The classic solutions to deal with unobserved information include case-deletion methods and imputation methods [12], [15]. The case-deletion method removes incomplete samples

Manuscript received September 17, 2019; revised May 11, 2020; accepted July 11, 2020. Date of publication July 30, 2020; date of current version July 7, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0102200, in part by the National Natural Science Foundation of China under Grant 61632007 and Grant 61876046, and in part by the Sichuan Science and Technology Program under Grant 2018GZDZX0032. *(Corresponding author: Xiaofeng Zhu.)*

Heng Tao Shen and Xiaofeng Zhu are with the Center for Future Media, University of Electronic Science and Technology of China, Chengdu 611731, China, and also with the School of Computer Science and Technology, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: shenhengtao@hotmail.com; seanzhuxf@gmail.com).

Yonghua Zhu and Wei Zheng are with the Guangxi Key Laboratory of Multi-Source Information Mining and Security, Guangxi Normal University, Guilin 541004, China.

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.2020.3009632

(i.e., the samples with unobserved information) from the data set and only uses the complete samples (i.e., the whole information of the samples is observed) for data analysis. Obviously, the case-deletion method overlooks the observed information in incomplete samples as the incomplete samples include observed information as well as unobserved information. Moreover, the case-deletion method uses limited information by discarding incomplete samples so that it cannot guarantee the effectiveness of data analysis [14], [15]. To address these issues, the imputation method first guesses a value for unobserved information (i.e., imputation) and then utilizes all complete samples (including the incomplete samples whose unobserved information has been imputed) for data analysis. Although the imputation method utilizes more information (i.e., the incomplete samples) compared with the case-deletion method, the imputed values of unobserved information may be unuseful or even noisy. The reason is that there is no ground truth for unobserved information so that the correctness of the imputed information cannot be evaluated [14].

This article extends our conference version [16] to propose a new UFS method by involving two following components: 1) using an indicator matrix to utilize all observed information in both complete samples and incomplete samples for conducting feature selection on incomplete data sets and 2) employing the half-quadratic minimization technique [9], [11] to flexibly reduce the influence of outliers. Specifically, the use of the indicator matrix prohibits unobserved information involving the process of feature selection and makes the use of observed information in incomplete samples, whereas the use of the half-quadratic minimization technique pushes outliers to reduce the influence for feature selection. As a result, the proposed method could simultaneously remove sample-level outliers and feature-level redundancy for conducting UFS on incomplete data sets.

Compared with previous UFS methods, we summarize the contribution of our proposed method as follows.

1) Previous UFS methods (see [17], [18]) use an $\ell_{2,1}$-norm loss function to reduce the influence of outliers. The use of the $\ell_{2,1}$-norm loss function is fixed and has no theoretical guarantee. On the contrary, the proposed method employs half-quadratic minimization [3], [19] to achieve the same goal so that it is flexible and has theoretical guarantee by robust statistics [9]. Specifically, the proposed method first automatically selects easy samples (i.e., the samples outputting top smallest residuals between the predictions and the ground truths[1]) to construct an initial feature selection model and then automatically detect other easy samples from the left samples to improve the robustness and generalization ability of this initial model. This process of easy sample selection will be repeated until this model achieves convergence or all samples have been selected. In this way, outliers will either be selected later than easy samples or never be selected. As a result, the influence of outliers is reduced. Moreover, the number of easy samples in each iteration is data-driven to make our method flexible.
2) To deal with incomplete data sets, the case-deletion method uses a part of the whole information, whereas the imputation method uses the whole information but it possibly introduces noise. By contrast, our proposed method introduces an indicator matrix to remove unobserved information from the process of feature selection so that all observed information can be used and less noise is involved.
3) To the best of our knowledge, the proposed method is the first work to integrate feature selection, dealing with unobserved data, and half-quadratic minimization, in a unified framework since previous works were only designed to focus on a part of these three tasks [20], [21]. Moreover, experimental results on real and synthetic incomplete data sets verified the effectiveness of the proposed method compared with previous state-of-the-art methods. Furthermore, it is easy to extend the proposed method for dealing with unobserved data to other tasks, such as other UFS methods, supervised feature selection, and semisupervised feature selection.

[1]Easy samples are defined as the samples outputting top smallest residuals in each iteration. Moreover, easy samples may change in different iterations and the selection of easy samples is dependent on the residual between the ground truth and the prediction for each sample. Furthermore, easy samples in the last iteration are called important samples.

II. Related Work

In this section, we review previous methods of feature selection as well as analyze the difference between our proposed method and them.

A. Feature Selection on Complete Data

The goal of feature selection is to find a useful subset from all original features. This makes feature selection interpretable [22], [23]. Specifically, the filter models conduct a two-step strategy for feature selection by first ranking the importance of every feature and then selecting the most important features. In the two-step strategy, the optimal results of the first step cannot guarantee to output the optimal of the second step [3], [10]. The wrapper model first employs an exhaustive search method to generate the subsets of all features, i.e., $2^d - 1$ subsets for d features, and then applies the model of data analysis to evaluate the importance of different subsets. Obviously, exhaustive search makes the wrapper model possibly obtain an optimized subset of features, but the time complexity of the wrapper model is very high. Thus, the wrapper model prohibits applying in data sets with a large number of features. Furthermore, the wrapper model is a two-step strategy for feature selection. The embedded model integrates feature ranking with the process of feature selection in a unified framework, i.e., a one-step strategy, to automatically select important features.

Based on the label information, the embedded model can be partitioned into three subgroups, i.e., unsupervised embedded model, supervised embedded model, and semisupervised embedded model. Specifically, unsupervised embedded model employs either dictionary learning methods or self-representation methods to conduct feature selection without

the supervision of labels. For example, the method of regularized self-representation (RSR) utilizes the $\ell_{2,1}$-norm loss function based on the self-representation property of features as well as uses the $\ell_{2,1}$-norm regularization to select features [24]. Supervised embedded model applies the $\ell_{2,r}$-norm loss function based on the difference between the labels and the predictions to reduce the influence of outliers as well as applies the $\ell_{2,p}$-norm regularization to conduct feature selection where both r and p are nonnegative tuning parameters. Moreover, their ranges are set as $0 < r \leq 2$ and $0 < p \leq 1$. For example, the method of general sparsity feature selection employs the $\ell_{2,r}$-norm loss function and the $\ell_{2,1}$-norm regularization [25], whereas the method of simultaneous capped ℓ_2-norm loss function and $\ell_{2,p}$-norm regularization minimization uses the capped ℓ_2-norm loss function and the $\ell_{2,p}$-norm regularization [26]. In semisupervised embedded model, the method of robust feature selection (RFS) with linear discriminant analysis [8] simultaneously finds sample- outliers and feature noises by using both labeled samples and unlabeled samples to conduct robust and discriminative semisupervised feature selection.

The aforementioned feature selection methods propose different strategies of robustness to reduce the influence of outliers, i.e., the $\ell_{2,r}$-norm loss function and the capped ℓ_2-norm loss function. In this article, we focus on employing half-quadratic minimization to conduct UFS.

B. Feature Selection on Incomplete Data

The literature (see [15], [20]) usually designs a two-stage strategy to conduct feature selection on incomplete data sets, i.e., using imputation methods to guess a value for unobserved information and then conducting feature selection with previous techniques on imputed data sets. For example, the method of feature selection based on conditional entropy [27] first uses entropy-based uncertainty measures to deal with unobserved information and then conducts feature selection. Recently, Zhang *et al.* [15] proposed first conducting feature selection and then imputing unobserved information by decision tree techniques. The method of multicriteria-based feature selection on cost-sensitive data with missing values [28] proposed first selecting features and then using a rough set theory to deal with unobserved information.

Different from previous literature which separately imputes unobserved information and conducts feature selection, our proposed method prohibits unobserved information to involve the process of feature selection by introducing an indicator matrix so that it simultaneously handles unobserved information, reducing the influence of outliers, and conducts feature selection by utilizing all observed information.

III. Approach

In this article, we use boldface uppercase letters, boldface lowercase letters, and normal italic letters, respectively, to denote matrices, vectors, and scalars.

A. Robust Feature Selection

Sparse theory has widely been used in embedded models since it selects features by pushing the weight coefficients of unimportant features to have small values or even zeros and the weight coefficients of important features to have large values [22]. Specifically, given a feature matrix $\mathbf{X} = [\mathbf{x}^1; \ldots; \mathbf{x}^n] = [\mathbf{x}_1, \ldots, \mathbf{x}_d] \in \mathbb{R}^{n\times d}$, traditional unsupervised sparse feature selection methods [24], [29], [30] can be formulated as

$$\min_{\mathbf{W}} ||\mathbf{X} - \mathbf{X}\mathbf{W}||_F^2 + \lambda||\mathbf{W}||_{2,1} \tag{1}$$

where λ is a nonnegative tuning parameter, $\mathbf{W} \in \mathbb{R}^{d\times d}$ is the weight coefficient matrix, $||\cdot||_F$ is a Frobenius norm, and $||\mathbf{W}||_{2,1}$ (i.e., $||\mathbf{W}||_{2,1} = \sum_i(\sum_j w_{ij}^2)^{1/2}$) is an $\ell_{2,1}$-norm sparsity regularizer, which selects important features by automatically assigning their corresponding rows of the weight coefficients $\mathbf{W}$ with nonzero values.

Equation (1) equivalently considers every sample, so the resulting model is easily influenced by outliers [9], [31]. To address this issue, Nie *et al.* [31] proposed to replace the Frobenius norm in (1) by an $\ell_{2,1}$-norm loss function to have

$$\min_{\mathbf{W}} ||\mathbf{X} - \mathbf{X}\mathbf{W}||_{2,1} + \lambda||\mathbf{W}||_{2,1}. \tag{2}$$

In (2), the residual of $\mathbf{X}-\mathbf{X}\mathbf{W}$ (i.e., $||\mathbf{X}-\mathbf{X}\mathbf{W}||_{2,1}$) does not have a squared operator, and thus, outliers have less influence compared with Frobrenious norm [31]. Compared with (1), (2) considers sample importance for feature selection. However, (2) is less flexible as it cannot guarantee the effectiveness without considering to reduce the influence of sample-level outliers. Moreover, both (1) and (2) cannot be used for dealing with incomplete data sets.

In this article, we employ the theory of half-quadratic minimization [9], [32], such as self-paced learning [33], [34] and M-estimation (i.e., maximum-likelihood-type estimation) [35] to replace the $\ell_{2,1}$-norm loss function in (2) with predefined robust loss functions, aimed at achieving flexibly RFS. In half-quadratic minimization, functions can be constructed so that they can first select easy samples to construct an initial model, and the left easy samples are automatically detected from the left samples joining with the former easy samples to revise the robustness and generalization of this initial model gradually until either this model is not further improved or all the samples are used up [33], [34]. By replacing the $\ell_{2,1}$-norm loss function in (2) with a general robust loss function $\phi(\cdot)$, the RFS framework can be formulated as

$$\min_{\mathbf{W}} \phi(||\mathbf{X} - \mathbf{X}\mathbf{W}||_F) + \lambda||\mathbf{W}||_{2,1}. \tag{3}$$

A number of robust loss functions have been designed in the literature [19], [36], such as Cauchy function, ℓ_1-ℓ_2 function, Welsch M-estimator, and GemanMcClure estimator. Every robust loss function $\phi(\cdot)$ has individual characteristics different from the others and can be flexibly selected in real applications, e.g., changing $\phi(\cdot)$ to achieve flexibility. In this way, (2) can be regarded as a special issue of (3).

B. Robust Feature Selection on Incomplete Data

A few feature selection methods have been designed to deal with incomplete data sets. A possible method of conducting feature selection on incomplete data sets is to construct feature

selection models only using complete samples by discarding incomplete samples. Obviously, the larger the number of incomplete samples, the lower the robustness and generalization ability of the feature selection model. To address this issue, an alternative method is to first impute unobserved values with imputation methods (such as mean-value method [37] and k nearest neighbors (kNNs) imputation method [15]) and then conduct feature selection using all the samples, including the imputed incomplete samples. However, the imputation strategy for feature selection is usually unpractical. Specifically, if the imputation models are estimated perfectly, then the constructed feature selection models may be overfitting. Otherwise, the models will be underfitting with a bad imputation. Moreover, we have no idea on the ground truth of unobserved information so that it is difficult to evaluate imputation models. Furthermore, the imputed information is obtained by the complete samples, so the imputation method does not utilize the observed information in incomplete samples, so does the case-deletion method.

Based on the above observations, it may be unnecessary to impute unobserved information for feature selection. To make the use of the observed information in incomplete samples, we propose to use an indicator matrix $\mathbf{D} \in \mathbb{R}^{n\times d}$ (which has the same size as the feature matrix $\mathbf{X}$) to avoid unobserved values involving the process of feature selection. We thus propose the following objective function to conduct RFS on incomplete data sets:

$$\min_{\mathbf{W}} \phi(||\mathbf{D} \circ (\mathbf{X} - \mathbf{X}\mathbf{W})||_F) + \lambda||\mathbf{W}||_{2,1} \tag{4}$$

where $\mathbf{D} = [\mathbf{d}^1; \ldots; \mathbf{d}^n]$ is an indicator matrix. Specifically, if the ith row and the jth column element x_{ij} are unobserved, the value of d_{ij} is 0, otherwise 1. The symbol $\circ$ is a Hadamard product operator that conducts the elementwise multiplication between two same size matrices.

According to (4), our proposed method conducts feature selection by using all available information (without imputing unobserved values) as well as reducing the influence of outliers. Moreover, the definition of $\mathbf{D}$ can be applied in any sparse feature selection models for conducting robust UFS using other embedded models, supervised feature selection, and semisupervised feature selection, on incomplete data sets.

C. Proposed Objective Function

Although a number of robust loss functions have been reported in the literature, the resulting objective functions in these predefined robust functions may be optimized difficultly or inefficiently or even nonconvex. To address this, the half-quadratic minimization technique introduces an auxiliary variable $\mathbf{v}$.

Lemma 1: Given a fixed scalar z, if a differentiable function $\phi(z)$ satisfies four conditions listed in [19], then the following holds:

$$\phi(z) = \inf_{z\in\mathbb{R}}\{\mathbf{v}z^2 + \psi(\mathbf{v})\} \tag{5}$$

where $\mathbf{v}$ is a variable determined by the minimization function of the dual potential function $\psi(\mathbf{v})$ of the differentiable loss function $\phi(z)$.

Based on Lemma 1, each robust loss function $\phi(z)$ theoretically has a corresponding potential function $\psi(\mathbf{v})$, which is optimized by its minimization function or directly optimized by other optimization methods. In this article, for a fixed scalar z, we select a specific robust loss function, i.e., GemanMcClure loss function [38]

$$\phi(z) = \frac{\mu z^2}{\mu + z^2} \tag{6}$$

to replace $\phi(\cdot)$ in (4) to have

$$\min_{\mathbf{W}} \frac{\mu||\mathbf{D} \circ (\mathbf{X} - \mathbf{X}\mathbf{W})||_F^2}{\mu + ||\mathbf{D} \circ (\mathbf{X} - \mathbf{X}\mathbf{W})||_F^2} + \lambda||\mathbf{W}||_{2,1} \tag{7}$$

where μ is used to tune the number of easy samples in each iteration [39]. Equation (7) can be optimized by any gradient methods. By the consideration of efficient and scalable optimization, we apply Lemma 1 to (7) and then obtain our final objective function

$$\min_{\mathbf{W},\mathbf{v}} \sum_{i=1}^{n}(v_i||\mathbf{d}^i \circ (\mathbf{x}^i - \mathbf{x}^i\mathbf{W})||_2^2 + \mu(\sqrt{v_i} - 1)^2) + \lambda||\mathbf{W}||_{2,1} \tag{8}$$

where $\psi(\mathbf{v}) = \mu(\sqrt{\mathbf{v}} - 1)^2$, $\mathbf{v} = [v_1, \ldots, v_n] \in \mathbb{R}^n$ is an auxiliary variable and μ and λ are two nonnegative tuning parameters. According to Lemma 1, (7) and (8) are equivalent with respect to the optimization of $\mathbf{W}$ and are more flexible than either (2) or (1) since $\mathbf{v}$ controls the number of easy samples as well as enables to define additional constraints to make the model [see (8)] more flexible. Specifically, by considering the optimization result of v_i in (10), if the residual (i.e., $\|\mathbf{d}^i \circ (\mathbf{x}^i - \mathbf{x}^i\mathbf{W})\|_2^2$) is small, then the ith sample $\mathbf{x}^i$ can be regarded an easy sample and, thus, its weight v_i will be large. By contrast, the weight of an outlier is small due to its large residual. Moreover, in alternative iterations of our proposed Algorithm 1 to solve (8), the value of every v_i will vary with the iteratively updated $\mathbf{W}$. In this way, the initial model constructed in the first iteration will be updated gradually with different weights for each sample until the model achieves stability. Furthermore, the number of easy samples in every iteration can be flexibly controlled by the tuning parameter μ according to the practical demand [39], i.e., $\mu \gg \tilde{f}$ where $\tilde{f}$ is the average of $\{f^i = ||\mathbf{d}^i \circ (\mathbf{x}^i - \mathbf{x}^i\mathbf{W})||_2^2, i = 1, \ldots, n\}$.

D. Optimization

The optimization of (8) is challenging due to the introduction of an auxiliary variable $\mathbf{v}$, the convex but no-smooth constraint on $\mathbf{W}$ (i.e., $||\mathbf{W}||_{2,1}$), and the indicator matrix $\mathbf{D}$. In this article, we propose an alternative optimization strategy based on the framework of iteratively reweighted least squares (IRLS) [40] to optimize (8): 1) update $\mathbf{v}$ by fixing $\mathbf{W}$ and 2) update $\mathbf{W}$ by fixing $\mathbf{v}$. We list the pseudo in Algorithm 1.

1) Update $\mathbf{v}$ *by Fixing* $\mathbf{W}$*:* While $\mathbf{W}$ is fixed, (8) can be changed to

$$\min_{\mathbf{v}} \sum_{i=1}^{n}\left(v_i\left|\left|\mathbf{d}^i \circ \mathbf{x}^i - \mathbf{d}^i \circ \left(\mathbf{x}^i\mathbf{W}\right)\right|\right|_2^2 + \mu(\sqrt{v_i} - 1)^2\right). \tag{9}$$

Algorithm 1 Proposed Algorithm for Optimizing (8)

Input: $\mathbf{X} \in \mathbb{R}^{n\times d}$ and λ;
Output: $\mathbf{W} \in \mathbb{R}^{d\times d}$, $\mathbf{v} \in \mathbb{R}^{n\times 1}$;
1. Initialize the binary matrix $\mathbf{D}$;
2. Initialize $\mathbf{W}$ by conducting ridge regression on observed samples;
3. Initialize $\mathbf{Q} = \mathbf{I} \in \mathbb{R}^{d\times d}$;
4. Initialize $\tilde{f}$ as $\tilde{f} = \sum_{i=1}^{n} f^i$;
5. Initialize μ as $\mu = 1$;
6. **repeat:**
   6.1. Update $\mathbf{v}$ via Eq. (10);
   6.2. Update $\mathbf{Q}$ via Eq. (13);
   6.3. Update $\mathbf{W}$ via Eq. (16);
   6.4. Update μ via $\mu = max(\frac{\mu}{2}, \frac{\tilde{f}}{2})$;
until Eq. (8) converges

According to either [39] or directly conducting the derivative with respect to v_i, the closed-form solution of v_i $(i = 1, \ldots, n)$ is

$$v_i = \left(\frac{\mu}{\mu + ||\mathbf{d}^i \circ (\mathbf{x}^i - \mathbf{x}^i\mathbf{W})||_2^2}\right)^2. \tag{10}$$

2) Update $\mathbf{W}$ *by Fixing* $\mathbf{v}$*:* While $\mathbf{v}$ is fixed, (8) becomes

$$\min_{\mathbf{W}} \sum_{i=1}^{n} v_i ||\mathbf{d}^i \circ \mathbf{x}^i - \mathbf{d}^i \circ (\mathbf{x}^i\mathbf{W})||_2^2 + \lambda||\mathbf{W}||_{2,1}. \tag{11}$$

By defining $\tilde{\mathbf{X}} = \text{diag}(\sqrt{\mathbf{v}})\mathbf{X}$ and $\mathbf{B} = \mathbf{D} \circ \tilde{\mathbf{X}}$ where diag($\mathbf{v}$) transfers the vector $\mathbf{v}$ to a diagonal matrix whose ith diagonal element is the value of v_i, (11) is thus changed to its matrix form as follows:

$$\min_{\mathbf{W}} ||\mathbf{B} - \mathbf{D} \circ (\tilde{\mathbf{X}}\mathbf{W})||_F^2 + \lambda tr(\mathbf{W}^T\mathbf{Q}\mathbf{W}) \tag{12}$$

where $\mathbf{Q}$ is a diagonal matrix whose diagonal element is defined as

$$q_{i,i} = \frac{1}{2||\mathbf{w}^i||_2}, \quad i = 1, \ldots, d. \tag{13}$$

According to the IRLS framework, (12) can be rewritten as

$$\min_{\mathbf{W}} tr(\mathbf{B}^T\mathbf{B} - 2\mathbf{D}^T \circ (\tilde{\mathbf{X}}\mathbf{W})^T\mathbf{B}) + tr(\mathbf{D}^T \circ (\tilde{\mathbf{X}}\mathbf{W})^T((\tilde{\mathbf{X}}\mathbf{W}) \circ \mathbf{D})) + \lambda tr(\mathbf{W}^T\mathbf{Q}\mathbf{W}). \tag{14}$$

The derivative of (14) with respect to $\mathbf{W}$ is

$$-2\mathbf{X}^T(\mathbf{D} \circ \mathbf{B}) + 2(\tilde{\mathbf{X}} \circ \mathbf{D})^T\tilde{\mathbf{X}}\mathbf{W} + 2\lambda\mathbf{Q}\mathbf{W} = 0. \tag{15}$$

Hence, the solution of $\mathbf{W}$ is

$$\mathbf{W} = (\mathbf{B}^T\tilde{\mathbf{X}} + \lambda\mathbf{Q})^{-1}\tilde{\mathbf{X}}^T\mathbf{B}. \tag{16}$$

E. Convergence Analysis

By denoting $\mathbf{v}^{(t)}$ and $\mathbf{W}^{(t)}$, respectively, as the tth iteration result of $\mathbf{v}$ and $\mathbf{W}$, according to Algorithm 1, (8) can be changed to

$$J(\mathbf{W}^{(t)}, \mathbf{v}^{(t)}) = \sum_{i=1}^{n}(v_i^{(t)}||\mathbf{d}^i \circ (\mathbf{x}^i - \mathbf{x}^i\mathbf{W}^{(t)})||_2^2 + \mu\left(\sqrt{v_i^{(t)}} - 1)^2\right) + \lambda||\mathbf{W}^{(t)}||_{2,1}. \tag{17}$$

To prove the convergence of the proposed optimization Algorithm 1, we first list the following lemma and theorem based on [10] and [22].

Lemma 2: For any positive real numbers p and q, the following inequality always holds:

$$\sqrt{p} - \frac{p}{2\sqrt{q}} \leq \sqrt{q} - \frac{q}{2\sqrt{q}}. \tag{18}$$

Theorem 1: The objective function value of (8) monotonically decreases until Algorithm 1 converges.

Proof: With the fixed $\mathbf{W}^{(t)}$ and according to the half-quadratic minimization theory [32], we have

$$\begin{aligned}
&\sum_{i=1}^{n}\left(v_i^{(t+1)}||\mathbf{d}^i \circ (\mathbf{x}^i - \mathbf{x}^i\mathbf{W}^{(t)})||_2^2 + \mu\left(\sqrt{v_i^{(t+1)}} - 1\right)^2\right) + \lambda||\mathbf{W}^{(t)}||_{2,1} \\
&\leq \sum_{i=1}^{n}\left(v_i^{(t)}||\mathbf{d}^i \circ \left(\mathbf{x}^i - \mathbf{x}^i\mathbf{W}^{(t)}\right)||_2^2 + \mu\left(\sqrt{v_i^{(t)}} - 1\right)^2\right) + \lambda||\mathbf{W}^{(t)}||_{2,1}.
\end{aligned} \tag{19}$$

While fixing $\mathbf{v}^{(t+1)}$, we have the following inequality according to the IRLS framework and (13):

$$\begin{aligned}
&\sum_{i=1}^{n}\left(v_i^{(t+1)}||\mathbf{d}^i \circ (\mathbf{x}^i - \mathbf{x}^i\mathbf{W}^{(t+1)})||_2^2 + \mu\left(\sqrt{v_i^{(t+1)}} - 1\right)^2\right) + \lambda\sum_{j=1}^{d}\frac{||\mathbf{w}^{j^{(t+1)}}||_2^2}{2||\mathbf{w}^{j^{(t)}}||_2} \\
&\leq \sum_{i=1}^{n}\left(v_i^{(t+1)}||\mathbf{d}^i \circ \left(\mathbf{x}^i - \mathbf{x}^i\mathbf{W}^{(t)}\right)||_2^2 + \mu\left(\sqrt{v_i^{(t+1)}} - 1\right)^2\right) + \lambda\sum_{j=1}^{d}\frac{||\mathbf{w}^{j^{(t)}}||_2^2}{2||\mathbf{w}^{j^{(t)}}||_2}
\end{aligned} \tag{20}$$

where $\mathbf{w}^{j^{(t)}}$ and $\mathbf{w}^{j^{(t+1)}}$, respectively, are the jth row of $\mathbf{W}^{(t)}$ and $\mathbf{W}^{(t+1)}$. According to Lemma 2, we have

$$||\mathbf{w}^{j^{(t+1)}}||_2 - \frac{||\mathbf{w}^{j^{(t+1)}}||_2^2}{2||\mathbf{w}^{j^{(t)}}||_2} \leq ||\mathbf{w}^{j^{(t)}}||_2 - \frac{||\mathbf{w}^{j^{(t)}}||_2^2}{2||\mathbf{w}^{j^{(t)}}||_2}. \tag{21}$$

TABLE I

SUMMARIZATION OF USED DATA SETS, WHERE 0.26% (FEATURES) INDICATES THAT THE RATIO OF INCOMPLETE FEATURES IS 0.26% OF ALL FEATURES AND 28.60% (SAMPLES) INDICATES THAT THE RATIO OF INCOMPLETE SAMPLES IS 28.60% OF ALL SAMPLES

Name	# (Samples)	# (Features)	# (Classes)	# (Incomplete feature/sample ratio)	# (Data types)
Advertisement	3279	1558	2	0.26% (features), 28.60% (samples)	Image data
Arrhythmia	452	279	13	1.79% (features), 84.96% (samples)	Medical data
Cvpu	2215	145	9	28.28% (features), 94.99% (samples)	Image data
Mice	1567	590	2	91.19% (features), 48.89% (samples)	Gene data
Cane	1080	856	9	Complete data set	Text data
Uspst	2007	256	10	Complete data set	Digit data
Yale	165	1024	15	Complete data set	Face image data
Yeast	1484	1470	10	Complete data set	Gene data
Palm	2000	256	100	Complete data set	Text data
Cifar	60000	3072	10	Complete data set	Images data
Connect	67557	126	3	Complete data set	Game data
Mnist	70000	784	10	Complete data set	Images data
Vehicle	78823	100	3	Complete data set	Sensor data
Aloi	108000	128	1000	Complete data set	Image data

By combining (20) with (21), we have

$$\begin{aligned}&\sum_{i=1}^{n}\left(v_i^{(t+1)}||\mathbf{d}^i\circ\left(\mathbf{x}^i-\mathbf{x}^i\mathbf{W}^{(t+1)}\right)||_2^2\right.\\&\quad\left.+\mu\left(\sqrt{v_i^{(t+1)}}-1\right)^2\right)+\lambda\sum_{j=1}^{d}||\mathbf{w}^{j^{(t+1)}}||_2\\&\le\sum_{i=1}^{n}\left(v_i^{(t+1)}||\mathbf{d}^i\circ\left(\mathbf{x}^i-\mathbf{x}^i\mathbf{W}^{(t)}\right)||_2^2\right.\\&\quad\left.+\mu\left(\sqrt{v_i^{(t+1)}}-1\right)^2\right)+\lambda\sum_{j=1}^{d}||\mathbf{w}^{j^{(t)}}||_2.\end{aligned}\tag{22}$$

Thus, we obtain final inequality by integrating (19) with (22) to have

$$\begin{aligned}&\sum_{i=1}^{n}\left(v_i^{(t+1)}||\mathbf{d}^i\circ\left(\mathbf{x}^i-\mathbf{x}^i\mathbf{W}^{(t+1)}\right)||_2^2\right.\\&\quad\left.+\mu\left(\sqrt{v_i^{(t+1)}}-1\right)^2\right)+\lambda||\mathbf{W}^{(t+1)}||_2\\&\le\sum_{i=1}^{n}\left(v_i^{(t)}||\mathbf{d}^i\circ\left(\mathbf{x}^i-\mathbf{x}^i\mathbf{W}^{(t)}\right)||_2^2\right.\\&\quad\left.+\mu\left(\sqrt{v_i^{(t)}}-1\right)^2\right)+\lambda||\mathbf{W}^{(t)}||_2.\end{aligned}\tag{23}$$

According to (23), (17) is nonincreasing at each iteration in Algorithm 1. Hence, Algorithm 1 converges. □

F. Complexity Analysis

The complexity of our proposed Algorithm 1 is focused on the variables of $\mathbf{v}$ and $\mathbf{W}$, which have closed-form solutions. Specifically, the optimization of the variables $\mathbf{v}$ is with linear complexity [i.e., $O(n)$] to the sample size n according to the literature [32], and the complexity of $\mathbf{W}$ is $O(r^3)$, where r denotes the number of the rank of the feature matrix, usually $r < d$, where d is the number of features. Moreover, our experimental results showed that our Algorithm 1 usually achieved convergence within 20 iterations, so the complexity of our proposed optimization Algorithm 1 is $\max\{O(tr^3), O(tn)\}$, where t is the iteration time. The space complexity of our method is linear to the sample size.

In our experiments, we found that our proposed Algorithm 1 is insensitive to the initialization of variables and our method achieved fast convergence with the initialization settings in Algorithm 1.

IV. EXPERIMENTS

In this section, we evaluated our proposed method by comparing with four previous feature selection methods on four real incomplete data sets and ten synthetic data sets in terms of their clustering performance on the low-dimensional data.

A. Data Sets

In our experiments, we used two kinds of data sets to evaluate the effectiveness of our proposed method, i.e., incomplete data sets and complete data sets. Specifically, we randomly missed different ratios of information to generate different data sets to demonstrate the robustness of our method, while we used real incomplete data sets to analyze the generalization ability of our proposed method.

The complete data sets (i.e., Cane, Uspst, Yale, Yeast, Palm, Cifar, Connect, Mnist, Vehicle, and Aloi) and incomplete data sets (i.e., Advertisement, Arrhthmia, Cvpu, and Mice) were downloaded from the LIBSVM data website[2] and the public UCI website[3]. The used data sets had different scales and have been applied in different domains. The diversity of used data sets makes our experimental results convincible. We listed the details of used data sets in Table I.

In this article, we denoted the incomplete feature ratio as the percentage of incomplete features out of all features and the incomplete sample ratio as the percentage of incomplete samples out of all samples. The incomplete sample ratios on incomplete data sets, i.e., Advertisement, Arrhythmia, Cvpu, and Mice, are 28.60%, 84.96%, 94.99%, and 48.89%, respectively, whereas the incomplete feature ratios on these data sets are 0.26%, 1.79%, 28.28%, and 91.19%, respectively. For complete data sets, we randomly marked observed information as unobserved information by setting the incomplete sample

[2]https://www.csie.ntu.edu.tw/cjlin/libsvmtools/datasets/

[3]https://archive.ics.uci.edu/ml/datasets.php

ratio as 0% (i.e., complete data set), 10%, 30%, 50%, 70%, and 90% and then analyzed the results of feature selection by removing 50% of all features as such a range of features makes all methods achieve good results in our experiments. It is noteworthy that we did not report the results of all methods with different incomplete feature ratios as such results are similar to the ones with different incomplete sample ratios.

B. Comparison Methods

We listed the details of the comparison methods as follows.

1) Laplacian score (LaPscore) [41] is a filter model to evaluate the importance of every feature based on the Laplacian score of every feature.
2) RSR [24] first uses the self-representation of the feature level to reconstruct each feature by all features and then uses an $\ell_{2,1}$-norm regularization to conduct feature selection.
3) General framework for sparsity regularized (GSR) [25] is a general sparse embedded model to simultaneously conduct feature selection and outliers reduce by tuning parameters.
4) RFS, i.e., (2), is a classic embedded model of feature selection. RFS is used to verify the effectiveness of reducing outlier influence compared with our proposed method.

In our experiments, we separated every incomplete data set into two subsets, i.e., incomplete set (IS) including all incomplete samples and observed set (OS) including all observed samples. First, we denoted the method of conducting k-means clustering with original features of OS as Baseline. We also employed a filter model (i.e., LaPscore) and three embedded feature selection methods, i.e., RSR, GSR, and RFS, to conduct feature selection on OS, and then conducted k-means clustering on OS with selected features. Second, we used the information in OS to impute unobserved values in IS by the imputation method, i.e., mean-value imputation method (Mean) and kNN imputation method (kNN) [12], [15], and then conducted feature selection on the combination of OS and IS (i.e., OS $\bigcup$ IS) by GSR, i.e., GSR_mean and GSR_knn, followed by conducting k-means clustering on OS with the selected features. Third, we used our proposed (8) to conduct feature selection on OS $\bigcup$ IS and then conducted k-means clustering on OS with the selected features.

C. Experimental Setting

The codes of all methods in our experiments were implemented with MATLAB, and all experiments were run under the environment of a Windows server with 128-GB RAM and Intel Core i9-7900x CPU.

We employed the tenfold cross-validation scheme to repeat every method on every data set ten times. We reported the average results of these ten times, each of which is the average of ten results of k-means clustering.[4] We set the ranges of the parameters of the comparison methods according to the corresponding literature to guarantee that all comparison methods achieved their best results and set the ranges of the parameter (i.e., λ) of our method in (8) as $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$. We further set the number of clusters in k-means clustering as the number of real classes of the data sets.

[4]Since k-means clustering is sensitive to the initialization, we need to repeat the clustering process many times to report the average result for avoiding the randomness.

D. Evaluation Metrics

We used two evaluation metrics [i.e., accuracy (ACC) and normalized mutual information (NMI)] to evaluate the clustering performance of all methods. ACC indicates the percentage of correctly classified samples, that is

$$\mathrm{ACC} = \frac{N_c}{N} \tag{24}$$

where N denotes the number of the samples and N_c is the number of the correctly classified samples.

NMI uncovers a correlation between the predictions and the ground truths, that is

$$\mathrm{NMI} = \frac{2I(\mathbf{X}, \mathbf{Y})}{H(\mathbf{X}) + H(\mathbf{Y})} \tag{25}$$

where $I(\mathbf{X}, \mathbf{Y})$ denotes mutual information between the predicted labels and the real labels, i.e., $I(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^{|\mathbf{X}|} \sum_{j=1}^{|\mathbf{Y}|} P(i, j) \log((P(i, j)/P(i)P'(j)))$, where $P(i, j) = (|\mathbf{X}_i \cap \mathbf{Y}_j|/N)$, and $|\mathbf{X}|$ and $|\mathbf{Y}|$ imply the cardinality of $\mathbf{X}$ and $\mathbf{Y}$, respectively. $H(\cdot)$ is the entropy, i.e., $H(\cdot) = -\sum_{i=1}^{|\cdot|} P(i) \log(P(i))$.

Besides, we conducted the paired-sample t-tests (at 95% significance level) between our method and every comparison method, in terms of ACC and NMI. Moreover, the symbols "*" and "**," respectively, indicate that our method has statistically significant difference with $p < 0.05$ and $p < 0.001$ on the paired-sample t-tests at 95% significance level compared with the comparison method.

E. Real Incomplete Data Sets

We reported clustering performance of all methods with different feature ratios (i.e., $\{10\%, 20\%, \ldots, 90\%\}$ of all the features) on four real incomplete data sets, i.e., Advertisement, Arrhythmia, Cvpu, and Mice, in Figs. 1 and 2.

1) Our method achieved the best clustering performance, followed by GSR_knn, GSR_mean, RFS, GSR, RSR, LapScore, and Baseline. Moreover, the results of our method are statistically significant based on the statistical analysis. For example, our proposed method on average improved by 2.15% and 0.48%, compared with the best comparison method (i.e., GSR_knn), as well as on average improved by 18.70% and 14.42%, compared with the worst comparison method (i.e., Baseline), in terms of ACC and NMI results. The reason may be that our method uses all the available information as well as considers to flexibly reduce the influence of outliers. It is noteworthy that imputation methods (i.e., GSR_mean and GSR_knn) used all the available information in complete samples, but they do not take the influence of outliers into account, whereas feature

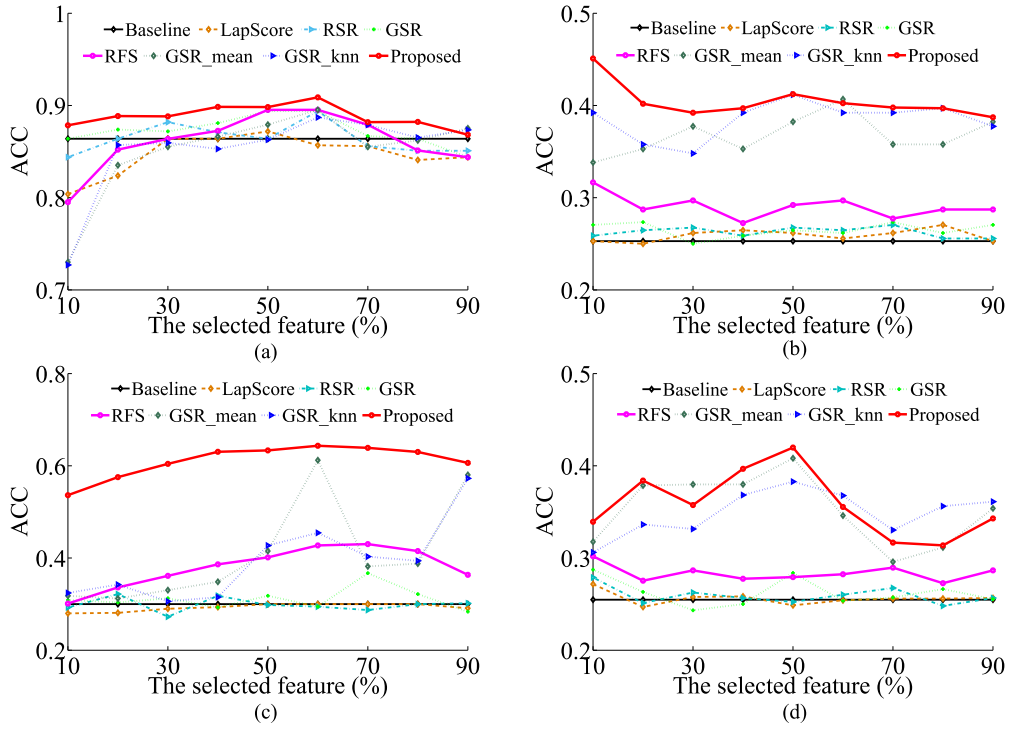

Fig. 1. ACC results of all methods with different features on different real incomplete data sets. (a) Advertisement. (b) Arrhythmia. (c) Cvpu. (d) Mice.

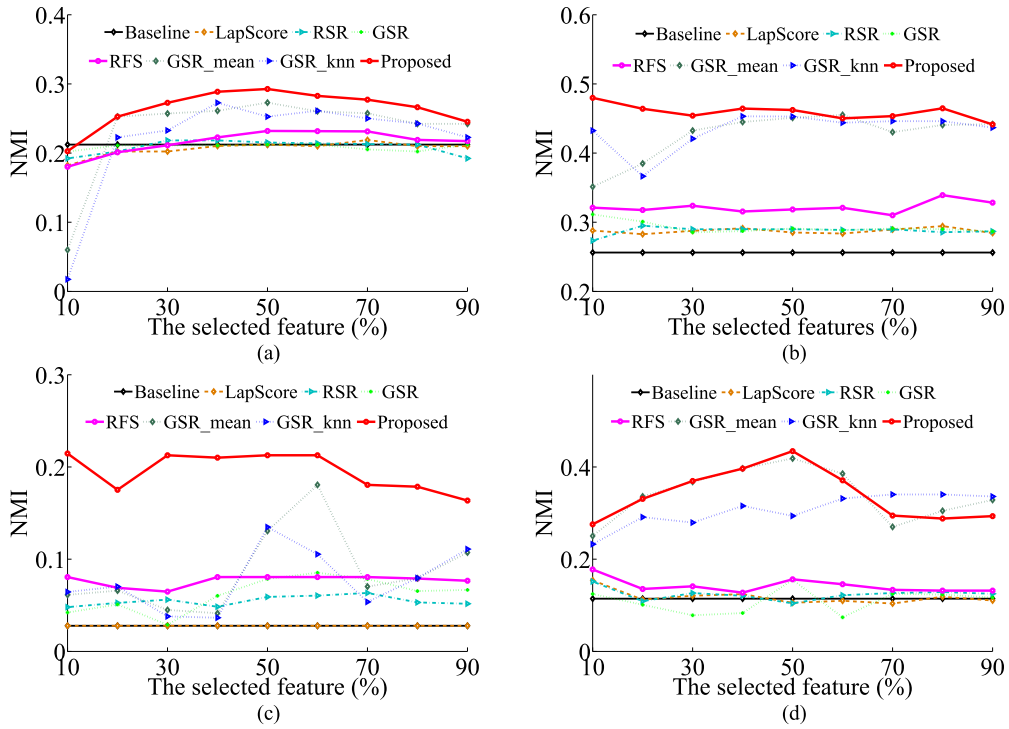

Fig. 2. NMI results of all methods with different features on different real incomplete data sets. (a) Advertisement. (b) Arrhythmia. (c) Cvpu. (d) Mice.

selection methods (i.e., LapScore, RSR, GSR, and RFS) only used the information in observed samples and do not use the observed information in incomplete samples.

2) Our method achieved the maximal improvement compared with four comparison methods on data set Cvpu because this data set only has 5.01% complete samples, i.e., the minimal ratio of incomplete samples on all data sets. This indicates that reliable feature selection models need enough training samples.
3) In all, our proposed method achieved the best performance for two evaluation metrics on the data sets

TABLE II

CLUSTERING RESULTS OF SYNTHETIC INCOMPLETE DATA SETS WITH DIFFERENT INCOMPLETE SAMPLE RATIOS WHILE SELECTING 50% OF ALL FEATURES FOR FEATURE SELECTION METHODS. THE BOLD NUMBERS INDICATE THE BEST RESULTS THROUGH THE WHOLE ROW FOR THE SAME EVALUATION METRIC AND THE UNDERLINED BOLD NUMBERS IMPLY THE BEST RESULT IN ONE BLOCK FOR THE SAME EVALUATION METRIC. THE SYMBOLS "*" AND "**," RESPECTIVELY, INDICATE THE STATISTICALLY SIGNIFICANT DIFFERENCE WITH $p < 0.05$ AND $p < 0.001$, ON THE PAIRED-SAMPLE t-TESTS AT 95% SIGNIFICANCE LEVEL OF THE RESULTS BETWEEN OUR METHOD AND EVERY COMPARISON METHOD

Data set	Ratio	Baseline		LapScore		RSR		GSR		RFS		GSR_mean		GSR_knn		Proposed	
		ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
Cane	0	48.2** ±2.6	41.4** ±2.6	49.9** ±2.6	42.3** ±2.2	48.5** ±3.0	41.3** ±2.5	48.8** ±3.1	42.0** ±2.4	54.5* ±2.1	46.9** ±1.7	46.2** ±2.6	40.2** ±2.1	49.3** ±2.5	41.5** ±2.2	**55.8** ±3.1	**48.9** ±2.9
	0.1	46.4** ±3.0	37.6** ±2.3	49.0** ±3.0	40.2** ±2.9	46.2** ±2.2	41.3** ±1.9	50.6** ±2.2	43.6** ±2.1	55.2** ±2.0	47.1* ±2.9	49.1** ±1.4	41.3** ±2.2	50.7** ±2.1	40.8** ±2.1	**58.1** ±1.9	**48.2** ±2.6
	0.3	45.3** ±3.0	37.4** ±2.7	45.7** ±1.9	39.8** ±1.9	53.2* ±2.2	47.3** ±2.4	52.5** ±1.9	45.6** ±2.1	60.2* ±3.2	52.8 ±2.9	48.8** ±3.7	42.9** ±2.8	52.9** ±2.9	44.6** ±1.5	**61.1** ±3.5	**53.3** ±2.6
	0.5	37.1** ±1.7	30.2** ±1.7	51.7** ±3.7	44.6** ±3.7	48.9** ±2.6	41.3** ±2.1	49.0** ±3.8	42.1** ±3.5	55.0** ±3.3	47.0 ±2.5	42.6** ±3.8	35.2** ±0.3	45.5** ±3.7	37.0** ±3.2	**56.9** ±3.5	**47.9** ±3.3
	0.7	42.9** ±3.6	36.6** ±3.3	49.6** ±2.8	42.7** ±2.4	41.7** ±3.1	36.0** ±2.9	44.1** ±3.3	39.7** ±3.6	**54.5** ±3.1	**47.8*** ±3.0	43.3** ±0.8	33.1** ±0.2	46.2** ±1.8	37.8** ±0.8	54.3 ±4.2	46.8 ±3.9
	0.9	42.1** ±2.1	39.1** ±2.1	45.3** ±2.0	44.6** ±2.0	48.7** ±2.0	45.2** ±2.4	44.9** ±2.9	41.9** ±2.6	53.4 ±3.2	**51.8**** ±3.0	45.1** ±1.1	37.0** ±2.8	45.8** ±3.0	37.4** ±2.9	**54.6** ±2.2	48.9 ±2.1
Uspst	0	64.4** ±1.2	59.5** ±0.7	59.7** ±1.7	57.2** ±1.0	62.4** ±2.3	58.2** ±0.9	64.3** ±1.9	58.6** ±0.9	64.2** ±1.1	58.2** ±0.7	62.4** ±2.3	58.2** ±0.9	64.3** ±1.9	58.6** ±0.9	**67.7** ±2.2	**60.9** ±0.8
	0.1	60.9** ±1.9	58.1** ±1.0	59.2** ±1.9	56.9** ±1.0	63.5** ±1.4	59.2* ±0.8	64.6** ±1.8	59.1 ±1.3	64.6** ±1.6	58.7 ±0.7	65.1** ±2.2	58.3** ±0.5	**67.5*** ±3.4	59.3 ±2.1	67.0 ±0.8	**59.4** ±0.7
	0.3	62.0** ±1.9	57.9** ±0.9	58.6** ±2.1	56.7 ±0.9	63.7** ±14	57.6** ±0.9	61.8** ±1.9	55.5** ±1.2	65.5** ±1.1	57.6** ±0.8	63.9** ±3.1	**61.3**** ±0.7	63.0** ±2.2	58.2** ±1.6	**67.9** ±0.7	56.7 ±0.2
	0.5	61.0** ±1.5	59.9** ±0.7	59.0** ±2.2	58.7** ±1.1	63.4** ±2.7	**62.0*** ±1.0	56.5** ±1.3	53.6** ±0.8	65.5* ±1.8	61.2** ±0.5	61.4** ±2.1	57.8** ±2.0	57.6** ±1.3	54.7** ±1.0	**65.9** ±1.8	61.3 ±0.6
	0.7	61.3** ±1.8	58.9** ±0.7	59.5** ±1.0	58.7** ±0.7	61.5** ±1.9	56.4** ±1.0	59.4** ±2.1	56.6 ±1.5	67.8 ±1.1	60.0** ±0.6	65.4* ±2.5	58.8** ±1.2	63.8** ±3.2	57.6** ±1.7	**68.1** ±2.3	**61.0** ±1.0
	0.9	61.1** ±2.0	59.4** ±0.9	58.0** ±2.3	59.2** ±1.2	56.5** ±1.5	57.5** ±1.1	58.6** ±2.2	59.3 ±1.4	62.0** ±1.3	61.9 ±0.5	61.6** ±1.9	57.6** ±1.4	64.3** ±2.8	59.4** ±2.6	**66.7** ±1.5	**61.9** ±0.7
Yale	0	48.4** ±2.1	57.8* ±1.9	53.6 ±2.4	57.7* ±2.0	54.4 ±2.3	**61.0**** ±2.7	46.3** ±1.6	55.2** ±1.4	46.9** ±2.1	54.4** ±1.7	51.4** ±2.3	59.0* ±1.3	46.3** ±1.6	55.2** ±1.6	**53.6** ±4.0	58.6 ±3.0
	0.1	50.4** ±1.7	61.9** ±1.8	51.0** ±1.9	67.0* ±1.7	**55.7**** ±1.2	**71.7**** ±1.8	55.1** ±1.3	68.8** ±1.3	41.8** ±1.6	48.0** ±1.6	48.7** ±0.5	57.9** ±1.5	54.0 ±2.0	59.5** ±1.9	53.3 ±1.9	65.6 ±1.3
	0.3	47.0** ±1.8	58.0** ±1.7	60.8** ±23	75.8** ±2.0	68.7** ±1.5	85.8** ±2.6	72.9** ±1.4	85.8** ±1.6	77.0** ±1.3	86.9 ±1.4	67.7** ±4.0	72.4** ±3.1	70.8** ±1.1	74.4** ±1.3	**79.1** ±1.7	**88.0** ±1.5
	0.5	46.3** ±1.9	57.5** ±2.1	48.3** ±1.5	60.5** ±1.4	48.3** ±2.7	60.8** ±3.3	47.9** ±2.2	61.1** ±1.5	50.8* ±1.4	60.2** ±1.5	48.7** ±1.5	55.4** ±1.9	47.7** ±2.5	54.6** ±2.0	**51.6** ±1.6	**62.2** ±1.7
	0.7	40.1** ±2.3	50.6* ±2.1	44.1* ±1.7	51.0** ±1.7	48.1** ±3.0	58.3** ±2.7	49.7** ±1.5	58.8** ±1.5	**56.4**** ±1.8	**69.0**** ±1.8	40.8* ±4.6	47.6** ±4.6	44.2** ±2.8	48.9* ±3.7	47.4 ±1.6	49.6 ±1.3
	0.9	41.5* ±2.0	51.9** ±2.2	44.3* ±1.6	**53.7**** ±1.8	43.7* ±2.5	53.0** ±3.0	43.7* ±1.7	49.5** ±1.0	44.6 ±1.5	51.2** ±1.1	38.3** ±2.1	44.8** ±0.6	**47.2**** ±1.5	49.5 ±1.9	45.1 ±1.4	50.2 ±1.0
Yeast	0	40.8** ±1.3	**30.3**** ±3.6	37.0** ±0.3	18.7* ±0.6	41.7** ±2.8	23.2** ±2.7	43.4* ±2.4	11.9** ±2.9	**44.8**** ±4.7	30.3** ±3.6	40.0** ±3.2	23.4** ±0.4	41.3** ±1.6	14.6** ±2.3	42.8 ±0.7	17.4 ±1.7
	0.1	38.6** ±1.6	27.5** ±4.0	37.8** ±0.6	29.9** ±1.1	42.3 ±2.1	**32.7**** ±2.1	43.2** ±2.3	31.8** ±1.3	**45.0**** ±3.1	32.4** ±1.6	38.1** ±2.4	24.6** ±0.5	40.9** ±0.1	16.1** ±3.1	42.0 ±1.0	19.7 ±1.6
	0.3	40.5** ±1.6	26.5** ±3.4	37.6** ±0.5	23.3** ±1.4	40.3** ±2.1	28.0** ±1.9	42.9** ±2.4	27.1** ±2.8	43.1** ±2.5	34.0** ±1.1	34.0** ±1.2	15.7** ±0.1	36.7** ±1.2	12.8** ±1.1	**45.7** ±0.7	**41.2** ±1.0
	0.5	35.9** ±1.5	14.8 ±3.2	31.0** ±0.7	14.1 ±1.0	37.0** ±2.1	25.7** ±1.7	**42.2**** ±2.5	24.3** ±1.3	38.2 ±2.4	**30.9**** ±1.5	27.9** ±1.2	12.9** ±0.1	31.0** ±0.6	5.9** ±0.1	38.4 ±0.4	15.1 ±1.5
	0.7	36.1** ±1.0	13.4** ±4.0	30.6** ±0.5	18.0** ±0.7	32.1** ±2.1	16.0** ±1.7	33.1** ±2.4	10.8** ±1.5	**42.8**** ±2.4	**40.7**** ±1.3	38.5* ±1.3	26.9** ±0.2	35.2** ±0.9	11.6** ±0.3	38.9 ±0.5	24.4 ±0.6
	0.9	36.5** ±1.8	15.5* ±2.5	31.6** ±0.3	15.6** ±0.2	32.6** ±2.1	13.2** ±1.9	34.7** ±1.7	**17.3**** ±1.9	38.0** ±3.0	12.3** ±1.0	29.4** ±0.9	8.1** ±0.2	35.0** ±0.8	11.0** ±0.8	**38.6** ±0.6	16.5 ±0.6
Palm	0	70.2** ±0.5	90.1 ±0.3	70.7** ±1.5	90.0** ±0.4	70.9** ±1.0	89.8** ±0.4	70.3** ±1.1	89.6** ±0.4	71.2** ±1.5	89.9* ±0.5	71.2** ±05	**90.3*** ±0.2	73.2 ±0.7	90.0** ±0.3	**73.3** ±0.9	90.2 ±0.3
	0.1	68.8** ±1.2	89.3** ±0.1	68.9** ±1.2	89.9* ±0.5	70.7** ±1.4	89.7** ±0.5	71.3** ±1.0	90.0* ±0.4	72.1** ±1.2	89.9** ±0.4	73.8** ±01	90.4 ±0.1	**74.4** ±0.4	**90.6** ±0.1	74.0 ±1.3	90.3 ±0.5
	0.3	68.7** ±1.6	89.7** ±0.4	71.2** ±0.6	90.4** ±0.2	71.0** ±1.2	89.5** ±0.5	73.1** ±0.9	90.6** ±0.3	70.7** ±1.2	89.5** ±0.4	74.0** ±06	90.4** ±0.2	74.4** ±0.6	90.5* ±0.2	**75.1** ±0.9	**90.8** ±0.3
	0.5	68.5** ±0.6	89.2** ±0.6	72.1** ±1.3	90.3** ±0.4	70.3** ±1.5	89.5** ±0.5	72.8** ±1.2	90.4** ±0.4	72.6** ±1.1	90.5** ±0.5	75.0** ±05	90.9** ±0.1	75.2* ±0.6	91.0** ±0.4	**76.9** ±0.7	**91.3** ±0.3
	0.7	71.5** ±0.3	90.7** ±0.3	73.2** ±0.9	91.2 ±0.3	72.3** ±1.5	91.4 ±0.6	76.4 ±1.0	**92.6**** ±0.4	74.1** ±0.9	91.6 ±0.3	76.5 ±05	91.9** ±0.8	**77.1*** ±0.6	92.2** ±0.3	76.3 ±0.6	91.3 ±0.3
	0.9	71.0** ±0.4	91.1** ±0.4	74.5** ±1.0	92.2** ±0.4	77.0** ±1.7	92.4** ±0.6	75.0** ±0.9	92.3** ±0.4	77.5** ±0.7	93.2** ±0.3	78.7** ±0.8	93.3** ±0.1	77.3** ±0.6	92.8** ±0.2	**79.1** ±0.8	**94.1** ±0.7

with different numbers of features. This indicated the robustness of our proposed method.

F. Synthetic Incomplete Data Sets

We randomly missed the observed information (i.e., randomly marked observed information as unobserved) in ten complete data sets to obtain different incomplete sample ratios, i.e., 0% (i.e., complete data sets), 10%, 30%, 50%, 70%, 90%, and for every data set, and we kept 50% of all the features for all feature selection methods on these incomplete data sets. We listed the clustering results in Tables II and III.

Obviously, our method achieved the best clustering results, in terms of ACC and NMI, at different missing ratios. Moreover, the clustering results of our method statistically significant outperform the ones of all comparison methods.

TABLE III

CLUSTERING RESULTS OF SYNTHETIC INCOMPLETE DATA SETS WITH DIFFERENT INCOMPLETE SAMPLE RATIOS WHILE SELECTING 50% OF ALL FEATURES FOR FEATURE SELECTION METHODS. THE BOLD NUMBERS INDICATE THE BEST RESULTS THROUGH THE WHOLE ROW FOR THE SAME EVALUATION METRIC AND THE UNDERLINED BOLD NUMBERS IMPLY THE BEST RESULT IN ONE BLOCK FOR THE SAME EVALUATION METRIC. THE SYMBOLS "*" AND "**," RESPECTIVELY, INDICATE THE STATISTICALLY SIGNIFICANT DIFFERENCE WITH $p < 0.05$ AND $p < 0.001$, ON THE PAIRED-SAMPLE t-TESTS AT 95% SIGNIFICANCE LEVEL OF THE RESULTS BETWEEN OUR METHOD AND EVERY COMPARISON METHOD

Data set	Ratio	Baseline		LapScore		RSR		GSR		RFS		GSR_mean		GSR_knn		Proposed	
		ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
Cifar	0	21.0** ±0.2	7.5** ±0.1	18.4** ±0.2	4.8** ±0.1	21.2 ±0.4	7.6** ±0.3	19.4** ±0.3	7.0** ±0.1	18.6** ±0.2	4.9** ±0.1	19.5** ±0.1	6.9** ±0.3	19.7** ±0.2	7.4** ±0.1	**21.3** ±0.4	**8.1** ±0.1
	0.1	21.2** ±0.2	7.5** ±0.1	18.3** ±0.3	4.7** ±0.1	20.9** ±0.2	7.2** ±0.2	19.7** ±0.5	6.6** ±0.2	18.6** ±0.2	4.9** ±0.1	20.6** ±0.4	7.1** ±0.2	21.0** ±0.2	7.9** ±0.1	**21.7** ±0.2	**8.4** ±0.2
	0.3	20.6** ±0.1	7.3** ±0.1	17.3** ±0.2	1.7** ±0.1	20.8** ±0.1	7.2 ±0.1	21.3** ±0.1	**7.7** ±0.1	18.6** ±0.2	4.9** ±0.1	21.3** ±0.1	7.5** ±0.1	21.3** ±0.4	7.5** ±0.1	**21.6** ±0.1	**7.7** ±0.2
	0.5	20.2** ±0.2	7.0** ±0.2	17.7** ±0.1	4.5** ±0.1	20.5** ±0.2	6.9** ±0.2	20.9** ±0.3	7.0** ±0.3	17.9** ±0.1	6.0** ±0.3	20.6** ±0.2	7.0** ±0.3	20.8** ±0.2	7.6** ±0.1	**21.3** ±0.1	**7.8** ±0.1
	0.7	19.8** ±0.3	6.2** ±0.1	17.9** ±0.4	4.3** ±0.3	19.6** ±0.2	6.5** ±0.2	20.0** ±0.1	6.4** ±0.1	17.6** ±0.2	5.9** ±0.3	20.2** ±0.3	6.2** ±0.5	20.1** ±0.2	6.6* ±0.4	**20.8** ±0.4	6.7 ±0.3
	0.9	18.9 ±0.7	5.7** ±0.4	17.3** ±0.1	3.6** ±0.2	18.8* ±0.5	5.5** ±0.4	19.4** ±0.7	6.1 ±0.4	17.0** ±0.6	4.6** ±0.5	**20.0**** ±0.6	6.0 ±0.2	**20.0**** ±0.4	6.1 ±0.2	**20.0** ±0.1	**6.2** ±0.3
Connect	0	37.5** ±0.6	10.2** ±0.1	38.5** ±1.3	10.3** ±0.1	38.1** ±0.6	10.2** ±0.1	42.6** ±1.0	**11.7**** ±0.3	40.6** ±1.3	10.4** ±0.1	42.7 ±2.1	11.4** ±0.2	42.3** ±1.4	11.4** ±0.5	**42.8** ±1.6	11.5 ±0.2
	0.1	37.8** ±0.7	10.2** ±0.1	38.8 ±0.9	10.3** ±0.1	38.9** ±0.6	10.2** ±0.1	41.2** ±2.9	**11.5*** ±0.2	40.4** ±1.3	10.5** ±0.1	**42.6**** ±3.1	**11.5** ±0.1	41.5** ±1.7	11.4 ±0.5	41.7 ±0.8	**11.5** ±0.2
	0.3	37.6** ±0.2	10.2** ±0.1	38.8 ±1.1	10.7** ±0.2	35.8* ±0.7	10.2** ±0.1	42.2** ±0.8	11.2 ±0.2	35.6** ±0.5	10.1** ±0.1	**42.5**** ±2.8	11.4** ±0.2	41.8 ±2.9	11.5** ±0.3	41.4 ±1.4	**11.8** ±0.3
	0.5	38.8** ±1.5	10.3** ±0.1	37.8** ±0.6	10.2** ±0.1	36.3** ±0.4	10.2** ±0.1	41.3** ±2.6	**12.2**** ±0.4	35.4* ±0.4	10.1** ±0.1	41.5** ±2.6	11.5* ±0.1	41.3** ±2.5	11.4 ±0.4	**41.7** ±1.1	11.3 ±0.2
	0.7	37.5** ±0.6	10.1** ±0.1	37.0** ±0.1	10.2** ±0.1	35.9* ±0.4	10.1** ±0.1	40.8** ±2.0	11.3 ±0.2	35.9* ±0.3	10.1** ±0.1	41.7** ±2.1	11.6** ±0.4	**41.8**** ±2.3	**11.7**** ±0.1	**41.8** ±1.1	11.3 ±0.2
	0.9	37.5** ±0.4	10.2** ±0.1	37.0 ±0.7	10.2** ±0.1	35.2** ±0.2	10.2** ±0.1	40.5* ±2.0	11.2 ±0.2	35.6** ±0.6	10.1** ±0.1	40.7** ±3.0	11.4* ±0.3	40.7** ±2.8	**11.5**** ±0.1	**40.9** ±0.8	11.2 ±0.1
Mnist	0	55.0** ±1.4	47.7** ±0.6	57.2** ±1.4	**48.6** ±1.0	54.8** ±2.5	42.5** ±3.0	55.0** ±0.7	44.7** ±0.8	53.9** ±2.0	42.4** ±0.7	54.7** ±2.1	43.8** ±1.1	55.0** ±0.6	44.9** ±0.6	**58.8** ±1.4	**48.6** ±0.7
	0.1	55.6** ±1.4	47.9** ±0.6	58.2** ±2.3	48.5 ±1.1	51.7** ±6.5	42.8** ±4.8	56.0** ±15	45.0** ±1.4	52.6** ±1.5	42.3** ±0.4	53.9** ±2.0	43.5** ±1.0	54.2** ±1.2	43.3** ±0.6	**58.8** ±1.3	**48.8** ±0.9
	0.3	54.5** ±1.6	47.4** ±0.6	56.4** ±1.6	47.9* ±0.9	51.1** ±0.2	40.5** ±1.8	54.3** ±1.0	45.5** ±0.7	52.0** ±1.7	41.6** ±0.6	53.8** ±2.0	43.4** ±1.0	52.1** ±0.5	42.6** ±0.7	**58.2** ±1.2	**48.4** ±0.5
	0.5	54.8** ±1.6	47.1* ±0.7	56.0* ±1.7	47.0** ±0.8	52.7** ±2.8	42.0** ±3.2	55.4** ±1.4	44.4** ±0.7	49.6** ±0.9	41.1** ±0.4	51.5** ±1.4	41.1** ±1.8	51.4** ±1.0	40.8** ±1.1	**57.2** ±1.5	**47.9** ±0.7
	0.7	53.3** ±1.4	46.9** ±0.6	54.2** ±1.5	45.7** ±0.7	50.9** ±0.5	40.5** ±2.5	55.8** ±1.5	45.0** ±0.9	49.0** ±1.9	41.6** ±0.6	52.0** ±3.0	40.6** ±2.0	53.2** ±0.6	41.3** ±0.6	**57.2** ±1.3	**48.1** ±0.8
	0.9	50.0** ±2.3	43.9** ±0.8	53.9** ±2.5	45.3 ±1.0	49.1* ±3.1	41.2** ±3.9	54.7** ±2.3	44.5* ±1.1	52.4* ±5.3	**45.5*** ±3.0	51.1** ±0.5	38.6** ±0.6	51.8** ±0.7	38.8** ±0.4	**55.8** ±1.7	44.8 ±0.9
Vehicle	0	54.5** ±1.2	17.1** ±0.5	57.2** ±0.3	15.7** ±0.1	45.6** ±1.6	9.5** ±1.0	55.8** ±2.7	15.0** ±0.9	55.3** ±1.4	15.1** ±1.1	55.9** ±1.7	14.7** ±1.0	57.5* ±0.4	16.1* ±0.2	**57.6** ±1.4	**16.2** ±0.2
	0.1	54.6** ±1.2	16.7** ±0.8	57.1 ±0.8	15.6* ±0.5	46.3** ±1.4	9.5** ±0.9	57.0* ±1.0	15.9 ±0.4	56.0** ±1.4	15.6* ±0.8	56.4** ±0.8	15.8 ±0.5	54.3** ±4.1	**19.7**** ±2.2	**57.1** ±1.2	15.7 ±0.6
	0.3	56.3** ±1.3	15.4** ±0.3	56.5 ±0.1	15.3* ±0.2	55.7** ±0.8	14.9* ±0.7	56.0** ±0.5	15.4 ±0.2	53.5** ±2.5	15.5* ±0.8	53.8** ±2.5	**15.5*** ±0.9	56.0 ±1.5	**15.5** ±1.3	**56.4** ±1.9	**15.5** ±0.9
	0.5	54.7** ±1.6	15.0** ±1.0	54.3** ±2.3	13.5** ±1.3	53.3** ±0.2	14.6 ±0.7	54.0** ±0.7	14.2 ±0.4	54.4** ±1.5	14.9** ±0.5	53.0** ±2.3	14.3 ±1.1	54.6** ±3.2	**16.1**** ±1.6	**56.3** ±0.8	14.3 ±0.5
	0.7	**55.2**** ±1.3	**14.8**** ±0.8	50.2** ±2.9	11.5** ±2.3	53.3* ±1.6	14.4** ±0.7	53.6 ±1.2	13.2** ±0.6	52.1* ±1.4	13.8 ±0.9	49.5** ±2.8	12.0** ±1.3	51.2** ±2.4	14.5** ±2.3	53.5 ±1.3	13.8 ±1.0
	0.9	51.2** ±1.5	12.7* ±0.6	50.9 ±0.5	11.6* ±0.2	50.1** ±1.0	11.9 ±1.0	50.3* ±1.5	11.7 ±0.7	50.6** ±0.7	**13.0**** ±0.2	49.6** ±1.2	10.9** ±0.8	48.5** ±2.3	11.6* ±3.0	**51.5** ±1.8	11.8 ±1.5
Aloi	0	50.4** ±0.1	79.5** ±0.1	48.5** ±0.1	73.1** ±0.1	50.5* ±0.2	80.3** ±0.1	50.3** ±0.2	80.6** ±0.1	50.6* ±0.6	**80.5*** ±0.2	50.1** ±0.5	80.3 ±0.1	**50.7** ±0.4	80.4** ±0.1	**50.7** ±0.5	**80.5** ±0.2
	0.1	50.5** ±0.4	79.6** ±0.1	48.5** ±0.1	73.1** ±0.1	50.0** ±0.4	80.4* ±0.1	50.7* ±0.5	**80.8**** ±0.2	50.5** ±0.2	80.7** ±0.1	50.3** ±0.4	80.5** ±0.1	50.0** ±0.2	80.4 ±0.1	**50.8** ±0.5	80.4 ±0.2
	0.3	48.4** ±0.1	78.0** ±0.1	**48.5*** ±0.1	73.1** ±0.1	48.0** ±0.4	79.2 ±0.1	47.8** ±0.3	79.3 ±0.1	48.2** ±0.3	**79.4*** ±0.1	47.9** ±0.8	79.2 ±0.3	47.6** ±0.2	78.9** ±0.1	**48.5** ±0.3	**79.4** ±0.1
	0.5	**41.9**** ±0.2	74.0** ±0.1	40.1** ±0.1	**76.3**** ±0.1	41.2** ±0.3	75.9** ±0.2	41.1** ±0.1	75.8** ±0.1	41.1** ±0.4	76.0** ±0.1	41.2** ±0.5	75.9** ±0.1	40.7** ±0.5	75.7 ±0.2	41.5 ±0.2	75.6 ±0.1
	0.7	22.5** ±0.2	69.7** ±0.1	32.0** ±0.1	77.5** ±0.1	32.8** ±0.3	73.0** ±0.1	**33.8**** ±0.2	**73.5*** ±0.2	32.8** ±0.1	73.1* ±0.3	33.2** ±0.1	73.2 ±0.1	33.2** ±0.1	73.2 ±0.1	33.6 ±0.1	73.2 ±0.1
	0.9	26.1** ±0.1	70.0** ±0.1	30.4** ±0.5	76.4** ±0.3	33.5** ±0.3	78.7* ±0.1	33.5** ±0.2	78.6* ±0.1	33.2* ±0.4	78.5** ±0.2	32.7** ±0.2	78.3** ±0.1	32.9** ±0.1	78.7 ±0.1	**33.8** ±0.2	**78.8** ±0.2

In particular, the improvement between our method and the comparison methods on big data sets is larger than the corresponding improvement on small data sets in terms of ACC and NMI. Besides, we have the following observations.

First, with the increase of the incomplete sample ratios, the clustering performance of all methods sharply reduced. For example, on data set Yale, all methods reduced on average by 8.01% at incomplete sample ratio 0.1, compared with the results at the incomplete sample ratio 0.9, in terms of ACC results. In particular, on data set Aloi, all methods reduced on average by 18.21% at incomplete sample ratio 0, compared to the results at the incomplete sample ratio 0.9, in terms of ACC results. Furthermore, most methods achieved the best results, while the data sets are with small missing ratios (i.e., ratio <= 0.3). This indicates that the number of training samples is very important for constructing reliable feature

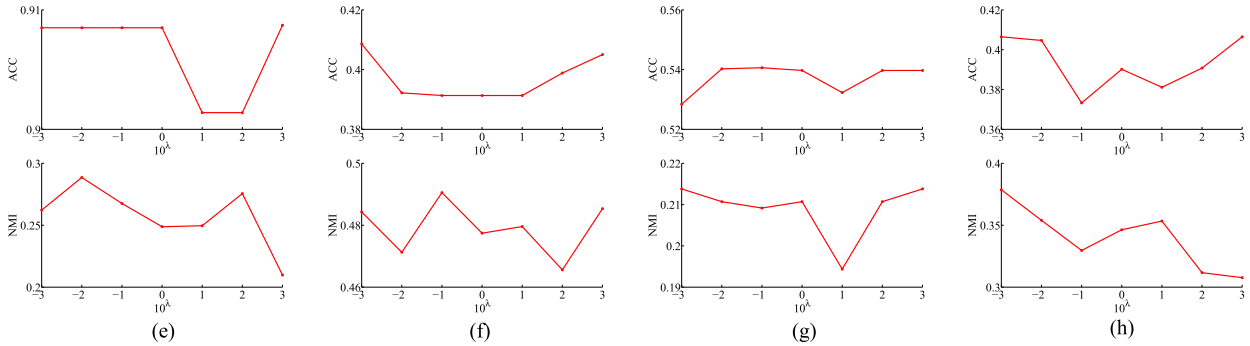

Fig. 3. ACC (top row) and NMI (bottom row) results of our method with varied parameter' setting (i.e., λ) on real incomplete data sets keeping 50% of all features. (a) Advertisement. (b) Arrhythmia. (c) Cvpu. (d) Mice.

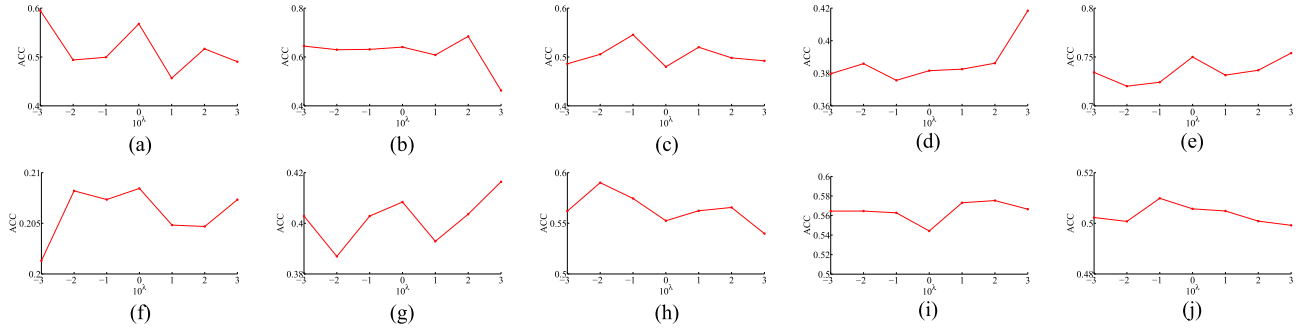

Fig. 4. ACC results of our method with varied parameter' setting (i.e., λ) on synthetic incomplete data sets keeping 50% of all features and 90% of all observed information. (a) Cane. (b) Uspst. (c) Yale. (d) Yeast. (e) Palm. (f) Cifar. (g) Connect. (h) Mnist. (i) Vehicle. (j) Aloi.

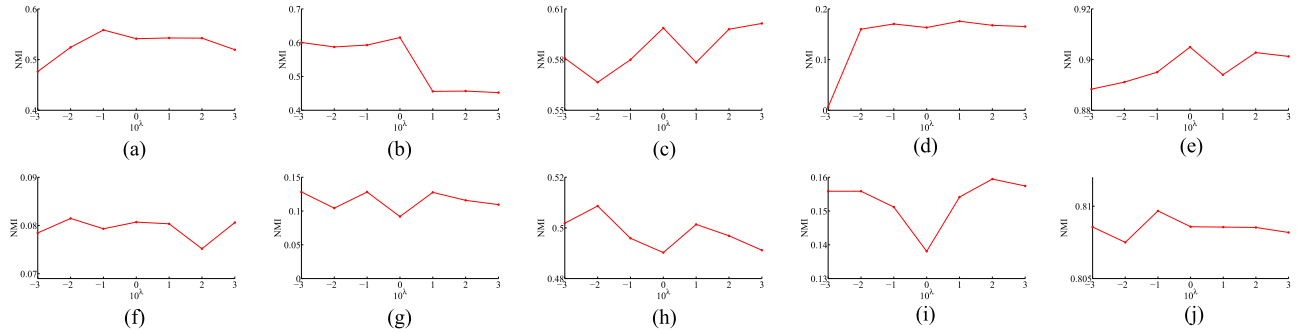

Fig. 5. NMI results of our method with varied parameter' setting (i.e., λ) on synthetic incomplete data sets keeping 50% of all features and 90% of all observed information. (a) Cane. (b) Uspst. (c) Yale. (d) Yeast. (e) Palm. (f) Cifar. (g) Connect. (h) Mnist. (i) Vehicle. (j) Aloi.

selection models. Second, with the increase of the incomplete sample ratio, the difference between our method and every comparison feature selection method was larger than the difference between our proposed method and every two imputation methods. For example, by considering the ACC results on data set Uspst, GSR_knn improved by 2.9% and 5.7% at the incomplete sample ratio 0.1 and 0.9, respectively, compared with GSR, whereas our method improved by 2.4% and 4.7% compared with RFS in the same incomplete sample ratios. The reason may be that both our method and imputation methods have more information to be used. Third, in imputation methods, GSR_knn outperformed GSR_mean. However, both of them were worse than our method. The reason is that imputed information in both GSR_knn and GSR_mean does not add the information of the whole data set (as the imputed information was derived from observed samples) and possibly introduces noise to degrade the effectiveness of feature selection.

G. Parameters' Sensitivity Analysis

Our proposed objective function in (8) only needs to tune one parameter, i.e., λ. We listed the variations of ACC and NMI, respectively, of our proposed method, at different values of the parameter λ (i.e., $\lambda \in \{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$) in Figs. 3–5.

Our method is sensitive to parameters' setting, but it may achieve stability and good clustering in some ranges. For example, the ACC results of our method varied from 68.43% to 46.27%, but it achieved the best results while $\lambda \in [10^{1}, 10^{2}]$ on data set Uspst. This enables our method to achieve reliable clustering performance via easy parameter tuning.

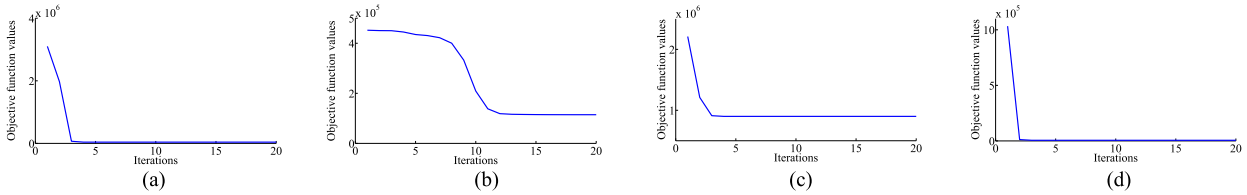

Fig. 6. Convergence analysis of our Algorithm 1 on real incomplete data sets keeping 50% of all features. (a) Advertisement. (b) Arrhythmia. (c) Cvpu. (d) Mice.

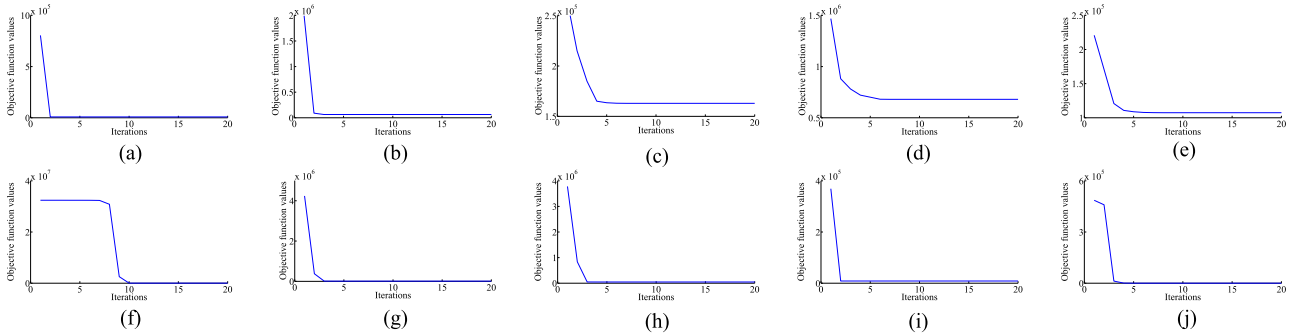

Fig. 7. Convergence analysis of our Algorithm 1 on synthetic incomplete data sets keeping 50% of all features and 90% of all observed information. (a) Cane. (b) Uspst. (c) Yale. (d) Yeast. (e) Palm. (f) Cifar. (g) Connect. (h) Mnist. (i) Vehicle. (j) Aloi.

H. Convergence Analysis

Section III-E theoretically proved the convergence of our proposed Algorithm 1 to solve (8). Figs. 6 and 7 reported the variations of the objective function value of (8) at different iterations. In our experiments, we set the stop criterion of Algorithm 1 as $(|\text{obj}(t+1) - \text{obj}(t)|/\text{obj}(t)) \leq 10^{-5}$, where $\text{obj}(t)$ represents the objective function value of the tth iteration.

Based on the results, Algorithm 1 achieved convergence as it monotonically decreased the objective function values of (8). Moreover, our experimental results indicated that our proposed Algorithm 1 can effectively optimize (8) with a fast convergence, i.e., within 20 iterations.

I. Comparison of Robust Loss Functions

We selected GemanMcClure loss function [see (6)] [38] in our proposed method. In this section, we compared it with other two robust loss functions (such as $\ell_1 - \ell_2$ estimator (i.e., $\phi(z) = (\mu + z^2)^{1/2}$) and the Welsch estimator (i.e., $\phi(z) = 1 - \exp(-(z^2/\mu^2))$) [42]) in this article. To do this, we denoted our (6) using the $\ell_1 - \ell_2$ estimator and the Welsch estimator, respectively, as Proposed1 and Proposed2. It is noteworthy that the difference among different robust loss functions is not either the main focus or the contribution of this work.

Specifically, we manually generated outliers by replacing the original values with the random maximum/minimum values of the features [43] at different outlier ratios in the range of {0, 5%, 10%, 15%, …, 45%, 50%} on four data sets, i.e., Cane, Uspst, Connect, and Vehicle. The comparison methods include Baseline, GSR_knn, Proposed, Proposed1, and Proposed2. Moreover, "Proposed 0" and "Proposed 0.7," respectively, indicate that our proposed method with zero and 70% missing ratios. Baseline and GSR_knn, respectively, are the worst and the best comparison methods. We reported the ACC results of all methods in Fig. 8.

First, all three methods with the robust loss function outperformed either GSR_knn or Baseline. Specifically, these methods improved on average by 4.71% and 2.12%, respectively, compared with Baseline and GSR_knn on four data sets. Second, with the increase of the outlier ratio, the clustering performance of every method with the robust loss function gradually decreases, while the clustering performance of both Baseline and GSR_knn drastically decreases. This implies that the methods with the robust loss function can effectively reduce the effect of outliers. Third, each method with the robust loss function has a tiny difference to the other two in terms of clustering performance. This indicates that there is not a significant difference among them in our proposed method. Moreover, it is difficult to find the best one among them since they achieved the best clustering performance on different data sets, different outlier ratios, and different missing ratios. This verified the conclusion again that different robust loss functions could result in different robustness [43].

J. Time and Space Costs

We listed the time cost (second) and space cost (MB) of the training process of all methods on the complete data sets in Table IV. It is noteworthy that three methods (i.e., GSR, GSR_knn, and GSR_mean) have the same time and space costs on the complete data sets. We did not list the time and space costs at a different missing ratio as the case-deletion methods (i.e., LPscore, RSR, GSR, and RFS) has less training samples compared with the proposed method for the incomplete data sets. As a result, RSR took the most time cost and GSR needed the most space cost as the time complexity of RSR is cubic to the sample size and the space complexity of GSR is cubic to the feature number.

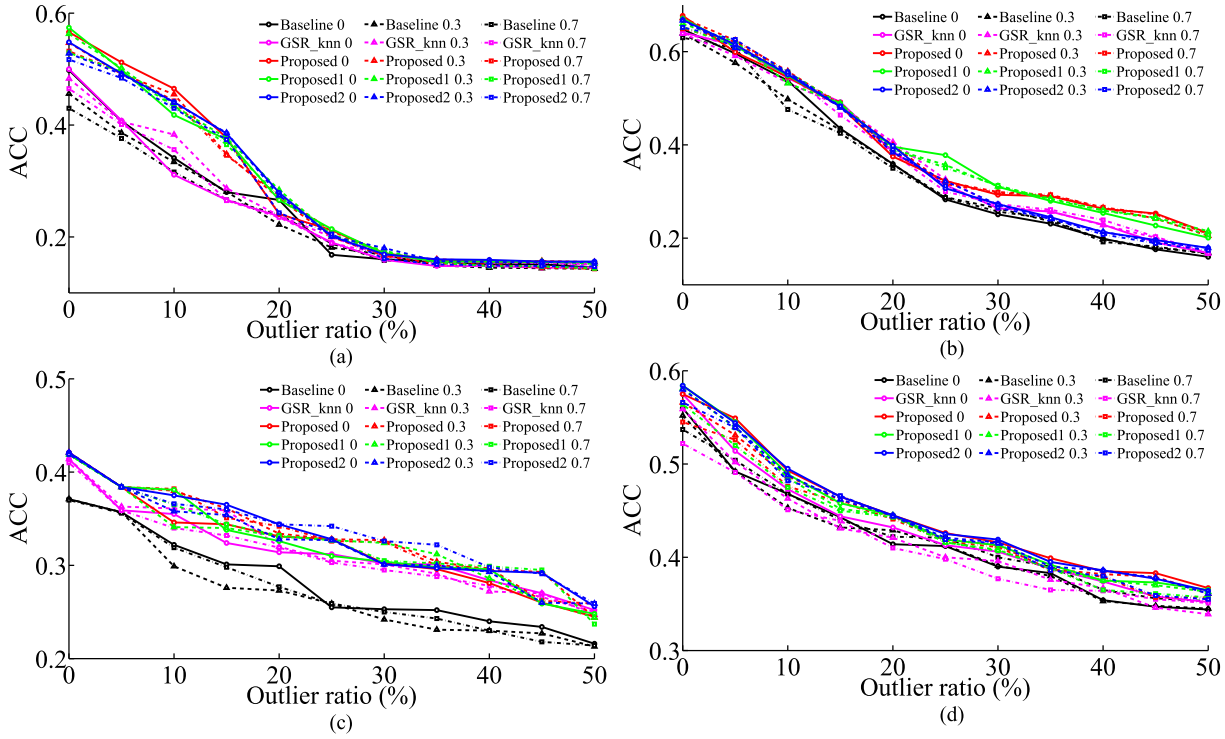

Fig. 8. Clustering results of different robust loss functions on four synthetic data sets keeping 50% of all features. (a) Cane. (b) Uspst. (c) Connect. (d) Vehicle.

TABLE IV
TIME AND SPACE COSTS OF ALL METHODS USING ALL SAMPLES FOR TRAINING

	Cane	Uspst	Yale	Yeast	Palm	Cifar	Connect	Mnist	Vechile	Aloi
Time cost (second)										
LPscore	0.86	0.63	0.38	1.91	0.72	203.80	13.62	45.381	14.33	22.34
RSR	2.81	2.56	1.23	5.49	2.77	6505.83	6674.03	7490.86	9752.71	61646.22
GSR	1.90	1.17	0.78	3.55	0.73	396.98	250.63	547.10	137.96	705.42
RFS	1.19	0.78	0.57	2.27	0.74	275.77	11.72	46.57	12.84	11.83
Proposed	3.20	1.35	0.62	3.16	0.80	593.50	21.14	157.5	23.82	30.88
Space cost (MB)										
LPscore	7.23	4.02	1.32	17.08	4.01	1442.82	66.64	387.13	81.37	108.21
RSR	95.03	7.76	75.94	108.35	4.01	1442.75	102.69	515.65	61.70	396.70
GSR	9.90	25.53	8.20	17.08	25.36	22825.63	28937.47	31068.50	39393.22	7355.70
RFS	42.67	4.02	8.21	17.08	4.01	1442.75	66.64	429.60	61.70	108.20
Proposed	10.08	4.02	8.21	17.08	6.17	1442.75	66.64	431.56	61.70	109.67

V. CONCLUSION

This article proposed a novel method to conduct UFS on incomplete data sets. Specifically, we employed half-quadratic minimization to consider sample importance for conducting feature selection and used an indicator matrix to avoid unobserved information taking participation in the process of feature selection as well as making the use of all the observed information. Experimental results showed that our method achieved the best clustering results compared with the comparison methods.

We will extend our proposed unsupervised framework to conduct supervised/semisupervised feature selection on incomplete data sets in our future work.

REFERENCES

[1] 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.

[2] Z. Zhang, L. Liu, F. Shen, H. T. Shen, and L. Shao, "Binary multi-view clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1774–1782, Jul. 2019.

[3] X. Zhu, S. Zhang, Y. Zhu, W. Zheng, and Y. Yang, "Self-weighted multi-view fuzzy clustering," *ACM Trans. Knowl. Discovery Data*, vol. 14, no. 4, p. 48, 2020, doi: 10.1145/3396238.

[4] R. Hu, X. Zhu, Y. Zhu, and J. Gan, "Robust SVM with adaptive graph learning," *World Wide Web*, vol. 23, pp. 1945–1968, Dec. 2019, doi: 10.1007/s11280-019-00766-x.

[5] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Electr. Eng.*, vol. 40, no. 1, pp. 16–28, Jan. 2014.

[6] S. Solorio-Fernández, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad, "A review of unsupervised feature selection methods," *Artif. Intell. Rev.*, vol. 53, no. 2, pp. 907–948, Feb. 2020.

[7] Y. Yang, Z. Ma, Y. Yang, F. Nie, and H. T. Shen, "Multitask spectral clustering by exploring intertask correlation," *IEEE Trans. Cybern.*, vol. 45, no. 5, pp. 1069–1080, May 2015.

[8] E. Adeli *et al.*, "Semi-supervised discriminative classification robust to sample-outliers and feature-noises," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 515–522, Feb. 2019.

[9] P. J. Huber, "Robust statistics," in *International Encyclopedia of Statistical Science*. 2011, pp. 1248–1251.

[10] X. Zhu, Y. Zhu, and W. Zheng, "Spectral rotation for deep one-step clustering," *Pattern Recognit.*, vol. 105, Sep. 2020, Art. no. 107175, doi: 10.1016/j.patcog.2019.107175.
[11] X. Zhu, J. Gan, G. Lu, J. Li, and S. Zhang, "Spectral clustering via half-quadratic optimization," *World Wide Web*, vol. 23, pp. 1969–1988, Nov. 2020, doi: 10.1007/s11280-019-00731-8.
[12] X. Zhu, J. Yang, C. Zhang, and S. Zhang, "Efficient utilization of missing data in cost-sensitive learning," *IEEE Trans. Knowl. Data Eng.*, early access, Nov. 28, 2019, doi: 10.1109/TKDE.2019.2956530.
[13] Y. Zhou, L. Tian, C. Zhu, X. Jin, and Y. Sun, "Video coding optimization for virtual reality 360-degree source," *IEEE J. Sel. Topics Signal Process.*, vol. 14, no. 1, pp. 118–129, Jan. 2020.
[14] J. Van Hulse and T. M. Khoshgoftaar, "Incomplete-case nearest neighbor imputation in software measurement data," *Inf. Sci.*, vol. 259, pp. 596–610, Feb. 2014.
[15] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, "Efficient kNN classification with different numbers of nearest neighbors," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1774–1785, May 2018.
[16] W. Zheng, X. Zhu, Y. Zhu, and S. Zhang, "Robust feature selection on incomplete data," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3191–3197.
[17] 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.
[18] P. Zhu, Q. Hu, C. Zhang, and W. Zuo, "Coupled dictionary learning for unsupervised feature selection," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 2422–2428.
[19] M. Nikolova and R. H. Chan, "The equivalence of half-quadratic minimization and the gradient linearization iteration," *IEEE Trans. Image Process.*, vol. 16, no. 6, pp. 1623–1627, Jun. 2007.
[20] G. Doquire and M. Verleysen, "Feature selection with missing data using mutual information estimators," *Neurocomputing*, vol. 90, pp. 3–11, Aug. 2012.
[21] D. Huang, R. Cabral, and F. De la Torre, "Robust regression," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 363–375, Feb. 2016.
[22] X. Zhu, X. Li, S. Zhang, Z. Xu, L. Yu, and C. Wang, "Graph PCA hashing for similarity search," *IEEE Trans. Multimedia*, vol. 19, no. 9, pp. 2033–2044, Sep. 2017.
[23] H. T. Shen *et al.*, "Exploiting subspace relation in semantic labels for cross-modal hashing," *IEEE Trans. Knowl. Data Eng.*, early access, Jan. 29, 2020, doi: 10.1109/TKDE.2020.2970050.
[24] 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.
[25] H. Peng and Y. Fan, "A general framework for sparsity regularized feature selection via iteratively reweighted least square minimization," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 2471–2477.
[26] G. Lan, C. Hou, F. Nie, T. Luo, and D. Yi, "Robust feature selection via simultaneous sapped norm and sparse regularizer minimization," *Neurocomputing*, vol. 283, pp. 228–240, Mar. 2018.
[27] L. Sun, J. Xu, and Y. Tian, "Feature selection using rough entropy-based uncertainty measures in incomplete decision systems," *Knowl.-Based Syst.*, vol. 36, pp. 206–216, Dec. 2012.
[28] W. Shu and H. Shen, "Multi-criteria feature selection on cost-sensitive data with missing values," *Pattern Recognit.*, vol. 51, pp. 268–280, Mar. 2016.
[29] J. Huang, T. Zhang, and D. Metaxas, "Learning with structured sparsity," *J. Mach. Learn. Res.*, vol. 12, pp. 3371–3412, Jan. 2011.
[30] S. Bengio, F. Pereira, Y. Singer, and D. Strelow, "Group sparse coding," in *Proc. NIPS*, 2009, pp. 82–89.
[31] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint ℓ2,1-norms minimization," in *Proc. NIPS*, 2010, pp. 1813–1821.
[32] M. Nikolova and M. K. Ng, "Analysis of half-quadratic minimization methods for signal and image recovery," *SIAM J. Sci. Comput.*, vol. 27, no. 3, pp. 937–966, Jan. 2005.
[33] M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Proc. NIPS*, 2010, pp. 1189–1197.
[34] Y. J. Lee and K. Grauman, "Learning the easy things first: Self-paced visual category discovery," in *Proc. CVPR*, Jun. 2011, pp. 1721–1728.
[35] S. Negahban, B. Yu, M. J. Wainwright, and P. K. Ravikumar, "A unified framework for high-dimensional analysis of *M*-estimators with decomposable regularizers," in *Proc. NIPS*, 2009, pp. 1348–1356.
[36] Y. Fan, R. He, J. Liang, and B. Hu, "Self-paced learning: An implicit regularization perspective," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1877–1883.
[37] R. J. Little and D. B. Rubin, *Statistical Analysis With Missing Data*. Hoboken, NJ, USA: Wiley, 2014.
[38] S. Ganan and D. McClure, "Bayesian image analysis: An application to single photon emission tomography," *Amer. Stat. Assoc.*, pp. 12–18, 1985.
[39] S. Geman and D. E. Mcclure, "Statistical methods for tomographic image reconstruction," in *Proc. Session ICI Bull. (ICI)*, vol. 4, 1987, pp. 1–68.
[40] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk, "Iteratively reweighted least squares minimization for sparse recovery," *Commun. Pure Appl. Math.*, vol. 63, no. 1, pp. 1–38, Jan. 2010.
[41] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Proc. NIPS*, 2005, pp. 507–514.
[42] Y. Zhang, Z. Sun, R. He, and T. Tan, "Robust subspace clustering via half-quadratic minimization," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3096–3103.
[43] R. He, W.-S. Zheng, T. Tan, and Z. Sun, "Half-quadratic-based iterative minimization for robust sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 261–275, Feb. 2014.

Heng Tao Shen (Senior Member, IEEE) received the B.Sc. (Hons.) and Ph.D. degrees from the Department of Computer Science, National University of Singapore, Singapore, in 2000 and 2004, respectively.

He then joined The University of Queensland, Brisbane, QLD, Australia, where he became a Professor in late 2011. He is currently a Professor and the Dean of the School of Computer Science and Engineering, the Executive Dean of the AI Research Institute, and the Director of the Centre for Future Media, University of Electronic Science and Technology of China, Chengdu, China. His research interests mainly include multimedia search, computer vision, artificial intelligence, and big data management. He has published over 280 peer-reviewed papers, including over 80 IEEE/ACM TRANSACTIONS.

Dr. Shen is a fellow of OSA and an ACM Distinguished Member. He received seven best paper awards from international conferences, including the Best Paper Award from the ACM Multimedia 2017 and the Best Paper Award–Honorable Mention from ACM SIGIR 2017. He is/was an Associate Editor of *ACM Transactions of Data Science*, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON MULTIMEDIA, and the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING.

Yonghua Zhu is currently pursuing the master's degree with Guangxi University, Nanning, China.

His current research interests include data mining and machine learning.

Wei Zheng is currently pursuing the master's degree with Guangxi Normal University, Guilin, China.

His current research interests include data mining and pattern recognition.

Xiaofeng Zhu (Senior Member, IEEE) is currently a Faculty Member with the University of Electronic Science and Technology of China, Chengdu, China. His current research interests include large-scale multimedia retrieval, feature selection, sparse learning, data preprocess, and medical image analysis.