

Article

RMFRASL: Robust Matrix Factorization with Robust Adaptive Structure Learning for Feature Selection

Shumin Lai ¹, Longjun Huang ^{1,*}, Ping Li ¹, Zhenzhen Luo ¹, Jianzhong Wang ² and Yugen Yi ^{1,*} ¹ School of Software, Jiangxi Normal University, Nanchang 330022, China² College of Information Science and Technology, Northeast Normal University, Changchun 130117, China

* Correspondence: 003721@jxnu.edu.cn (L.H.); yiyg510@jxnu.edu.cn (Y.Y.)

Abstract: In this paper, we present a novel unsupervised feature selection method termed robust matrix factorization with robust adaptive structure learning (RMFRASL), which can select discriminative features from a large amount of multimedia data to improve the performance of classification and clustering tasks. RMFRASL integrates three models (robust matrix factorization, adaptive structure learning, and structure regularization) into a unified framework. More specifically, a robust matrix factorization-based feature selection (RMFFS) model is proposed by introducing an indicator matrix to measure the importance of features, and the L_{21} -norm is adopted as a metric to enhance the robustness of feature selection. Furthermore, a robust adaptive structure learning (RASL) model based on the self-representation capability of the samples is designed to discover the geometric structure relationships of original data. Lastly, a structure regularization (SR) term is designed on the learned graph structure, which constrains the selected features to preserve the structure information in the selected feature space. To solve the objective function of our proposed RMFRASL, an iterative optimization algorithm is proposed. By comparing our method with some state-of-the-art unsupervised feature selection approaches on several publicly available databases, the advantage of the proposed RMFRASL is demonstrated.

Keywords: feature selection; matrix factorization; adaptive structure learning; structure regularization



Citation: Lai, S.; Huang, L.; Li, P.; Luo, Z.; Wang, J.; Yi, Y. RMFRASL: Robust Matrix Factorization with Robust Adaptive Structure Learning for Feature Selection. *Algorithms* **2023**, *16*, 14. <https://doi.org/10.3390/a16010014>

Academic Editors: Frank Werner and Alicia Cordero

Received: 31 October 2022

Revised: 3 December 2022

Accepted: 22 December 2022

Published: 26 December 2022



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, with the rapid development of the Internet and the popularization of mobile devices, it has become more and more convenient to analyze, search, transmit, and process large amounts of data, which can also promote the development of society and the sharing of resources. However, with the continuous advancement of information technology, mobile Internet data are becoming enormous [1], inevitably increasing the amount of data needing to be processed in real life. Although these high-dimensional data can bring us helpful information, they also cause problems such as data redundancy and noisy data [2–4]. Dimensionality reduction technology can not only avoid the “curse of dimensionality” by removing redundant and irrelevant features, but also reduce the time-consuming nature of data processing and the storage space of data [5]. Therefore, it is widely used to process large quantities of Internet mobile data [6]. The most commonly used dimensionality reduction techniques are feature extraction and feature selection. Feature extraction projects the original high-dimensional data into a low-dimensional subspace [7–9]. Unlike feature extraction, feature selection selects an optimal subset from the original feature set. Since feature selection can preserve the semantics of original features, it is more explicable and physical; thus, it has been widely used in many fields, e.g., intrusion detection of industrial Internet [10], simulated attack of Internet of things [11], and computer vision [12].

According to the presence or absence of labels, feature selection methods can be divided into supervised, semi-supervised, and unsupervised methods. Among them,

unsupervised feature selection is a significant challenge research because few labeled data are available in real life [13]. Unsupervised feature selection can be divided into three categories: (1) filtering methods, (2) wrapping methods, and (3) embedding-based methods. The filtering method involves performing feature selection on the original dataset in advance and then training a learner. In other words, the procedure of feature selection is independent of the learner training [14]. The wrapping method continuously selects feature subsets from the initial dataset and trains the learner until the optimal feature subset is selected [15,16]. Embedding automatically performs feature selection during the learning process. Compared with the filtering and wrapping methods, embedded methods can significantly reduce the computational cost and achieve good results [17,18].

He et al. [19] proposed an unsupervised feature selection method named Laplacian score (LS), which selects optimal feature subsets to preserve the local manifold structure of the original data. However, the LS method ignores the correlation between features. In order to overcome this disadvantage, Liu et al. [20] proposed an unsupervised feature selection method with robust neighborhood embedding (RNEFS). RNEFS adopts the local linear embedding (LLE) algorithm to compute the feature weight matrix. Then, the L_1 -norm-based reconstructing error was utilized to suppress the influence of noise and outliers. Subsequently, Wang et al. [21] proposed an efficient soft label feature selection (SLFS) method, which first performed soft label learning and then selected features on the basis of the learned soft labels. However, weight matrices of the above methods were defined in advance by considering the similarity of features, which may have affected their performances due to the quality of weight matrices. Therefore, Yuan et al. [22] proposed an adaptive graph convex non-negative matrix analysis method for unsupervised feature selection (CNAFS) which embedded self-expression and pseudo-label information into a joint model. CNAFS can select the most representative features by computing top-ranked features. Shang et al. [23] proposed a non-negative matrix factorization adaptive constraint-based feature selection (NNSAFS) method, which introduced a feature map into matrix factorization to combine manifold learning and feature selection. By combining feature transformation with adaptive constraints, NNSAFS can significantly improve the accuracy of feature selection. However, the above methods take Euclidean distance as the metric. Euclidean distance is sensitive to noise and redundant data, which may cause the performance to be unstable and not robust. To address this issue, Zhao et al. [24] proposed a joint adaptive and discriminative unsupervised feature selection method, which adaptively learns similar graphs while employing irrelevance constraints to enhance the discriminability of the selected features. Zhu et al. [25] proposed a regularized self-representation (RSR) model, which efficiently performs feature selection by using the self-representation ability of features with the L_{21} -norm error to guarantee robustness. Shi et al. [26] proposed a robust spectral learning framework, which utilized graph embedding and spectral regression to deal with noise during unsupervised feature selection. Du et al. [27] proposed a robust unsupervised feature selection based on the matrix factorization (RUFMS) method, which adopted the L_{21} -norm to improve robustness and preserve the local manifold structure, so that an optimal feature subset could be found to retain the manifold structure information. Miao et al. [28] proposed an unsupervised feature selection method named graph-regularized local linear embedding (GRLLE), which combined linear embedding and graph regularization into a unified framework while maintaining local spatial structure. However, the correlation between the selected features was still ignored in the above methods.

This paper proposes an unsupervised feature selection method called robust matrix factorization with robust adaptive structure learning (RMFRASL) to solve the above problems. First, this paper presents a robust matrix factorization-based feature selection model (RMFFS), which introduces a feature indicator matrix into matrix factorization with an L_{21} -norm-based error metric. Next, a robust adaptive structure learning (RASL) model based on self-representation capability is proposed to better describe the local geometry of the samples, which can reduce redundant and noisy features to learn a more discriminative

graph structure. Then, to preserve the geometric structure information during the feature selection, a structure regularization (SR) term is constructed. Finally, the above three models are combined into a unified framework and shown in Figure 1. In summary, the main contributions of this paper are as follows:

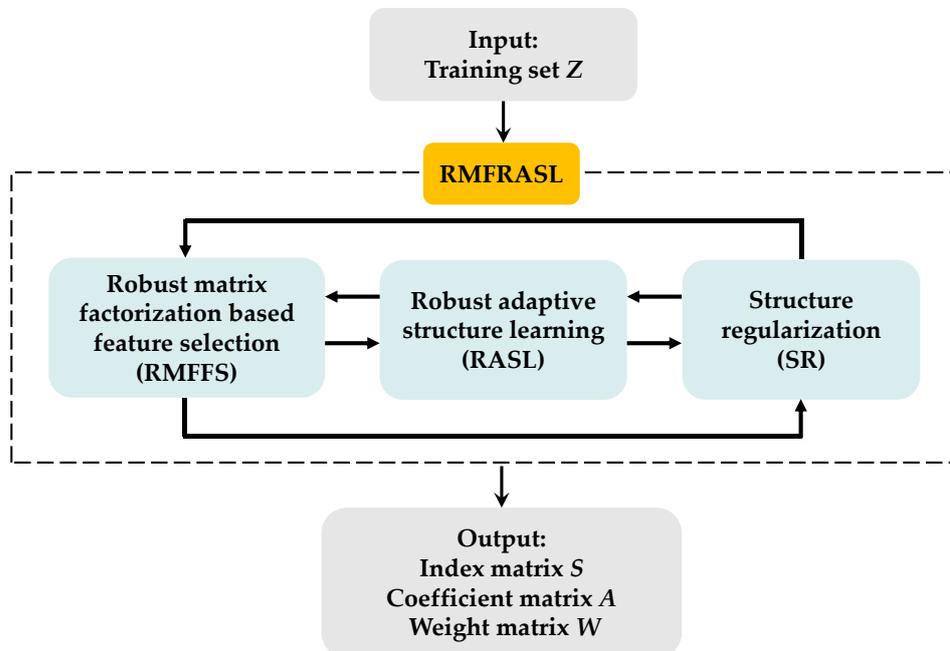


Figure 1. The structure flowchart of the proposed RMFRASL method.

(1) In order to overcome the shortcoming of existing methods in that they neglect the correlation of features, we introduce an indicator matrix into matrix factorization and design a robust matrix factorization-based feature selection (RMFFS).

(2) To consider the structure information during the process of feature selection, a structure regularization term is integrated into the RMFFS model.

(3) To solve the disadvantages of the existing methods needing to predefine a graph structure, a robust adaptive structure learning model based on the self-representation capability of the sample is proposed, which adopts L_{21} -norm as an error metric rather than L_2 -norm to reduce the influence of noise or outliers.

(4) To further improve the performance of feature selection methods, an unified framework is proposed by combining the process of the feature selection and structure learning.

(5) An iterative optimization algorithm is designed to solve the objective function, and the convergence of the optimization algorithm is verified from mathematical theory and numerical experiments.

This paper is organized as follows: Section 2 briefly reviews the related work. The proposed RMFRASL model is introduced in Section 3. The convergence analysis is provided in Section 4. The experimental results and analysis are provided in Section 5. Section 6 gives the conclusions and future research works.

2. Related Work

2.1. Notations

Assume that $Z = [z_1; z_2; \dots; z_j; \dots; z_n] \in R^{n \times d}$ is a matrix with the size of $n \times d$ to represent high-dimensional original data, where n and d denote the number of data and

features, and $z_j = [z_{j1}, z_{j2}, \dots, z_{jd}] \in R^{1 \times d}$ is a row vector with the size of d to denote the j -th data of Z . L_1 -norm, L_2 -norm, and L_{21} -norm of the matrix Z can be defined as follows:

$$\|Z\|_1 = \sum_{i=1}^n \sum_{j=1}^d |z_{ij}|, \tag{1}$$

$$\|Z\|_2 = \sqrt{\sum_{i=1}^n \sum_{j=1}^d z_{ij}^2}, \tag{2}$$

$$\|Z\|_{2,1} = \sum_{i=1}^n \|z_i\|_2 = \sum_{i=1}^n \sqrt{\sum_{j=1}^d z_{ij}^2}. \tag{3}$$

According to the above definitions and matrix correlation operations [29], we can obtain that $\|Z\|_{2,1} = tr(Z^T U Z)$, where $U \in R^{n \times n}$ denotes the diagonal matrix whose diagonal elements are $u_{ii} = 1/(\|z_i\|_2)$. For ease of reading, a description of the commonly used symbols is listed in Table 1.

Table 1. Description of commonly used symbols.

Symbol	Description	Symbol	Description
$Z \in R^{n \times d}$	Sample matrix	d	Dimension of sample
$S \in R^{d \times k}$	Indicator matrix	n	Number of samples
$A \in R^{k \times d}$	Low-dimensional feature matrix	k	Number of selected features
$W \in R^{n \times n}$	Weight matrix	$\ \cdot\ _1$	L_1 -norm
$I \in R^{k \times k}$	Identity matrix	$\ \cdot\ _2$	L_2 -norm
$U \in R^{n \times n}$	Diagonal matrix	$\ \cdot\ _{2,1}$	L_{21} -norm
$C \in R^{d \times d}$	Diagonal matrix	$tr(\cdot)$	Matrix trace operation

2.2. Introduction of Indicator Matrix

The indicator matrix $S = [S_1, S_2, \dots, S_k] \in R^{d \times k}$ is a binary matrix, where S_i denotes a column vector with only one element value equal to 1. To understand the indicator matrix S for feature selection more clearly, we establish an example. First, we assume that the data matrix contains four samples, and that each sample has five features; the data matrix can be described as follows:

$$X = \begin{matrix} & & f_1 & f_2 & f_3 & f_4 & f_5 \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{matrix} & \left(\begin{matrix} 0.9 & 0.1 & 0.7 & 1.0 & 0.2 \\ 0.7 & 0.4 & 0.5 & 0.3 & 0.4 \\ 0.3 & 0.8 & 0.4 & 0.6 & 0.1 \\ 1.0 & 0.6 & 0.6 & 0.8 & 1.0 \end{matrix} \right) . \end{matrix} \tag{4}$$

Then, supposing that three features are selected (i.e., f_1, f_4 and f_5), we can obtain the selected feature subset as follows:

$$X' = \begin{matrix} & & f_1 & f_4 & f_5 \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{matrix} & \left(\begin{matrix} 0.9 & 1.0 & 0.2 \\ 0.7 & 0.3 & 0.4 \\ 0.3 & 0.6 & 0.1 \\ 1.0 & 0.8 & 1.0 \end{matrix} \right) . \end{matrix} \tag{5}$$

Meanwhile, the indicator matrix S can be assumed as follows:

$$S = \begin{matrix} & & & 1 & 4 & 5 \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{matrix} . \tag{6}$$

Thus, we can obtain

$$\begin{matrix} X & S & = & X' \\ \begin{pmatrix} 0.9 & 0.1 & 0.7 & 1.0 & 0.2 \\ 0.7 & 0.4 & 0.5 & 0.3 & 0.4 \\ 0.3 & 0.8 & 0.4 & 0.6 & 0.1 \\ 1.0 & 0.6 & 0.6 & 0.8 & 1.0 \end{pmatrix} & \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} & = & \begin{pmatrix} 0.9 & 1.0 & 0.2 \\ 0.7 & 0.3 & 0.4 \\ 0.3 & 0.6 & 0.1 \\ 1.0 & 0.8 & 1.0 \end{pmatrix} \end{matrix} . \tag{7}$$

Equation (7) can describe the process of feature selection using the indicator matrix S . Finally, we can also obtain $S^T S = I_{3 \times 3}$ as follows:

$$S^T S = I \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} . \tag{8}$$

3. Robust Matrix Factorization with Robust Adaptive Structure Learning

In this section, a robust matrix factorization with robust adaptive structure learning (RMFRASL) model for unsupervised feature selection is developed to deal with multimedia data collected from Internet of things. Then, an iterative optimization method is designed to solve the objective function of the proposed method. Lastly, a description and the time complexity of the proposed algorithm are provided.

3.1. The Proposed RMFRASL Model

The RMFRASL effectively integrates three models, i.e., robust matrix factorization-based feature selection (RMFFS), robust adaptive structure learning (RASL), and structure regularization (SR), into a unified framework.

3.1.1. RMFFS Model

In order to take the correlation of features into consideration, the self-representation of data or features is widely used for feature representation learning [30]. Therefore, a new feature selection model named robust matrix factorization-based feature selection (RMFFS) is proposed. In the proposed RMFFS model, we first utilize an indicator matrix to find a suitable subset of features for capturing the most critical information to represent the original features approximately. Then, unlike the previous methods, the L_{21} -norm metric is imposed on the reconstruction errors to improve the effectiveness and enhance robustness of feature selection. As a result, the objective function of the RMFFS model is defined as follows:

$$\begin{aligned} \min \phi(Z, A, S) &= \|Z - ZSA\|_{2,1} \\ \text{s.t. } S &\geq 0, A \geq 0, S^T S = I_{k \times k} \end{aligned} , \tag{9}$$

where $I_{k \times k}$ is an identity matrix with the size of k , $A \in R^{k \times d}$ is a coefficient matrix which maps the original features from high-dimensional space into a new low-dimensional space, $S = [s_1, s_2, \dots, s_k] \in R^{d \times k}$ is defined as a feature weight matrix, and k is the number of

selected features. The constraint conditions of $S^T S = I_{k \times k}$ and $S \geq 0$ are to force each element of matrix S to be either one or zero and at most a nonzero element in any row or column during the iterative update process. Thus, the matrix S can be considered as an indicator matrix for the selected features under these constraint conditions.

3.1.2. RASL Model

Although the proposed RMFFS model can implement feature selection, it does not sufficiently consider the local structure relationships among original data, which may lead to unsatisfactory performance. Therefore, a robust structure learning method based on the self-representation capability of the original data is proposed, which can learn non-negative representation coefficients to indicate the structural relationships among original data. In order to learn more discriminative non-negative representation coefficients, an error metric function with L_{21} -norm is imposed. Hence, the objective function of RASL is defined as follows:

$$\begin{aligned} \min \partial(Z, W) &= \|Z^T - Z^T W\|_{2,1}, \\ \text{s.t. } W &\geq 0 \end{aligned} \quad (10)$$

where $W = [w_1, w_2, \dots, w_n] \in R^{n \times n}$ denotes the non-negative representation coefficient matrix. The element of matrix W with a larger value means that the correlation between two data is more remarkable.

3.1.3. SR Model

Preserving the structure of selected feature space is very important for feature learning [31]. Thus, a structure regularization (SR) term is designed, which can be defined as follows:

$$\begin{aligned} \min \varepsilon(Z, S, W) &= \sum_{i,j=1}^n \|y_i - y_j\|_2^2 w_{ij} = \text{tr}(YDY^T) - \text{tr}(YWY^T) \\ &= \text{tr}(YLY^T) = \text{tr}(S^T Z^T LZS) \\ \text{s.t. } S &\geq 0, W \geq 0 \end{aligned} \quad (11)$$

where $y_i = S^T z_i^T$ denotes the selected feature of sample z_i , L is a Laplacian matrix defined by $L = D - W$ [32], and D is a diagonal matrix with elements of $d_{ii} = \sum_{j=1}^n w_{ij}$.

3.1.4. The Framework of RMFRASL

The three modules (feature selection, robust structure learning, and structure regularization) are integrated into a unified framework to obtain the final objective function of RMFRASL as follows:

$$\begin{aligned} \min \psi(Z, S, A, W) &= \phi(Z, A, S) + \alpha \varepsilon(Z, S, W) + \beta \partial(Z, W) \\ &= \|Z - ZSA\|_{2,1} + \alpha \text{tr}(S^T Z^T LZS) + \beta \|Z^T - Z^T W\|_{2,1} \\ \text{s.t. } S &\geq 0, A \geq 0, W \geq 0, S^T S = I_{k \times k} \end{aligned} \quad (12)$$

where $\alpha > 0$ and $\beta > 0$ are tradeoff parameters that adjust the importance of each term in the model.

3.2. Model Optimization

The proposed model has three variables (S , A , and W) to be solved. The objective function of Equation (12) is not a convex optimization problem for all variables [32]. Therefore, we cannot obtain the globally optimal solution for the objective function. In order to solve the above problem, we need to design an iterative optimization algorithm. Specifically, we can optimize one of the variables while fixing the remainder, with alternate iterative updates until the objective function converges.

3.2.1. Fix A and W ; Update S

For Equation (12), removing the irrelevant term with S yields the following objective function:

$$\begin{aligned} \varphi(S) &= \|Z - ZSA\|_{2,1} + \alpha \operatorname{tr}(S^T Z^T LZS) \\ \text{s.t. } S &\geq 0, S^T S = I_{k \times k} \end{aligned} \tag{13}$$

However, the constraint conditions in Equation (13) make it difficult to optimize; hence, we can alleviate the constraint condition $S^T S = I_{k \times k}$ to $\|S^T S - I_{k \times k}\|_2^2$ as a penalty term. Thus, Equation (13) is transformed into the following objective function:

$$\begin{aligned} \varphi(S) &= \|Z - ZSA\|_{2,1} + \alpha \operatorname{tr}(S^T Z^T LZS) + \lambda \|S^T S - I\|_2^2, \\ \text{s.t. } S &\geq 0 \end{aligned} \tag{14}$$

where the parameter $\lambda > 0$ is used to constrain the orthogonality of vector S .

Let $M = Z^T LZ$; according to the above definitions and matrix correlation operations [29], Equation (14) can be simplified to Equation (15) through a series of algebraic operations:

$$\begin{aligned} \varphi(S) &= \|Z - ZSA\|_{2,1} + \alpha \operatorname{tr}(S^T MS) + \lambda \operatorname{tr}((S^T S - I)(S^T S - I)^T) \\ &= \operatorname{tr}((Z - ZSA)^T U(Z - ZSA)) + \alpha \operatorname{tr}(S^T MS) + \lambda \operatorname{tr}((S^T S - I)(S^T S - I)) \\ &= \operatorname{tr}((Z^T - A^T S^T Z^T)U(Z - ZSA)) + \alpha \operatorname{tr}(S^T MS) + \lambda \operatorname{tr}(S^T S S^T S - 2S^T S) \\ &= \operatorname{tr}(Z^T UZ - 2A^T S^T Z^T UZ + A^T S^T Z^T UZSA + \alpha S^T MS + \lambda S^T S S^T S - 2\lambda S^T S) \\ \text{s.t. } S &\geq 0 \end{aligned} \tag{15}$$

where $U \in R^{n \times n}$ denotes the diagonal matrix defined as follows:

$$U_{ii} = \frac{1}{\|(Z - ZSA)^i\|_2 + \varepsilon} \tag{16}$$

where ε is a small constant.

To solve Equation (15), a Lagrange multiplier Λ with non-negative constraint $S \geq 0$ is introduced, and the Lagrangian function $\varphi(S, \Lambda)$ can be defined as follows:

$$\begin{aligned} \varphi(S, \Lambda) &= \operatorname{tr}(Z^T UZ - 2A^T S^T Z^T UZ + A^T S^T Z^T UZSA + \alpha S^T MS) \\ &\quad + \lambda \operatorname{tr}(S^T S S^T S - 2S^T S) + \operatorname{tr}(\Lambda S) \\ \text{s.t. } S &\geq 0 \end{aligned} \tag{17}$$

The partial derivative of Equation (17) with respect to matrix S can be obtained as follows:

$$\frac{\partial \varphi(S, \Lambda)}{\partial S} = -2Z^T UZ A^T + 2Z^T UZ S A A^T + 2\alpha MS + 4\lambda S S^T S - 4\lambda S + \Lambda. \tag{18}$$

Equation (19) can be obtained according to the KKT condition ($\Lambda_{ij} S_{ij} = 0$) [33].

$$(-2Z^T UZ A^T + 2Z^T UZ S A A^T + 2\alpha MS + 4\lambda S S^T S - 4\lambda S + \Lambda) S_{ij} = 0. \tag{19}$$

The update formula of the matrix S is as follows:

$$S = S \odot \frac{Z^T UZ A^T + \alpha M^- S + 2\lambda S}{Z^T UZ S A A^T + \alpha M^+ S + 2\lambda S S^T S} \tag{20}$$

where $M = M^+ - M^-$, $M^+ = \frac{|M|+M}{2}$, and $M^- = \frac{|M|-M}{2}$.

3.2.2. Fix S and W; Update A

For Equation (12), removing the irrelevant term with A yields the following objective function:

$$\begin{aligned} \varphi(A) &= \|Z - ZSA\|_{2,1} \\ \text{s.t. } & A \geq 0 \end{aligned} \quad (21)$$

Through a series of algebraic formulas [29], Equation (21) can be simplified as

$$\begin{aligned} \varphi(A) &= \|Z - ZSA\|_{2,1} = \text{tr}((Z - ZSA)^T U (Z - ZSA)) \\ &= \text{tr}(Z^T U Z - 2A^T S^T Z^T U Z + A^T S^T Z^T U Z S A) \\ \text{s.t. } & A \geq 0 \end{aligned} \quad (22)$$

Likewise, the Lagrangian multiplier ω with a non-negative constraint $A \geq 0$ is introduced to solve Equation (22), and the Lagrangian function $\varphi(A, \omega)$ can be expressed as follows:

$$\varphi(A, \omega) = \text{tr}(Z^T U Z - 2A^T S^T Z^T U Z + A^T S^T Z^T U Z S A) + \text{tr}(\omega A). \quad (23)$$

The partial derivative of Equation (23) with respect to A is defined as follows:

$$\frac{\partial \varphi(A, \omega)}{\partial A} = -2S^T Z^T U Z + 2S^T Z^T U Z S A + \omega = 0. \quad (24)$$

As the complementary relaxation of the KKT condition leads to $\omega_{ij} A_{ij} = 0$, we can get Equation (25).

$$(-2S^T Z^T U Z + 2S^T Z^T U Z S A + \omega) A_{ij} = 0. \quad (25)$$

According to Equation (25), the update rule of the matrix A is as follows:

$$A = A \odot \frac{S^T Z^T U Z}{S^T Z^T U Z S A}. \quad (26)$$

3.2.3. Fix S and A; Update W

For Equation (12), the following objective function can be obtained by removing the irrelevant term with W:

$$\begin{aligned} \varphi(W) &= \beta \|Z^T - Z^T W\|_{2,1} + \alpha \sum_{i,j=1}^n \|y_i - y_j\|_2^2 w_{ij} \\ &= \beta \|Z^T - Z^T W\|_{2,1} + \alpha \sum_{i,j=1}^n \text{Dist} Y_{ij} w_{ij} \end{aligned} \quad (27)$$

where $\text{Dist} Y = [\text{Dist} Y_{ij}] \in R^{n \times n}$, and $\text{Dist} Y_{ij} = \|y_i - y_j\|_2^2$.

Through a series of algebraic formulas [29], Equation (27) can be simplified as

$$\begin{aligned} \varphi(W) &= \|Z^T - Z^T W\|_{2,1} + \mu \sum_{i,j=1}^n \text{Dist} Y_{ij} w_{ij} \\ &= \text{tr}((Z^T - Z^T W)^T C (Z^T - Z^T W)) + \mu \text{tr}(\text{Dist} Y W) \\ &= \text{tr}(Z C Z^T - 2W^T Z C Z^T + W^T Z C Z^T W + \mu \text{Dist} Y W) \\ \text{s.t. } & W \geq 0 \end{aligned} \quad (28)$$

where $\mu = \alpha / \beta$ ($\beta \neq 0$) and $C \in R^{d \times d}$ is a diagonal matrix defined as follows:

$$C_{ii} = \frac{1}{\|(Z^T - Z^T W)^i\|_2 + \varepsilon}. \quad (29)$$

To solve Equation (28), a Lagrange multiplier τ for the non-negative constraint $W \geq 0$ is introduced, and the Lagrangian function $\varphi(W, \tau)$ is defined as follows:

$$\varphi(W, \tau) = \text{tr}(Z C Z^T - 2W^T Z C Z^T + W^T Z C Z^T W + \mu \text{Dist} Y W) + \text{tr}(\tau W). \quad (30)$$

The partial derivative Equation (30) with respect to W can be shown as follows:

$$\frac{\partial \varphi(W, \tau)}{\partial W} = -2ZCZ^T + 2ZCZ^TW + \mu DistY + \tau = 0. \tag{31}$$

As the complementary relaxation of the KKT condition leads to $\tau_{ij}W_{ij} = 0$, we can obtain Equation (32).

$$(-2ZCZ^T + 2ZCZ^TW + \mu DistY + \tau)W_{ij} = 0. \tag{32}$$

The update formula for the matrix W is as follows:

$$W = W \odot \frac{2ZCZ^T}{2ZCZ^TW + \mu DistY}. \tag{33}$$

3.3. Algorithm Description

The pseudo code of RMFRASL is listed in Algorithm 1, and a flowchart of the proposed method is shown Figure 2. Moreover, the termination condition of the algorithm is that the change in objective function values between two successive iterations is less than a threshold or has reached the predefined maximum number of iterations in our work.

Algorithm 1 RMFRASL algorithm

Input: The matrix of sample dataset $Z = [z_1; z_2; \dots; z_n] \in R^{n \times d}$, parameters $\alpha > 0, \beta > 0, \lambda > 0$

1: Initialization: matrices S_0, A_0 , and W_0 are initial nonnegative matrices, $t = 0$

2: Calculate matrices U_t , and C_t according to Equations (16) and (29),

3: Repeat

4: Update S_t according to Equation (20)

5: Update A_t according to Equation (26)

6: Update W_t according to Equation (33)

7: Update U_t and C_t according to Equations (16) and (29)

8: Update $t = t + 1$

9: **Until** the objective function is convergent.

Output: Index matrix S , coefficient matrix A , and weight matrix W

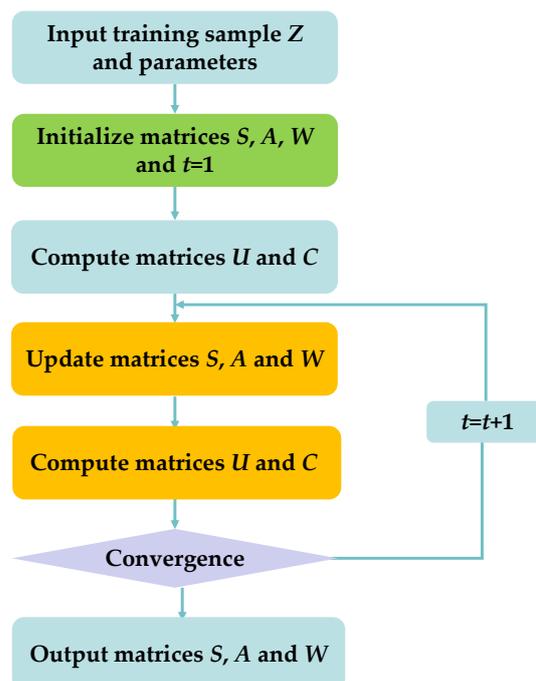


Figure 2. The flowchart of the proposed RMFRASL method.

3.4. Computational Complexity Analysis

The computational complexity can be calculated in light of the above flowchart of the proposed RMFRASL algorithm. First, the computational complexities of all variables in one iteration are shown in Table 2, where d , n , and k represent the dimensions of the original sample, the number of samples, and the number of selected features, respectively. Since $k \ll d$ and $k \ll n$, we only need to compare the size of d and n . (1) If $d > n$, S , A , and W need $O(nd^2)$, $O(kdn)$, and $O(nd^2)$, respectively. Then, the computational complexity of the RMFRASL algorithm is $O(t \times nd^2)$ where t is the number of iterations. (2) If $n \geq d$, S , A , and W need $O(dn^2)$, $O(kn^2)$, and $O(dn^2)$, respectively. Then, the computational complexity of the RMFRASL algorithm is $O(t \times dn^2)$. Finally, we can conclude that the computational complexity of the RMFRASL algorithm is $O(t \times \max(dn^2, nd^2) + dn^2)$.

Table 2. Computational complexity of each variable.

Matrix	Complexity	Matrix	Complexity
S	$O(\max(dn^2, nd^2))$	U	$O(dn^2)$
A	$O(\max(kn^2, kdn))$	C	$O(kn^2)$
W	$O(\max(nd^2, dn^2))$		

Meanwhile, the running time of the proposed RMFRASL and other compared feature selection methods was also tested when the number of experimental iterations was set to 10. From the experimental results listed in Table 3, the running time of the proposed method was lower than that of the other compared methods.

Table 3. Running time (s) of each method on different databases.

Method	AR	CMU PIE	Extended YaleB	ORL	COIL20
RNEFS [20]	11.57	15.43	13.89	1.698	4.187
USFS [21]	36.12	30.09	18.27	18.29	8.524
CNAFS [22]	11.81	14.19	11.48	4.292	5.382
NNSAFS [23]	34.83	52.95	43.70	4.261	8.775
RSR [25]	3.607	4.920	4.377	1.553	1.805
SPNFSR [34]	15.21	18.24	16.24	7.291	8.924
RMFRASL	2.250	2.687	2.493	1.190	1.504

4. Convergence Analysis

In order to analyze the convergence of the proposed RMFRASL method, we firstly establish some theorems and definitions below.

Theorem 1. For variables S , A , and W , the value of Equation (12) is nonincreasing under the update rules of Equations (20), (26) and (33).

We can refer to the convergence analysis of NMF to solve the above theorem. Therefore, some auxiliary functions are introduced below.

Definition 1. $\phi(m, m')$ is an auxiliary function of $\varphi(m)$ if $\phi(m, m')$ and $\varphi(m)$ satisfy the following conditions [35]:

$$\begin{aligned} \phi(m, m') &\geq \varphi(m) \\ \phi(m, m) &= \varphi(m) \end{aligned} \tag{34}$$

Lemma 1. If $\phi(m, m')$ is an auxiliary function of $\varphi(m)$, then it is nonincreasing under the following conditions:

$$m^{t+1} = \underset{m}{\operatorname{argmin}} \phi(m, m^t). \tag{35}$$

Proof.

$$\varphi(m^{t+1}) \leq \varphi(m^{t+1}, m^t) \leq \varphi(m^t, m^t) = \varphi(m^t). \tag{36}$$

In order to apply Lemma 1 to prove Theorem 1, we need to find suitable auxiliary functions for the variables S , A , and W . Therefore, the functions, first-order derivatives, and second-order derivatives of the three variables are specified. \square

$$\varphi_{ij}(S_{ij}) = [Z^T UZ - 2A^T S^T Z^T UZ + A^T S^T Z^T UZSA + \alpha S^T Z^T LZS + \lambda S^T S^T S - 2\lambda S^T S]_{ij}, \tag{37}$$

$$\varphi'_{ij}(S_{ij}) = 2[-Z^T UZA^T + Z^T UZSAA^T + \alpha Z^T LZS + 2\lambda SS^T S - 2\lambda S]_{ij}, \tag{38}$$

$$\varphi''_{ij}(S_{ij}) = 2[Z^T UZ]_{ii}[AA^T]_{jj} + 2\alpha[M]_{ij} + 4\lambda[SS^T]_{ij}, \tag{39}$$

$$\varphi_{ij}(A_{ij}) = [Z^T UZ - 2A^T S^T Z^T UZ + A^T S^T Z^T UZSA]_{ij}, \tag{40}$$

$$\varphi'_{ij}(A_{ij}) = 2[-S^T Z^T UZ + S^T Z^T UZSA]_{ij}, \tag{41}$$

$$\varphi''_{ij}(A_{ij}) = 2[S^T Z^T UZS^T]_{ii}, \tag{42}$$

$$\varphi_{ij}(W_{ij}) = [ZCZ^T - 2W^T ZCZ^T + W^T ZCZ^T W + \mu DistYW]_{ij}, \tag{43}$$

$$\varphi'_{ij}(W_{ij}) = [-2ZCZ^T + 2ZCZ^T W + \mu DistY]_{ij}, \tag{44}$$

$$\varphi''_{ij}(W_{ij}) = 2[ZCZ^T]_{ii}, \tag{45}$$

where $\varphi'_{ij}(\cdot)$ and $\varphi''_{ij}(\cdot)$ denote the first-order and second-order derivatives, respectively. Then, we can obtain the three lemmas described below.

Lemma 2. *The function*

$$\begin{aligned} \varphi(S_{ij}, S_{ij}^{(t)}) &= \varphi_{ij}(S_{ij}, S_{ij}^{(t)}) + \varphi'_{ij}(S_{ij})(S_{ij} - S_{ij}^{(t)}) \\ &\quad + \frac{[Z^T UZSAA^T + \alpha M^+ S + 2\lambda SS^T S]_{ij}}{S_{ij}^{(t)}} (S_{ij} - S_{ij}^{(t)})^2 \end{aligned} \tag{46}$$

is an auxiliary function of $\varphi(S_{ij})$.

Lemma 3. *The function*

$$\varphi(A_{ij}, A_{ij}^{(t)}) = \varphi_{ij}(A_{ij}, A_{ij}^{(t)}) + \varphi'_{ij}(A_{ij})(A_{ij} - A_{ij}^{(t)}) + \frac{[S^T Z^T UZSA]_{ij}}{A_{ij}^{(t)}} (A_{ij} - A_{ij}^{(t)})^2 \tag{47}$$

is an auxiliary function of $\varphi(A_{ij})$.

Lemma 4. *The function*

$$\begin{aligned} \varphi(W_{ij}, W_{ij}^{(t)}) &= \varphi_{ij}(W_{ij}, W_{ij}^{(t)}) + \varphi'_{ij}(W_{ij})(W_{ij} - W_{ij}^{(t)}) \\ &\quad + \frac{[2ZCZ^T W + \mu DistY]_{ij}}{W_{ij}^{(t)}} (W_{ij} - W_{ij}^{(t)})^2 \end{aligned} \tag{48}$$

is an auxiliary function of $\varphi(W_{ij})$.

Next, we can prove Lemma 2 using the steps below.

Proof the Lemma 2. The Taylor series expansion of $\varphi(S_{ij})$ can be given as follows:

$$\begin{aligned} \varphi_{ij}(S_{ij}) &= \varphi_{ij}(S_{ij}^{(t)}) + \varphi'_{ij}(S_{ij})(S_{ij} - S_{ij}^{(t)}) + \frac{1}{2}\varphi''_{ij}(S_{ij})(S_{ij} - S_{ij}^{(t)})^2 \\ &= \varphi_{ij}(S_{ij}^{(t)}) + \varphi'_{ij}(S_{ij})(S_{ij} - S_{ij}^{(t)}) + (2[Z^T UZ]_{ii}[AA^T]_{jj} + 2\alpha[M]_{ij} + 4\lambda[SS^T]_{ij})(S_{ij} - S_{ij}^{(t)})^2. \end{aligned} \tag{49}$$

□

We can obtain the following inequalities through a series of calculations of the matrix:

$$\begin{aligned} [Z^T UZSAA^T]_{ij} &= \sum_{l=1}^k [Z^T UZS^l]_{il}[AA^T]_{lj} \geq [Z^T UZS^t]_{ij}[AA^T]_{jj} \\ &\geq \left(\sum_{l=1}^d [Z^T UZ]_{il}S_{ij}^l \right) [AA^T]_{jj} \geq [Z^T UZ]_{ii}S_{ij}^t [AA^T]_{jj} = [Z^T UZ]_{ii}[AA^T]_{jj}S_{ij}^t, \end{aligned} \tag{50}$$

$$[M^+ S]_{ij} = \sum_{l=1}^d [M^+]_{il}S_{ij}^l \geq M_{ij}^+ S_{ij}^t \geq [M_{ij}^+ - M_{ij}^-]S_{ij}^t = [M]_{ij}S_{ij}^t, \tag{51}$$

$$[SS^T S]_{ij} = \sum_{l=1}^k [SS^T]_{il}S_{ij}^l \geq [SS^T]_{ii}S_{ij}^t. \tag{52}$$

To sum up, we can obtain the inequality

$$\frac{[Z^T UZSAA^T + \alpha M^+ S + 2\lambda SS^T S]_{ij}}{S_{ij}^{(t)}} \geq [Z^T UZS^t]_{ij}[AA^T]_{jj} + \alpha[M]_{ij} + 2\lambda[SS^T]_{ii}. \tag{53}$$

Therefore, we can obtain Equation (54).

$$\phi(S_{ij}, S_{ij}^{(t)}) \geq \varphi_{ij}(S_{ij}). \tag{54}$$

Similarly, we can prove Lemma 3 and Lemma 4. Finally, according to Lemma 1, we can obtain the update scheme of variables S , A , and W as follows:

$$\begin{aligned} S_{ij}^{t+1} &= \underset{S_{ij}}{\operatorname{argmin}} \phi(S_{ij}, S_{ij}^t) \\ &= S_{ij}^t - S_{ij}^t \frac{\varphi'(S_{ij})}{2[Z^T UZSAA^T + \alpha M^+ S + 2\lambda SS^T S]_{ij}} = S_{ij}^t \frac{[Z^T UZAA^T + \alpha M^- S + 2\lambda S]_{ij}}{[Z^T UZSAA^T + \alpha M^+ S + 2\lambda SS^T S]_{ij}}, \end{aligned} \tag{55}$$

$$A_{ij}^{t+1} = \underset{A_{ij}}{\operatorname{argmin}} \varphi(A_{ij}, A_{ij}^t) = A_{ij}^t - A_{ij}^t \frac{\varphi'_{ij}(A_{ij})}{2[S^T Z^T UZSA]_{ij}} = A_{ij}^t \frac{[S^T Z^T UZ]_{ij}}{[S^T Z^T UZSA]_{ij}}, \tag{56}$$

$$W_{ij}^{t+1} = \underset{W_{ij}}{\operatorname{argmin}} \varphi(W_{ij}, W_{ij}^t) = W_{ij}^t - W_{ij}^t \frac{\varphi'_{ij}(W_{ij})}{2[2ZCZ^T W + \mu \operatorname{Dist} Y]_{ij}} = W_{ij}^t \frac{[2ZCZ^T]_{ij}}{[2ZCZ^T W + \mu \operatorname{Dist} Y]_{ij}}. \tag{57}$$

In summary, Theorem 1 is proven.

Next, the convergence of the proposed iterative procedure in Algorithm 1 is proven. For any nonzero vectors $p \in R^m$ and $q \in R^m$, the following inequality can be obtained:

$$\|p\|_2 - \frac{\|p\|_2^2}{2\|q\|_2} \leq \|q\|_2 - \frac{\|q\|_2^2}{2\|q\|_2}. \tag{58}$$

More detailed proof of Equation (58) is similar to that in [36].

Theorem 2. According to the iterative update approach depicted in Algorithm 1, the objective function value in Equation (12) monotonically decreases in each iteration until it converges to the global optimum [36].

Proof. First, matrices U_t and C_t are denoted as the t -th iteration of matrices U and C . Then, when fixing U_t and C_t , matrices S_{t+1} , A_{t+1} , and W_{t+1} can be updated by solving the following inequality:

$$\Phi(S_{t+1}, A_{t+1}, W_{t+1}, U_t, C_t) \leq \Phi(S_t, A_t, W_t, U_t, C_t). \tag{59}$$

□

The above inequality can be changed to

$$\begin{aligned} & \left\{ \begin{aligned} & \text{tr}((Z - ZS^{(t+1)}A^{(t+1)})^T U^{(t)}(Z - ZS^{(t+1)}A^{(t+1)})) + \alpha \text{tr}(S^{(t+1)T} Z^T LZS^{(t+1)}) \\ & + \beta \text{tr}((Z^T - Z^T W^{(t+1)})^T C^{(t)}(Z^T - Z^T W^{(t+1)})) \end{aligned} \right. \\ & \leq \left\{ \begin{aligned} & \text{tr}((Z - ZS^{(t)}A^{(t)})^T U^{(t)}(Z - ZS^{(t)}A^{(t)})) + \alpha \text{tr}(S^{(t)T} Z^T LZS^{(t)}) \\ & + \beta \text{tr}((Z^T - Z^T W^{(t)})^T C^{(t)}(Z^T - Z^T W^{(t)})) \end{aligned} \right. \end{aligned} \tag{60}$$

On the basis of the definitions of matrices U_t and C_t , the following inequalities can be obtained:

$$\begin{aligned} & \left\{ \begin{aligned} & \sum_i \frac{\|(Z - ZS^{(t+1)}A^{(t+1)})^i\|_2^2}{2\|(Z - ZS^{(t)}A^{(t)})^i\|_2} + \alpha \text{tr}(S^{(t+1)T} Z^T LZS^{(t+1)}) + \beta \sum_i \frac{\|(Z^T - Z^T W^{(t+1)})^i\|_2^2}{2\|(Z^T - Z^T W^{(t)})^i\|_2} \\ & \leq \left\{ \begin{aligned} & \sum_i \frac{\|(Z - ZS^{(t)}A^{(t)})^i\|_2^2}{2\|(Z - ZS^{(t)}A^{(t)})^i\|_2} + \alpha \text{tr}(S^{(t)T} Z^T LZS^{(t)}) + \beta \sum_i \frac{\|(Z^T - Z^T W^{(t)})^i\|_2^2}{2\|(Z^T - Z^T W^{(t)})^i\|_2} \end{aligned} \right. \end{aligned} \right. \end{aligned} \tag{61}$$

$$\begin{aligned} & \left\{ \begin{aligned} & \sum_i \|(Z - ZS^{(t+1)}A^{(t+1)})^i\|_2 - (\sum_i \|(Z - ZS^{(t+1)}A^{(t+1)})^i\|_2 - \sum_i \frac{\|(Z - ZS^{(t+1)}A^{(t+1)})^i\|_2^2}{2\|(Z - ZS^{(t)}A^{(t)})^i\|_2}) \\ & + \alpha \text{tr}(S^{(t+1)T} Z^T LZS^{(t+1)}) \\ & + \beta \sum_i \|(Z^T - Z^T W^{(t+1)})^i\|_2 - \beta (\sum_i \|(Z^T - Z^T W^{(t+1)})^i\|_2 - \sum_i \frac{\|(Z^T - Z^T W^{(t+1)})^i\|_2^2}{2\|(Z^T - Z^T W^{(t)})^i\|_2}) \end{aligned} \right. \\ & \leq \left\{ \begin{aligned} & \sum_i \|(Z - ZS^{(t)}A^{(t)})^i\|_2 - (\sum_i \|(Z - ZS^{(t)}A^{(t)})^i\|_2 - \sum_i \frac{\|(Z - ZS^{(t)}A^{(t)})^i\|_2^2}{2\|(Z - ZS^{(t)}A^{(t)})^i\|_2}) \\ & + \alpha \text{tr}(S^{(t)T} Z^T LZS^{(t)}) \\ & + \beta \sum_i \|(Z^T - Z^T W^{(t)})^i\|_2 - \beta (\sum_i \|(Z^T - Z^T W^{(t)})^i\|_2 - \sum_i \frac{\|(Z^T - Z^T W^{(t)})^i\|_2^2}{2\|(Z^T - Z^T W^{(t)})^i\|_2}) \end{aligned} \right. \end{aligned} \tag{62}$$

According to Equation (58), the following inequalities can be obtained:

$$\begin{aligned} & \sum_i \|(Z - ZS^{(t+1)}A^{(t+1)})^i\|_2 - \sum_i \frac{\|(Z - ZS^{(t+1)}A^{(t+1)})^i\|_2^2}{2\|(Z - ZS^{(t)}A^{(t)})^i\|_2} \\ & \leq \sum_i \|(Z - ZS^{(t)}A^{(t)})^i\|_2 - \sum_i \frac{\|(Z - ZS^{(t)}A^{(t)})^i\|_2^2}{2\|(Z - ZS^{(t)}A^{(t)})^i\|_2} \end{aligned} \tag{63}$$

$$\begin{aligned} & \sum_i \|(Z^T - Z^T W^{(t+1)})^i\|_2 - \sum_i \frac{\|(Z^T - Z^T W^{(t+1)})^i\|_2^2}{2\|(Z^T - Z^T W^{(t)})^i\|_2} \\ & \leq \sum_i \|(Z^T - Z^T W^{(t)})^i\|_2 - \sum_i \frac{\|(Z^T - Z^T W^{(t)})^i\|_2^2}{2\|(Z^T - Z^T W^{(t)})^i\|_2} \end{aligned} \tag{64}$$

Combining Equation (61) to Equation (64), the following inequality can be obtained:

$$\begin{aligned} & \left\{ \begin{aligned} & \sum_i \|(Z - ZS^{(t+1)}A^{(t+1)})^i\|_2 + \alpha \text{tr}(S^{(t+1)T} Z^T LZS^{(t+1)}) + \beta \sum_i \|(Z^T - Z^T W^{(t+1)})^i\|_2 \\ & \leq \left\{ \begin{aligned} & \sum_i \|(Z - ZS^{(t)}A^{(t)})^i\|_2 + \alpha \text{tr}(S^{(t)T} Z^T LZS^{(t)}) + \beta \sum_i \|(Z^T - Z^T W^{(t)})^i\|_2 \end{aligned} \right. \end{aligned} \right. \end{aligned} \tag{65}$$

From Equation (65), we can clearly see that Equation (12) is monotonically decreasing during iterations and has a lower bound. Therefore, the proposed iterative optimization

algorithm will converge. Moreover, numerical experiments verified that the objective function values of our proposed method tend to converge.

5. Experiments and Analysis

In this section, we compare the proposed RMFRASL method with some advanced unsupervised feature selection methods in recent years for classification and clustering tasks, mainly consisting of LSFS [19], RNEFS [20], USFS [21], CNAFS [22], NNSAFS [23], RSR [25], and SPNFSR [34].

5.1. Description of Compared Methods

RNEFS [20] is a robust neighborhood embedding unsupervised feature selection method that employs the locally linear embedding (LLE) to compute the feature weight matrix and minimizes the reconstruction error using the L_1 -norm. USFS [21] guides feature selection by combining the feature selection matrix with soft labels learned in the low-dimensional subspace. CNAFS [22] combines adaptive graph learning with pseudo-label matrices and constructs two different manifold regularizations with pseudo-label and projection matrices. NNSAFS [23] uses residual terms from sparse regression and introduces feature graphs and maximum entropy theory to construct the manifold structure. RSR [25] uses a linear combination to represent the correlation among each feature. Meanwhile, it also imposes an L_{21} -norm on the loss function and weight matrix to select representative features. SPNFSR [34] constructs a low-rank graph to maintain the global and local structure of data and employs the L_{21} -norm to constrain the representation coefficients for feature selection. The summarization of the methods is shown in Table 4.

Table 4. The summarization of compared methods in this paper.

Method	Objective Function	L_{21} -Norm
LSFS [19] (2005)	$\frac{\sum_{ij} (z_{ri} - z_{ri})^2 W_{ij}}{\text{Var}(z_r)}$	No
RNEFS [20] (2020)	$\ (I_n - A^T)ZS\ _1 + \frac{\alpha}{4}\ S^T S - I_m\ _2^2 + \text{tr}(\beta S^T)$	No
USFS [21] (2021)	$\sum_{i=1}^n \sum_{j=1}^c \ W^T z_i - o_j\ _2^2 y_{ij} + \alpha \ ZS - Y\ _F^2 + \gamma \ S\ _{2,1} + \beta \ Y\ _2^2$	No
CNAFS [22] (2020)	$\left\{ \begin{aligned} &\ Z - ZSA\ _2^2 + \ C_n(Z^T W - Y_p)\ _2^2 + \lambda \ W\ _{2,1} + \alpha \text{tr}(Y_p^T L Y_p) \\ &+ \beta \sum_{ij} g_{ij} \log g_{ij} + \gamma \text{tr}(ALA^T) + \epsilon \text{tr}(A^T Q A) \end{aligned} \right.$	No
NNSAFS [23] (2022)	$\left\{ \begin{aligned} &\ Z^T W - Y\ _2^2 + \alpha_1 \ W\ _1 + \alpha_2 \text{Tr}(W^T L^W W) \\ &+ 2\lambda (\text{Tr}(Y^T L_S Y) + \beta \sum_{ij} g_{ij} \log g_{ij}) \end{aligned} \right.$	No
RSR [24] (2015)	$\ Z - ZS\ _{2,1} + \lambda \ S\ _{2,1}$	Yes
SPNFSR [34] (2017)	$\ Z - ZS\ _{2,1} + \alpha \text{tr}(S^T MS) + \beta \ S\ _{2,1}$	Yes
RMFRASL (2022)	$\ Z - ZSA\ _{2,1} + \alpha \text{tr}(S^T Z^T LZS) + \beta \ Z^T - Z^T W\ _{2,1}$	Yes

5.2. Description of Experimental Data

In this experiment, we performed classification and clustering tasks [37] on five publicly available datasets, including four publicly available face datasets (AR, CMU PIE, Extended YaleB, and ORL) and one object dataset (COIL20). The specific details of the five datasets are shown in Table 5, where Tr and Te represent the number of training samples and testing samples selected from each class.

Table 5. Details of five databases.

Database	Image Size	Number of Classes	Each Class Size	Tr	Te
AR	32×32	100	14	7	7
CMU PIE	32×32	68	24	12	12
Extended YaleB	32×32	38	64	20	44
ORL	32×32	40	10	7	3
COIL20	32×32	20	72	20	52

The AR [38] database consists of 4000 images corresponding to face images of 70 males and 56 females. The photos have different facial expressions, lighting conditions, and occlusions. There are 26 images for each object. Some examples of the AR database are shown in Figure 3a.

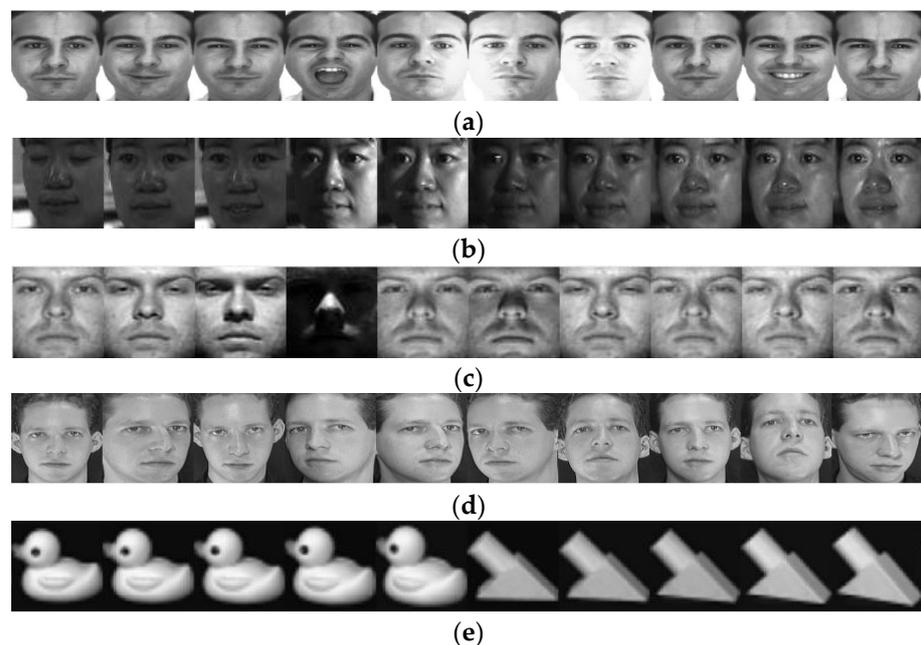


Figure 3. Selected images from the five databases: (a) AR; (b) CMU PIE; (c) Extended YaleB; (d) ORL; (e) COIL20.

The CMU PIE [39] database consists of 41,368 multi-pose, illumination, and expression face images, including face images with different pose conditions, illumination conditions, and expressions for each object. Some instances of the CMU PIE database are shown in Figure 3b.

The Extended YaleB [40] database consists of 2414 images containing 38 objects with 64 photographs taken for each object, with strict control over pose, lighting, and the shooting angle. Some face images of the Extended YaleB database are shown in Figure 3c.

The ORL [41] dataset consists of 400 face images containing 40 subjects taken at different times of day, under different lighting, and with different facial expressions. Some examples of the ORL dataset are shown in Figure 3d.

The COIL20 [42] dataset contains 1440 images, taken from 20 objects at different angles, with 72 images per object. Some images the COIL20 dataset are shown in Figure 3e.

5.3. Experimental Evaluation

For the classification task, the nearest neighbor classifier (NNC) [43] was employed to classify the selected features of each method, and the classification accuracy rate was used to evaluate the performance, which is defined as

$$Accuracyrate = \frac{N_{cor}}{N_{tot}} \times 100\%, \quad (66)$$

where N_{cor} and N_{tot} denote the number of the samples that are accurately classified and the number of all samples for classification, respectively.

For the clustering task, we applied the k -means clustering to cluster the selected features of different methods, and we adopted clustering accuracy (ACC) [44–46] to evaluate their performances. The ACC is defined as follows:

$$ACC = \frac{1}{n} \sum_{k=1}^n \vartheta(l_k, map(c_k)), \quad (67)$$

where n is total number of the test samples, l_k and c_k denote the ground truth and the corresponding clustering result of the test sample x_k , respectively, and $\vartheta(\cdot, \cdot)$ is a function which is defined as follows:

$$\vartheta(x, y) = \begin{cases} 1, & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}. \quad (68)$$

$map(\cdot)$ is a mapping function that projects each clustering label c_k to its corresponding truth label l_k using the Kuhn–Munkres algorithm. Moreover, normalized mutual information (NMI) [46] was also used to test the clustering performance, which is defined as follows:

$$NMI(U, V) = \frac{I(U, V)}{\sqrt{H(U)H(V)}}, \quad (69)$$

where U and V denotes two arbitrary variables, $H(U)$ and $H(V)$ are defined as the entropies of U and V , and $I(U, V)$ is defined as the mutual information of variables U and V .

5.4. Experimental Setting

In the experiments of this paper, we randomly selected T sample data from each database as the training sample set and the remaining C sample data as the test samples. All experiments were repeated 10 times, and the average of their classification accuracy was calculated. In order to find the best parameters for the proposed method, a grid search method [47] was used. For the three balance parameters α , β , and λ , the values of α and β were set in the range $\{0, 0.01, 0.1, 1, 10, 100, 1000, \text{ and } 10,000\}$, and λ was set as large as possible (i.e., 100,000) in this work. For the feature dimensions, we set the range from 60 to 560 with intervals of every 100 dimensions.

5.5. Analysis of Experimental Results

5.5.1. Classification Performance Analysis

In the first experiment, we tested the influence of parameter values on the classification performance of the proposed method. First, we analyzed the impact of each balance parameter (i.e., α and β) on the accuracy of the RMFRASL method. From the experimental results shown in Figure 4, we can observe that (1) when the value of parameters is set to 0, the performance of the proposed RMFRASL method is relatively low. Thus, adding these two terms is necessary for feature selection. (2) With the parameter value increasing, the performance of the method also improves, which indicates that introducing the two terms can enhance the discriminatory capability of the selected features. (3) When the performance of the proposed method reaches the optimal value, it decreases or remains stable as the value of the parameter increases. This phenomenon may be due to the parameters being set as larger values such that the importance of the second and third terms of the objective function is overemphasized, while the first term is neglected. As a result, the proposed method is unable to extract the discriminatory features. Then, we evaluated the impact of the number of iterations on the performance of the proposed

method. Figure 5 shows the classification accuracy of RMFRASL with different numbers of iterations on five datasets, where the number of iterations was set to {20, 100, 200, 300, 400, and 500}. The experimental results show that the accuracy rate increased with the number of iterations between 20 and 100 and then stabilized or decreased slightly. The best combination of parameters for the classification task is listed in Table 6.

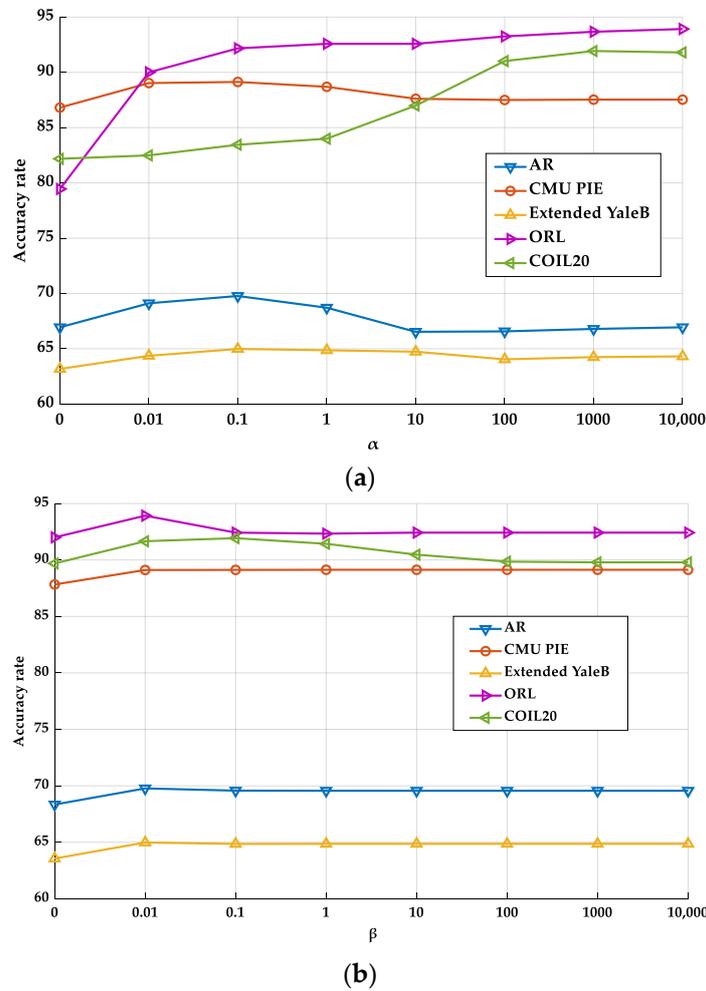


Figure 4. The classification accuracy rate of different parameter values: (a) parameter α ; (b) parameter β .

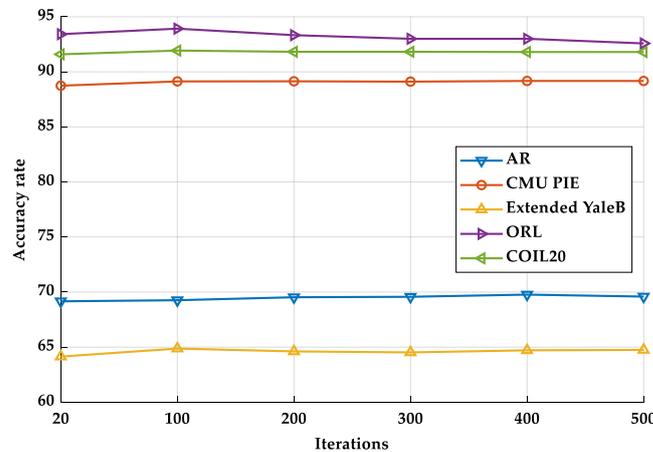


Figure 5. Accuracy rate of the proposed RMFRASL method on each database with different iteration.

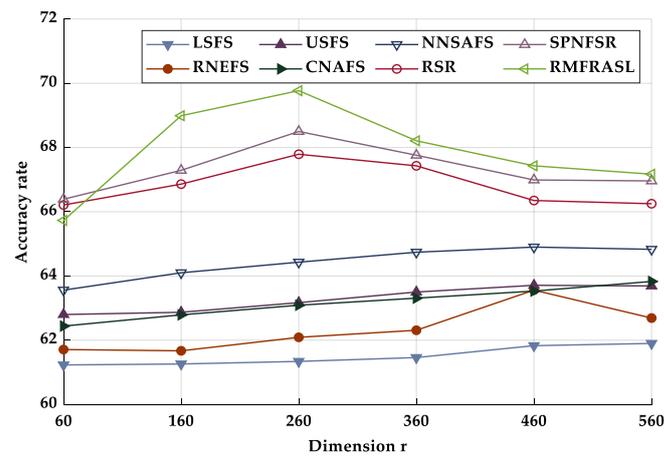
Table 6. The optimal combination of parameters and dimension on the classification task.

Databases	Accuracy Rate	Parameter $\{\alpha, \beta\}$	Dimension r
AR	69.77 ± 0.79	{0.1, 0.01}	260
CMU PIE	89.17 ± 0.80	{0.1, 1}	360
Extended YaleB	64.98 ± 0.75	{0.1, 0.01}	360
ORL	93.92 ± 1.56	{10000, 0.01}	460
COIL20	95.93 ± 1.17	{1000, 0.1}	160

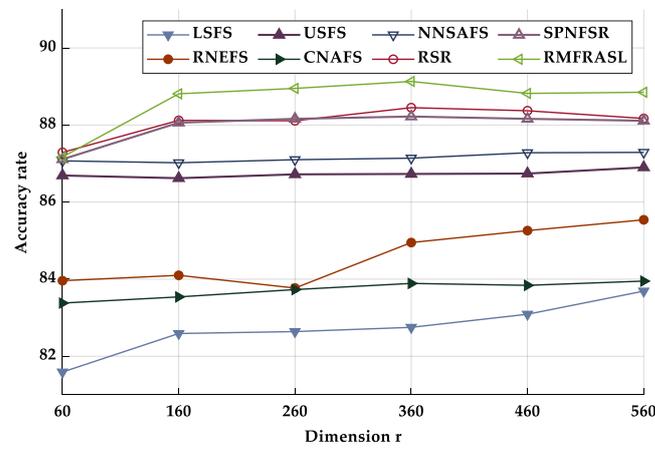
In the second experiment, we compared the performance of different feature selection methods. The classification accuracies of each unsupervised feature selection method on the five databases are given in Table 7. From the experimental results, we can find that, due to the LSFS computing the feature scores without considering the correlation of the selected feature, its performance was the worst. The RNEFS considers the feature self-representation capability; hence, its performance was better than LSFS. The performance of USFS was superior to RNEFS and LSFS since it incorporates soft label learning and constructs the regression relationships between the soft label matrix and features to guides feature selection. CNAFS and NNSAFS combine the geometric structure of the data or features and can select more discriminatory features to obtain slightly higher classification accuracy than RNEFS and USFS. Since RSR and SPNFSR exploit the feature self-representation capability and define the reconstruction error with L_{21} -norm to significantly improve robustness to noise, their performances are better than other compared methods. However, SPNFSR outperforms RSR due to it considering the geometric structure of features. By fully considering the feature self-representation, geometric structure, adaptive graph learning, and robustness together, our proposed RMFRASL achieved the best results among all compared methods. The classification accuracy rate curves of different feature selection methods with different numbers of selected features are shown in Figure 6. First, with the increase in the number of selected features, the classification accuracy rates of all methods improved. Then, the classification accuracy rates of most methods began to stabilize after achieving their best performances.

Table 7. The best accuracy rate and STD of different algorithms on the five databases.

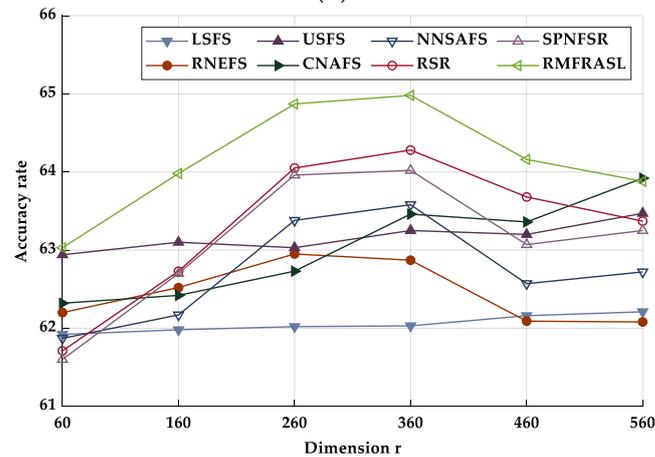
Method	AR	CMU PIE	Extended YaleB	ORL	COIL20
LSFS	61.90 ± 2.35 (560)	83.69 ± 1.36 (560)	61.21 ± 1.56 (560)	91.42 ± 3.07 (560)	90.34 ± 1.48 (560)
RNEFS	63.56 ± 1.48 (460)	85.54 ± 1.29 (560)	62.95 ± 0.75 (260)	92.33 ± 2.00 (460)	91.68 ± 0.93 (560)
USFS	63.71 ± 2.71 (460)	86.90 ± 1.17 (560)	63.57 ± 1.38 (560)	92.67 ± 1.93 (560)	92.66 ± 1.39 (60)
CNAFS	63.83 ± 1.66 (560)	83.95 ± 1.56 (560)	63.92 ± 1.12 (560)	93.13 ± 1.85 (560)	94.29 ± 1.22 (160)
NNSAFS	64.90 ± 2.12 (460)	87.29 ± 0.64 (560)	63.58 ± 1.39 (360)	92.17 ± 1.81 (560)	93.56 ± 1.32 (160)
RSR	67.79 ± 2.03 (260)	88.45 ± 0.90 (360)	64.28 ± 1.10 (360)	93.33 ± 1.47 (560)	94.88 ± 1.15 (560)
SPNFSR	68.50 ± 0.91 (260)	88.22 ± 1.02 (360)	64.02 ± 1.95 (360)	92.83 ± 1.81 (560)	94.21 ± 1.42 (460)
RMFRASL	69.77 ± 0.97 (260)	89.17 ± 0.80 (360)	64.98 ± 0.75 (360)	93.92 ± 1.56 (460)	95.93 ± 1.17 (160)



(a)



(b)



(c)

Figure 6. Cont.

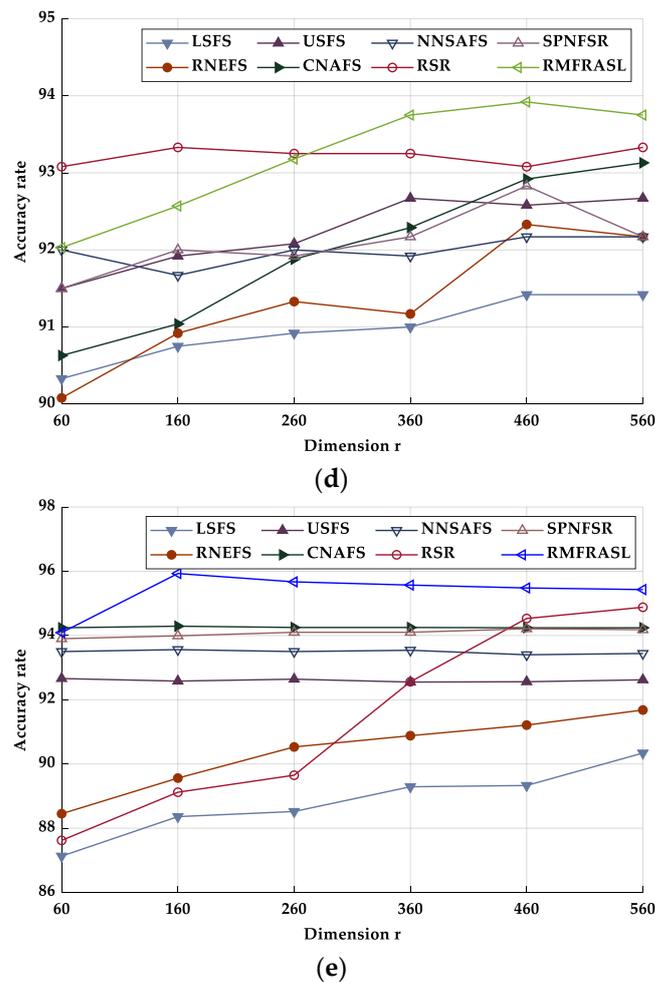


Figure 6. Accuracy rate of each database in different dimensions: (a) AR; (b) CMU PIE; (c) Extended YaleB; (d) ORL; (e) COIL20.

5.5.2. Clustering Performance Analysis

In this subsection, the *k*-means method was used for the clustering task. To reduce the randomness of initialization in *k*-means, we report the mean of 10 experiments. First, the effect of different parameters on the performance of the proposed method was tested. The experimental results of each balance parameter (i.e., α and β) on the clustering of the RMFRASL method are shown in Figure 7. The results show that the influences of the balance parameters on the performance of our method were similar to those for the classification experiments. Then, after setting the optimal parameter settings for RMFRASL in Table 8, the clustering results on ACC and NMI were as shown in Tables 9 and 10, consistent with the classification experiments.

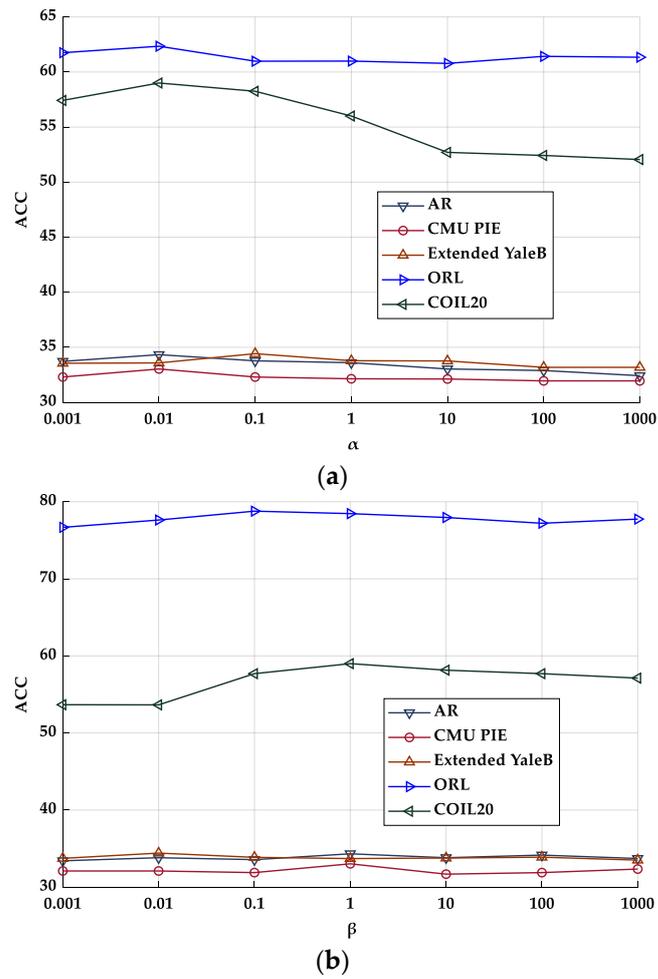


Figure 7. The clustering accuracy (ACC) of different parameter values: (a) parameter α ; (b) parameter β .

Table 8. The optimal combination of parameters and dimension on the clustering task.

Databases	ACC%	NMI%	Parameter $\{\alpha, \beta\}$	Dimension r
AR	34.33 \pm 1.33	65.56 \pm 1.04	{0.01, 1}	380
CMU PIE	33.03 \pm 0.90	56.66 \pm 0.97	{0.01, 1}	480
Extended YaleB	34.42 \pm 1.16	57.26 \pm 1.87	{0.1, 0.01}	350
ORL	61.75 \pm 2.42	78.78 \pm 1.24	{0.01, 0.1}	270
COIL20	59.90 \pm 1.56	72.56 \pm 1.07	{0.01, 1}	240

Table 9. The best ACC% and STD of different algorithms on the five databases.

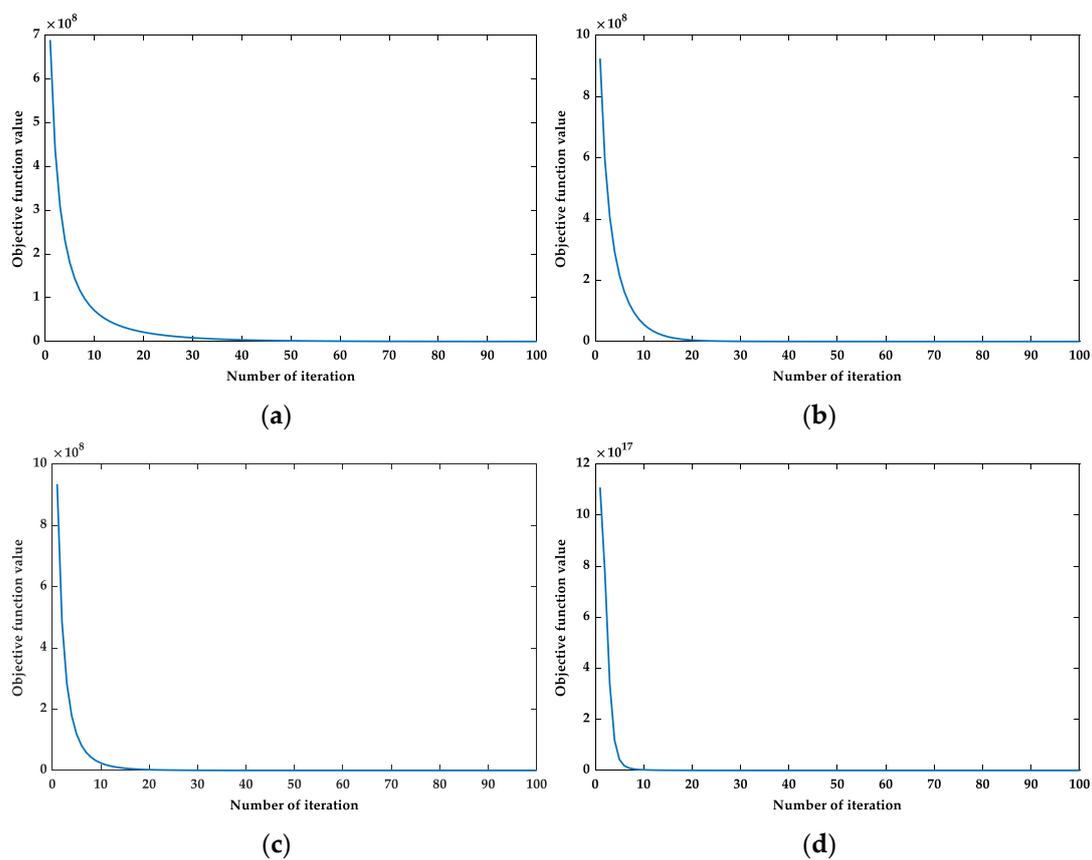
Method	AR	CMU PIE	Extended YaleB	ORL	COIL20
LSFS	29.89 \pm 1.37	26.88 \pm 0.94	32.82 \pm 0.93	55.57 \pm 2.65	53.42 \pm 2.77
RNEFS	30.37 \pm 1.34	28.13 \pm 0.59	33.99 \pm 0.60	59.79 \pm 2.22	55.67 \pm 1.99
USFS	31.00 \pm 1.17	28.50 \pm 1.59	33.39 \pm 0.90	57.79 \pm 2.15	58.73 \pm 3.66
CNAFS	31.10 \pm 1.67	30.38 \pm 1.38	34.33 \pm 0.77	57.53 \pm 0.00	58.27 \pm 3.67
NNSAFS	31.10 \pm 1.70	31.32 \pm 1.43	33.76 \pm 0.62	57.79 \pm 5.01	58.55 \pm 3.25
RSR	32.91 \pm 1.22	32.17 \pm 1.42	33.52 \pm 0.80	58.79 \pm 3.52	55.77 \pm 1.65
SPNFSR	33.13 \pm 0.80	32.07 \pm 1.01	33.57 \pm 0.91	59.57 \pm 2.85	57.10 \pm 3.46
RMFRASL	34.33 \pm 1.33	33.03 \pm 0.90	34.42 \pm 1.16	61.75 \pm 2.42	59.90 \pm 1.56

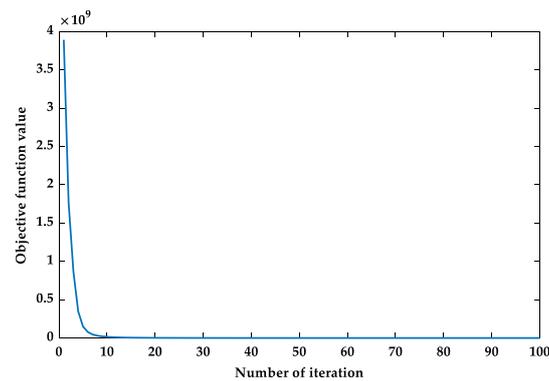
Table 10. The best NMI% and STD of different algorithms on the five databases.

Method	AR	CMU PIE	Extended YaleB	ORL	COIL20
LSFS	63.00 ± 0.75	52.81 ± 0.51	55.13 ± 1.40	75.96 ± 1.18	64.92 ± 1.71
RNEFS	62.87 ± 0.77	53.64 ± 0.55	55.51 ± 0.82	78.13 ± 0.82	65.55 ± 1.42
USFS	63.04 ± 1.14	54.01 ± 1.10	56.10 ± 0.91	76.71 ± 1.61	67.92 ± 1.69
CNAFS	63.03 ± 1.01	54.63 ± 0.82	56.34 ± 1.37	75.90 ± 0.00	68.66 ± 2.15
NNSAFS	63.99 ± 1.17	53.78 ± 1.25	56.26 ± 1.38	72.26 ± 2.43	69.19 ± 2.38
RSR	63.53 ± 0.91	54.98 ± 1.06	56.66 ± 1.69	76.09 ± 1.47	66.60 ± 1.91
SPNFSR	64.68 ± 0.66	55.18 ± 0.86	57.18 ± 1.05	78.68 ± 1.41	70.89 ± 1.53
RMFRASL	65.56 ± 1.04	56.66 ± 0.97	57.26 ± 1.87	78.78 ± 1.24	72.56 ± 1.07

5.5.3. Convergence Analysis

In this subsection, we set the parameters to their respective optimal combinations in each of the five databases; then, iterative convergence analysis was performed. Figure 8 shows that RMFRASL achieved convergence with a small number of iterations on each database. Moreover, we provide the number of iterations needed by the compared methods to achieve convergence in Table 11. According to this table, the number of iterations of the proposed RMFRASL method was lower than that of the other methods in most cases.

**Figure 8.** Cont.



(e)

Figure 8. The convergence of the RMFRASL algorithm: (a) AR; (b) CMU PIE; (c) Extended YaleB; (d) ORL; (e) COIL20.

Table 11. The number of iterations of different algorithms on the five databases.

Method	AR	CMU PIE	Extended YaleB	ORL	COIL20
RNEFS	15	23	14	12	15
USFS	13	30	17	15	20
CNAFS	409	510	398	24	30
NNSAFS	15	45	5	5	24
SPNFSS	396	410	412	462	490
RMFRASL	37	21	20	8	11

6. Discussion and Future Work

Feature selection is an effective dimensionality reduction technique, and previous methods did not consider the performance of adaptive graph learning and robustness simultaneously. To solve the above problems, this paper proposed a robust adaptive structure learning-based matrix factorization (RMFRASL) method, which integrates feature selection and adaptive graph learning into a unified framework. A large number of experiments showed that RMFRASL could achieve better performance than previous methods. Deep learning and transfer learning show excellent promise in feature learning tasks. In future work, it may be possible to introduce deep learning and transfer learning schemes into feature selection to develop a more efficient feature selection method.

Author Contributions: Data curation, S.L., L.H., P.L. and Z.L.; formal analysis, S.L., P.L. and Z.L.; methodology, S.L., L.H. and Y.Y.; resources, S.L. and Z.L.; supervision, L.H. and Y.Y.; writing—original draft, S.L., L.H. and J.W.; writing—review and editing, Y.Y. and J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China (Nos. 62062040, 62006174, and 61967010), the Outstanding Youth Project of Jiangxi Natural Science Foundation (No. 20212ACB212003), the Jiangxi Province Key Subject Academic and Technical Leader Funding Project (No. 20212BCJ23017), and the Fund of Jilin Provincial Science and Technology Department (No. 20210101187JC).

Data Availability Statement: The data were derived from public domain resources.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Cheng, X.; Fang, L.; Hong, X.; Yang, L. Exploiting mobile big data: Sources, features, and applications. *IEEE Netw.* **2017**, *31*, 72–79. [\[CrossRef\]](#)
- Cheng, X.; Fang, L.; Yang, L.; Cui, S. Mobile big data: The fuel for data-driven wireless. *IEEE Internet Things J.* **2017**, *4*, 1489–1516. [\[CrossRef\]](#)

3. Lv, Z.; Lou, R.; Li, J.; Singh, A.K.; Song, H. Big data analytics for 6G-enabled massive internet of things. *IEEE Internet Things J.* **2021**, *8*, 5350–5359. [[CrossRef](#)]
4. Tang, C.; Zheng, X.; Liu, X.; Zhang, W.; Zhang, J.; Xiong, J.; Wang, L. Cross-view locality preserved diversity and consensus learning for multi-view unsupervised feature selection. *IEEE Trans. Knowl. Data Eng.* **2021**, *34*, 4705–4716. [[CrossRef](#)]
5. Jin, J.; Xiao, R.; Daly, I.; Miao, Y.; Wang, X.; Cichocki, A. Internal feature selection method of CSP based on L1-norm and Dempster-Shafer theory. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4814–4825. [[CrossRef](#)] [[PubMed](#)]
6. Zhu, F.; Ning, Y.; Chen, X.; Zhao, Y.; Gang, Y. On removing potential redundant constraints for SVOR learning. *Appl. Soft Comput.* **2021**, *102*, 106941. [[CrossRef](#)]
7. Li, Z.; Nie, F.; Bian, J.; Wu, D.; Li, X. Sparse pca via L2,p-norm regularization for unsupervised feature selection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, 1–8. [[CrossRef](#)] [[PubMed](#)]
8. Luo, F.; Zou, Z.; Liu, J.; Lin, Z. Dimensionality reduction and classification of hyperspectral image via multistructure unified discriminative embedding. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–16. [[CrossRef](#)]
9. Lee, D.D.; Seung, H.S. Learning the parts of objects by non-negative matrix factorization. *Nature* **1999**, *401*, 788–791. [[CrossRef](#)]
10. Awotunde, J.B.; Chakraborty, C.; Adeniyi, A.E. Intrusion detection in industrial internet of things network-based on deep learning model with rule-based feature selection. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 7154587. [[CrossRef](#)]
11. Aminanto, M.E.; Choi, R.; Tanuwidjaja, H.C.; Yoo, P.D.; Kim, K. Deep abstraction and weighted feature selection for Wi-Fi impersonation detection. *IEEE Trans. Inf. Secur.* **2017**, *13*, 621–636. [[CrossRef](#)]
12. Zhang, F.; Li, W.; Feng, Z. Data driven feature selection for machine learning algorithms in computer vision. *IEEE Internet Things J.* **2018**, *5*, 4262–4272. [[CrossRef](#)]
13. Qi, M.; Wang, T.; Liu, F.; Zhang, B.; Wang, J.; Yi, Y. Unsupervised feature selection by regularized matrix factorization. *Neurocomputing* **2018**, *273*, 593–610. [[CrossRef](#)]
14. Zhai, Y.; Ong, Y.S.; Tsang, I.W. The emerging “big dimensionality”. *IEEE Comput. Intell. Mag.* **2014**, *9*, 14–26. [[CrossRef](#)]
15. Bermejo, P.; Gámez, J.A.; Puerta, J.M. Speeding up incremental wrapper feature subset selection with Naive Bayes classifier. *Knowl.-Based Syst.* **2014**, *55*, 140–147. [[CrossRef](#)]
16. Cheng, L.; Wang, Y.; Liu, X.; Li, B. Outlier detection ensemble with embedded feature selection. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 3503–3512.
17. Lu, Q.; Li, X.; Dong, Y. Structure preserving unsupervised feature selection. *Neurocomputing* **2018**, *301*, 36–45. [[CrossRef](#)]
18. Li, X.; Zhang, H.; Zhang, R.; Liu, Y.; Nie, F. Generalized uncorrelated regression with adaptive graph for unsupervised feature selection. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *30*, 1587–1595. [[CrossRef](#)]
19. He, X.; Cai, D.; Niyogi, P. Laplacian score for feature selection. *Adv. Neural Inf. Process. Syst.* **2005**, *18*, 1–8.
20. Liu, Y.; Ye, D.; Li, W.; Wang, H.; Gao, Y. Robust neighborhood embedding for unsupervised feature selection. *Knowl.-Based Syst.* **2020**, *193*, 105462. [[CrossRef](#)]
21. Wang, F.; Zhu, L.; Li, J.; Chen, H.; Zhang, H. Unsupervised soft-label feature selection. *Knowl.-Based Syst.* **2021**, *219*, 106847. [[CrossRef](#)]
22. Yuan, A.; You, M.; He, D.; Li, X. Convex non-negative matrix factorization with adaptive graph for unsupervised feature selection. *IEEE Trans. Cybern.* **2020**, *52*, 5522–5534. [[CrossRef](#)] [[PubMed](#)]
23. Shang, R.; Zhang, W.; Lu, M.; Jiao, L.; Li, Y. Feature selection based on non-negative spectral feature learning and adaptive rank constraint. *Knowl.-Based Syst.* **2022**, *236*, 107749. [[CrossRef](#)]
24. Zhao, H.; Li, Q.; Wang, Z.; Nie, F. Joint Adaptive Graph Learning and Discriminative Analysis for Unsupervised Feature Selection. *Cogn. Comput.* **2022**, *14*, 1211–1221. [[CrossRef](#)]
25. Zhu, P.; Zuo, W.; Zhang, L.; Hu, Q.; Shiu, S.C. Unsupervised feature selection by regularized self-representation. *Pattern Recognit.* **2015**, *48*, 438–446. [[CrossRef](#)]
26. Shi, L.; Du, L.; Shen, Y.D. Robust spectral learning for unsupervised feature selection. In Proceedings of the 2014 IEEE International Conference on Data Mining, Shenzhen, China, 14–17 December 2014; pp. 977–982.
27. Du, S.; Ma, Y.; Li, S.; Ma, Y. Robust unsupervised feature selection via matrix factorization. *Neurocomputing* **2017**, *241*, 115–127. [[CrossRef](#)]
28. Miao, J.; Yang, T.; Sun, L.; Fei, X.; Niu, L.; Shi, Y. Graph regularized locally linear embedding for unsupervised feature selection. *Pattern Recognit.* **2022**, *122*, 108299. [[CrossRef](#)]
29. Hou, C.; Nie, F.; Li, X.; Yi, D.; Wu, Y. Joint embedding learning and sparse regression: A framework for unsupervised feature selection. *IEEE Trans. Cybern.* **2013**, *44*, 793–804.
30. Wang, S.; Pedrycz, W.; Zhu, Q.; Zhu, W. Subspace learning for unsupervised feature selection via matrix factorization. *Pattern Recognit.* **2015**, *48*, 10–19. [[CrossRef](#)]
31. Zhu, F.; Gao, J.; Yang, J.; Ye, N. Neighborhood linear discriminant analysis. *Pattern Recognit.* **2022**, *123*, 108422. [[CrossRef](#)]
32. Cai, D.; He, X.; Han, J.; Huang, T.S. Graph regularized nonnegative matrix factorization for data representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *33*, 1548–1560.
33. Dreves, A.; Facchinei, F.; Kanzow, C.; Sagratella, S. On the solution of the KKT conditions of generalized Nash equilibrium problems. *SIAM J. Optim.* **2011**, *21*, 1082–1108. [[CrossRef](#)]
34. Zhou, W.; Wu, C.; Yi, Y.; Luo, G. Structure preserving non-negative feature self-representation for unsupervised feature selection. *IEEE Access* **2017**, *5*, 8792–8803. [[CrossRef](#)]

35. Yi, Y.; Zhou, W.; Liu, Q.; Luo, G.; Wang, J.; Fang, Y.; Zheng, C. Ordinal preserving matrix factorization for unsupervised feature selection. *Signal Process. Image Commun.* **2018**, *67*, 118–131. [[CrossRef](#)]
36. Yang, Y.; Shen, H.T.; Ma, Z.; Huang, Z.; Zhou, F. $\ell_2, 1$ -norm regularized discriminative feature selection for unsupervised learning. *Int. Jt. Conf. Artif. Intell.* **2011**, *22*, 1589–1594.
37. Yi, Y.; Wang, J.; Zhou, W.; Fang, Y.; Kong, J.; Lu, Y. Joint graph optimization and projection learning for dimensionality reduction. *Pattern Recognit.* **2019**, *92*, 258–273. [[CrossRef](#)]
38. Martinez, A.; Benavente, R. *The AR Face Database: Cvc Technical Report*; Universitat Autònoma de Barcelona: Bellaterra, Spain, 1998; Volume 24.
39. Sim, T.; Baker, S.; Bsat, M. The CMU pose, illumination, and expression (PIE) database. In Proceedings of the Fifth IEEE International Conference on Automatic Face Gesture Recognition, Washington, DC, USA, 21–21 May 2002; pp. 53–58.
40. Georghiades, A.S.; Belhumeur, P.N.; Kriegman, D.J. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 643–660. [[CrossRef](#)]
41. Samaria, F.; Harter, A. Parameterisation of a Stochastic Model for Human Face Identification. In Proceedings of the 1994 IEEE Workshop on Applications of Computer Vision, Sarasota, FL, USA, 5–7 December 1994; pp. 138–142.
42. Nene, S.; Nayar, S.; Murase, H. Columbia object image library (COIL-20). In *Technical Report, CUICS-005-96*; Columbia University: New York, NY, USA, 1996.
43. Dai, J.; Chen, Y.; Yi, Y.; Bao, J.; Wang, L.; Zhou, W.; Lei, G. Unsupervised feature selection with ordinal preserving self-representation. *IEEE Access* **2018**, *6*, 67446–67458. [[CrossRef](#)]
44. Tang, C.; Li, Z.; Wang, J.; Liu, X.; Zhang, W.; Zhu, E. Unified One-step Multi-view Spectral Clustering. *IEEE Trans. Knowl. Data Eng.* **2022**, 1–11. [[CrossRef](#)]
45. Tang, C.; Zhu, X.; Liu, X.; Li, M.; Wang, P.; Zhang, C.; Wang, L. Learning a joint affinity graph for multiview subspace clustering. *IEEE Trans. Multimed.* **2018**, *21*, 1724–1736. [[CrossRef](#)]
46. Li, Z.; Tang, C.; Liu, X.; Zheng, X.; Zhang, W.; Zhu, E. Consensus graph learning for multi-view clustering. *IEEE Trans. Multimed.* **2021**, *24*, 2461–2472. [[CrossRef](#)]
47. Zhu, F.; Gao, J.; Xu, C.; Yang, J.; Tao, D. On selecting effective patterns for fast support vector regression training. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *29*, 3610–3622. [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.