information

Article

SFS-AGGL: Semi-Supervised Feature Selection Integrating Adaptive Graph with Global and Local Information

Yugen Yi [1], Haoming Zhang [1], Ningyi Zhang [1], Wei Zhou [2,*], Xiaomei Huang [1], Gengsheng Xie [1] and Caixia Zheng [3,*]

[1] School of Software, Jiangxi Normal University, Nanchang 330022, China; yiyg510@jxnu.edu.cn (Y.Y.); 202140100838@jxnu.edu.cn (H.Z.); zny@jxnu.edu.cn (N.Z.); 002218@jxnu.edu.cn (X.H.); xiegengsheng@jxnu.edu.cn (G.X.)
[2] College of Computer Science, Shenyang Aerospace University, Shenyang 110136, China
[3] College of Information Science and Technology, Northeast Normal University, Changchun 130117, China
* Correspondence: wei.zhou@sau.edu.cn (W.Z.); zhengcx789@nenu.edu.cn (C.Z.)

Abstract: As the feature dimension of data continues to expand, the task of selecting an optimal subset of features from a pool of limited labeled data and extensive unlabeled data becomes more and more challenging. In recent years, some semi-supervised feature selection methods (SSFS) have been proposed to select a subset of features, but they still have some drawbacks limiting their performance, for e.g., many SSFS methods underutilize the structural distribution information available within labeled and unlabeled data. To address this issue, we proposed a semi-supervised feature selection method based on an adaptive graph with global and local constraints (SFS-AGGL) in this paper. Specifically, we first designed an adaptive graph learning mechanism that can consider both the global and local information of samples to effectively learn and retain the geometric structural information of the original dataset. Secondly, we constructed a label propagation technique integrated with the adaptive graph learning in SFS-AGGL to fully utilize the structural distribution information of both labeled and unlabeled data. The proposed SFS-AGGL method is validated through classification and clustering tasks across various datasets. The experimental results demonstrate its superiority over existing benchmark methods, particularly in terms of clustering performance.

Keywords: semi-supervised learning; feature selection; adaptive graph learning; sparse regularization; label propagation

check for updates

Citation: Yi, Y.; Zhang, H.; Zhang, N.; Zhou, W.; Huang, X.; Xie, G.; Zheng, C. SFS-AGGL: Semi-Supervised Feature Selection Integrating Adaptive Graph with Global and Local Information. *Information* **2024**, *15*, 57. https://doi.org/10.3390/info15010057

Academic Editor: Roberto Posenato

Received: 1 November 2023
Revised: 30 December 2023
Accepted: 14 January 2024
Published: 17 January 2024

1. Introduction

High-dimensional data can describe real-world things more realistically and effectively. However, these data might include vast redundant and irrelevant information. If we process these data directly, it not only consumes a large amount of storage space and computational resources but also leads to the performance degradation of existing models [1]. Therefore, it is necessary to mine the potential relationships between the data to select and learn useful feature information.

Feature representation learning (FRL) is one of the most effective methods of learning useful feature information. Among the existing FRL methods, feature extraction (FE) [2] and feature selection (FS) [3] are two representative methods. FE aims to map the original high-dimensional feature space to a low-dimensional subspace according to some pre-defined criteria [4]. FS selects an optimal feature subset from the original feature set based on evaluation metrics [5]. In comparison, FS is more interpretable than FE since it can remove irrelevant and redundant features from the original features and retain a small number of relevant features. Therefore, FS is widely used in image classification, bioinformatics, face recognition, medical image analysis, natural language processing, and other fields [6].

FS methods can be divided into unsupervised feature selection (UFS), supervised feature selection (SFS), and semi-supervised feature selection (SSFS). UFS methods can achieve feature selection by only using unlabeled data; they have received widespread

attention since they do not require any labeled data. However, a lack of label-guided learning in UFS methods will lead to poor performance on practical application tasks [7]. Thus, SFS methods have been devised to leverage the label information of the sample to guide the process of FS, enhancing the distinctiveness and consequently of selected features, improving the performance of the classification and clustering [8]. However, obtaining ample labeled data is very challenging and time consuming in practical situations. For this reason, many SSFS methods have been proposed in the past decades. SSFS methods employ semi-supervised learning (SSL) to leverage the information of limited labeled data and a substantial volume of unlabeled data, enhancing the feature selection ability of the model [9]. The existing SSFS methods can be classified as filtered, wrapped, and embedded methods [10]. Filtered methods first evaluate each feature based on the principles of statistical or information theory and then perform the process of FS in terms of the calculated weights. A major benefit of filtering methods is that they are more applicable to large-scale datasets since they have high speed and computational efficiency. However, filtered methods may ignore the amount of redundant information generated by the combination of multiple features [11]. Thus, some wrapped approaches have been proposed to exploit the interrelationship of features to mine the best combination of features. However, these approaches have high computational complexity. This makes them unsuitable for processing large-scale data [12]. In contrast to the above-mentioned methods, embedded methods combine FS and model training together. That is, FS is automatically executed during the process of model training, which makes embedded methods improve the efficiency of FS by reducing runtime [13]. Therefore, they have become mainstream and widely used in various scenarios.

In recent years, several semi-supervised embedded feature selection (SSEFS) methods have emerged. For example, Zhao et al. [14] introduced an SSFS method using both labeled and unlabeled data. Recognizing sparse regularization is an effective strategy for selecting useful features and reducing feature representation dimensions [15]. Chen et al. [16] introduced an efficient semi-supervised feature selection (ESFS) method. ESFS first combines SSL and sparse regularization to obtain feature subsets. Then, it uses probability matrices of unlabeled data to measure feature relevance to the class, aiming to identify the globally optimal feature subset. Least squares regression (LSR) with complete statistical theory can handle noisy data effectively and thus improve computational efficiency [17]. Therefore, Chang et al. [18] proposed a convex sparse feature selection (CSFS) method based on LSR, which employs the convex optimization theory to fit samples and predict labels to select the most critical features using constraint terms. Chen et al. [19] contended that LSR-based feature selection lacks interpretability and struggles to identify a global sparse solution. Hence, they proposed an embedded SSFS method based on rescaled linear regression, which exploits the L_{21} norm to obtain both global and sparse solutions. Moreover, they also introduced a sparse regularization with an implicit L2p norm to obtain sparser and more interpretable solutions [20]. Therefore, this approach effectively constrains regression coefficients, achieving feature ranking. Besides, Liu et al. [21] combined sparse features and considered the correlation of samples in the original high- and low-dimensional spaces to improve the performance of feature learning. Despite the good results achieved by the sparse model-based methods, there are still some problems.

The first problem is that most of the methods do not consider constructing graphs to better preserve the geometric structural information of the data during the FS process. Initially, KNN was adopted by some FS methods to construct graphs based on Euclidean distances [22–25]. To minimize the influence of the redundant features and noise in the original high-dimensional data on the constructing graph process, Chen et al. [26] employed local discriminant analysis (LDA) to map the data from high-dimensional space to low-dimensional space. Subsequently, numerous graph construction methods based on data correlation have been presented, including L_1 graph [27], low-rank representation (LRR) [28], local structure learning [29], and sparse subspace clustering (SSC) [30], to construct high-quality graphs. The above-mentioned graph construction methods are

integrated into FS models, proposing a large number of improvements for feature selection [31–37]. However, the processes of adaptive graph construction and FS in the above-mentioned methods are independent of each other, so the influence of graph construction on the FS process is limited. To this end, some methods have been constructed to unify adaptive graph learning (AGL) and FS into a single framework [38–41].

The second problem is that the spatial distribution of the sample label information is not sufficiently considered, resulting in the weak discriminative ability of the selected features, which further leads to poor classification or clustering performance. To alleviate this issue, label propagation (LP) has been incorporated into the FS methods [42–44]. However, since LP is also a graph-learning-based algorithm, the quality of the learned graph affects the performance to some extent. Therefore, numerous methods have emerged to merge AGL and LP [45–48]. However, these methods still have the following limitations: (1) the process of AGL is based on the original data; (2) the process of adaptive graph construction only considers the local structure or the global structure. Therefore, these methods are inevitably affected by high-dimensional features or noisy data.

To address the above-mentioned issues, this study develops a novel SSFS framework, SFS-AGGL, which integrates FS, AGL, and LP to capture both global and local data structural information for selecting an optimal feature subset with maximum discrimination and minimum redundancy. In AGL, global and local constraints are imposed on the construction coefficient obtained by the self-representation of low-dimensionally selected features. Meanwhile, the similarity matrix obtained by AGL is integrated into LP, enhancing label prediction performance. To improve the discriminative ability of the selected features, the predicted label matrix is introduced into the sparse feature selection (SFS) process. SFS is performed through the mutual promotion of the three models. The framework of the proposed SFS-AGGL is shown in Figure 1.

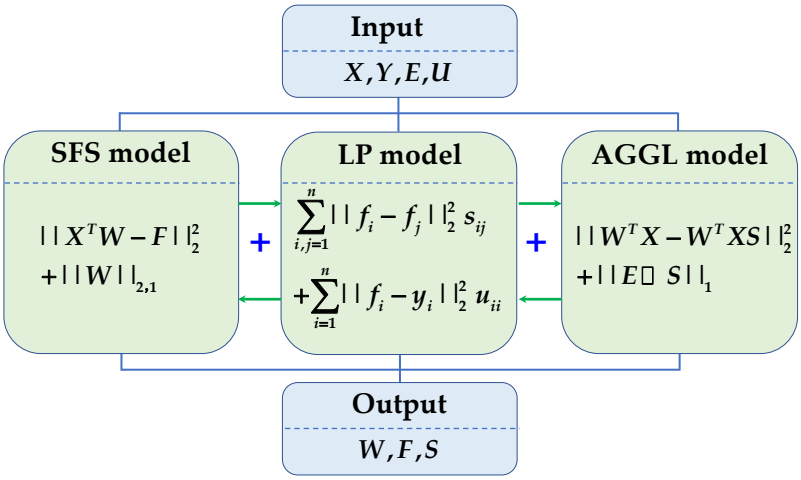


Figure 1. The illustration of the SFS-AGGL framework.

The primary contributions of this paper are as follows:

(1) An efficient SSFS framework is proposed by combining the advantages of FS, adaptive learning, and LP.

(2) An adaptive learning strategy based on low-dimensional features is designed to counteract the influence of high-dimensional features or noise data. Moreover, global and local constraints are introduced.

(3) An LP based on an adaptive similarity matrix is introduced to enhance label prediction accuracy.

(4) Comprehensive experiments conducted on multiple real datasets demonstrate that the proposed SFS-AGGL method surpasses existing representative methods in classification and clustering tasks.

The rest of this paper is organized as follows: Section 2 describes some related work; Section 3 outlines the details of the proposed method and the iterative minimization strategy employed to optimize the objective function; Section 4 introduces the experimental setup and provides a comprehensive analysis of the obtained results, including comparisons with eight state-of-the-art methods on five real datasets; and Section 5 provides a summary of our work in this paper.

2. Related Work

In this section, we have first provided some commonly used notations. Then, sparse representation and graph construction methods are introduced. Finally, some semi-supervised feature selection methods are briefly reviewed.

2.1. Notations

Let $X = [X_l, X_u] = [x_1, \cdots, x_l, x_{l+1}, \cdots, x_{l+1+u}] \in R^{m \times n}$ denote the training samples, where $x_i \in R^m$ denotes the i-th sample. $Y = [Y_l\ Y_u]^T \in R^{c \times n}$ is the label matrix, and Y_l denotes the true label of the labeled sample. If the sample x_i belongs to the class j, then its corresponding class label is $Y_{ij} = 1$; otherwise, $Y_{ij} = 0$. Y_u denotes the true label of the unlabeled sample. Since Y_u is unknown during the training process, it is set as a 0 matrix during training [49]. The main symbols in this paper are presented in Table 1.

Table 1. Definition of the main symbols in this paper.

Notation	Description	Notation	Description
$X \in R^{d \times n}$	Sample matrix	$0 \in R^{u \times c}$	Zero matrix
$X_l \in R^{d \times l}$	Labeled sample matrix	d	Sample dimension
$X_u \in R^{d \times (n-l)}$	Unlabeled sample matrix	n	Sample size
$Y \in R^{n \times c}$	Label matrix	k	Number of selected features
$F \in R^{n \times c}$	Predictive labeling matrix	c	Number of categories
$S \in R^{n \times n}$	Weighting matrix	l	Number of label samples
$E \in R^{n \times n}$	Local adaptation matrix	$+\infty$	Infinitely large numbers
$W \in R^{d \times c}$	Weighting matrix	$\odot$	Matrix dot product
$I \in R^{u \times u}$	Unit matrix	$tr(\cdot)$	Traces of matrix

Common matrix norms include L_1, L_2, F, and L_{21} norms. Their detailed definitions are as follows:

$$||B||_1 = \sum_{i=1}^{m} \sum_{j=1}^{n} |b_{ij}| \tag{1}$$

$$||B||_2 = \sqrt{\sum_{i=1}^{m} b_i^2} \tag{2}$$

$$||B||_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} b_{ij}^2} \tag{3}$$

$$||B||_{2,1} = \sum_{i=1}^{m} ||B^i||_2 = \sum_{i=1}^{m} \sqrt{\sum_{j}^{n} b_{ij}^2} \tag{4}$$

where B^i is the i-th row vector of the matrix B. According to the matrix computation theory, $||B||_{2,1} = tr(B^T U B)$, $U \in R^{m \times m}$ is a matrix consisting of diagonal elements $u_{ii} = 1/\|B^i\|_2$.

2.2. Sparse Representation

Sparse representation is a method that was first developed in signal processing. The core idea of sparse representation is to find a target dictionary to describe the signal. To be specific, the original signal can be decomposed into linear combinations of elements in the dictionary. Only a few non-zero elements are used to represent signal information, while

the rest can be ignored. Given a sample $X \in R^n$ and a target dictionary D, it is desired to find a coefficient vector a such that the signal X can be represented as a linear combination of the basic elements of the target dictionary D.

$$\begin{aligned} &\min_{\alpha} \|\alpha\|_0 \\ &s.t.\ X = D \times \alpha. \end{aligned} \tag{5}$$

where $\alpha \in R^d$ is a one-dimensional vector [50] and $||\alpha||_0$ is the L_0 norm of α. Due to the non-convexity and discontinuity of the L_0 norm, the L_1 norm is usually used to replace the L_0 norm to obtain an approximate solution, as shown in the following formula:

$$\begin{aligned} &\min_{\alpha} \|\alpha\|_1 \\ &\text{s.t. } X = D \times \alpha. \end{aligned} \tag{6}$$

Compared with the L_1 norm, the continuous derivability property of the L_2 norm can make the optimization algorithm more intuitive. Hence, the L_2 norm is commonly used to control overfitting, which can make the weight parameters of the model smoother and avoid overly complex models, as shown in the following formula:

$$\begin{aligned} &\min_{\alpha} \|\alpha\|_2 \\ &\text{s.t. } X = D \times \alpha. \end{aligned} \tag{7}$$

However, the disadvantage of the L_2 norm is that the model parameters will be close to 0, but most of them cannot be 0. Therefore, the L_{21} norm, which is between the L_1 and L_2 norms, is proposed as an effective scheme, as shown in Equation (8):

$$\begin{aligned} &\min_{\alpha} \|\alpha\|_{21} \\ &\text{s.t. } X = D \times \alpha. \end{aligned} \tag{8}$$

The advantage of L_{21} norm is that it can make the elements of the whole row 0, thus achieving a similar sparse effect as L_1 and more robustness.

2.3. Constructing Graph Methods

KNN graph is a widely used method for constructing a similarity matrix. S_{ij} is the similarity of the sample x_i and x_j, which is defined as:

$$S_{ij} = \begin{cases} \exp\left(-\frac{||x_i - x_j||_2^2}{\delta^2}\right) & \text{if } x_i \in N_k(x_j) \text{ or } x_j \in N_k(x_i), \\ 0. & \text{else.} \end{cases} \tag{9}$$

where $N_k(x_j)$ is a set that contains k nearest neighbor samples of the sample x_j and δ is a parameter.

From Equation (9), it can be seen that as the samples get closer, their similarity also increases. In addition, there are some similar methods, such as the ϵ-neighborhood method [51] and the fully connected method [52], which can also be utilized to construct graphs.

Unlike the KNN graph, the L_1 graph is an adaptive graph learning mechanism method that aims to reconstruct each sample by find the best sparse linear combination of other samples. The objective function of the L_1 graph can be described as follows:

$$\begin{aligned} &\min \|\alpha_i\|_1 \\ &\text{s.t. } \ x_i = X\alpha_i, \alpha_{ii} = 0. \end{aligned} \tag{10}$$

Then, the weight matrix formed by the L_1 graph is expressed as $S = [\alpha_1, \alpha_2, \ldots, \alpha_N]$. Compared with KNN graphs, L_1 graphs can adaptively select the nearest samples for each sample.

2.4. Label Propagation Algorithm

The label propagation (LP) algorithm is a graph-based semi-supervised classification method that can effectively classify unknown samples using a small number of labeled samples. In the LP algorithm, similar samples should have similar labels. Therefore, the objective function of LP can be expressed as:

$$\begin{aligned} &\min_{F\geq 0} \sum_{i=1}^{N}\sum_{j=1}^{N} ||f_i - f_j||_2^2 s_{ij} + \sum_{i=1}^{N} \|f_i - y_i\|_2^2 u_{ii} \\ &= tr(F^T L F) + tr((F-Y)U(F-Y)^T) \end{aligned} \tag{11}$$

where s_{ij} can be computed by Equation (9) or Equation (10). $U \in R^{m\times m}$ is a diagonal matrix that effectively utilizes category information from all samples in SSL. The diagonal elements of this matrix are defined as follows:

$$u_{ii} = \begin{cases} \infty \text{ if } x_i \text{ is unlabeled,} \\ 0 \quad \text{otherwise.} \end{cases} \tag{12}$$

where the symbol ∞ represents a relatively large constant. The first term in Equation (11) is based on the similarity of the data, which assigns similar labels to the neighboring samples to keep the graph as smooth as possible. The second term aims to minimize the difference between the matrix F and the label Y, i.e., the sample labels predicted by the trained model should be as consistent as possible with the true labels.

2.5. The Graph-Based Semi-Supervised Sparse Feature Selection

Sparse learning is widely used in machine learning due to its superior feature extraction capabilities. In this context, sparse regularization terms are used to penalize the projection matrix with the aim of selecting features with high sparsity and high discriminative properties. The following equation is commonly used for sparse feature selection:

$$\min_{W} Loss(X, W, Y) + \theta R(W) \tag{13}$$

where the $Loss(X, W, Y)$ is defined as a regression term and $R(W)$ is a sparse regularization term, and $\theta \geq 0$ is a regularization weight to constrain both terms.

As we know, selecting features only using the information of the labeled sample is inaccurate and unreliable since the labeled samples are insufficient in SSL. Therefore, it is also necessary to make full use of the information of unlabeled samples to improve the performance. The following model can achieve feature selection by introducing an LP algorithm into the semi-supervised sparse model.

$$\min_{W,F} Loss_1(X, W, F) + \theta R(W) + \alpha Loss_2(F, Y) \tag{14}$$

where $Loss_2(F, Y)$ is the objective function of the LP algorithm shown in Equation (11).

It can be seen that when constructing the model above, the merits of the similarity matrix construction directly determine the performance. To alleviate the issue, the following graph-based semi-supervised sparse feature selection model has been developed.

$$\min_{W,F\geq 0,S\geq 0} Loss_1(X, W, Y) + \theta R(W) + \alpha Loss_2(F, Y, S) + \beta Loss_3(X, S) \tag{15}$$

where $\alpha, \beta, \theta \geq 0$ are model parameters. $Loss_3(X, S)$ is the objective function of the graph learning. Some adaptively constructed graph methods have been designed [38–41,48,53].

3. The Proposed Method

In this part, a detailed introduction of the SFS model is first presented. Second, a new AGL mechanism is introduced to make full use of the global and local information between the samples, which can acquire the geometric structural information of the original data well. Next, the similarity matrix learned by the AGL mechanism is integrated into the LP

algorithm, which enhances label prediction performance and allows the model to classify and cluster unlabeled samples more accurately. Finally, the SFS, AGL, and LP models are fused in a unified framework to propose a novel SFS-AGGL method. Moreover, a new iterative updated algorithm is introduced to optimize the proposed model, and its convergence is confirmed through both theoretical and experimental testing.

3.1. Methodology Model

3.1.1. SFS Model

The L_{21} sparsity constraint is applied to achieve the process of FS. In combination with LSR, a basic SFS model can be obtained as follows:

$$\begin{aligned} &\min_{W} \|X^T W - Y\|_2^2 + \theta \|W\|_{2,1} \\ &s.t.\ W \geq 0. \end{aligned} \tag{16}$$

where $W \in R^{d \times c}$ denotes the feature projection matrix and θ is a regularization parameter.

3.1.2. Global and Local Adaptive Graph Learning (AGGL) Model

Although the sparse model-based approach has achieved good results in FS, there are still some problems, for e.g., the above-mentioned sparse model only focuses on the sample–label relationship and ignores the geometric structural information among the samples. To better preserve the original data's geometric structural information, the method of adaptively constructing the nearest neighbor graph is usually adopted. However, the nearest neighbor information in the original feature space may be disturbed by redundant and noisy features. Previous research has shown that feature projection can effectively mitigate the negative impact of redundant and noisy features [54]. Therefore, when learning the nearest neighbor graph, the similarity matrix should be constructed through adaptive updates of sample similarities and their neighboring samples in the projected feature space. Hence, in this paper, the similarity of samples in the original high-dimensional space and the low-dimensional space is utilized to describe the local distribution structure more accurately, thus enhancing the effectiveness of the graph learning task. Specifically, we have used a coefficient reconstruction method to construct the graph, leading to the subsequent model:

$$\begin{aligned} &\min_{W,S} \|W^T X - W^T X S\|_2^2 \\ &s.t.\ W \geq 0, S \geq 0. \end{aligned} \tag{17}$$

where $S \in R^{n \times n}$ and $s_i = [s_{i1}, s_{i2}, \cdots, s_{in}] \in \mathbb{R}^{n \times 1}$ denotes the reconstructed coefficient vector of the sample.

The maintenance of global and local sample information is crucial for sample reconstruction. That is, the similarity between the sample that needs to be reconstructed and its surrounding samples should be maintained in the process of sample reconstruction. To achieve this goal, we have incorporated global and local constraints into the sample reconstruction process. This ensures that the sample points are better reconstructed by the most adjacent sample points, thereby improving the quality of the construction graph. Specifically, we have combined global and local constraints with sparse learning to reconstruct samples, as shown in the following formula:

$$\|S \odot E\|_1 \tag{18}$$

where $E = [e_{ij}] \in R^{n \times n}$ and each element e_{ij} in E is defined as:

$$e_{ij} = \exp\left(\frac{\|x_i - x_j\|^2}{\sigma^2}\right) \tag{19}$$

By combining Equations (17) and (18), the following adaptive graph construction model with global and local constraints is obtained:

$$\begin{aligned} &\min_{W,S} \|W^T X - W^T XS\|_2^2 + \lambda \|S \odot E\|_1 \\ &s.t.\ W \geq 0, S \geq 0. \end{aligned} \tag{20}$$

where $\lambda > 0$ is the balance coefficient, which aims to balance the effects of the coefficient reconstruction term and the global and local constraint terms. By constructing the above model, we can effectively maintain the global and local information of the sample, thereby enhancing the similarity matrix of the graph.

3.1.3. Objective Function

As can be seen in Equation (17), the SFS model only utilizes the labeling information of the data. It ignores the spatial distribution of the labels, making it difficult to select the ideal subset of features. It has been shown that the structural distribution information embedded in unlabeled data is very important for FS when there is less labeling information [55]. For this reason, we have introduced the LP algorithm. Meanwhile, to make the LP process more efficient, we have introduced the adaptive graph coefficient matrix obtained by Equation (20) into LP. Therefore, a new SFS-AGGL algorithm is proposed by integrating SFS, AGGL, and LP into a unified learning framework. SFS-AGGL can account for both global and local sample information, and it is robust for FS. The objective function of SFS-AGGL is:

$$\begin{aligned} \min \varepsilon(W, F, S) \quad &= \beta \|W^T X - W^T XS\|_2^2 + \lambda \|S \odot E\|_1 \\ &+ \alpha \sum_{i,j=1}^{n} \|f_i - f_j\|_2^2 S_{ij} + \sum_{i=1}^{n} \|f_i - y_i\|_2^2 u_{ii} \\ &+ \|X^T W - F\|_2^2 + \theta \|W\|_{2,1} \\ &s.t.\ W \geq 0, F \geq 0, S \geq 0. \end{aligned} \tag{21}$$

where $\alpha, \beta, \theta, \lambda > 0$ are the equilibrium control parameters to be adjusted in the experiment, and $\odot$ denotes the product of matrix elements in their corresponding positions.

As shown in Equation (21), we first efficiently obtained the constructive coefficients by imposing global and local constraints while self-representing the low-dimensional features. Therefore, it can avoid possible redundant information to affect the learning performance due to predefined matrices not being introduced. Second, we introduced the similarity matrix obtained by AGL into the LP process to improve the accuracy of label prediction. In addition, to enhance the discriminative performance of the selected features, we introduced a predictive labeling matrix into the SFS process and completed the FS by mutual reinforcement of the three models: SFS, AGGL, and LP.

3.2. Model Optimization

The objective function of the SFS-AGGL method involves three variables, i.e., the feature projection matrix W, the prediction label matrix F, and the similarity matrix S. Since the objective functions of all three variables are non-convex, they cannot be optimized directly. However, the objective function exhibits convexity with respect to a single variable. Therefore, we can solve it step-by-step by performing convex optimization on each variable separately. The specific process of solving the objective function is as follows:

(1) Fixed variables F and S update variable W

Simplifying Equation (21) by removing the terms unrelated to the variable W, the following optimization function is obtained:

$$\min \varepsilon(W) = \|X^T W - F\|_2^2 + \beta \|W^T X - W^T XS\|_2^2 + \theta \|W\|_{2,1} \tag{22}$$

From the definition of matrix trace, Equation (23) can be derived from Equation (22) by using a simple algebraic transformation as follows:

$$\begin{array}{l} \min\varepsilon(W) = tr((X^TW - F)(X^TW - F)^T) \\ +\beta tr((W^TX - W^TXS)(W^TX - W^TXS)^T) + \theta tr(W^THW) \\ = tr(X^TWW^TX - 2FW^TX + FF^T) \\ +\beta tr\left(\begin{array}{l} W^TXX^TW - 2W^TXS^TX^TW \\ +W^TXSS^TX^TW \end{array} \right) + \theta tr(W^THW) \end{array} \tag{23}$$

To solve Equation (23), a Lagrange multiplier and the corresponding Lagrange functions are introduced, which can be constructed as follows:

$$\varepsilon(W, \vartheta) = tr\left(\begin{array}{l} X^TWW^TX - 2FW^TX + FF^T + \beta W^TXX^TW \\ -2\beta W^TXS^TX^TW + \beta W^TXSS^TX^TW + \theta W^THW \end{array} \right) + tr(\vartheta W) \tag{24}$$

Next, the partial derivative regarding the variable W is computed and then set to 0 as follows:

$$\frac{\partial\varepsilon(W, \vartheta)}{\partial W} = \left(\begin{array}{l} 2XX^TW - 2XF + 2\beta XX^TW - 4\beta XS^TX^TW \\ +2\beta XSS^TX^TW + 2\theta W^THW + \vartheta \end{array} \right) = 0 \tag{25}$$

Meanwhile, by combining the Karush–Kuhn–Tucker (KKT) condition ($\vartheta_{ij}W_{ij} = 0$), we can obtain Equation (26) as follows:

$$\left(\begin{array}{l} 2XX^TW - 2XF + 2\beta XX^TW - 4\beta XS^TX^TW \\ +2\beta XSS^TX^TW + 2\theta HW \end{array} \right)_{ij} W_{ij} = 0 \tag{26}$$

Therefore, an updated rule for the variable W can be obtained:

$$W_{ij} = W_{ij}\frac{[XF + 2\beta XS^TX^TW]_{ij}}{[XX^TW + \beta XX^TW + \beta XSS^TX^TW + \theta HW]_{ij}} \tag{27}$$

(2) Fixed variables W and S update variable F

We first remove the terms unrelated to the variable F from Equation (21), and the optimization function on the variable F is acquired as:

$$\min\varepsilon(F) = \|X^TW - F\|_2^2 + \alpha\sum_{i,j=1}^{n}\|f_i - f_j\|_2^2S_{ij} + \sum_{i=1}^{n}\|f_i - y_i\|_2^2u_{ii} \tag{28}$$

According to the definition of matrix trace, we can use a simple algebraic transformation to obtain Equation (29) as follows:

$$\begin{array}{l} \min\varepsilon(F) \\ = tr((X^TW - F)(X^TW - F)^T) + \alpha tr(F^TLF) + tr((F - Y)(F - Y)^T) \\ = tr(X^TWW^TX - 2FW^TX + FF^T) + \alpha tr(F^TLF) + tr(FUF^T - 2FUY^T + YUY^T) \\ = tr(X^TWW^TX - 2FW^TX + FF^T + \alpha F^TLF + FUF^T - 2FUY^T + YUY^T) \end{array} \tag{29}$$

Next, we have introduced a Lagrange multiplier to optimize Equation (29), and the corresponding Lagrange function can be defined as follows:

$$\varepsilon(F, \mu) = tr\left(\begin{array}{l} X^TWW^TX - 2FW^TX + FF^T \\ +\alpha F^TLF + FUF^T - 2FUY^T + YUY^T \end{array} \right) + tr(\mu F) \tag{30}$$

Then, we have calculated the partial derivative with respect to the variable F and set it to 0 as follows:

$$\frac{\partial\varepsilon(F, \mu)}{\partial F} = (-2X^TW + 2F + 2\alpha LF + 2FU - 2YU^T + \mu) = 0 \tag{31}$$

Following the KKT condition ($\mu_{ij}F_{ij} = 0$), we can derive Equation (32) as shown:

$$(-2X^TW + 2F + 2\alpha LF + 2FU - 2YU^T)_{ij}F_{ij} = 0 \tag{32}$$

Finally, we have provided an iterative updated rule for the variable F as follows:

$$F_{ij} = F_{ij} \frac{[X^T W - YU^T]_{ij}}{[F + \alpha LF + FU]_{ij}} \tag{33}$$

(3) Fixed variables W and F update variable S

Likewise, by removing the terms unrelated to the variable S, the optimization function becomes the following form:

$$\min \varepsilon(S) = \alpha \sum_{i,j=1}^{n} \|f_i - f_j\|_2^2 S_{ij} + \beta \|W^T X - W^T XS\|_2^2 + \lambda \|S \odot E\|_1 \tag{34}$$

Equation (34) can be reduced to Equation (35) as follows:

$$\begin{aligned} &\min \varepsilon(S) = \alpha tr(F^T LF) + \beta tr((W^T X - W^T XS)(W^T X - W^T XS)^T) + \lambda S \odot E \\ &= \alpha tr(F^T LF) + \beta tr(W^T XX^T W - 2W^T XS^T X^T W + W^T XSS^T X^T W) + \lambda SE \\ &= tr(\alpha F^T DF - \alpha F^T SF + \beta W^T XX^T W - 2\beta W^T XS^T X^T W + \beta W^T XSS^T X^T W) + \lambda SE \end{aligned} \tag{35}$$

Here, a Lagrange multiplier is utilized to determine the optimal solution of Equation (35), and the related Lagrange function is formulated as:

$$\varepsilon(S, \xi) = \begin{pmatrix} tr(\alpha F^T DF - \alpha F^T SF + \beta W^T XX^T W - 2\beta W^T XS^T X^T W + \beta W^T XSS^T X^T W) \\ +\lambda SE + tr(\xi S) \end{pmatrix} \tag{36}$$

The partial derivative regarding the variable S is then set to 0 as follows:

$$\frac{\partial \varepsilon(S, \xi)}{\partial S} = (-\alpha FF^T - 2\beta X^T WW^T X + 2\beta X^T WW^T XS + \lambda E + \xi) = 0 \tag{37}$$

Since the KKT condition ($\xi_{ij} S_{ij} = 0$) exists, we can obtain Equation (38) as follows:

$$(-\alpha FF^T - 2\beta X^T WW^T X + 2\beta X^T WW^T XS + \lambda E)_{ij} S_{ij} = 0 \tag{38}$$

Therefore, an expression of the following form for the variable S can be obtained:

$$S_{ij} = S_{ij} \frac{[\alpha FF^T + 2\beta X^T WW^T X]_{ij}}{[2\beta X^T WW^T XS + \lambda E]_{ij}} \tag{39}$$

3.3. Algorithm Description

Algorithm 1 describes the SFS-AGGL method in detail, while Figure 2 depicts its flowchart. Moreover, the SFS-AGGL algorithm stops iterating when the alteration of the objective function value between consecutive iterations is below a threshold or the maximum number of iterations is reached.

3.4. Computational Complexity and Convergence Analysis

3.4.1. Computational Complexity Analysis

Based on Algorithm 1, the SFS-AGGL algorithm's computational complexity comprises two parts. The first part is the computation of the diagonal auxiliary matrix U in step 2, and the second part is the updating of three matrices (W, F, and S) during each iteration and the computation of the local matrix E in step 7. The computational or updating components of each matrix are defined in Table 2. Therefore, the total complexity of the SFS-AGGL algorithm is $O(\max(kn^2, cn^2) + (iter \times \max(cmn, cn^2))$, where *iter* is the iteration count. Furthermore, the computational complexities of other related FS methods are also presented in Table 3.

Algorithm 1: SFS-AGGL

Input: Sample Matrix:$X = [X_L, X_U] \in R^{d \times n}$
Label Matrix:$Y = [Y_l; Y_u]^T \in R^{n \times c}$
Parameters:$\alpha \geq 0, \beta \geq 0, \theta \geq 0, \lambda \geq 0$
Output: Feature Projection Matrix W
Predictive Labeling Matrix F
Similarity Matrix S

1: Initialization: the initial non-negative matrix W_0, F_0, S_0, $iter = 0$;
2: Calculation of the matrix U_{iter} according to Equation (12);
3: Repeat
4: According to Equation (27) **update** W_{iter} as
$$W_{iter} \leftarrow \frac{XF+2\beta XS^TX^TW}{XX^TW+\beta XX^TW+\beta XSS^TX^TW+\theta HW};$$
5: According to Equation (33) **update** F_{iter} as $F_{iter} \leftarrow \frac{X^TW-YU^T}{F+\alpha LF+FU}$;
6: According to Equation (39) **update** S_{iter} as $S_{iter} \leftarrow \frac{\alpha FF^T+2\beta X^TWW^TX}{2\beta X^TWW^TXS+\lambda E}$;
7: According to Equation (19) **update** E;
8: Update $iter = iter + 1$;
9: Until converges

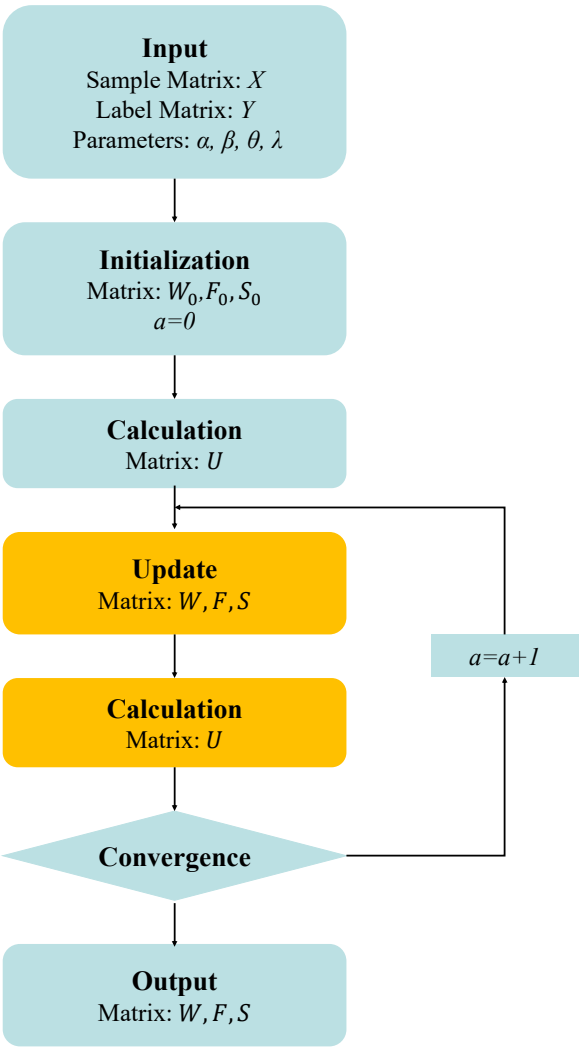


Figure 2. Flow chart of SFS-AGGL algorithm.

Table 2. The time complexity of each matrix in our proposed algorithm.

Matrix	Formula	Time Complexity
U	$U = [u_{ii}] \in R^{n \times n}$	$O(n^2)$
E	$E = [e_{ij}] \in R^{n \times n}$	$O(kn^2)$
W	$W_{ij} = W_{ij} \frac{[XF+2\beta XS^T X^T W]_{ij}}{[XX^T W+\beta XX^T W+\beta XSS^T X^T W+\theta HW]_{ij}}$	$O(cmn)$
F	$F_{ij} = F_{ij} \frac{[X^T W-YU^T]_{ij}}{[F+\alpha LF+FU]_{ij}}$	$O(cmn)$
S	$S_{ij} = S_{ij} \frac{[\alpha FF^T+2\beta X^T WW^T X]_{ij}}{[2\beta X^T WW^T XS+\lambda E]_{ij}}$	$O(cn^2)$

Table 3. Computational complexity of each iteration for FS methods.

Method	Number of Variables	Algorithm Complexity
RLSR [19]	2	$O(iter \times \max(ndc, n^3))$
FDEFS [49]	3	$O(\max(cmn, cn^2))$
GS^{3}FS [43]	4	$O(iter \times \max(d^3, n^3))$
S2LFS [44]	3	$O(cd^2n + cd^3 + cn^2)$
AGLRM [47]	4	$O(iter \times \max(d^3, n^3))$
ASLCGLFS [48]	4	$O(iter \times \max(n^3, d^3))$
SFS-AGGL	3	$O(\max(kn^2, cn^2) + iter \times \max(cmn, n^2))$

3.4.2. Proof of Convergence

Definition 1. *If functions $\varphi(q, q')$ and $\psi(q)$ meet these two conditions, as shown in Equation (40), $\varphi(q, q')$ is an auxiliary function of $\psi(q)$.*

$$\begin{aligned} \varphi(q, q') &\geq \psi(q), \\ \varphi(q, q) &= \psi(q), \end{aligned} \tag{40}$$

Lemma 1. *If Definition 1 holds, $\psi(q)$ is non-increasing in Equation (41).*

$$q^{(iter+1)} = \underset{q}{\operatorname{argmin}} \varphi(q, q^{(iter)}) \tag{41}$$

Proof.

$$\psi(q^{(iter+1)}) \leq \varphi(q^{(iter+1)}, q^{(tier)}) \leq \varphi(q^{(iter)}, q^{(iter)}) = \psi(q^{(iter)}) \tag{42}$$

It is only necessary to show that the variables W, F, and S are non-decreasing under the update rule as shown in Equation (42). For this purpose, we have computed and presented the first- and second-order derivatives of each formula in Table 4. □

Lemma 2.

$$\begin{aligned} \varphi(W_{ij}, W_{ij}^{(iter)}) &= \psi_{ij}(W_{ij}, W_{ij}^{(iter)}) + \psi'_{ij}(W_{ij})(W_{ij} - W_{ij}^{(iter)}) \\ &+ \frac{[XX^T W+\beta XX^T W+\beta XSS^T X^T W+\theta HW]_{ij}}{W_{ij}^{(iter)}} (W_{ij} - W_{ij}^{(iter)})^2 \end{aligned} \tag{43}$$

$$\begin{aligned} \varphi(F_{ij}, F_{ij}^{(iter)}) &= \psi_{ij}(F_{ij}, F_{ij}^{(iter)}) + \psi'_{ij}(F_{ij})(F_{ij} - F_{ij}^{(iter)}) \\ &+ \frac{[F+\alpha LF \quad +FU]_{ij}}{F_{ij}^{(iter)}} (F_{ij} - F_{ij}^{(iter)})^2 \end{aligned} \tag{44}$$

$$\begin{aligned} \varphi(S_{ij}, S_{ij}^{(iter)}) &= \psi_{ij}(S_{ij}, S_{ij}^{(iter)}) + \psi'_{ij}(S_{ij})(S_{ij} - S_{ij}^{(iter)}) \\ &+ \frac{[2\beta X^T WW^T XS+\lambda E]_{ij}}{S_{ij}^{(iter)}} (S_{ij} - S_{ij}^{(iter)})^2 \end{aligned} \tag{45}$$

Equations (43)–(45) are both auxiliary functions of ψ_{ij}.

Table 4. First- and second-order derivatives of each formula.

W	$\psi_{ij}(W_{ij})$	$\psi_{ij}(W_{ij}) = [X^TWW^TX - 2FW^TX + \beta W^TXX^TW - 2\beta W^TXS^TX^TW + \beta W^TXSS^TX^TW + \theta W^THW]_{ij}$
	$\psi'_{ij}(W_{ij})$	$\psi'_{ij}(W_{ij}) = 2[XX^TW - XF + \beta XX^TW - 2\beta XS^TX^TW + \beta XSS^TX^TW + \theta HW]_{ij}$
	$\psi''_{ij}(W_{ij})$	$\psi''_{ij}(W_{ij}) = 2[XX^T + \beta XX^T - 2\beta XSX^T + \beta XSS^TX^T + \theta H^T]_{ii}$
F	$\psi_{ij}(F_{ij})$	$\psi_{ij}(F_{ij}) = [-2FW^TX + FF^T + \alpha F^TLF + FUF^T - 2FUY^T]_{ij}$
	$\psi'_{ij}(F_{ij})$	$\psi'_{ij}(F_{ij}) = 2[X^TW + F + \alpha LF + FU - YU^T]_{ij}$
	$\psi''_{ij}(F_{ij})$	$\psi''_{ij}(F_{ij}) = 2[E + \alpha L^T + U^T]_{ii}$
S	$\psi_{ij}(S_{ij})$	$\psi_{ij}(S_{ij}) = [-\alpha F^TSF - 2\beta W^TXS^TX^TW + \beta W^TXSS^TX^TW + \lambda SE]_{ij}$
	$\psi'_{ij}(S_{ij})$	$\psi'_{ij}(S_{ij}) = [-\alpha FF^T - 2\beta X^TWW^TX + 2\beta X^TWW^TXS + \lambda E]_{ij}$
	$\psi''_{ij}(S_{ij})$	$\psi''_{ij}(S_{ij}) = 2[2\beta X^TWW^TX]_{ii}$

Proof. Taylor series expansion of $\psi_{ij}(W_{ij}, W_{ij}^{(iter)})$:

$$\begin{aligned}\psi_{ij}(W_{ij}) &= \psi_{ij}(W_{ij}^{(iter)}) + \psi'_{ij}(W_{ij})(W_{ij} - W_{ij}^{(iter)}) + \tfrac{1}{2}\psi''_{ij}(W_{ij})(W_{ij} - W_{ij}^{(iter)})^2 \\ &= \psi_{ij}(W_{ij}^{(iter)}) + \psi'(V_{ij})(W_{ij} - W_{ij}^{(iter)}) \\ &+ \left(XX^T + \beta XX^T - 2\beta XSX^T + \beta XSS^TX^T + \theta H^T\right)_{ii}(W_{ij} - W_{ij}^{(iter)})^2\end{aligned} \tag{46}$$

$\varphi(W_{ij}, W_{ij}^{(iter)}) \geq \psi_{ij}(W_{ij})$ Equivalent to:

$$\begin{aligned}&\frac{[XX^TW + \beta XX^TW + \beta XSS^TX^TW + \theta HW]_{ij}}{W_{ij}^{(iter)}} \\ &\geq (XX^T + \beta XX^T - 2\beta XSX^T + \beta XSS^TX^T + \theta H^T)_{ii}\end{aligned} \tag{47}$$

By comparing Equations (43) and (46), we get that $\varphi(W_{ij}, W_{ij}^{(iter)}) \geq \psi_{ij}(W_{ij})$ holds, therefore, $\varphi(W_{ij}, W_{ij}^{(iter)}) = \psi_{ij}(W_{ij})$ also holds.

$$[XX^TW]_{ij} = \sum_{k=1}^{r}(XX^T)_{ik}W_{kj}^{(iter)} \geq (XX^T)_{ij}W_{ij}^{(iter)} \tag{48}$$

$$[XSS^TX^TW]_{ij} = \sum_{k=1}^{r}(XSS^TX^T)_{ik}W_{kj}^{(iter)} \geq (XSS^TX^T)_{ij}W_{ij}^{(iter)} \tag{49}$$

$$[HW]_{ij} = \sum_{k=1}^{r}H_{ik}W_{kj}^{(iter)} \geq H_{ii}W_{ij}^{(iter)} \tag{50}$$

□

Similarly, it is possible to prove Equations (44) and (45). Finally, based on Lemma 1, the update schemes for the variables W, F, S are derived in this paper as shown in Equations (51)–(53).

Theorem 1. *$W \geq 0$, $F \geq 0$, and $S \geq 0$, updating iterative Formulas (27), (33), and (39) are non-increasing.*

Proof.

Bringing Equation (43) into Equation (41):

$$\begin{aligned} W_{ij}^{(iter+1)} &= \underset{W_{ij}}{\operatorname{argmin}} \varphi(W_{ij}, W_{ij}^{(iter)}) \\ &= W_{ij}^{(iter)} - W_{ij}^{(iter)} \frac{\psi''(W_{ij})}{[XX^TW+\beta XX^TW+\beta XSS^TX^TW+\theta HW]_{ij}} \\ &= W_{ij}^{(iter)} \frac{[XF+2\beta XS^TX^TW]_{ij}}{[XX^TW+\beta XX^TW+\beta XSS^TX^TW+\theta HW]_{ij}} \end{aligned} \tag{51}$$

Bringing Equation (44) into Equation (41):

$$\begin{aligned} F_{ij}^{(iter+1)} &= \underset{F_{ij}}{\operatorname{argmin}} \varphi(F_{ij}, F_{ij}^{(iter)}) \\ &= F_{ij}^{(iter)} - F_{ij}^{(iter)} \frac{\psi''(F_{ij})}{[F+\alpha LF+FU]_{ij}} = F_{ij}^{(iter)} \frac{[X^TW-YU^T]_{ij}}{[F+\alpha LF+FU]_{ij}} \end{aligned} \tag{52}$$

Bringing Equation (45) into Equation (41):

$$\begin{aligned} S_{ij}^{(iter+1)} &= \underset{S_{ij}}{\operatorname{argmin}} \varphi(S_{ij}, S_{ij}^{(iter)}) \\ &= S_{ij}^{(iter)} - S_{ij}^{(iter)} \frac{\psi''(S_{ij})}{[2\beta X^TWW^TXS+\lambda E]_{ij}} = S_{ij}^{(iter)} \frac{[\alpha FF^T+2\beta X^TWW^TX]_{ij}}{[2\beta X^TWW^TXS+\lambda E]_{ij}} \end{aligned} \tag{53}$$

It is obvious that Equations (51)–(53) are auxiliary functions of ψ_{ij}, resulting in a non-increasing ψ_{ij} under their respective update rules.

Next, the upcoming focus will be on demonstrating the convergence of iteration-based Algorithm 1.

For any non-zero vectors $u \in R^m$ and $v \in R^m$, the following inequalities exist:

$$\|u\|_2 - \frac{\|u\|_2^2}{2\|v\|_2} \leq \|u\|_2 - \frac{\|u\|_2^2}{2\|u\|_2} \tag{54}$$

The proof of Equation (54) can be found in the literature [55]. □

Theorem 2. *Referring to Algorithm 1, Equation (21) decreases in each iteration until it converges.*

Proof. Let H^{iter} denote the process of the *iter*-th iteration, then updating W^{iter+1}, F^{iter+1} and S^{iter+1} involves solving the given inequality:

$$\phi(W^{iter+1}, F^{iter+1}, S^{iter+1}, H^{iter}) \leq \phi(W^{iter}, F^{iter}, S^{iter}, H^{iter}) \tag{55}$$

□

According to Equation (55), we obtain:

$$\begin{aligned} &tr\Big((X^TW^{(iter+1)} - F^{(iter+1)})(X^TW^{(iter+1)} - F^{(iter+1)})^T\Big) + \alpha tr\Big((F^{(iter+1)})^T LF^{(iter+1)}\Big) \\ &+tr\Big((F^{(iter+1)} - Y)U(F^{(iter+1)} - Y)^T\Big) \\ &+\beta tr\Big(((W^{(iter+1)})^TX - (W^{(iter+1)})^TXS^{(iter+1)})((W^{(iter+1)})^TX - (W^{(iter+1)})^TXS^{(iter+1)})^T\Big) \\ &+\theta tr\Big((W^{(iter+1)})^TH^{(iter)}W^{(iter+1)}\Big) + \lambda S^{(iter+1)}E \\ &\leq tr\Big((X^TW^{(iter)} - F^{(iter)})(X^TW^{(iter)} - F^{(iter)})^T\Big) + \alpha tr\Big((F^{(iter)})^T LF^{(iter)}\Big) \\ &+tr\Big((F^{(iter)} - Y)U(F^{(iter)} - Y)^T\Big) \\ &+\beta tr\Big((W^{(iter)})^TX - (W^{(iter)})^TXS^{(iter)})((W^{(iter)})^TX - (W^{(iter)})^TXS^{(iter)})^T\Big) \\ &+\theta tr\Big((W^{(iter)})^TH^{(iter)}W^{(iter)}\Big) + \lambda S^{(iter)}E \end{aligned} \tag{56}$$

Again, based on the definition of matrix H^{iter}, Equation (56) can be rewritten as:

$$\begin{aligned}
&tr\Big((X^TW^{(iter+1)}-F^{(iter+1)})(X^TW^{(iter+1)}-F^{(iter+1)})^T\Big)+\alpha tr\Big((F^{(iter+1)})^TLF^{(iter+1)}\Big)\\
&+tr\Big((F^{(iter+1)}-Y)U(F^{(iter+1)}-Y)^T\Big)\\
&+\beta tr\Big(((W^{(iter+1)})^TX-(W^{(iter+1)})^TXS^{(iter+1)})((W^{(iter+1)})^TX-(W^{(iter+1)})^TXS^{(iter+1)})^T\Big)\\
&+\theta\sum_{i=1}^{m}\frac{||(W^{(iter+1)})^i||_2^2}{2||(W^{(iter+1)})^i||_2}+\lambda S^{(iter+1)}E\\
&\le tr\Big((X^TW^{(iter)}-F^{(iter)})(X^TW^{(iter)}-F^{(iter)})^T\Big)+\alpha tr\Big((F^{(iter)})^TLF^{(iter)}\Big)\\
&+tr\Big((F^{(iter)}-Y)U(F^{(iter)}-Y)^T\Big)\\
&+\beta tr\Big(((W^{(iter)})^TX-(W^{(iter)})^TXS^{(iter)})((W^{(iter)})^TX-(W^{(iter)})^TXS^{(iter)})^T\Big)\\
&+\theta\sum_{i=1}^{m}\frac{||(W^{(iter)})^i||_2^2}{2||(W^{(iter)})^i||_2}+\lambda S^{(iter)}E
\end{aligned}\tag{57}$$

Thus, there is the following inequality:

$$\begin{aligned}
&tr\Big((X^TW^{(iter+1)}-F^{(iter+1)})(X^TW^{(iter+1)}-F^{(iter+1)})^T\Big)+\alpha tr\Big((F^{(iter+1)})^TLF^{(iter+1)}\Big)\\
&+tr\Big((F^{(iter+1)}-Y)U(F^{(iter+1)}-Y)^T\Big)\\
&+\beta tr\Big(((W^{(iter+1)})^TX-(W^{(iter+1)})^TXS^{(iter+1)})((W^{(iter+1)})^TX-(W^{(iter+1)})^TXS^{(iter+1)})^T\Big)\\
&+\theta\sum_{i=1}^{m}\|(W^{(iter+1)})^i||_2-\theta\Big(\sum_{i=1}^{m}||(W^{(iter+1)})^i||_2-\sum_{i=1}^{m}\frac{||(W^{(iter+1)})^i||_2^2}{2||(W^{(iter+1)})^i||_2}\Big)+\lambda S^{(iter+1)}E\\
&\le tr\Big((X^TW^{(iter)}-F^{(iter)})(X^TW^{(iter)}-F^{(iter)})^T\Big)+\alpha tr\Big((F^{(iter)})^TLF^{(iter)}\Big)\\
&+tr\Big((F^{(iter)}-Y)U(F^{(iter)}-Y)^T\Big)\\
&+\beta tr\Big(((W^{(iter)})^TX-(W^{(iter)})^TXS^{(iter)})((W^{(iter)})^TX-(W^{(iter)})^TXS^{(iter)})^T\Big)\\
&+\theta\sum_{i=1}^{m}\|(W^{(iter)})^i||_2-\theta\Big(\sum_{i=1}^{m}||(W^{(iter)})^i||_2-\sum_{i=1}^{m}\frac{||(W^{(iter)})^i||_2^2}{2||(W^{(iter)})^i||_2}\Big)+\lambda S^{(iter)}E
\end{aligned}\tag{58}$$

From Equation (58), we have:

$$\sum_{i=1}^{m}\|(W^{(iter+1)})^i||_2-\sum_{i=1}^{m}\frac{\|(W^{(iter+1)})^i||_2^2}{2\|(W^{(iter+1)})^i||_2}\le\sum_{i=1}^{m}\|(W^{(iter)})^i||_2-\sum_{i=1}^{m}\frac{\|(W^{(iter)})^i||_2^2}{2\|(W^{(iter)})^i||_2}\tag{59}$$

Considering Equations (55)–(59) together, the following results can be obtained:

$$\begin{aligned}
&tr\Big((X^TW^{(iter+1)}-F^{(iter+1)})(X^TW^{(iter+1)}-F^{(iter+1)})^T\Big)+\alpha tr\Big((F^{(iter+1)})^TLF^{(iter+1)}\Big)\\
&+tr\Big((F^{(iter+1)}-Y)U(F^{(iter+1)}-Y)^T\Big)\\
&+\beta tr\Big(((W^{(iter+1)})^TX-(W^{(iter+1)})^TXS^{(iter+1)})((W^{(iter+1)})^TX-(W^{(iter+1)})^TXS^{(iter+1)})^T\Big)\\
&+\theta\sum_{i=1}^{m}\|(W^{(iter+1)})^i||_2+\lambda S^{(iter+1)}E\\
&\le tr\Big((X^TW^{(iter)}-F^{(iter)})(X^TW^{(iter)}-F^{(iter)})^T\Big)+\alpha tr((F^{(iter)})^TLF^{(iter)}\Big)\\
&+tr\Big((F^{(iter)}-Y)U(F^{(iter)}-Y)^T\Big)\\
&+\beta tr\Big(((W^{(iter)})^TX-(W^{(iter)})^TXS^{(iter)})((W^{(iter)})^TX-(W^{(iter)})^TXS^{(iter)})^T\Big)\\
&+\theta\sum_{i=1}^{m}\|(W^{(iter)})^i||_2+\lambda S^{(iter)}E
\end{aligned}\tag{60}$$

The inequality in Equation (60) shows the value of the objective function is decreased per iteration, indicating the optimization algorithm's progress toward a more optimal solution at each step. In addition, since there is a lower bound on the objective function, our

proposed optimization algorithm will converge. We also adopted numerical experiments to further verify the effectiveness of the optimization algorithm, and the experimental result demonstrates that the objective function value consistently decreases as the number of iterations increases.

4. Experiment and Analysis

In this section, the effectiveness of the proposed method is validated on classification and clustering tasks, respectively. We first used five image classification datasets to test the classification performance of the proposed method and then employed two image datasets and two subsets of UCI data to verify the clustering performance of the proposed method. In the experiment, we compared our proposed method with some contemporary UFS and SSFS methods, including two UFS methods (SPNFSR [56] and NNSAFS [57]) and six SSFS methods (RLSR [19], FDEFS [50], GS^{3}FS [43], S2LFS [44], AGLRM [47], and ASLCGLFS [48]).

4.1. Description of the Comparison Methods

In order to verify the effectiveness of our method and comprehensively evaluate the strengths and weakness of our proposed method, we compared it with some classical and novel benchmark methods for unsupervised and semi-supervised FS, which are similar to our method. Compared to these existing methods, our method is an improvement and innovation of them, which is with a general tendency toward continuous improvement.

(1) SPNFSR is a UFS algorithm that uses a low-rank representation graph for maintaining feature structures, and it achieves FS by using the L_{21} norm and non-negative constraints on the reconstruction coefficient matrix. The objective function of the SPNFSR method can be defined as follows:

$$\begin{aligned} &\min \|X - XW||_{2,1} + \alpha tr(W^T MW) + \beta \|W||_{2,1} \\ &s.t.\ W \geq 0. \end{aligned} \tag{61}$$

where the matrix M is obtained by solving the low-rank representation. In the SPNFSR method, the processes of graph construction and feature selection are performed independently, so the quality of the matrix M will directly affect the performance of feature selection.

(2) NNSAFS is a UFS algorithm that employs adaptive rank constraints and non-negative spectral feature learning. It employs sparse regression and feature mapping to mine the local structural information of the feature space to improve the adaptability of manifold learning. The objective function of NNSAFS can be defined as follows:

$$\begin{aligned} &\min \|X^T W - F\|_2^2 + \alpha_1 \|W\|_1 + \alpha_2 Tr(W^T L^W W) + \lambda Tr(F^T L_S F) + \beta \textstyle\sum_{ij} (s_{ij} \log s_{ij}) \\ &s.t.\ W \geq 0, F^T F = I, \sum_{j=1}^{n} s_{ij} = 1, s_{ij} > 0. \end{aligned} \tag{62}$$

where $\sum_{ij} (s_{ij} \log s_{ij})$ is an entropy regularization term to estimate the uniformity of matrix S. Compared with the SPNFSR method, NNSAFS integrates graph learning and feature selection into a framework to overcome the shortcomings of the SPNFSR method. Moreover, local structural information of learned feature is also considered. However, since NNSAFS and SPNFSR are unsupervised methods and do not consider the label information of the data, they cannot select the features with good discriminability.

(3) RLSR is an SSFS method, which identifies key features by learning the global and sparse solutions of the feature projection matrix. It also redefines regression coefficients with a deflation factor, as shown in Equation (63):

$$\begin{aligned} &\min \|X^T W + 1b^T - Y||_F^2 + \gamma \|W||_{2,1}^2 \\ &s.t.\ W,\ b,\ Y_U \geq 0, Y_U 1 = 1. \end{aligned} \tag{63}$$

Different from the SPNFSR and NNSAFS methods, RLSR is a semi-supervised selection method that can use both labeled and unlabeled samples to improve the discriminability

of features. Moreover, it also uses the L_{21} norm instead of the L_1 norm to reduce the redundancy of selected features.

(4) FDEFS is a supervised or semi-supervised FS method that combines margin discriminant embedding, manifold embedding, and sparse regression to achieve feature selection.

$$\begin{array}{l} \min \mu \ (\|W\|_{2,1} + \gamma \|X^T W + 1_n b^T - F\|_2^2) + tr(Z^T L_1 Z) \\ s.t.\ L_1 = L + \lambda \widetilde{M}_l. \end{array} \tag{64}$$

where $\widetilde{M}_l$ is a square matrix, and the detailed calculation procedure is provided in [50]. FDEFS can be regarded as an extension of RLSR by combining discriminant embedding terms and manifold embedding terms to enhance the discriminability of selected features.

(5) GS^{3}FS is a robust graph-based SSFS method that selects relevant and sparse features through manifold learning and the $L_{2\mathrm{p}}$ norm imposed on the regularization and loss functions.

$$\begin{array}{l} \min tr(F^T LF) + tr((F - Y)U(F - Y)^T) + \|X^T W + 1_n b^T - F\|_{2,p}^p + \lambda \|W\|_{2,p}^p \\ s.t.\ F \geq 0, W, p \in (0, 1]. \end{array} \tag{65}$$

Compared with the FDEFS method, GS^{3}FS first integrates the LP into FDEFS. Moreover, GS^{3}FS uses the $L_{2\mathrm{p}}$ norm instead of the L_{21} norm to highlight the robustness of the selected features.

(6) S2LFS is a novel SSFS that can select different subsets for different categories rather than selecting one subset for all categories.

$$\begin{array}{l} \min \sum\limits_{k=1}^{c} \|g_k - X^T w_k\|^2 + \lambda \sum\limits_{k=1}^{c} w_k^T diag(z_k^{-1}) w_k + \beta(tr(G^T LG) + tr((G - Y)^T U(G - Y)) \\ s.t.\ G \geq 0, G^T G = I_c, z_k \geq 0, z_k^T 1_d = 1. \end{array} \tag{66}$$

where z_k is an indicator vector representing whether a feature is chosen or not for the k-th class, and w_k is the prediction function for the k-th class based on the selected features.

(7) AGLRM uses AGL techniques to enhance similarity matrix construction and mitigate the adverse impact of redundant features by minimizing redundant terms.

$$\begin{array}{l} \min \left\{ \begin{array}{l} \gamma tr(F^T LF) + tr((F - Y)U(F - Y)^T) + \alpha ||S||_F^2 + tr(W^T XLW^T X) \\ +\theta tr(W^T AW) + ||X^T W + 1_n b^T - F||_F^2 + \lambda ||W||_{2,1} \end{array} \right. \\ s.t.\ 0 \leq \mathrm{S}_{ij} \leq 1, \mathrm{S}_i 1_n = 1. \end{array} \tag{67}$$

where A is a matrix of correlation coefficients for evaluating feature correlations.

Although the performance of the AGLRM method is superior to other methods, it still has shortcomings. First, the weight matrix of the graph is constrained by the L_2 norm, which results in the graph lacking a sparse structure. Second, global constraints are not considered in the graph learning process, which leads to neglect of the distribution of the data and failure to explore more effective feature similarity metrics, thus affecting the performance of the method.

(8) ASLCGLFS improves similarity matrix quality by integrating label information into AGL. Additionally, it considers both local and global structures of the samples, thereby reducing redundancy in the selected features.

$$\begin{array}{l} \min \left\{ \begin{array}{l} ||X^T W - F||_F^2 + \sum\limits_{ij}^{n} ||W^T (X_i - X_j)||_2^2 S_{ij} + \alpha ||S - A||_F^2 + tr(F^T LF) \\ +tr((F - Y)U(F - Y)^T) + ||W^T X - W^T XZ||_F^2 + \beta ||Z||_{2,1} + \lambda ||W||_{2,1} \end{array} \right. \\ s.t.\ 0 \leq \mathrm{S}_{ij} \leq 1, \mathrm{S}_i^T 1_n = 1, \alpha, \beta, \lambda \geq 0. \end{array} \tag{68}$$

As an improvement to AGLRM, ASLCGLFS considers global information. However, the introduction of a predefined similarity matrix may bring in redundant information, which affects the learning performance. Therefore, instead of introducing predefined matrices, we will consider using the introduction of brand-new constraints to learn global

and local information and reduce redundancy in order to improve the performance of feature selection.

4.2. Classification Experiments

4.2.1. Classification Datasets

Five publicly available image datasets were used in the classification experiment, which includes four face classification datasets (AR [58], CMU PIE [59], Extended YaleB [60], ORL [61]) and one object classification dataset (COIL20 [62]). Table 5 presents the detailed information of these datasets, in which P_1 and P_2 indicate training and test samples per category, respectively.

Table 5. Details of the five image datasets.

Dataset	Size of Image	Size of Classes	Size per Class	P_1	P_2	Type
AR	32×32	100	14	7	7	Face
CMU PIE	32×32	68	24	12	12	Face
Extended YaleB	32×32	38	64	20	44	Face
ORL	32×32	40	10	7	3	Face
COIL20	32×32	20	72	20	52	Object

The AR dataset is a widely used standard database consisting of more than 4000 color facial images. These images are from 126 faces, including 56 females and 70 males. The images in this dataset have variable expressions, lighting changes, and external occlusions. Figure 3a shows some images from this database.

The CMU PIE dataset consists of 41,368 grayscale facial images of 68 individuals. These images cover subjects of different ages, genders, and skin tones with different postural conditions, lighting environments, and expressions. Figure 3b shows some examples in this dataset.

The Extended YaleB dataset was taken from 38 subjects, and each subject was selected from 64 photos in different poses, different lighting environments, and 5 different shooting angles. This dataset has a total of 2414 face images. Figure 3c shows some images from the Extended YaleB dataset.

The ORL dataset contains 400 images of faces from 40 volunteers. Each volunteer provided 10 images with different facial postures, facial expressions, and facial ornaments obscured, such as serious or smiling, eyes up or squinting, and wearing or not wearing accessories. Some of the examples from this dataset can be observed in Figure 3d.

The COIL20 dataset comprises 1440 images featuring 20 different subjects. A total of 72 images were taken for each subject at 5-degree intervals. Some of the images from COIL20 are shown in Figure 3e.

It should be mentioned that in most existing work, these face databases (AR, CMU PIE, Extended YaleB, and ORL) are commonly used to evaluate the performance of the method because of the following aspects: (1) each database has different numbers of original data and categories; (2) each database contains different types of face variations; (3) each database has different conditions and environments for image acquisition. By using these classical facial datasets to evaluate our proposed method, we can ensure that our experimental results are adequately comparable to previous findings, thus better assessing the novelty and effectiveness of our proposed method in the field of face recognition.

(**a**) AR

(**b**) CMU PIE

(**c**) Extended YaleB

(**d**) ORL

(**e**) COIL20

Figure 3. Sample images of five datasets.

4.2.2. Evaluation Metric

The accuracy rate [63] is employed to measure the performance of SFS-AGGL on the classification task, which is represented as:

$$ACC = \frac{TP + TN}{TP + FP + FN + TN} \times 100\% \tag{69}$$

where TP and TN represent the numbers of correctly identified positive and negative samples. Additionally, false positive (FP) and false negative (FN) signify the misclassification of negative samples as positive and positive samples as negative, respectively. A higher accuracy rate value indicates improved classification performance.

4.2.3. Experimental Setup for Classification Task

In this experiment, P_1 samples are randomly selected from each class for training, and the remaining P_2 samples are used for testing. Then, an FS model is used to select a limited number of relevant features from the training data, and the model's effectiveness is assessed using KNN on the testing samples with only a subset of features. For the sake of experiment fairness and reliability, each experiment is conducted 10 times using diverse training data, and the final experimental results are represented as average classification accuracy and standard deviation. In addition, to select the optimal parameters, we used the grid search method to find the optimal values of parameters α, β, θ, and λ in the range {0.001, 0.01, 0.1, 1, 10, 100, 1000} and the optimal number of iterations a in {100, 200, 300, 400, 500, 600}. The dimensions of selected features vary from 50 to 500 in increments of 50.

4.2.4. Analysis of Classification Results

(1) Parameter sensitivity analysis of classification

The effects of feature dimension (d), number of iterations ($iter$), and four balance parameters (α, β, θ, λ) on the performance of SFS-AGGL in the classification task are investigated. To assess SFS-AGGL's performance across varied experimental scenarios, the number of feature dimensions, iteration times, and the values of four balancing parameters were adjusted.

First, we have demonstrated the influence of different iteration times on the performance of SFS-AGGL, with the remaining parameters set to their optimal values. As shown in Figure 4, the classification accuracy varies with the iterations, showing an increasing trend. However, the classification accuracy will decrease or remain stable with an increasing iteration after reaching its peak. This demonstrates that SFS-AGGL can reduce the impact of noise and redundant features and effectively overcome overfitting problems.

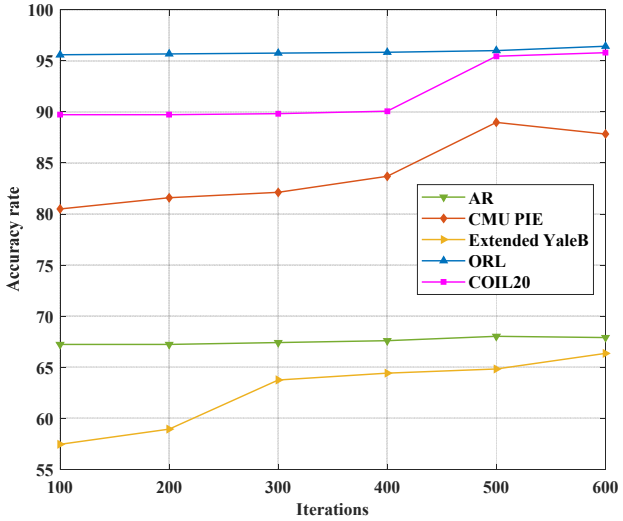
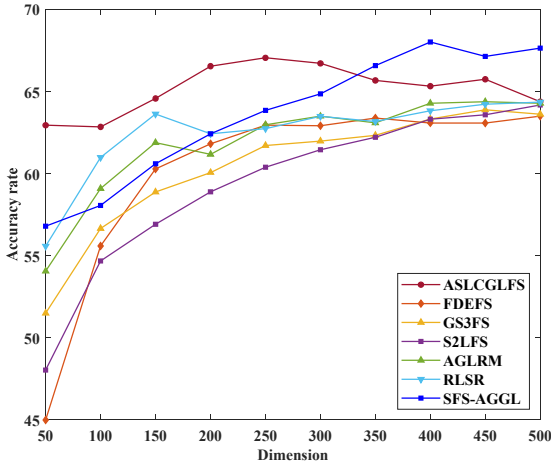
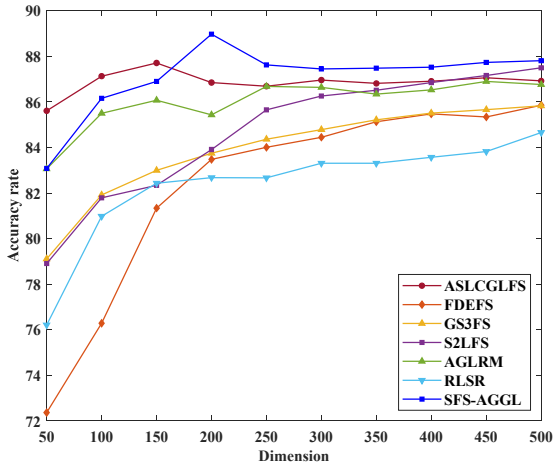


Figure 4. Classification accuracy of SFS-AGGL under different iterations.

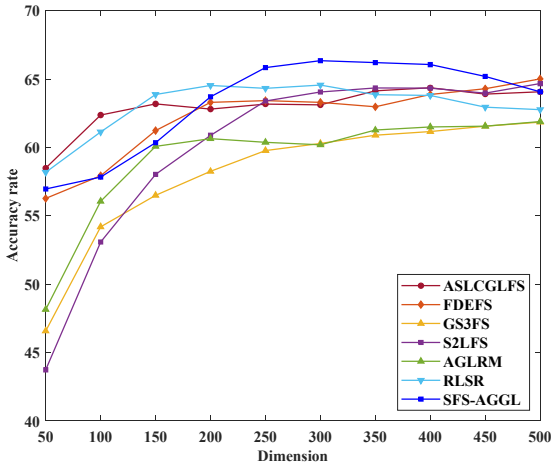
Second, the performance of different methods in different feature dimensions is shown in Figure 5. From Figure 5, we can find that the accuracy obtained by all methods is relatively lower when the feature dimensions are smaller. On the contrary, the performance of all methods gradually improves as the number of selected features increases. In most cases, the proposed SFS-AGGL outperforms the comparison methods, which indicates the stronger discriminative ability of the features selected by SFS-AGGL. However, the performance of some methods decreases as the number of selected features increases. This may be due to the presence of redundant or noisy information features in higher dimensions. Nevertheless, SFS-AGGL still surpasses the comparison methods in classification accuracy. The experimental results further validate the enhanced robustness of the features chosen by the SFS-AGGL method.



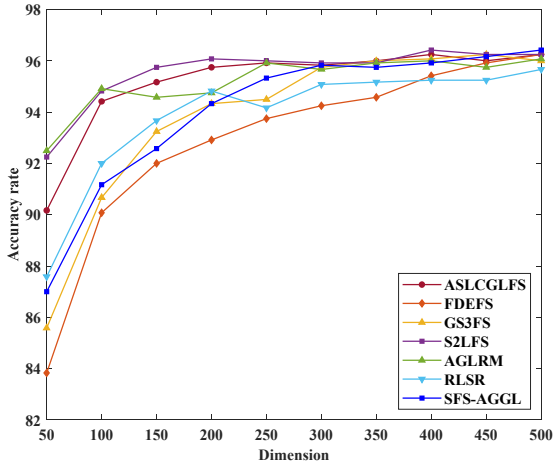
(**a**) AR



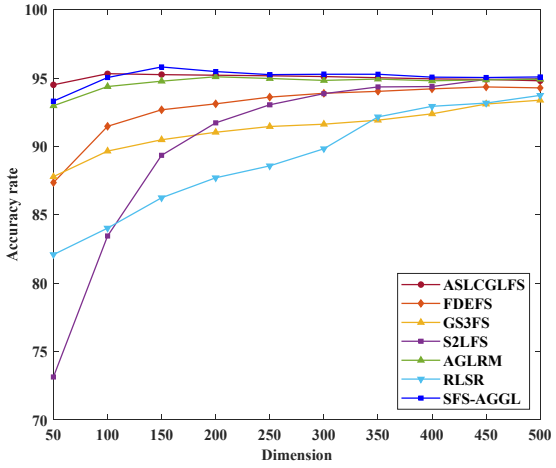
(**b**) CMU PIE



(**c**) Extended YaleB



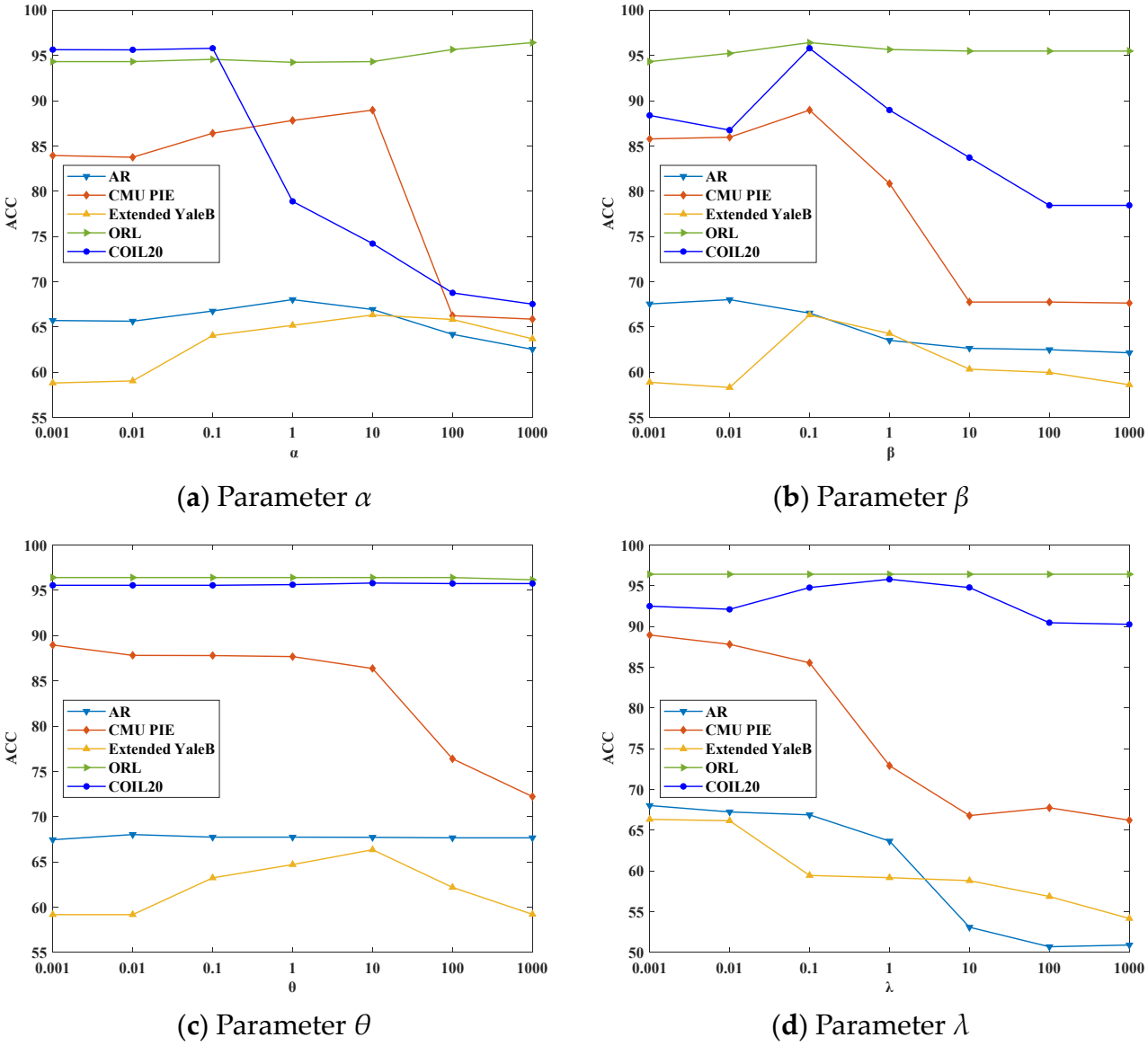
(**d**) ORL



(**e**) COIL20

Figure 5. Classification accuracy of different methods under different feature dimensions.

Third, the performance of the proposed SFS-AGGL with different values of the four balancing parameters α, β, θ, and λ on different datasets is tested. The classification results for each balance parameter are depicted in Figure 6. From Figure 6, the following conclusions can be drawn:



(**a**) Parameter α (**b**) Parameter β

(**c**) Parameter θ (**d**) Parameter λ

Figure 6. Classification results of SFS-AGGL under different parameter values.

(1) The parameter α is used to control LP. The performance of SFS-AGGL is very sensitive to parameter α on different datasets.

(2) The parameter β affects the performance of AGL. SFS-AGGL achieves the best performance when β is set to 0.01 for the AR dataset and β is set to 0.1 for other datasets. In addition, the classification accuracy of SFS-AGGL on the ORL dataset is insensitive to different values of β. In contrast, the classification performance is very sensitive to the parameter β on other datasets. Therefore, β should be set to a smaller value to obtain better classification results.

(3) The parameter θ determines the significance of the sparse feature projection terms. The performance of SFS-AGGL is insensitive to parameter θ on the ORL, COIL20, and AR datasets, but it is very sensitive on the Extended YaleB and CMU PIE datasets.

(4) The parameter λ determines the importance of global and local constraint terms. SFS-AGGL achieves high accuracy on each dataset when the value of λ is small. However, the performance of SFS-AGGL decreases with increasing λ for the CMU PIE, Extended YaleB, and AR datasets. This indicates that there is significant variation among intraclass samples in these datasets. Therefore, λ should be set to a smaller value in the case of large differences between intraclass samples.

In summary, different values of the balancing parameters will have different effects on different datasets. The optimal parameter combinations for each dataset are listed in Table 6.

Table 6. Optimal parameter combination for SFS-AGGL on the five datasets.

Dataset	$\{d, t, \alpha, \beta, \theta, \lambda\}$
AR	{400, 200, 1, 0.01, 0.01, 0.001}
CMU PIE	{200, 200, 10, 0.1, 0.001, 0.001}
Extended YaleB	{300, 100, 10, 0.1, 10, 0.001}
ORL	{500, 100, 1000, 0.1, 0.001, 0.001}
COIL20	{150, 100, 0.1, 0.1, 10, 1}

(2) Comparative analysis of classification performance

First, this section validates the classification performance of SFS-AGGL compared to other methods on the five image datasets. Table 7 presents the optimal average classification accuracy and their corresponding standard deviations for the different methods. The results in Table 7 show that: (1) SSFS methods outperform the UFS method, which indicates that the guidance of a small number of labels is crucial to improving the performance; (2) the joint FS algorithms achieve better performances than that of the RLSR method, which indicates that the correlation information among features is important for improving the FS performance; (3) the semi-supervised methods RLSR and FDEFS are inferior to other semi-supervised methods, which demonstrates that introducing the LP algorithm into semi-supervised methods is favorable for selecting discriminative features; (4) the proposed SFS-AGGL method outperforms the ASLCGLF method, notably since it integrates global and local constraints into AGL. Therefore, it is beneficial to fully consider LP and AGL in the SSFS approach to improve performance.

Table 7. Best results of each method on five image datasets (ACC).

Method	AR	CMU PIE	Extended YaleB	ORL	COIL20
NNSAFS	63.90 ± 2.12 (400)	85.29 ± 0.64 (500)	62.58 ± 1.39 (300)	92.17 ± 1.81 (500)	93.56 ± 1.32 (100)
SPNFSR	64.50 ± 0.91 (200)	86.22 ± 1.02 (300)	64.02 ± 1.95 (300)	92.83 ± 1.81 (500)	94.21 ± 1.42 (400)
RLSR	64.37 ± 1.58 (500)	84.66 ± 1.25 (500)	64.57 ± 0.87 (300)	95.67 ± 1.75 (500)	93.73 ± 1.25 (500)
FDEFS	63.51 ± 1.29 (500)	85.85 ± 1.06 (500)	65.01 ± 1.00 (500)	96.25 ± 1.37 (500)	94.35 ± 1.42 (450)
GS^3FS	63.90 ± 1.37 (450)	85.83 ± 0.68 (500)	61.85 ± 1.18 (500)	96.25 ± 1.48 (450)	93.38 ± 1.35 (500)
S2LFS	64.20 ± 1.48 (500)	87.50 ± 0.85 (500)	64.67 ± 0.82 (500)	96.42 ± 1.62 (400)	94.95 ± 1.18 (500)
AGLRM	64.39 ± 1.58 (450)	86.90 ± 0.72 (450)	61.89 ± 1.13 (500)	96.08 ± 1.42 (500)	95.09 ± 1.48 (200)
ASLCGLFS	67.07 ± 1.62 (250)	87.71 ± 1.12 (150)	64.36 ± 1.19 (400)	96.25 ± 1.37 (500)	95.31 ± 1.13 (100)
SFS-AGGL	68.03 ± 1.58(400)	88.97 ± 1.11(200)	66.35 ± 1.22 (300)	96.42 ± 1.31(500)	95.80 ± 1.16(500)

Numbers in parentheses denote the feature dimensions yielding the optimal results.

Then, to demonstrate the superiority of SFS-AGGL, we employed one-tailed *t*-tests to determine if SFS-AGGL significantly outperformed the comparison methods. Both the null hypothesis and alternative hypotheses assumed that the results achieved by SFS-AGGL were equal to or greater than the results obtained by the comparison methods. For instance, in comparing SFS-AGGL with RLSR (SFS-AGGL vs. RLSR), the hypotheses are defined as $H0$: SFS-AGGL = RLSR and $H1$: SFS-AGGL > RLSR, where SFS-AGGL and RLSR represent average classification results obtained by SFS-AGGL and RLSR on different datasets, respectively. The experiment sets a statistical significance level of 0.05, and Table 8 presents the *p* values of pairwise one-tailed *t*-tests on different datasets.

From Table 8, it can be seen that the performance of all methods is comparable on ORL and COIL datasets since these two datasets are relatively simple compared with other datasets, but the accuracy of our method is still slightly higher than that of other methods. Moreover, for AR, CMU PIE, and Extended YaleB databases, our method was able to significantly outperform the other comparative methods, indicating that our method is more advantageous in dealing with complex datasets.

Table 8. p values of the pairwise one-tailed t-tests on five image datasets.

Method	AR	CMU PIE	Extended YaleB	ORL	COIL20
RLSR vs. SFS-AGGL	3.14×10^{-5}	4.40×10^{-8}	6.97×10^{-4}	7.03×10^{-1}	6.24×10^{-4}
FDEFS vs. SFS-AGGL	7.82×10^{-7}	1.03×10^{-6}	0.74×10^{-2}	9.44×10^{-1}	1.12×10^{-2}
GS^3FS vs. SFS-AGGL	3.36×10^{-6}	6.90×10^{-8}	5.95×10^{-8}	9.36×10^{-1}	2.14×10^{-4}
S2LFS vs. SFS-AGGL	1.29×10^{-5}	1.10×10^{-3}	9.62×10^{-4}	9.53×10^{-1}	6.23×10^{-2}
AGLRM vs. SFS-AGGL	1.55×10^{-5}	1.96×10^{-5}	5.10×10^{-8}	8.84×10^{-1}	1.23×10^{-1}
ASLCGLFS vs. SFS-AGGL	9.87×10^{-2}	7.50×10^{-3}	8.17×10^{-4}	9.44×10^{-1}	1.76×10^{-1}

4.3. Clustering Experiments

This section validates the effectiveness of the SFS-AGGL method for clustering tasks. For this purpose, we used the face dataset ORL and the object dataset COIL20, as well as two UCI datasets (Libras Movement and Landsat [64]) in the experiment.

4.3.1. Clustering Datasets

The Libras Movement dataset contains 15 gestures with a total of 360 samples and 89 attributes, while the Landsat dataset contains multispectral images of six different geographic regions with a total of 296 samples and 36 attributes. The details of all clustering datasets used are shown in Table 9.

Table 9. Details of four clustering datasets.

Dataset	Number of Samples	Dimension	Category
ORL	400	1024	40
COIL20	1440	1024	20
Libras Movement	360	89	15
Landsat	296	36	6

4.3.2. Evaluation Metrics

Multiple metrics, such as *ACC*, *NMI*, *purity*, *ARI*, *F-score*, *precision*, and *recall* [65], are applied to evaluate the clustering performance.

ACC represents clustering accuracy, which is defined as:

$$ACC = \frac{\sum_{i=1}^{n} \delta(y_i, map(\overline{y}_i))}{n} \tag{70}$$

where $\delta(x, y) = \begin{cases} 1, \ if \ x = y \\ 0, \ otherwise \end{cases}$, n is the total number of samples, y_i and $\overline{y}_i$ denote the ground truth label and clustering label of the i-th sample, respectively, and where $map(\cdot)$ is a function that maps the learned clustering labels to align with the ground-truth labels.

NMI is the normalized mutual information for clustering, which is defined as:

$$NMI = \frac{MI(H, V)}{\sqrt{H(U) \cdot H(V)}} \tag{71}$$

where *MI* denotes the mutual information, i.e., the entropy of the two sets, U and V. *MI* has been normalized to ensure fair comparisons between sets of different sizes.

ARI is the adjusted Rand index, which is defined as:

$$ARI = \frac{RI - Expected\_RI}{\max(RI\_max) - Expected\_RI} \tag{72}$$

where *RI* (Rand index) denotes the number of sample pairs that are correctly clustered by the clustering algorithm out of all sample pairs; *Expected_RI* denotes the expected Rand index obtained through random clustering; and max(*RI_max*) indicates the maximum

possible Rand index. The RI is adjusted to account for randomness, with values ranging between -1 and 1, where a value closer to 1 indicates better clustering performance.

Purity measures the proportion of true categories that dominate each cluster.

$$Purity = \frac{1}{N}\sum_k \max_j |C_k \cap G_j| \tag{73}$$

where C_k denotes the k-th cluster, G_j denotes the j-th true category, and N denotes total number of samples.

Precision reflects the ratio of correctly clustered positive samples to all samples identified as positive.

$$Precision = \frac{TP}{TP + FP} \tag{74}$$

Recall indicates the proportion of positive samples that were correctly clustered with all actual positive samples.

$$Recall = \frac{TP}{TP + FN} \tag{75}$$

F-score is the harmonic mean of precision and recall, providing a comprehensive assessment of both performance metrics.

$$F - score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{76}$$

4.3.3. Experimental Setup for Clustering

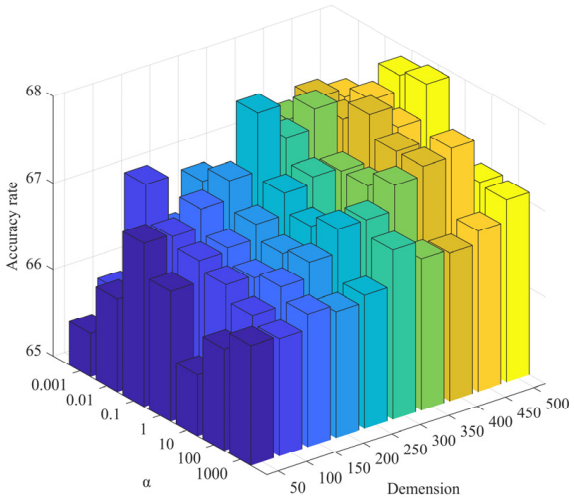
In this experiment, we set the four parameters (α, β, θ, and λ) with range {0.001, 0.01, 0.1, 1, 10, 100, 1000} for all datasets and the dimensions (d) with range {50, 100, 150, 200, 250, 300, 350, 400, 450, 500}, {8, 16, 24, 32, 40, 48, 56, 64, 72, 80}, and {3, 6, 9, 12, 15, 18, 21, 24, 27, 30} for different datasets, respectively.

4.3.4. Analysis of Clustering Results

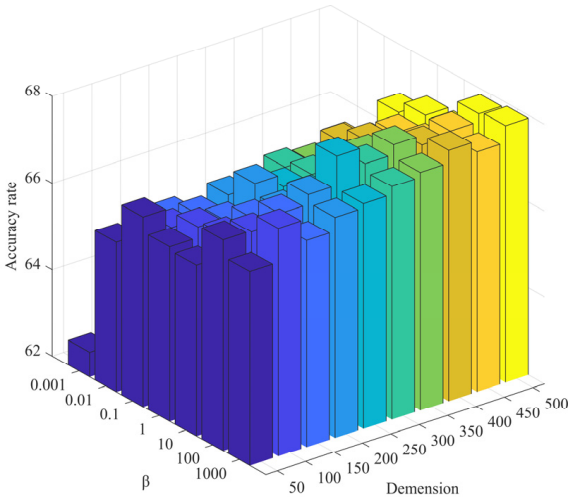
(1) Parameter sensitivity analysis of clustering

Figure 7 illustrates the clustering results of SFS-AGGL on four datasets with varying parameters. When the selected feature dimension is unchanged, the parameter α first increases, then decreases, and finally rises again. The performance of SFS-AGGL is sensitive to different parameter values on different datasets, which underscores the importance of adjusting these values to achieve optimal clustering performance. Smaller values of regularization parameters β and λ can yield improved overall performance on diverse datasets. This demonstrates that our proposed SFS-AGGL can not only acquire neighboring information in the projected feature space but also capture the global and local sparse structures in the original feature space, ultimately leading to good performance. The performance of SFS-AGGL first improves and then decreases as the regularization parameter λ increases on the COIL20 and Landsat datasets. This indicates that SFS-AGGL is more sensitive to sparse learning in space. In summary, setting all balance parameters to smaller values enhances the clustering results of SFS-AGGL. Furthermore, it is advisable to adjust parameter values tailored to each dataset to achieve optimal outcomes.

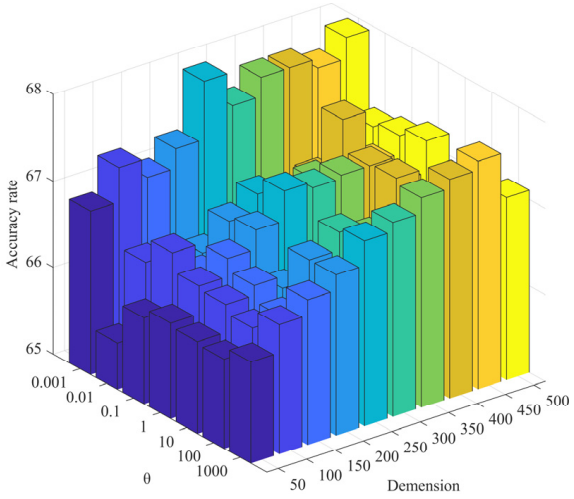
Figure 8 shows the clustering results obtained by sequentially setting each balancing parameter to different values while keeping all other conditions at optimal values. It can be found that the performance of SFS-AGGL is insensitive to all parameters in most cases. Notably, the clustering accuracy of SFS-AGGL on the ORL dataset is relatively sensitive to an increase in the parameter β. Therefore, it is recommended to set β to a larger value for optimizing clustering performance.



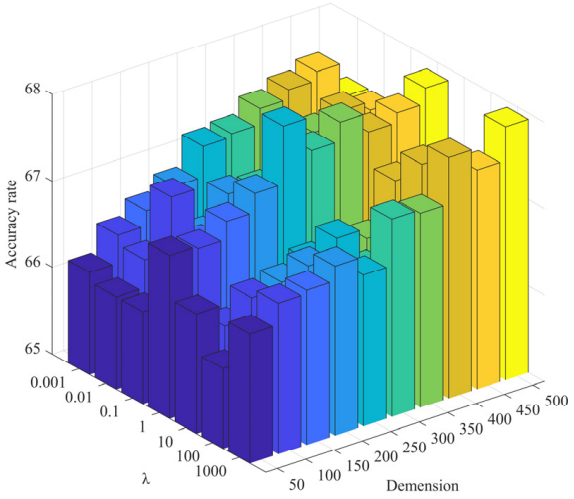
α on ORL



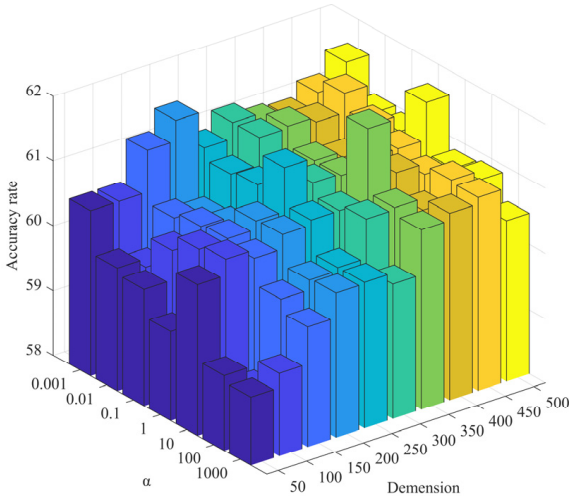
β on ORL



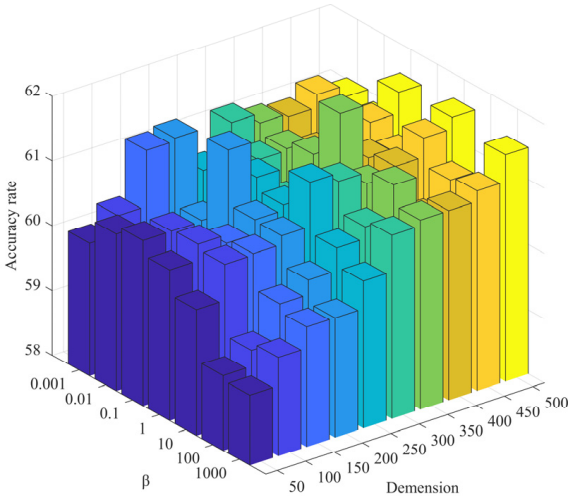
θ on ORL



λ on ORL

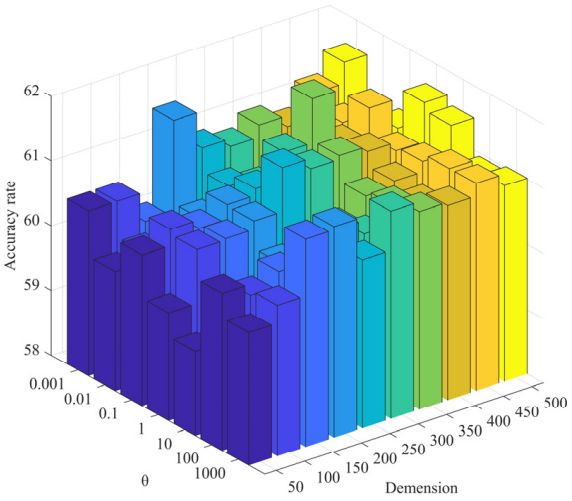


α on COIL20

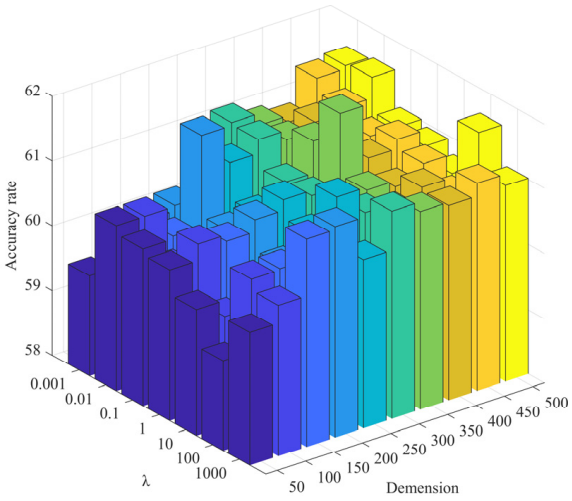


β on COIL20

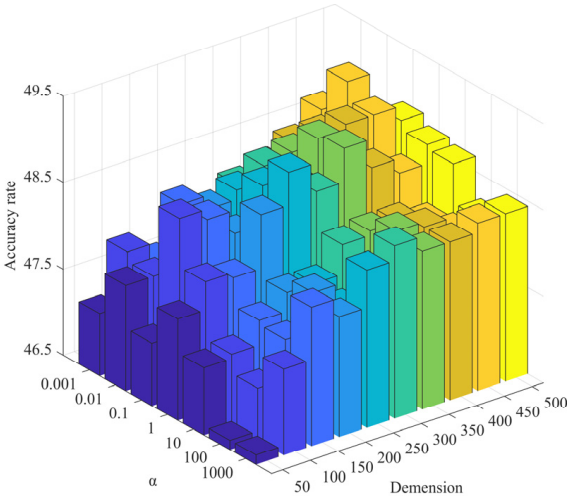
Figure 7. *Cont.*



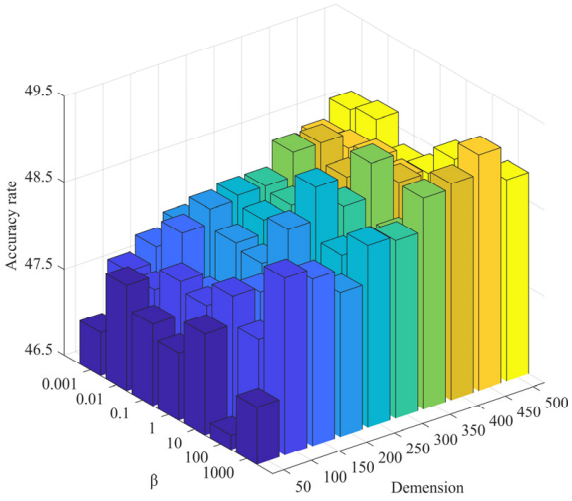
θ on COIL20



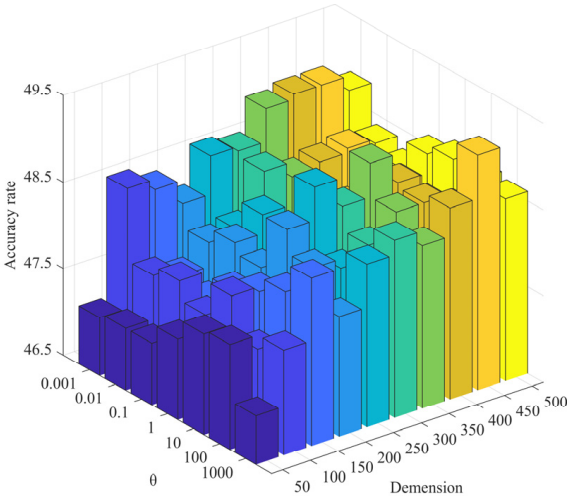
λ on COIL20



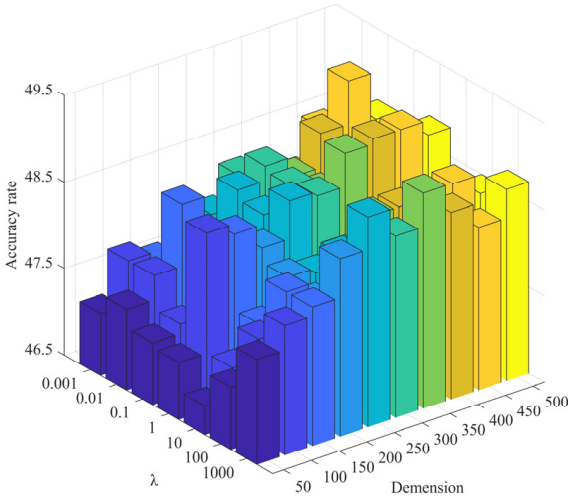
α on Libras Movement



β on Libras Movement

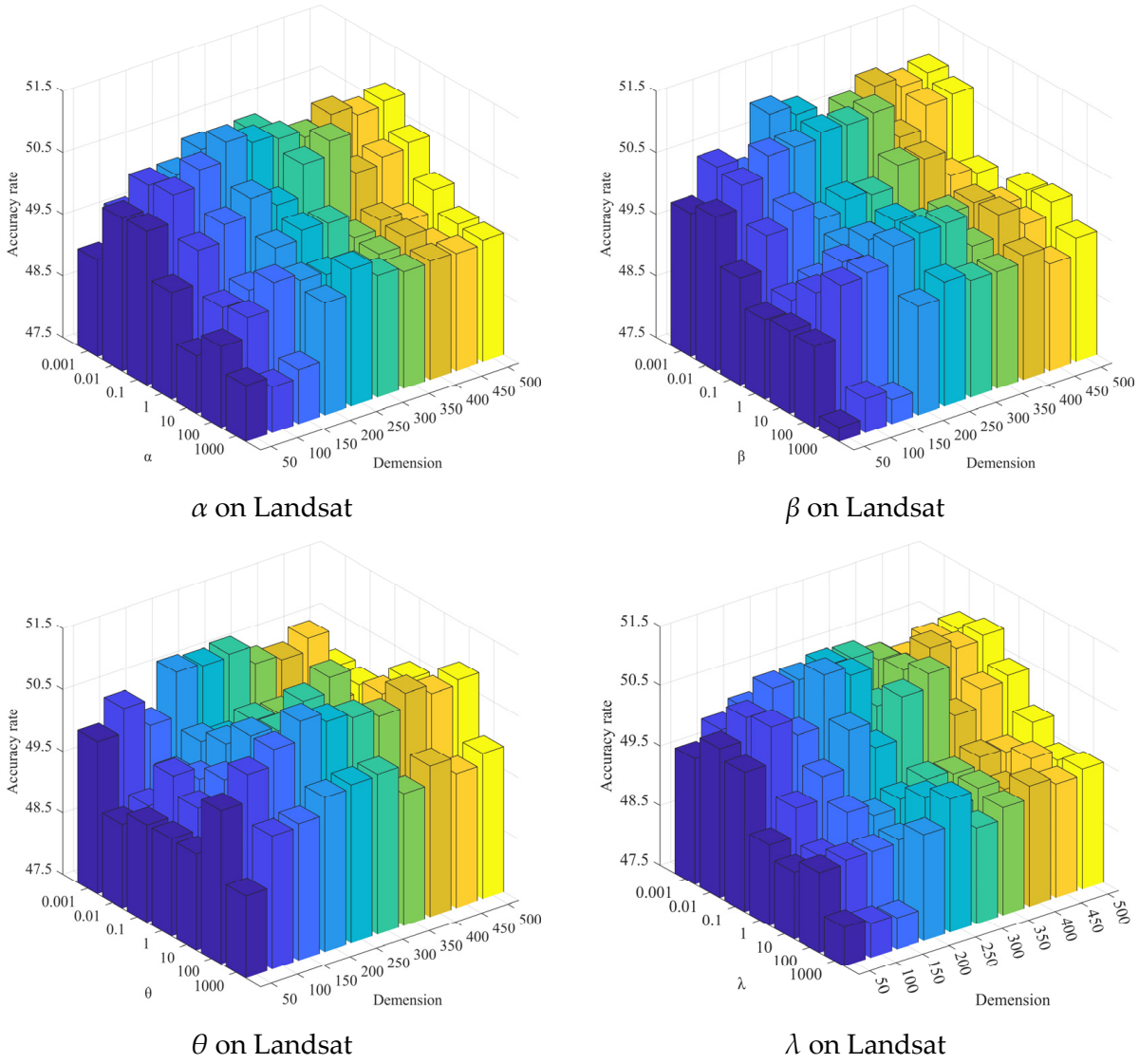


θ on Libras Movement



λ on Libras Movement

Figure 7. *Cont.*



α on Landsat β on Landsat

θ on Landsat λ on Landsat

Figure 7. Clustering results of SFS-AGGL under different parameter values and different feature dimensions, where different colors represent different feature dimensions.

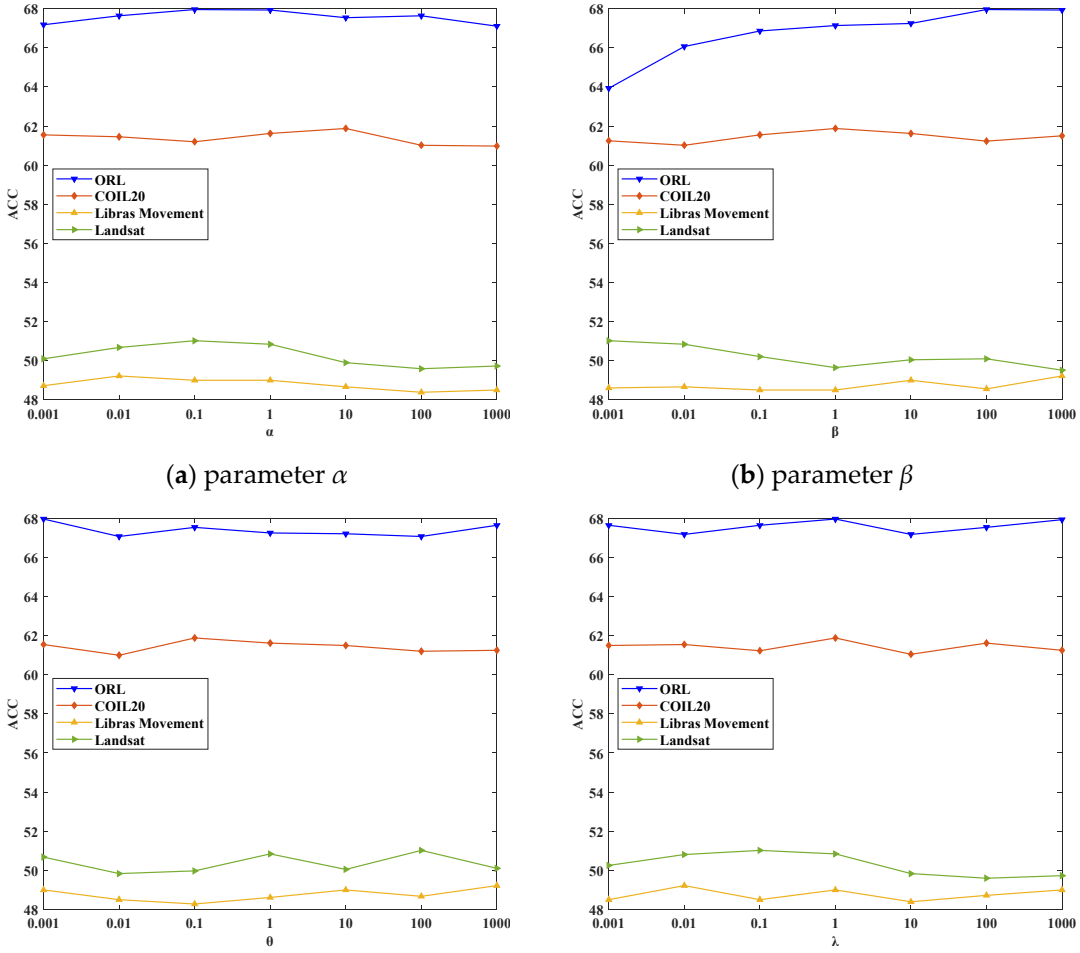
(2) Comparative analysis of clustering performance

In this experiment, the k-means method is adopted to cluster the low-dimensional features selected by each FS method. To minimize the impact of initialization on the k-means method, we performed 10 clustering experiments with varied random initializations. Tables 10–13 display the average values and standard deviations of ACC, NMI, purity, ARI, F-score, precision, and recall for the RLSR, FDEFS, GS^{3}FS, S2LFS, AGLRM, ASLCGLFS, and SFS-AGGL methods on the ORL, COIL20, Libras Movement, and Landsat datasets. These results further illustrate the superiority of the proposed SFS-AGGL compared to other comparative methods.

4.4. Convergence and Runtime Analysis

In this section, experiments were performed on seven databases to assess the convergence and runtime of the proposed SFS-AGGL method. Figure 9 shows the convergence curve of SFS-AGGL. From Figure 9, we can see that the objective function values of the SFS-AGGL methods only require less than 100 iterations to reach convergence, which validates the efficiency of the proposed iterative optimization method. Table 14 displays the runtime of SFS-AGGL when iteration is set to 100 and feature dimensions are set to 500. The results

in Table 14 clearly indicate that the runtime of our proposed method is slightly higher than that of AGLRM but lower than that of other methods. It is noteworthy that the runtime of SFS-AGGL is lower than that of all comparative methods after GPU optimization.



(**a**) parameter α (**b**) parameter β

(**c**) parameter θ (**d**) parameter λ

Figure 8. Clustering results of SFS-AGGL under different parameter values.

Table 10. The best clustering results of different methods on ORL dataset.

Method	ACC	NMI	Purity	ARI	F-Score	Precision	Recall
RLSR	62.79 ± 2.89 (500)	81.04 ± 1.83 (500)	66.93 ± 2.19 (500)	49.88 ± 3.78 (100)	51.13 ± 3.64 (100)	44.28 ± 4.33 (100)	60.75 ± 2.65 (500)
FDEFS	62.82 ± 3.69 (200)	81.27 ± 1.59 (100)	67.25 ± 3.08 (100)	50.13 ± 3.71 (100)	51.37 ± 3.60 (100)	44.55 ± 3.85 (100)	60.88 ± 3.64 (50)
GS^3FS	62.21 ± 1.55 (50)	80.99 ± 0.74 (50)	66.18 ± 1.33 (50)	49.86 ± 1.58 (50)	51.11 ± 1.53 (50)	44.17 ± 1.79 (50)	60.79 ± 2.32 (150)
S2LFS	61.93 ± 3.35 (350)	80.62 ± 1.45 (350)	66.82 ± 2.37 (350)	48.55 ± 3.61 (350)	49.82 ± 3.51 (350)	43.55 ± 3.60 (350)	58.74 ± 4.44 (400)
AGLRM	64.21 ± 3.70 (50)	81.84 ± 1.89 (50)	68.00 ± 3.14 (50)	51.16 ± 4.40 (50)	52.36 ± 4.26 (50)	45.80 ± 4.72 (50)	61.29 ± 4.00 (50)
ASLCGLFS	58.32 ± 3.68 (250)	78.56 ± 2.33 (250)	63.32 ± 3.10 (250)	44.22 ± 5.03 (250)	45.62 ± 4.85 (250)	39.37 ± 5.58 (300)	54.62 ± 3.95 (300)
SFS-AGGL	67.96 ± 2.30 (250)	84.17 ± 1.50 (400)	71.89 ± 1.94 (500)	56.89 ± 3.34 (400)	57.95 ± 3.25 (400)	50.69 ± 3.47 (400)	67.71 ± 3.11 (400)

Table 11. The best clustering results of different methods on COIL20 dataset.

Method	ACC	NMI	Purity	ARI	F-Score	Precision	Recall
RLSR	60.45 ± 3.98 (150)	72.19 ± 2.00 (250)	63.27 ± 3.19 (50)	50.84 ± 3.42 (300)	53.42 ± 3.19 (300)	48.67 ± 3.97 (300)	59.38 ± 2.56 (50)
FDEFS	58.52 ± 3.44 (50)	70.67 ± 2.62 (400)	61.55 ± 3.23 (50)	48.14 ± 4.12 (50)	50.93 ± 3.85 (50)	45.32 ± 4.36 (50)	58.42 ± 3.27 (150)
GS^3FS	59.98 ± 2.85 (150)	72.31 ± 1.41 (250)	63.38 ± 2.52 (150)	50.23 ± 1.84 (250)	52.86 ± 1.70 (250)	47.65 ± 2.60 (250)	59.47 ± 1.61 (250)
S2LFS	58.42 ± 3.90 (250)	70.19 ± 3.15 (250)	61.58 ± 3.61 (250)	46.40 ± 5.23 (450)	49.41 ± 4.74 (450)	42.64 ± 6.35 (250)	59.72 ± 2.47 (450)
AGLRM	59.85 ± 4.34 (150)	72.17 ± 2.59 (150)	63.17 ± 4.12 (150)	50.03 ± 4.46 (150)	52.71 ± 4.15 (150)	47.16 ± 5.17 (150)	60.05 ± 3.18 (300)
ASLCGLFS	60.02 ± 3.59 (50)	71.52 ± 1.67 (50)	62.95 ± 3.35 (50)	50.05 ± 2.56 (50)	52.67 ± 2.34 (50)	48.11 ± 3.81 (100)	58.55 ± 1.90 (50)
SFS-AGGL	61.88 ± 3.70 (350)	73.30 ± 1.78 (500)	64.67 ± 3.62 (350)	52.37 ± 1.80 (500)	54.85 ± 1.69 (500)	50.36 ± 3.16 (200)	62.16 ± 2.28 (500)

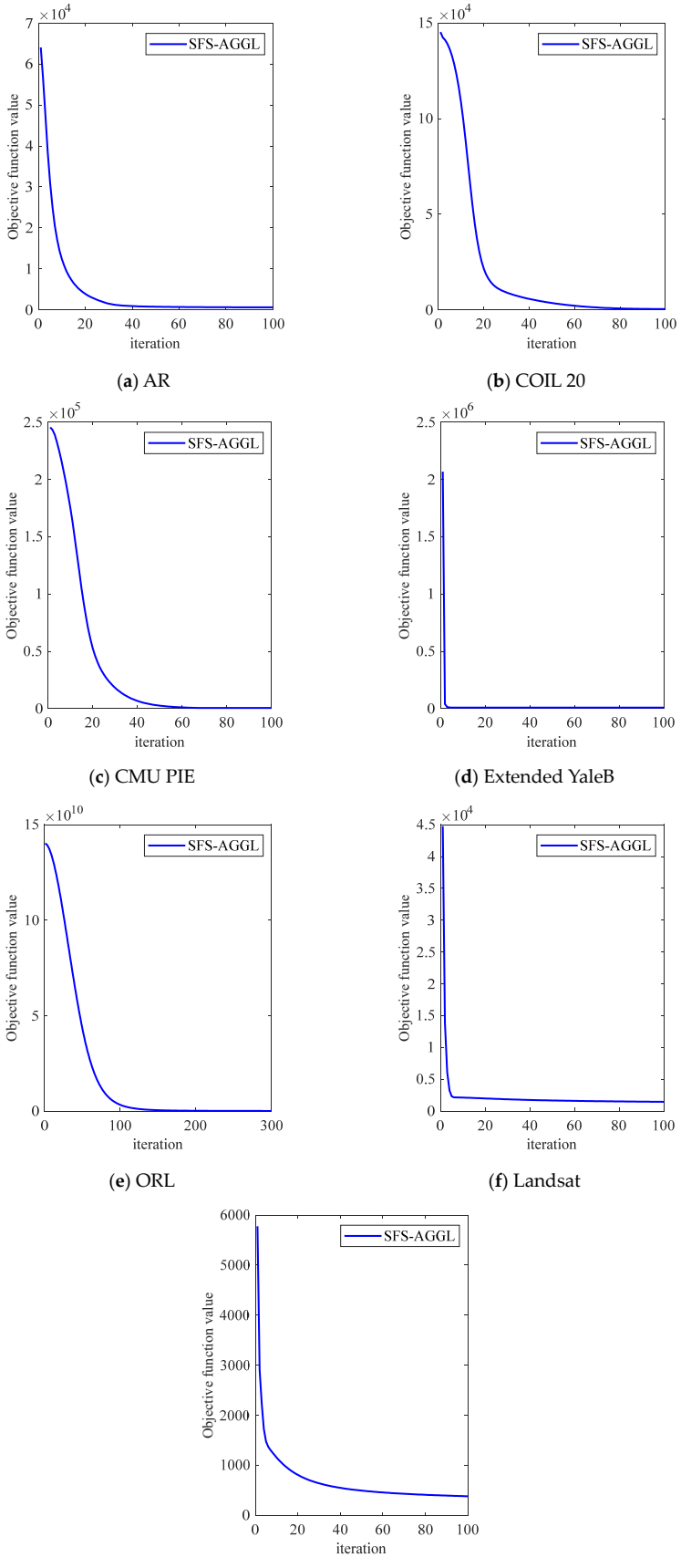
Table 12. The best clustering results of different methods on Libras Movement dataset.

Method	ACC	NMI	Purity	ARI	F-Score	Precision	Recall
RLSR	47.50 ± 2.21 (40)	60.07 ± 2.13 (56)	50.00 ± 1.85 (56)	30.04 ± 2.88 (56)	34.82 ± 2.69 (56)	31.38 ± 2.56 (56)	39.22 ± 3.70 (56)
FDEFS	46.33 ± 3.22 (32)	60.36 ± 2.88 (32)	50.22 ± 2.60 (24)	30.73 ± 3.77 (72)	35.58 ± 3.41 (72)	31.60 ± 3.65 (32)	41.28 ± 3.95 (72)
GS^3FS	46.72 ± 3.24 (80)	60.57 ± 1.97 (56)	50.94 ± 2.24 (80)	31.20 ± 2.68 (56)	36.01 ± 2.53 (56)	31.79 ± 2.32 (80)	41.93 ± 3.79 (56)
S2LFS	46.56 ± 1.92 (80)	59.95 ± 1.12 (64)	50.72 ± 1.34 (64)	30.28 ± 1.80 (80)	35.13 ± 1.82 (80)	30.97 ± 1.17 (80)	40.82 ± 4.28 (80)
AGLRM	46.00 ± 2.89 (56)	60.35 ± 1.32 (56)	50.72 ± 1.87 (72)	30.80 ± 1.92 (56)	35.62 ± 1.88 (56)	31.42 ± 1.45 (56)	41.33 ± 4.05 (56)
ASLCGLFS	46.28 ± 3.41 (40)	59.72 ± 2.30 (40)	50.17 ± 2.33 (40)	29.93 ± 2.97 (40)	34.84 ± 2.77 (40)	30.60 ± 2.68 (40)	41.04 ± 4.67 (80)
SFS-AGGL	49.22 ± 2.88 (72)	62.33 ± 2.34 (72)	53.11 ± 2.70 (72)	33.04 ± 2.65 (56)	37.75 ± 2.46 (56)	33.57 ± 3.13 (72)	44.42 ± 4.83 (80)

Table 13. The best clustering results of different methods on Landsat dataset.

Method	ACC	NMI	Purity	ARI	F-Score	Precision	Recall
RLSR	48.30 ± 2.10 (6)	45.97 ± 1.02 (18)	50.59 ± 1.88 (18)	33.94 ± 1.46 (27)	47.53 ± 1.91 (27)	38.53 ± 1.49 (3)	63.72 ± 8.11 (27)
FDEFS	47.89 ± 2.65 (30)	45.68 ± 1.59 (30)	50.60 ± 2.30 (30)	33.49 ± 1.83 (30)	47.10 ± 1.99 (24)	38.12 ± 1.25 (30)	63.03 ± 6.57 (18)
GS^3FS	49.10 ± 1.88 (15)	46.34 ± 1.05 (30)	51.37 ± 1.82 (21)	34.02 ± 1.33 (21)	47.69 ± 1.56 (21)	38.23 ± 1.33 (30)	64.44 ± 6.57 (21)
S2LFS	47.81 ± 2.63 (15)	45.99 ± 1.27 (30)	49.86 ± 2.53 (15)	34.03 ± 0.82 (15)	47.46 ± 1.24 (15)	38.83 ± 1.60 (30)	62.37 ± 7.02 (15)
AGLRM	49.06 ± 3.14 (9)	45.83 ± 1.40 (15)	50.97 ± 3.01 (9)	34.03 ± 1.62 (27)	47.23 ± 2.13 (27)	39.39 ± 1.48 (15)	60.78 ± 8.67 (27)
ASLCGLFS	48.79 ± 1.78 (30)	46.51 ± 1.41 (18)	50.59 ± 2.03 (24)	34.77 ± 0.83 (21)	48.33 ± 0.99 (21)	38.67 ± 1.17 (30)	65.35 ± 4.89 (21)
SFS-AGGL	51.02 ± 1.99 (12)	47.21 ± 1.18 (18)	52.81 ± 1.76 (12)	35.49 ± 1.23 (27)	49.04 ± 1.23 (18)	40.17 ± 1.32 (30)	69.26 ± 4.51 (15)

The numbers in parentheses denote the feature dimensions that yield the optimal results.



(**a**) AR (**b**) COIL 20

(**c**) CMU PIE (**d**) Extended YaleB

(**e**) ORL (**f**) Landsat

(**g**) Libras_Movement

Figure 9. Convergence curves of SFS-AGGL on different datasets.

Table 14. Runtime(s) of different methods on different datasets.

Method	AR	Extended YaleB	CMU PIE	ORL	COIL20
RLSR	28.3706	28.8286	27.0407	27.0545	23.2237
FDEFS	154.3282	190.8570	210.9357	55.6699	71.4754
GS^3FS	41.5550	44.3966	49.7174	28.2427	27.0868
S2LFS	52.7492	52.1703	54.4472	50.3727	47.9846
AGLRM	11.7974	13.1744	15.9342	3.4292	4.5080
ASLCGLFS	2690.2222	2477.4907	3735.1528	126.7054	340.7762
SFS-AGGL	15.5277	13.7843	17.4384	5.5204	6.7449
SFS-AGGL(GPU)	10.2069	9.3128	11.0843	3.3713	3.9999

5. Conclusions and Discussion

This paper proposes the semi-supervised feature selection based on an adaptive graph with global and local constraints (SFS-AGGL) algorithm. This algorithm considers the sample neighborhood structure within the projected feature space, dynamically learns the optimal nearest neighbor graph among samples, and maintains global and local sparse structures within the selected feature subset. This ensures the preservation of the original data's geometric structural information. Moreover, it can effectively leverage structural distribution information from labeled data to derive label information from unlabeled samples. The incorporation of the L_{21} norm in the SFS model enhances its resilience to noisy features. The iterative optimization approach employed to solve parameter optimal solutions is validated, confirming the convergence of the SFS-AGGL algorithm. Extensive experiments on real datasets validate the classification and clustering performance of the proposed SFS-AGGL method. Although our method can achieve good performance, there are still several issues that need to be pointed out, which are as follows:

1. Since the proposed method has considered the correlation and geometric structure of the data, it is suitable for the features of data with significant correlation, meanwhile, the distribution of data has a certain local structure.

2. Since our proposed method only considers the local and global structural information of the data, its application will be limited in certain datasets.

3. The proposed method cannot effectively extract effective features from the data with complex nonlinear structures because it is a linear feature selection method.

To overcome the above-mentioned shortcomings, we will try to do the following work in the future:

1. We will introduce other constraints to comprehensively capture and represent the structural information of the data.

2. We will integrate the idea of deep learning into the feature selection process to extract effective features from highly unstructured data.

Author Contributions: Data curation, Y.Y., H.Z., N.Z., G.X. and X.H.; Formal analysis, X.H., H.Z., N.Z., X.H. and G.X.; Methodology, Y.Y., H.Z., W.Z. and C.Z.; Resources, Y.Y., H.Z., N.Z. and W.Z.; Supervision, W.Z. and C.Z.; Writing—original draft, Y.Y. and H.Z.; Writing—review and editing, Y.Y., H.Z., W.Z. and C.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported in part by grants from the National Natural Science Foundation of China (Nos. 62062040 and 62006174), the Outstanding Youth Project of Jiangxi Natural Science Foundation (No. 20212ACB212003), the Jiangxi Province Key Subject Academic and Technical Leader Funding Project (No. 20212BCJ23017), the Science and Technology Research Project of Jiangxi Provincial Department of Education (No. GJJ210330), and the Fund of the Jilin Provincial Science and Technology Department (No. 20220201157GX).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

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

References

1. Wen, J.; Yang, S.; Wang, C.D.; Jiang, Y.; Li, R. Feature-splitting Algorithms for Ultrahigh Dimensional Quantile Regression. *J. Econom.* **2023**, *2023*, 105426. [CrossRef]
2. Lue, X.; Long, L.; Deng, R.; Meng, R. Image feature extraction based on fuzzy restricted Boltzmann machine. *Measurement* **2022**, *204*, 112063. [CrossRef]
3. Sheikhpour, R.; Sarram, M.A.; Gharaghani, S.; Chahooki, M.A.Z. A survey on semi-supervised feature selection methods. *Pattern Recognit.* **2017**, *64*, 141–158. [CrossRef]
4. Mafarja, M.; Qasem, A.; Heidari, A.A.; Aljarah, I.; Faris, H.; Mirjalili, S. Efficient hybrid nature-inspired binary optimizers for feature selection. *Cogn. Comput.* **2020**, *12*, 150–175. [CrossRef]
5. Huang, G.Y.; Hung, C.Y.; Chen, B.W. Image feature selection based on orthogonal $\ell_{2,0}$ norms. *Measurement* **2022**, *199*, 111310. [CrossRef]
6. Cai, J.; Luo, J.; Wang, S.; Yang, S. Feature selection in machine learning: A new perspective. *Neurocomputing* **2018**, *300*, 70–79. [CrossRef]
7. Solorio-Fernández, S.; Carrasco-Ochoa, J.A.; Martínez-Trinidad, J.F. A systematic evaluation of filter Unsupervised Feature Selection methods. *Expert Syst. Appl.* **2020**, *162*, 113745. [CrossRef]
8. Bhadra, T.; Bandyopadhyay, S. Supervised feature selection using integration of densest subgraph finding with floating forward–backward search. *Inf. Sci.* **2021**, *566*, 1–18. [CrossRef]
9. Mann, G.S.; McCallum, A. Generalized Expectation Criteria for Semi-Supervised Learning with Weakly Labeled Data. *J. Mach. Learn. Res.* **2010**, *11*, 955–984.
10. 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.
11. Wang, L.; Jiang, S.; Jiang, S. A feature selection method via analysis of relevance, redundancy, and interaction. *Expert Syst. Appl.* **2021**, *183*, 115365. [CrossRef]
12. Dokeroglu, T.; Deniz, A.; Kiziloz, H.E. A comprehensive survey on recent metaheuristics for feature selection. *Neurocomputing* **2022**, *494*, 2966. [CrossRef]
13. Nie, F.; Zhu, W.; Li, X. Structured graph optimization for unsupervised feature selection. *IEEE Trans. Knowl. Data Eng.* **2019**, *33*, 1210–1222. [CrossRef]
14. Zhao, Z.; Liu, H. Semi-supervised feature selection via spectral analysis. In Proceedings of the 2007 SIAM International Conference on Data Mining; Society for Industrial and Applied Mathematics, Minneapolis, MN, USA, 26–28 April 2007; pp. 641–646.
15. Toğaçar, M.; Ergen, B.; Cömert, Z. Classification of flower species by using features extracted from the intersection of feature selection methods in convolutional neural network models. *Measurement* **2020**, *158*, 107703. [CrossRef]
16. Chen, X.; Song, L.; Hou, Y.; Shao, G. Efficient semi-supervised feature selection for VHR remote sensing images. In Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2016; pp. 1500–1503.
17. Peng, S.; Lu, J.; Cao, J.; Peng, Q.; Yang, Z. Adaptive graph regularization method based on least square regression for clustering. *Signal Process. Image Commun.* **2023**, *114*, 116938. [CrossRef]
18. Chang, X.; Nie, F.; Yang, Y.; Huang, H. A convex formulation for semi-supervised multi-label feature selection. In Proceedings of the AAAI Conference on Artificial Intelligence, Québec City, QC, Canada, 27–31 July 2014; Volume 28.
19. Chen, X.; Yuan, G.; Nie, F.; Huang, J.Z. Semi-supervised feature selection via rescaled linear regression. In Proceedings of the Twenty Sixth International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; pp. 1525–1531.
20. Chen, X.; Chen, R.; Wu, Q.; Nie, F.; Yang, M.; Mao, R. Semi supervised feature selection via structured manifold learning. *IEEE Trans. Cybern.* **2021**, *52*, 5756–5766. [CrossRef]
21. Liu, Z.; Lai, Z.; Ou, W.; Zhang, K.; Zheng, R. Structured optimal graph based sparse feature extraction for semi-supervised learning. *Signal Process.* **2020**, *170*, 107456. [CrossRef]
22. Akbar, S.; Hayat, M.; Tahir, M.; Chong, K.T. cACP-2LFS: Classification of anticancer peptides using sequential discriminative model of KSAAP and two-level feature selection approach. *IEEE Access* **2020**, *8*, 131939–131948. [CrossRef]
23. Bakir-Gungor, B.; Hacilar, H.; Jabeer, A.; Nalbantoglu, O.U.; Aran, O.; Yousef, M. Inflammatory bowel disease biomarkers of human gut microbiota selected via ensemble feature selection methods. *PeerJ* **2022**, *10*, e13205. [CrossRef]
24. Ahmed, N.; Rafiq, J.I.; Islam, M.R. Enhanced human activity recognition based on smartphone sensor data using hybrid feature selection model. *Sensors* **2020**, *20*, 317. [CrossRef]
25. López, D.; Ramírez-Gallego, S.; García, S.; Xiong, N.; Herrera, F. BELIEF: A distance-based redundancy-proof feature selection method for Big Data. *Inf. Sci.* **2021**, *558*, 124–139. [CrossRef]
26. Chen, X.; Yuan, G.; Wang, W.; Nie, F.; Chang, X.; Huang, J.Z. Local adaptive projection framework for feature selection of labeled and unlabeled data. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 6362–6373. [CrossRef] [PubMed]
27. Cheng, B.; Yang, J.; Yan, S.; Fu, Y.; Huang, T.S. Learning with l1-graph for image analysis. *IEEE Trans. Image Process.* **2009**, *19*, 858–866. [CrossRef]
28. Liu, G.; Lin, Z.; Yan, S.; Sun, J.; Yu, Y.; Ma, Y. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 171–184. [CrossRef]
29. Singh, R.P.; Ojha, D.; Jadon, K.S. A Survey on Various Representation Learning of Hypergraph for Unsupervised Feature Selection. In *Data, Engineering and Applications: Select Proceedings of IDEA 2021*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 71–82.

30. Elhamifar, E.; Vidal, R. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 2765–2781. [CrossRef]
31. Zhong, G.; Pun, C.M. Subspace clustering by simultaneously feature selection and similarity learning. *Knowl. Based Syst.* **2020**, *193*, 105512. [CrossRef]
32. Wan, Y.; Sun, S.; Zeng, C. Adaptive similarity embedding for unsupervised multi-view feature selection. *IEEE Trans. Knowl. Data Eng.* **2020**, *33*, 3338–3350. [CrossRef]
33. Shang, R.; Song, J.; Jiao, L.; Li, Y. Double feature selection algorithm based on low-rank sparse non-negative matrix factorization. *Int. J. Mach. Learn. Cybern.* **2020**, *11*, 1891–1908. [CrossRef]
34. Zhu, J.; Jang-Jaccard, J.; Liu, T.; Zhou, J. Joint spectral clustering based on optimal graph and feature selection. *Neural Process. Lett.* **2021**, *53*, 257–273. [CrossRef]
35. Sha, Y.; Faber, J.; Gou, S.; Liu, B.; Li, W.; Schramm, S.; Stoecker, H.; Steckenreiter, T.; Vnucec, D.; Wetzstein, N.; et al. An acoustic signal cavitation detection framework based on XGBoost with adaptive selection feature engineering. *Measurement* **2022**, *192*, 110897. [CrossRef]
36. Zhu, P.; Hou, X.; Tang, K.; Liu, Y.; Zhao, Y.P.; Wang, Z. Unsupervised feature selection through combining graph learning and $\ell 2$, 0-norm constraint. *Inf. Sci.* **2023**, *622*, 68–82. [CrossRef]
37. Mei, S.; Zhao, W.; Gao, Q.; Yang, M.; Gao, X. Joint feature selection and optimal bipartite graph learning for subspace clustering. *Neural Netw.* **2023**, *164*, 408–418. [CrossRef] [PubMed]
38. Zhou, P.; Du, L.; Li, X.; Shen, Y.D.; Qian, Y. Unsupervised feature selection with adaptive multiple graph learning. *Pattern Recognit.* **2020**, *105*, 107375. [CrossRef]
39. Bai, X.; Zhu, L.; Liang, C.; Li, J.; Nie, X.; Chang, X. Multi-view feature selection via nonnegative structured graph learning. *Neurocomputing* **2020**, *387*, 110–122. [CrossRef]
40. Zhou, P.; Chen, J.; Du, L.; Li, X. Balanced spectral feature selection. *IEEE Trans. Cybern.* **2022**, *53*, 4232–4244. [CrossRef]
41. 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]
42. Xie, G.B.; Chen, R.B.; Lin, Z.Y.; Gu, G.S.; Yu, J.R.; Liu, Z.; Cui, J.; Lin, L.; Chen, L. Predicting lncRNA–disease associations based on combining selective similarity matrix fusion and bidirectional linear neighborhood label propagation. *Brief. Bioinform.* **2023**, *24*, bbac595. [CrossRef]
43. Sheikhpour, R.; Sarram, M.A.; Gharaghani, S.; Chahooki, M.A.Z. A robust graph-based semi-supervised sparse feature selection method. *Inf. Sci.* **2020**, *531*, 13–30. [CrossRef]
44. Li, Z.; Tang, J. Semi-supervised local feature selection for data classification. *Sci. China Inf. Sci.* **2021**, *64*, 192108. [CrossRef]
45. Jiang, B.; Wu, X.; Zhou, X.; Liu, Y.; Cohn, A.G.; Sheng, W.; Chen, H. Semi-supervised multiview feature selection with adaptive graph learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, 1–15. [CrossRef]
46. Shang, R.; Zhang, X.; Feng, J.; Li, Y.; Jiao, L. Sparse and low-dimensional representation with maximum entropy adaptive graph for feature selection. *Neurocomputing* **2022**, *485*, 57–73. [CrossRef]
47. Lai, J.; Chen, H.; Li, T.; Yang, X. Adaptive graph learning for semi-supervised feature selection with redundancy minimization. *Inf. Sci.* **2022**, *609*, 465–488. [CrossRef]
48. Lai, J.; Chen, H.; Li, W.; Li, T.; Wan, J. Semi-supervised feature selection via adaptive structure learning and constrained graph learning. *Knowl.-Based Syst.* **2022**, *251*, 109243. [CrossRef]
49. Luo, T.; Hou, C.; Nie, F.; Tao, H.; Yi, D. Semi-supervised feature selection via insensitive sparse regression with application to video semantic recognition. *IEEE Trans. Knowl. Data Eng.* **2018**, *30*, 1943–1956. [CrossRef]
50. Moosaei, H.; Hladík, M. Sparse solution of least-squares twin multi-class support vector machine using $\ell 0$ and ℓp-norm for classification and feature selection. *Neural Netw.* **2023**, *166*, 471–486. [CrossRef]
51. Favati, P.; Lotti, G.; Menchi, O.; Romani, F. Construction of the similarity matrix for the spectral clustering method: Numerical experiments. *J. Comput. Appl. Math.* **2020**, *375*, 112795. [CrossRef]
52. Qu, J.; Zhao, X.; Xiao, Y.; Chang, X.; Li, Z.; Wang, X. Adaptive Manifold Graph representation for Two-Dimensional Discriminant Projection. *Knowl.-Based Syst.* **2023**, *266*, 110411. [CrossRef]
53. Ma, Z.; Wang, J.; Li, H.; Huang, Y. Adaptive graph regularized non-negative matrix factorization with self-weighted learning for data clustering. *Appl. Intell.* **2023**, *53*, 28054–28073. [CrossRef]
54. Yang, S.; Wen, J.; Zhan, X.; Kifer, D. ET-lasso: A new efficient tuning of lasso-type regularization for high-dimensional data. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 607–616.
55. Huang, S.; Xu, Z.; Wang, F. Nonnegative matrix factorization with adaptive neighbors. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 486–493.
56. 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]
57. 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]
58. Martinez, A.; Benavente, R. *The AR Face Database: CVC Technical Report*; Computer Vision Center: Barcelona, Spain, 1998; Volume 24.

59. 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, 20–21 May 2002; pp. 53–58.
60. Zhang, L.; Zhang, L.; Zhang, D.; Zhu, H. Online finger-knuckle-print verification for personal authentication. *Pattern Recognit.* **2010**, *43*, 2560–2571. [CrossRef]
61. Samaria, F.S.; Harter, A.C. Parameterisation of a stochastic model for human face identification. In Proceedings of the 1994 IEEE Workshop on Applications of Computer Vision, Seattle, WA, USA, 21–23 June 1994; pp. 138–142.
62. Nene, S.A.; Nayar, S.K.; Murase, H. *Columbia Object Image Library (COIL-20)*; Columbia University: New York, NY, USA, 1996.
63. Yi, Y.; Lai, S.; Li, S.; Dai, J.; Wang, W.; Wang, J. RRNMF-MAGL: Robust regularization non-negative matrix factorization with multi-constraint adaptive graph learning for dimensionality reduction. *Inf. Sci.* **2023**, *640*, 119029. [CrossRef]
64. Blake, C.L.; Merz, C.J. *UCI Repository of Machine Learning Databases*; Department of Information and Computer Science, University of California: Irvine, CA, USA, 1998; p. 55.
65. Li, Z.; Tang, C.; Zheng, X.; Liu, X.; Zhang, W.; Zhu, E. High-order correlation preserved incomplete multi-view subspace clustering. *IEEE Trans. Image Process.* **2022**, *31*, 2067–2080. [CrossRef] [PubMed]