Journal of Mathematical Modeling
Vol. -, No. -, 2026, pp. -. Research Article

JMM

A robust unsupervised feature selection based on subspace learning and adaptive graph structure

Hazhir Sohrabi[†], **Shahrokh Esmaeili**[†*], **Parham Moradi**[‡]

[†]*Department of Applied Mathematics, University of Kurdistan, Sanandaj, Iran*
[‡]*School of Engineering, RMIT University, Melbourne, Australia*
Email(s): h.sohrabi@uok.ac.ir, sh.esmaeili@uok.ac.ir, parham.moradi@rmit.edu.au

Abstract. Feature selection is vital for improving high-dimensional data analysis by identifying a subset of representative and uncorrelated features. This paper presents an unsupervised feature selection algorithm based on subspace learning and adaptive graph structure (UFSAG). The UFSAG uses matrix factorization to preserve global data structure and incorporates local correlations into its objective function. It also integrates sample similarity graph learning to maintain data geometry. Unlike prior methods, UFSAG employs adaptive local structure learning to reduce noise and enhance feature selection. By inducing row sparsity in the feature coefficient matrix using the $\ell_{2,1}$-norm, UFSAG identifies representative features. Comparative experiments on six datasets show UFSAG's superior clustering performance over twelve state-of-the-art methods.

Keywords: Matrix factorization, feature selection, local correlation, data manifold, clustering.
AMS Subject Classification 2020: 15A23, 68T05, 68T30

1 Introduction

Recent technological advancements have led to an explosion of high-dimensional datasets across diverse domains including bioinformatics and social network analysis. These datasets suffer from the well-known "curse of dimensionality" [3]. This phenomenon poses significant challenges for traditional machine learning methods, including increased computational complexity, sparsity issues, and the presence of redundant and irrelevant features [19], which can hinder model performance and increase overfitting risks.

To address these challenges, dimensionality reduction techniques, particularly feature selection (FS), have become essential. Feature selection methods identify relevant and non-redundant features, improving model interpretability while reducing computational costs and enhancing generalization perfor-

[*] Corresponding author

Received: 30 March 2025/ Revised: 30 September 2025/ Accepted: 30 September 2025
DOI: 10.22124/jmm.2025.30225.2714

mance [8]. These approaches have widespread applications across various fields, including medicine, artificial intelligence, engineering, and finance [1, 2, 11, 24, 25].

Unsupervised feature selection (UFS) is crucial for high-dimensional data analysis, identifying feature subsets that preserve intrinsic data structure without label information. We review existing UFS methods, categorizing them thematically to highlight contributions and limitations, setting the stage for our proposed unsupervised feature selection algorithm (UFSAG).

In recent years, subspace-based feature selection methods [6, 15] have gained prominence by exploiting the underlying structure of high-dimensional data. Unlike traditional methods that evaluate features individually, these approaches consider feature relationships within lower-dimensional subspaces where intrinsic structure is more discernible. However, existing subspace-based methods like MFFS [33], RMFFS [27], and UFGOR [28] suffer from several key limitations. First, they often fail to preserve local structural integrity. Second, they typically rely on heuristic search strategies [22] that may not guarantee optimal solutions. Third, their performance can degrade significantly in the presence of noise and outliers [34].

Matrix factorization (MF) has proven effective for UFS by reconstructing data spaces while identifying informative features. Wang et al. [33] proposed MFFS, formulating feature selection as a matrix factorization problem with orthogonal constraints. However, MFFS struggles with orthogonal constraints and ignores feature correlations and local data structure. Qi et al. [27] developed RMFFS, incorporating ℓ_1 and ℓ_2 regularization to reduce feature redundancy. While improving on MFFS, RMFFS neglects sample correlations important for data structure. Parsa et al. [26] introduced DLUFS, using low-rank constraints to eliminate noisy features and spectral analysis for local structure, but lacks a unified local-global structure framework.

Graph-based methods excel at capturing data geometry. Cai et al. [5] pioneered this with GNMF, using Laplacian matrices to preserve local structure. Extensions include GDNMF [17] emphasizing discriminative power, and DSNMF [9] incorporating supervised constraints. Sohrabi et al. [30] recently proposed global threshold-based graph construction, outperforming local threshold methods. While effective for local structures, these methods rely on fixed similarity graphs vulnerable to noise. Wu et al. [36] developed DENMF with dual regularization, but like others, suffers from inflexible graph structures.

Recent UFS advances focus on robustness and adaptability. Lim et al. [18] proposed DUFS prioritizing feature dependence over local structures, but neglecting global preservation. Adaptive graph learning represents significant progress, with Tang et al. [31] developing outlier-resistant adaptive graphs, and [35] proposing deep NMF with adaptive graphs. However, these often lack comprehensive local-global structure integration.

Current UFS methods face three key challenges: (1) inadequate integration of local and global structures, (2) sensitivity to noise due to fixed graphs, and (3) interpretability and scalability issues in adaptive methods.

The proposed method addresses these limitations through several key innovations:

- Preserves global structure via matrix factorization while capturing local relationships through Laplacian graph regularization.

- Employs an adaptive graph learning mechanism that dynamically adjusts to the data, ensuring robustness to outliers and outperforming static approaches such as DLUFS [26].

- Enhances feature discriminability by reducing redundancy and leveraging the $\ell_{2,1}$-norm to induce row sparsity in feature coefficients.
- Improves noise resilience through robust regularization techniques, leading to more stable feature selection.
- Unifies subspace learning, adaptive graph learning, and robust local structure preservation within the UFSAG framework, enabling superior high-dimensional data analysis.

The UFSAG framework bridges these gaps by unifying subspace learning, adaptive graph learning, and robust local structure preservation. Its adaptive graph dynamically captures data structure, ensuring robustness to outliers, while matrix factorization maintains global structure. This comprehensive approach enables superior performance in high-dimensional data analysis. Experimental results on benchmark datasets show our method consistently outperforms state-of-the-art techniques in clustering accuracy and feature selection quality.

The remainder of this paper is organized as follows: Section 2 details our methodology, Section 3 present the optimization algorithm, analyze its complexity and convergence, Section 4 presents experimental results, and Section 5 concludes with future research directions.

2 Proposed method

We use the notation $\mathbb{R}_+^{n\times m}$ to denote the space of $n \times m$ nonnegative matrices. Given a data set $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, the notation $X = [\mathbf{x}_1; \mathbf{x}_2; \ldots; \mathbf{x}_n] = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_m] \in \mathbb{R}_+^{n\times m}$ represents the data matrix, where n signifies the number of samples and m denotes the number of features.

2.1 Feature selection via matrix factorization

An innovative UFS criterion rooted in principles of subspace learning was introduced by Wang et al. [33], formulating it as a matrix factorization problem. In contrast to previous representation-based approaches that utilize all features for self-reconstruction, the method operates under the assumption that a small discriminative feature subset of size $k < m$, suffices to effectively represent the entire feature set. The index set of chosen features is denoted by I, where $|I| = k$. In this context, a matrix factorization problem quantifies the separation between the initial data space X and the feature subspace corresponding to the specified feature subset X_I:

$$\min_{W,H} \frac{1}{2}\|X - XWH\|_F^2 \qquad \text{s.t. } W \geq 0,\ H \geq 0,\ W^\top W = I_k, \tag{1}$$

where I_k is the $k \times k$ identity matrix. Here, $W \in \mathbb{R}_+^{m\times k}$ is an indicator matrix representing the selected features, while $H \in \mathbb{R}_+^{k\times m}$ serves as the coefficient matrix mapping the original feature space to the subspace formed by the selected features. In this approach, it is assumed that the feature subset X_I can be expressed as the matrix product $X_I = XW$. This assumption justifies the imposition of the constraint $W^\top W = I_k$, wherein each entry of W is either 1 or 0, and each row or column of W has at most one nonzero element. This is an optimization problem on a manifold. Specifically, constraint sets of the form $\{W \in \mathbb{R}_+^{m\times k} : W^\top W = I_k\}$ form an embedded submanifold of $\mathbb{R}_+^{m\times k}$, known as the (orthogonal) Stiefel manifold [7].

It appears that subspace learning is intricately linked to the *concept factorization*, wherein NMF for data clustering is extended [37]. In concept factorization, each cluster center (concept) $[X_I]_{:c}$ is modeled as a nonnegative linear combination of data points $X_{:j}$, such that $[X_I]_{:c} = \sum_{j=1}^{m} W_{jc} X_{:j}$. Additionally, each data point $X_{:j}$ is modeled as a nonnegative linear combination of the cluster centers $[X_I]_{:c}$, given by $X_{:j} \approx \sum_{c=1}^{k} H_{cj} [X_I]_{:c}$. By combining these two relations, we obtain

$$X_{:i} \approx \sum_{c=1}^{k} H_{ci} \left(\sum_{j=1}^{m} W_{jc} X_{:j} \right).$$

In this manner, we obtain the approximation $X \approx XWH$, upon which the formulation of the objective function in problem (1) relies.

In recent years, there has been a trend to reformulate the objective function represented in equation (1) using QR factorization to achieve more advantageous outcomes [29]. Typically, the reduced QR factorization of matrix X is represented as $X = QR$, where Q is $n \times m$ with orthonormal columns and R is $m \times m$ and upper triangular. From this standpoint, due to the unitary invariance property of the Frobenius norm, we also propose a straightforward method whereby the optimization problem defined in (1) is redefined as follows:

$$\min_{W,H} \frac{1}{2} \|R - RWH\|_F^2 \qquad \text{s.t. } W \geq 0,\ H \geq 0,\ W^\top W = I_k.$$

This alternative formulation may offer computational advantages compared to the original problem. However, we do not pursue this approach.

2.2 Data structure learning

The internal geometric structure information of the data space can be learned by constructing the graph associated with data points. Let us view the row representation of the matrix $X_I = XW$ as a mapping from $\mathbb{R}_+^{n \times m}$ to $\mathbb{R}_+^{n \times k}$:

$$[\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_n]^\top \mapsto [W\mathbf{x}_1 \quad W\mathbf{x}_2 \quad \cdots \quad W\mathbf{x}_n]^\top.$$

In manifold learning theory [10], it is commonly assumed that when two data samples are close in the original space, their corresponding representations in a projected space should also be close. Following this premise, if two data samples, denoted as $\mathbf{x}_i$ and $\mathbf{x}_j$, exhibit similarity, it is expected that their corresponding mapped vectors $\mathbf{x}_i W$ and $\mathbf{x}_j W$ in a reduced dimensionality space will also display similarity.

Allowing $G \in \mathbb{R}_+^{n \times n}$ to be a matrix representing the similarity between each pair of nodes $\mathbf{x}_i$ and $\mathbf{x}_j$, the task of learning the manifold for the linear transformation $X \mapsto XW$ can be formulated as follows:

$$\min_W \mathscr{B}(W) \equiv \frac{1}{2} \sum_{i,j=1}^{n} \|\mathbf{x}_i W - \mathbf{x}_j W\|^2 G_{ij}, \tag{2}$$

to ensure that $\|\mathbf{x}_i W - \mathbf{x}_j W\|$ is anticipated to have a small value when G_{ij} is large. But then we can write

$$\mathscr{B}(W) = \frac{1}{2} \sum_{i,j=1}^{n} (\mathbf{x}_i W - \mathbf{x}_j W)(\mathbf{x}_i W - \mathbf{x}_j W)^\top G_{ij} = \sum_{i=1}^{n} (\mathbf{x}_i W W^\top \mathbf{x}_i^\top) D_{ii} - \sum_{i,j=1}^{n} (\mathbf{x}_i W W^\top \mathbf{x}_j^\top) G_{ij},$$

where $D_{ii} = \sum_{j=1}^{n} G_{ij}$. Subsequently, employing trace notation, we can express (2) as

$$\min_{W} \mathscr{B}(W) \equiv \mathrm{Tr}(W^{\top} X^{\top} L X W), \tag{3}$$

where $L = D - G$ represents the Laplacian matrix and $D = \mathrm{diag}(D_{11}, \dots, D_{nn})$. It is noteworthy to mention that in this paper, G represents the similarity matrix and is defined using various approaches, including the adaptive graph method.

There is an ongoing aspiration and endeavor to either initially choose a set of orthogonal features or guide them toward meeting orthogonality conditions. Essentially, the columns of a data matrix exhibit less redundancy when they are less similar, and achieving this involves constructing a nearly orthogonal set from the original data matrix. To this end, the following criterion can be applied to a given submatrix of features $X_I = [\mathbf{f}_{i_1}, \dots, \mathbf{f}_{i_k}]$:

$$\mathrm{Corr}(X_I) = \frac{1}{k^2} \sum_{r=1}^{k} \sum_{s=1}^{k} \mathbf{f}_{i_r}^{\top} \mathbf{f}_{i_s} = \frac{1}{k^2} \mathrm{Tr}(X_I^{\top} X_I \mathbf{1}\mathbf{1}^{\top}). \tag{4}$$

Here, $\mathbf{1}$ is a column vector with all entries equal to 1. It is crucial to acknowledge that when the angle between two distinct features approaches 90 degrees, the corresponding term in Corr will tend towards zero.

2.3 Adaptive local structure graph

Various adaptive strategies have been proposed to learn local structures and improve clustering performance [23]. For example, [14] employs information entropy to construct adaptive graphs, overcoming the limitations of k-NN-based methods. Recent approaches, such as RRNMF-MAGL [39] and DPSS-RBFNN [42], also focus on learning local graph structures, each with distinct characteristics.

However, most existing methods rely on individual data points, which can result in inaccurate affinity matrices in noisy environments. As demonstrated in [40], the number of neighbors should be adjusted based on their proximity to clusters. Motivated by this insight, we adopt an adaptive local structure learning technique that dynamically captures the data manifold while mitigating the influence of outliers.

In accordance with the findings presented in [31], we construct the adaptive manifold graph $S \in \mathbb{R}_{+}^{n \times n}$. In particular, we initially establish $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|^2$, and arrange the set $\{d_{i1}, \dots, d_{in}\}$ in ascending order. Secondly, unlike the conventional local structure learning approach that assigns an individual threshold for each node, we introduce a global threshold $\varepsilon_p = \frac{1}{n} \sum_{i=1}^{n} d_{i,p+2}$, where p is a parameter that controls the sparsity of S. The manual selection of parameter p plays a crucial role in mitigating overfitting and enhancing noise control in noisy environments. By judiciously adjusting p, we can effectively regulate the sparsity of the adaptive graph S, thereby improving the robustness of our model. Subsequently, we specify that only the pairs of nodes $(\mathbf{x}_i, \mathbf{x}_j)$ for which the resulting value d_{ij} is less than the threshold ε_p are eligible to become neighbors, i.e., $S_{ij} = 0$ if $d_{ij} \geq \varepsilon_p$, and $S_{ij} > 0$ otherwise. Here, the matrix V is defined as $V_{ij} = d_{ij} - \varepsilon_p$. Then, the following strategy is introduced for learning an adaptive local structure graph

$$\min_{S \geq 0} \|V \odot S\|_{\diamond} + \eta \|S\|_F^2, \tag{5}$$

where $\|A\|_{\diamond} = \sum_{i,j} A_{ij}$ represents the sum of the entries of A and the notation $\odot$ is used to indicate the Hadamard product between two matrices. The second term serves as a regularization term, with $\eta \geq 0$

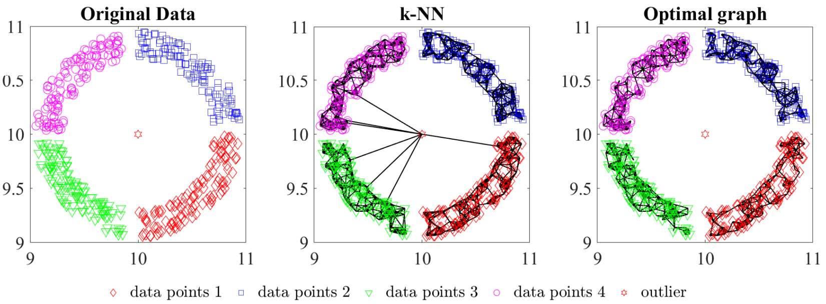

Figure 1: Comparison of robustness in adaptive local structure learning strategies

acting as a flexible parameter that controls the density of the adaptive graph S. Some implicit conditions, such as $\sum_{j=1}^{n} S_{ij} = 1$ and $S_{ij} \geq 0$, are implied by this problem. Based on these constraints, we partition problem (5) into n subproblems, each corresponding to an individual sample. Subsequently, to address each of these subproblems, we construct the Lagrangian function as follows:

$$\mathscr{L}_i(S_{i:},\tau,\beta_i) = \frac{1}{2}\left\|S_{i:}+\frac{1}{2\eta_i}V_{i:}\right\|^2 - \tau(S_{i:}\mathbf{1}-1) - S_{i:}\beta_i,$$

where $\tau \geq 0$ and $\beta_i \geq 0$ are the Lagrangian multipliers. Applying the Karush-Kuhn-Tucker (KKT) conditions, the solution to the corresponding subproblem can be determined as $S_{ij} = \max(-V_{ij}/(2\eta_i)+\tau,0)$. As indicated in [23], if the optimal $S_{i:}$ consist of only p nonzero elements, the multiplier τ and the parameter η_i can be derived. For computational convenience, the overall parameter η can be set as the mean of $\eta_1,\eta_2,\ldots,\eta_n$, denoted as

$$\eta = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{p}{2}d_{i,p+2} - \frac{1}{2}\sum_{j=2}^{p+1} d_{ij}\right) = \frac{p}{2}\varepsilon_p - \frac{1}{2n}\sum_{i=1}^{n}\sum_{j=2}^{p+1} d_{ij}.$$

Certainly, this adjustment to the adaptive graph S can also be achieved by assigning probabilistic interpretations, such that $\frac{1}{n}\sum_{i=1}^{n}\sum_{j=2}^{p+1} S_{ij} \approx 1$. Ultimately, the solution to optimization problem (5) is attained through

$$S_{ij} = \max\left(\frac{\varepsilon_p - d_{ij}}{2\eta}, 0\right). \tag{6}$$

The adaptive graph S, formed in this manner, lacks symmetry. Therefore, in constructing the Laplacian matrix, we employ the formula $L_S = D - (S + S^\top)/2$.

As stated in [31], the adaptive graph strategy (5) exhibits notable robustness against outliers. Testing this concept on a synthetic dataset comprised of three clusters plus noise. The k-NN strategy is designated as the control group. Figure 1 presents specific results, indicating that only equation (5) accurately constructs the spatial network of non-outliers. Conversely, k-NN fails to correctly identify the spatial structure under the influence of an outlier.

2.4 Significance of the coefficient matrix

Remember that H represents the coefficient matrix of the original feature space within the selected feature space. It is a main component of the feature selection problem (1). In the subsequent discussion, we delve deeper into the significance of H, excerpted from [29].

It can be deduced from $X \approx X_I H$ that the columns of matrix X can be expressed approximately as a linear combination of the columns of X_I:

$$\mathbf{f}_j \approx \sum_{r=1}^{k} H_{rj}\mathbf{f}_{i_r}, \quad j = 1, \ldots, m. \tag{7}$$

Each column vector $H_{:j}$ in H represents coefficients for the corresponding feature vector $\mathbf{f}_j$ in (7), with H_{rj} denoting the coefficient of $\mathbf{f}_{i_r}$. Greater sparsity in $H_{:j}$ helps identify key feature vectors while eliminating less relevant ones.

In the feature selection framework (1), enforcing sparsity in $H_{:j}$ aids in detecting redundant features and selecting a distinctive subset of $\mathbf{f}_j$. Among sparsity-inducing methods, $\ell_{2,1}$-norm regularization is particularly effective for promoting sparsity in matrix columns or rows [22]. Thus, it is beneficial to incorporate $\ell_{2,1}$-norm regularization for $H \in \mathbb{R}_+^{k\times m}$ as

$$\min_H \|H\|_{2,1} \equiv \mathrm{Tr}(HUH^\top), \tag{8}$$

where U is a diagonal $m \times m$ matrix with entries

$$U_{jj} = \frac{1}{\max(\|H_{:j}\|, \varepsilon)}, \quad j = 1, \ldots, m, \tag{9}$$

and ε is a small constant preventing overflow errors.

2.5 Objective function

This research hypothesizes that replacing data matrix X with its subspace improves performance. Building on feature correlation and manifold learning, we incorporate matrix factorization (1) with regularization terms (3), (4), (5), and (8), yielding

$$\begin{aligned}\mathscr{F}(W,H,S) =& \frac{1}{2}\|X - XWH\|_F^2 + \frac{\alpha}{2}\mathrm{Tr}(W^\top X^\top XW\mathbf{1}\mathbf{1}^\top)\\ &+ \frac{\beta}{2}\mathrm{Tr}(W^\top X^\top L_S XW) + \frac{\gamma}{2}(\|V \odot S\|_\diamond + \eta\|S\|_F^2) + \frac{\lambda}{2}\|H\|_{2,1}.\end{aligned} \tag{10}$$

In this context, positive balancing parameters, denoted as α, β, γ, and λ, play a pivotal role as weighting factors within the objective function. These parameters intricately influence the trade-offs between various components of the optimization problem, allowing for fine-tuning and customization of the model's behavior. The derived objective function, along with the associated constraints, constitutes the following optimization problem:

$$\min_{W,H,S} \mathscr{F}(W,H,S) \quad \text{s.t. } W \geq 0,\ H \geq 0,\ S \geq 0,\ W^\top W = I_k. \tag{11}$$

Upon solving the optimization problem, we can obtain the matrix W. Subsequently, we calculate $\|W_{i:}\|$ for each feature, allowing the assessment of the importance of the ith feature. Finally, a new data matrix $X_{\text{new}} \in \mathbb{R}_+^{n\times k}$ is formed by selecting the first k features based on the sorted values of $\|W_{i:}\|$, arranged in descending order. The illustration of subspace learning and the USFAG framework is given in Figure 2.

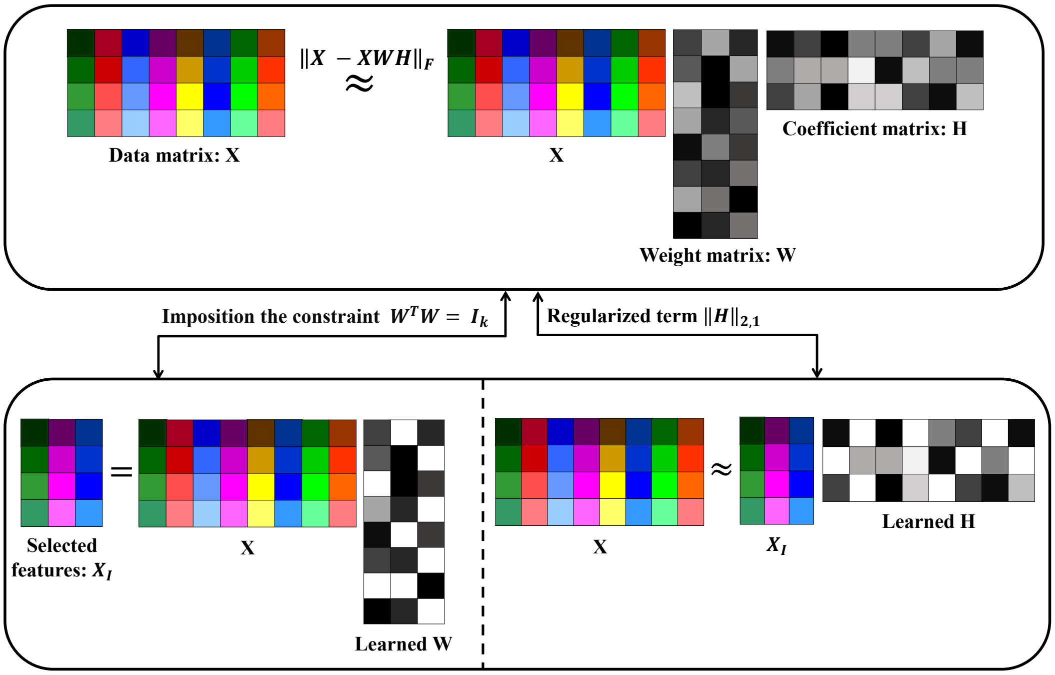

Figure 2: Illustration of the proposed USFAG framework

3 Optimization algorithm and theoretical analysis

This section details our proposed approach, analyzes its computational complexity compared to existing methods, and examines the algorithm's convergence properties.

3.1 Optimization algorithm

For the multi-variable optimization problem (11), simultaneous optimization of all variables is often computationally impractical. We therefore employ an alternating optimization approach, iteratively updating each variable while fixing others until convergence.

The adaptive graph S is particularly crucial as it determines the Laplacian matrix L_S. We begin by updating S while holding other variables fixed. To this end, taking into account two terms of the objective function (10) that involve S, we formulate and tackle the following optimization subproblem:

$$\min_{S\geq 0} \mathscr{E}(S) \equiv \beta\,\mathrm{Tr}(W^\top X^\top L_S X W) + \gamma\left(\|V\odot S\|_\diamond + \eta\|S\|_F^2\right). \tag{12}$$

To address problem (12), we reframe its objective function using (2) as follows:

$$\mathscr{E}(S) = \gamma\left(\|V\odot S\|_\diamond + \eta\|S\|_F^2 + \mu\|E\odot S\|_\diamond\right) + \text{unrelated terms to } S,$$

where $\mu = \beta/\gamma$ and $E_{ij} = \|\mathbf{x}_i W - \mathbf{x}_j W\|^2$. In this manner, we express the optimization subproblem (12) in the following equivalent form:

$$\min_{S\geq 0} \|Z \odot S\|_{\diamond} + \eta \|S\|_F^2,$$

where $Z = V + \mu E$. Hence, similar to problem (5), the solution of optimization subproblem is as follows:

$$S_{ij} = \max\left(\frac{\varepsilon_p - Z_{ij}}{2\eta}, 0\right). \tag{13}$$

By normalizing $d_{ij} = d_{ij}/\max_i d_{ij}$ and $E_{ij} = E_{ij}/\max_i E_{ij}$, we ensure that the values remain within a consistent and manageable range, thereby enhancing the stability and robustness of the numerical calculations.

With the matrix S updated, attention now shifts to updating other model variables within the objective function (10). As is customary, we commence the procedure by constructing the Lagrangian corresponding to the optimization problem (11), introducing the Lagrange multipliers $\delta \geq 0$, $\Phi \in \mathbb{R}_+^{m\times k}$, and $\Psi \in \mathbb{R}_+^{k\times m}$ as follows:

$$\mathscr{L}(W,H,S,\delta,\Phi,\Psi) = \mathscr{F}(W,H,S) + \frac{\delta}{4}\|W^\top W - I_k\|_F^2 + \mathrm{Tr}(\Phi W^\top) + \mathrm{Tr}(\Psi H^\top).$$

Critical points of the Lagrangian $\mathscr{L}$ correspond to critical points of the optimization problem (11). Given that the objective and constraint functions have continuous first partial derivatives in (11), we apply the gradient form of the KKT Theorem [4] as outlined below:

$$\begin{aligned}&\nabla\mathscr{F}(W,H,S) + \frac{\delta}{4}\nabla\|W^\top W - I_k\|_F^2 + \nabla\mathrm{Tr}(\Phi W^\top) + \nabla\mathrm{Tr}(\Psi H^\top) = 0,\\ &\Phi \odot W = 0, \quad \Psi \odot H = 0, \quad \Phi \geq 0, \quad \Psi \geq 0.\end{aligned} \tag{14}$$

By the KKT conditions (14), a minimizer (W,H) must satisfy

$$\begin{aligned}&X^\top X H^\top - X^\top X W H H^\top - \alpha X^\top X W \mathbf{1}\mathbf{1}^\top - \beta X^\top L_S X W - \delta(W W^\top W - W) = \Phi,\\ &W^\top X^\top X - W^\top X^\top X W H - \lambda H U = \Psi,\\ &\Phi \odot W = 0, \quad \Psi \odot H = 0, \quad \Phi \geq 0, \quad \Psi \geq 0.\end{aligned}$$

Given that $L_S = D - G$, the updating formulas are derived by examining these conditions in element-wise form

$$W_{ij} \longleftarrow W_{ij}\sqrt[4]{\frac{(X^\top X H^\top + \beta X^\top G X W + \delta W)_{ij}}{(X^\top X W H H^\top + \alpha X^\top X W \mathbf{1}\mathbf{1}^\top + \beta X^\top D X W + \delta W W^\top W)_{ij}}}, \tag{15}$$

$$H_{ij} \longleftarrow H_{ij}\sqrt{\frac{(W^\top X^\top X)_{ij}}{(W^\top X^\top X W H + \lambda H U)_{ij}}}. \tag{16}$$

The detailed iterative updating procedures are provided in Algorithm 1, which outlines the step-by-step process for updating the variables iteratively until convergence is achieved. The computational complexity of UFSAG (Algorithm 1) per iteration involves three main operations: Updating W via (15) requires $\mathscr{O}(nm^2 + n^2m + km^2)$ operations, updating H through (16) takes $\mathscr{O}(knm + k^2m + km^2)$ operations,

Algorithm 1: The training process UFSAG

```
Require: Data matrix $X \in \mathbb{R}_{+}^{n\times m}$; the number of selected features $k$; parameters $\alpha$, $\beta$, $\lambda$, $\gamma$; penalty
    coefficient $\delta$; domain parameter $p$; maximum iterations maxIter.
Ensure: Index set of selected features $I$ with $|I| = k$.
Initialize $W \in \mathbb{R}_{+}^{m\times k}$, $H \in \mathbb{R}_{+}^{k\times m}$, and $S \in \mathbb{R}^{n\times n}$ by (6).
while iteration $\leq$ maxIter do
    Update $G \leftarrow (S + S^{\top})/2$ and $D_{ii} = \sum_j G_{ij}$.
    $W \leftarrow WP^{-1}$ and $H \leftarrow PH$ where $P = \sqrt{\mathrm{diag}(W^{\top}W)}$.
    Update $W$ using (15) (fixing other variables).
    Update $H$ using (16) (fixing other variables).
    Update $U$ using (9) (fixing other variables).
    Update $S$ using (13).
end while
Sort $\|W_{i:}\|$ in descending order and select top $k$ features as $I$.
```

and updating S using (9) demands $\mathcal{O}(kn^2)$ operations. Additionally, matrix normalization contributes $\mathcal{O}(k^2m)$ operations. Since typically n is constant and $k \ll m$, the overall time complexity reduces to $\mathcal{O}(m^2)$ operations.

While the empirical evaluations presented in Section 4 demonstrate the effectiveness of UFSAG on datasets with considerable dimensions, including datasets with over $n = 2400$ samples and featuring more than $m = 4400$ features, it is crucial to acknowledge the potential limitations when deploying the algorithm on ultra-high dimensional datasets. Specifically, when considering datasets with approximately a million features ($m \approx 10^6$), challenges related to runtime and memory requirements may arise. Given that n is typically constant and $k \ll m$, the overall time complexity per iteration is $\mathcal{O}(m^2)$ operations. This quadratic dependency on the number of features (m) suggests that the computational overhead can become significant for extremely high-dimensional data.

The aforementioned complexity analysis implies that the algorithm's applicability to datasets with millions of features may be constrained by the substantial computational resources required. Specifically, the memory footprint associated with storing intermediate matrices, such as X and the coefficient matrices, can pose a practical limitation. Furthermore, the runtime needed to perform the iterative updates may become prohibitively long, especially in resource-constrained environments.

These limitations notwithstanding, the core framework of UFSAG, which effectively integrates global structures preservation, local manifold structures and feature correlations reduction, holds considerable promise for adaptation to high-dimensional data regimes. Future research directions could explore algorithmic optimizations, such as employing sparse matrix techniques or data partitioning strategies, to enhance UFSAG's efficiency and scalability. Additionally, investigating approximation techniques may offer a pathway to reduce the computational burden without significantly compromising the quality of feature selection. As shown in Table 1, the $\mathcal{O}(m^2)$ complexity of UFSAG is comparable to other FS and MF methods, offering a balance between performance and computational cost. The table summarizes the theoretical computational complexity with respect to the number of features m, samples n, and selected features k.

Table 1: Computational cost comparison of FS and MF methods

RRQR NDFS MCFS	SDFS	RSR	MFFS	IUFS EUFS	DISR	UDFS	DUFS	UFGOR UFSAG
$\mathscr{O}(n^2m)$	$\mathscr{O}(n^3+n^2m)$	$\mathscr{O}(m^3+nm^2)$	$\mathscr{O}(nm^2)$	$\mathscr{O}(mn+k^2(m+n))$	$\mathscr{O}(n^2m+nmk)$	$\mathscr{O}(n^2m+m^3)$	$\max\{\mathscr{O}(m^3),\mathscr{O}(m^2n)\}$	$\mathscr{O}(m^2)$

3.2 Convergence analysis

Following conventional approaches, we analyze the convergence of the UFSAG algorithm.

Since the objective function $\mathscr{F}$ in (10) is nonnegative and the matrix S-related terms $\mathscr{E}(S)$ in (12) admit closed-form solution (13), we only need to prove that for any W and H, the objective value decreases monotonically under Algorithm 1's update rules. The proof uses the auxiliary function method from [13].

Lemma 1 ([32])**.** *For any matrices $A \in \mathbb{R}_+^{n\times n}$, $B \in \mathbb{R}_+^{k\times k}$, $C \in \mathbb{R}_+^{n\times k}$, and $C' \in \mathbb{R}_+^{n\times k}$, where A and B are symmetric, the following inequalities hold*

$$\begin{aligned}\mathrm{Tr}(C^\top ACB) &\le \sum_{i=1}^{n}\sum_{j=1}^{k}(AC'B)_{ij}\frac{C_{ij}^2}{C'_{ij}},\\ \mathrm{Tr}(CC^\top CC^\top) &\le \sum_{i=1}^{n}\sum_{j=1}^{k}(C'C'^\top C')_{ij}\frac{C_{ij}^4}{C'^3_{ij}}.\end{aligned}$$

First, assuming that the matrix H is fixed, by removing the unrelated terms to W from the objective function, the following function results:

$$\begin{aligned}\mathscr{F}_1(W) = &\frac{1}{2}\mathrm{Tr}(H^\top W^\top X^\top XWH) - \mathrm{Tr}(H^\top W^\top X^\top X) + \frac{\alpha}{2}\mathrm{Tr}(W^\top X^\top XW\mathbf{1}\mathbf{1}^\top)\\ &+\frac{\beta}{2}\mathrm{Tr}(W^\top X^\top(D-G)XW) + \frac{\delta}{4}\mathrm{Tr}(W^\top WW^\top W) - \frac{\delta}{2}\mathrm{Tr}(W^\top W).\end{aligned}$$

Following the procedure outlined in [28], one can derive the corresponding auxiliary function:

$$\begin{aligned}\mathscr{G}_1(W,W') =&\frac{1}{4}\sum_{i,j}(X^\top XW'HH^\top)_{ij}\frac{W_{ij}^4+W'^4_{ij}}{W'^3_{ij}} - \sum_{i,j}(X^\top XH^\top)_{ij}W'_{ij}\left(1+\log\frac{W_{ij}}{W'_{ij}}\right)\\ &+\frac{\alpha}{4}\sum_{i,j}(X^\top XW'\mathbf{1}\mathbf{1}^\top)_{ij}\frac{W_{ij}^4+W'^4_{ij}}{W'^3_{ij}} - \frac{\beta}{2}\sum_{i,j,k}(X^\top GX)_{ik}W'_{kj}W'_{ij}\left(1+\log\frac{W_{kj}W_{ij}}{W'_{kj}W'_{ij}}\right)\\ &+\frac{\beta}{4}\sum_{i,j}(X^\top DXW')_{ij}\frac{W_{ij}^4+W'^4_{ij}}{W'^3_{ij}} + \frac{\delta}{4}\sum_{i,j}(W'W'^\top W')_{ij}\frac{W_{ij}^4}{W'^3_{ij}} - \frac{\delta}{2}\sum_{i,j}W'_{ij}W'_{ij}\left(1+\log\frac{W_{ij}^2}{W'^2_{ij}}\right).\end{aligned}$$

The function $\mathscr{G}_1$ serves as an auxiliary function for $\mathscr{F}_1$ as it fulfills the conditions

$$\mathscr{G}_1(W,W') \geq \mathscr{F}_1(W) \quad \text{and} \quad \mathscr{G}_1(W,W) = \mathscr{F}_1(W).$$

As a result, the function $\mathscr{F}_1(W)$ is nonincreasing under the update rule

$$W^{t+1} = \underset{W}{\operatorname{argmin}} \mathscr{G}_1(W,W^t),$$

where t is the number of iterations. Now, we proceed to solve this optimization problem. By taking the derivative of $\mathscr{G}_1$ with respect to W and setting it equal to zero, we obtain the following equation:

$$(X^\top X H^\top + \beta X^\top G X W' + \delta W')_{ij} \frac{W'_{ij}}{W_{ij}}$$
$$= (X^\top X W' H H^\top + \alpha X^\top X W' \mathbf{1}\mathbf{1}^\top + \beta X^\top D X W' + \delta W' W'^\top W')_{ij} \frac{W_{ij}^3}{W_{ij}'^3}.$$

Now, substituting W' with W^t and W with W^{t+1} allows us to retrieve (15). So far, it has been demonstrated that

$$\mathscr{F}_1(W^{t+1}) \leq \mathscr{G}_1(W^{t+1},W^t) \leq \mathscr{G}_1(W^t,W^t) = \mathscr{F}_1(W^t).$$

The distinction between $\mathscr{F}$ and $\mathscr{F}_1$ lies solely in a positive constant; hence, it can be deduced that

$$\mathscr{F}(W^{t+1},H^t,S^t) \leq \mathscr{F}(W^t,H^t,S^t).$$

The preceding content can be summarized in the following proposition.

Proposition 1. *The objective function $\mathscr{F}$ in* (10) *is nonincreasing by updating W with the updating rule*

(15).

In the next step, assuming the matrix W is fixed, the following function is extracted from the objective function:

$$\mathscr{F}_2(H) = \frac{1}{2}\text{Tr}(H^\top W^\top X^\top X W H) - \text{Tr}(H^\top W^\top X^\top X) + \frac{\lambda}{2}\text{Tr}(H U H^\top).$$

As done previously, one can derive the corresponding auxiliary function as follows:

$$\mathscr{G}_2(H,H') = \frac{1}{2}\sum_{i,j}(W X^\top X W H')_{ij}\frac{H_{ij}^2}{H'_{ij}} - \sum_{i,j}(W^\top X^\top X)_{ij} H'_{ij}\left(1 + \log\frac{H_{ij}}{H'_{ij}}\right) + \frac{\lambda}{2}\sum_{i,j}(H'U)_{ij}\frac{H_{ij}^2}{H'_{ij}}.$$

Hence, the function $\mathscr{F}_2(H)$ is nonincreasing under the update rule

$$H^{t+1} = \underset{H}{\operatorname{argmin}} \mathscr{G}_2(H,H^t).$$

To address this optimization problem, we take the derivative of $\mathscr{G}_2$ with respect to H and equate it to zero, yielding the following equation:

$$(W^\top X^\top X)_{ij}\frac{H'_{ij}}{H_{ij}} = (W X^\top X W H' + \lambda H' U)_{ij}\frac{H_{ij}}{H'_{ij}}.$$

Table 2: Detailed information about the datasets

Dataset	instances	features	classes	Type of Data
COIL20	1440	1024	20	Object images
Isolet	1560	617	26	Letter image
ORL	400	1024	40	Face image
YaleB	2414	1024	38	Face image
Lung-discrete	73	325	7	Biological microarray
GLIOMA	50	4434	4	Biological microarray

Now, substituting H' with H^t and H with H^{t+1} allows us to retrieve (16). Therefore, it can be deduced that

$$\mathscr{F}_2(H^{t+1}) \leq \mathscr{G}_2(H^{t+1}, H^t) \leq \mathscr{G}_2(H^t, H^t) = \mathscr{F}_2(H^t)$$

holds true. The difference between $\mathscr{F}$ and $\mathscr{F}_2$ is only a positive constant; therefore

$$\mathscr{F}(W^t, H^{t+1}, S^t) \leq \mathscr{F}(W^t, H^t, S^t).$$

The material covered so far can be summarized in the following proposition.

Proposition 2. *The objective function $\mathscr{F}$ in* (10) *is nonincreasing by updating H with the updating rule*

(16).

In summary, the proposed UFSAG algorithm consistently decreases the objective function in each iteration. Figure 5 provides practical demonstrations illustrating the decrease of the objective function and the convergence of the method.

4 Experiments

In this section, we evaluate the effectiveness of the proposed approach through experimentation on six publicly available datasets. Table 2 outlines the specifics of the datasets employed in our investigation. We use the k-means clustering algorithm to evaluate unsupervised feature selection methods, setting the parameter k to correspond to the number of classes in each dataset. Following [18], we repeated clustering 20 times with random initialization and record the average metrics obtained to minimize initialization effects. Extended results, tables, figures, and codes are available at https://github.com/sohrabi94/UFSAG.

4.1 Comparison of conventional feature selection approaches

The UFSAG was compared with various established UFS techniques, including MCFS [6], UDFS [38], MFFS [33], NDFS [15], DUFS, IUFS [12], EUFS [34], DISR [16], RSR [27], UFGOR [28], RRQR [21], and SDFS

Table 3: Comparison of ACC and NMI for unsupervised feature selection techniques

Dataset	Coil20	Isolet	ORL	YaleB	Lung-discrete	GLIOMA
ALL	63.1 ± 3.6 / 77.1 ± 2.1	63.1 ± 3.1 / 77.8 ± 1.6	56.7 / 73.9	10.5 ± 0.6 / 10.6 ± 0.7	71.7 ± 4.5 / 65.6 ± 5.3	61.1 ± 4.7 / 50.1 ± 6.5
MCFS	61.2 ± 2 / 75.6 ± 1.2 (300)	58.5 ± 3 / 75.3 ± 1.5 (300)	55.4 ± 3.7 / 73.4 ± 2 (100)	11.9 ± 0.5 / 13.8 ± 0.9 (50)	74.8 ± 1.6 / 66.2 ± 2.5 (90)	54.8 ± 6.4 / 30.1 ± 7.7 (90)
MFFS	61.4 ± 2.7 / 70.9 ± 1.6 (100)	62.7 ± 4.6 / 76.3 ± 2.4 (300)	48.5 ± 2 / 67.5 ± 1.6 (100)	10.1 ± 0.0 / 10.5 ± 0.4 (50)	73.7 ± 5.1 / 67.2 ± 4.7 (100)	57.2 ± 3.6 / 47.4 ± 3.6 (90)
UDFS	59.4 ± 3 / 71.1 ± 2.2 (200)	57.2 ± 1.8 / 70 ± 0.8 (300)	51.1 ± 2 / 69.6 ± 0.7 (100)	11.4 ± 0.4 / 14.4 ± 1 (150)	72.1 ± 3.8 / 62.5 ± 3.9 (90)	55.3 ± 3.4 / 30.2 ± 4.6 (90)
NDFS	62 ± 1.9 / 74.3 ± 2.7 (150)	63.9 ± 1.9 / 75.6 ± 1.7 (250)	53.4 ± 2.7 / 72.4 ± 2 (100)	13.9 ± 0.6 / 9.8 ± 0.6 (50)	69.6 ± 2.4 / 66.5 ± 4.2 (100)	56.6 ± 2.9 / 48.8 ± 1.2 (60)
EUFS	56.4 ± 3.1 / 69.7 ± 1.9 (300)	58.1 ± 1.4 / 72.9 ± 1.4 (300)	49.4 ± 2.5 / 68.4 ± 1.9 (100)	11.2 ± 0.6 / 10.5 ± 0.8 (300)	68.1 ± 4.5 / 61.3 ± 6.3 (90)	59.6 ± 2.9 / 49.9 ± 4.1 (70)
DUFS	61.9 ± 3.1 / 76.9 ± 1.9 (150)	64.7 ± 3.7 / 76.7 ± 1.4 (300)	56.6 ± 2.9 / 74.6 ± 1.9 (100)	18.9 ± 0.7 / 10.7 ± 1 (50)	72.4 ± 2.1 / 66.1 ± 2.1 (50)	56.3 ± 5.2 / 32.2 ± 7.2 (90)
RSR	60.5 ± 3.8 / 75.6 ± 2.5 (100)	62.3 ± 1.2 / 76 ± 2.1 (300)	52.6 ± 2.6 / 71.9 ± 1.8 (100)	10.1 ± 0.7 / 12 ± 2.4 (300)	72.6 ± 2.3 / 64.4 ± 1.9 (40)	54.7 ± 3.1 / 43.1 ± 4.2 (90)
DISR	61.2 ± 3.7 / 76.6 ± 2.5 (100)	58.6 ± 2.7 / 75.6 ± 1.1 (300)	51.3 ± 1.8 / 70.9 ± 0.8 (100)	9.8 ± 0.4 / 11.8 ± 0.4 (50)	65.7 ± 3.3 / 56.7 ± 3.4 (90)	56.4 ± 3.4 / 47.3 ± 1.6 (90)
IUFS	56.2 ± 2.9 / 73.2 ± 0.3 (100)	55.6 ± 4.6 / 69.1 ± 0.8 (300)	54.5 ± 2 / 73 ± 1 (100)	18.1 ± 0.7 / 25.6 ± 0.7 (100)	75.1 ± 3.4 / 67.8 ± 3.5 (90)	55.8 ± 2.4 / 45.3 ± 2.1 (90)
UFGOR	61 ± 2.4 / 71.8 ± 2.6 (100)	65.1 ± 3.8 / 78.3 ± 1.4 (250)	51.5 ± 2.5 / 71.5 ± 1.8 (100)	19.6 ± 0.9 / 25.9 ± 1.4 (50)	74.1 ± 5.4 / 69.4 ± 3.1 (90)	47.9 ± 2.2 / 40.9 ± 3.7 (90)
SDFS	62.3 ± 1.7 / 74.1 ± 1.9 (100)	64.3 ± 2.4 / 74 ± 2.6 (250)	58.1 ± 2.5 / 75.4 ± 1.9 (100)	21.1 ± 0.9 / 27.4 ± 1.1 (30)	73.7 ± 3.6 / 65.9 ± 2.2 (90)	62.1 ± 2.1 / 50.8 ± 2.3 (90)
RRQR	61.9 ± 2.1 / 73 ± 2.3 (100)	66.1 ± 1.9 / 77 ± 1.4 (250)	56 ± 2.1 / 73.3 ± 1.7 (100)	20.2 ± 0.6 / 27 ± 1.1 (30)	76.1 ± 2.7 / 68.2 ± 2.1 (90)	61.3 ± 1.9 / 50.3 ± 2.7 (90)
UFSAG	64.7 ± 3.1 / 77 ± 2.1 (100)	67.3 ± 2.5 / 79 ± 1.6 (250)	57 ± 3.1 / 74.5 ± 2.2 (100)	23 ± 0.7 / 30 ± 1.2 (30)	74.5 ± 4.7 / 67.1 ± 2.6 (90)	64 ± 4.5 / 53.3 ± 6.3 (90)

Various UFS methodologies were evaluated using clustering accuracy (ACC) and normalized mutual information (NMI) metrics [20], with results summarized in Table 3. While UFSAG demonstrates competitive results, achieving relatively high ACC and NMI scores compared to methods like MCFS and EUFS, methods like RRQR sometimes show superior performance on certain datasets.

4.2 Result and analysis

Figure 3 shows ACC/NMI curves across feature selection sizes, demonstrating UFSAG's consistent superiority over other methods.

4.3 Parameter settings

We analyze the impact of parametersα, β, γ, and λ on UFSAG performance using 3D histograms (Figure 4), testing values from $10^{-6:2:6}$ for α, β, and γ and $10^{0:2:6}$ for λ, with feature counts k ranging from $10-300$. Key findings are as follows:

- **Global stability**: Parameters α (feature correlation reduction) and λ (sparsity control) exhibit broad stability ranges ($10^{-6} \leq \alpha \leq 10^{6}$, $10^{-6} \leq \lambda \leq 10^{4}$ across datasets), where performance

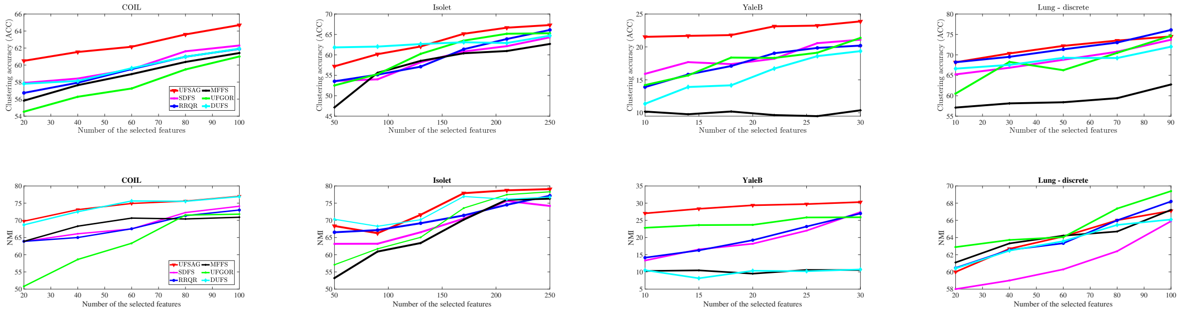

Figure 3: Results of ACC and NMI of compared methods on datasets

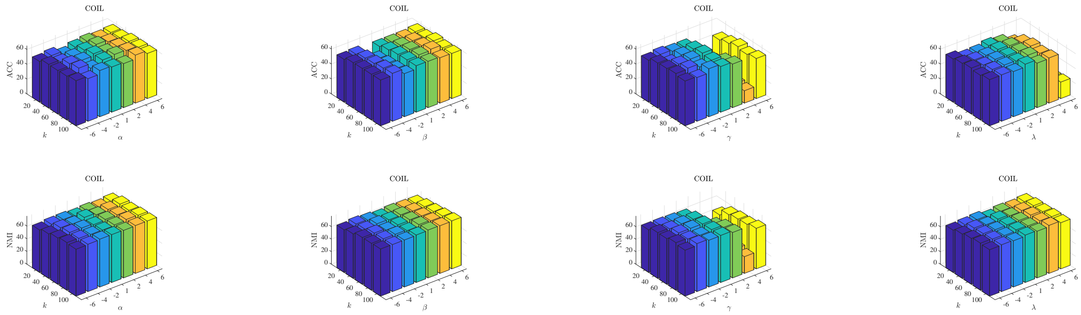

Figure 4: The ACC and NMI of UFSAG with different values of parameters and k on COIL

metrics (ACC/NMI) vary by $< 5\%$ despite order-of-magnitude parameter changes. This aligns with the orthogonality constraints in equation(4) and the $\ell_{2,1}$-norm's effect on feature redundancy minimization (Section 2.4).

- **Controlled sensitivity to γ**: Our analysis reveals a remarkable robustness property of the adaptive graph strategy parameter γ. While UFSAG maintains stable performance across an extensive range ($10^{-6} \leq \gamma \leq 10^{6}$), we observe a precise critical zone ($10^{2} < \gamma < 10^{4}$) where performance degrades significantly. The optimal operating range ($10^{-2} \leq \gamma \leq 10^{1}$) balances these effects, preserving the data manifold's topological structure while maintaining sparsity a key advantage over static graph methods like k-NN (Table 5).

- **Local structure preservation (β)**: The neighborhood preservation parameter β demonstrates remarkable stability across six orders of magnitude ($10^{-6} \leq \beta \leq 10^{6}$), with minimal performance variation ($< 3\%$ ACC change) in the range $10^{-3} \leq \beta \leq 10^{0}$. This consistency stems from three key design features: (1) the adaptive graph formulation in equation(5) intrinsically preserves local manifolds through its distance-aware thresholding, (2) the Laplacian regularization term $\mathrm{Tr}(W^{\top}X^{\top}L_{S}XW)$ in equation (3) maintains neighborhood relationships regardless of exact β values, and (3) the joint optimization automatically balances local and global structures.

Based on extensive experiments conducted on hyperparameter values, the optimal values for each

Table 4: The optimal values of hyperparameters for each dataset

Parameter	Coil20	Isolet	ORL	YaleB	Lung-discrete	GLIOMA
α	10^{-4}	10^{-2}	10^{-2}	10^{1}	10^{-2}	10^{2}
β	10^{4}	10^{1}	10^{4}	10^{2}	10^{2}	10^{-2}
γ	10^{-2}	10^{4}	10^{-2}	10^{1}	10^{2}	10^{1}
λ	10^{-1}	10^{2}	10^{2}	10^{2}	10^{2}	10^{4}

Table 5: Impact of graph construction methods on UFSAG performance

Method	Isolet		YaleB		ORL	
	ACC	NMI	ACC	NMI	ACC	NMI
k-NN	61.5 ± 1.4	71.2 ± 1.7	18.6 ± 0.6	24.1 ± 1.3	52.4 ± 1.1	70.4 ± 1.2
Gaussian Kernel	65.3 ± 1.9	75.6 ± 1.2	21 ± 1.1	26.5 ± 0.7	55.1 ± 2.8	71.8 ± 1.5
Cosine Similarity	63.1 ± 2.4	73.4 ± 1.4	19 ± 1.3	25.1 ± 1.1	52.1 ± 1.9	69.6 ± 1.7
UFSAG	67.3 ± 2.5	79 ± 1.6	23 ± 0.7	30 ± 1.2	57 ± 3.1	74.5 ± 2.2

dataset are reported in Table 4. In real-world scenarios where datasets lack ground truth labels, appropriate hyperparameter values can be determined by utilizing datasets with similar characteristics.

This study proposes a novel approach for adjacency matrix construction that effectively preserves local data structures. We evaluate its performance through empirical experiments measuring ACC and NMI across four distinct graph construction strategies: *k*-NN, Gaussian Kernel, Cosine Similarity, and our proposed UFSAG method. As demonstrated in Table 5 across three benchmark datasets, UFSAG consistently outperforms conventional techniques in both accuracy and cluster quality metrics.

4.4 Ablation study

To validate the contribution of each component in UFSAG, we conduct an ablation study by systematically removing key elements from the objective function in equation (10). Table 6 reports the performance on three representative datasets.

The ablation study reveals that **local structure preservation** (**β**) is the most critical component, with the largest performance drop (5.8–7.9% ACC reduction across datasets), confirming the essential role of manifold learning in equation (10). The **feature correlation reduction** (**α**) ranks second in importance (4.2–5.5% ACC reduction), particularly vital for high-dimensional data like GLIOMA as discussed in Section 2.2. Notably, while **graph adaptivity** (**γ**) shows moderate impact (2.9–3.8% ACC drop), its adaptive thresholding mechanism remains crucial for noisy environments. The $\ell_{2,1}$-norm (λ) exhibits the smallest but still significant effect (1.8–2.4% ACC drop), validating its role in feature selection stability. The full model's consistent superiority (4.7–7.9% ACC improvement) demonstrates the complementary nature of these components in (10).

Table 6: Ablation study of UFSAG components (ACC/NMI % ± std)

Variant	**COIL20**	**YaleB**	**GLIOMA**
Full UFSAG	64.7±3.1 / 77.0±2.1	23.0±0.7 / 30.0±1.2	64.0±4.5 / 53.3±6.3
w/o β (Local structure)	58.9±2.7 / 70.8±2.1	18.2±0.8 / 22.5±1.2	56.1±3.8 / 44.3±5.2
w/o α (Corr. reduction)	60.5±2.9 / 73.6±1.8	19.7±0.9 / 24.8±1.3	59.3±4.0 / 47.9±5.5
w/o γ (Graph adaptivity)	61.8±3.0 / 74.2±2.0	20.5±0.7 / 25.6±1.1	60.2±4.2 / 49.1±5.8
w/o λ ($\ell_{2,1}$-norm)	62.9±3.0 / 75.8±1.9	21.8±0.8 / 27.9±1.2	62.3±4.4 / 51.5±6.1

Table 7: Average rankings obtained from the Friedman test (lower is better)

	Method												
Metric	UFSAG	RRQR	SDFS	UFGOR	DUFS	IUFS	MCFS	MFFS	RSR	NDFS	DISR	UDFS	EUFS
ACC	1.83	3.00	3.33	4.50	4.17	5.33	7.17	6.50	6.67	6.83	8.50	8.17	9.00
NMI	1.50	2.83	3.50	3.83	4.50	5.00	6.33	6.83	6.00	7.17	7.83	8.50	9.17

4.5 Statistical analysis by Friedman test

We employ the Friedman test to compare methods across all datasets, with methods as treatments and metrics as measurements. The test ranks methods while testing the null hypothesis of no significant difference.

The results of the Friedman test, as shown in Table 7, indicate the superiority of the UFSAG method. It's important to note that a lower rank indicates better performance. Upon scrutinizing Table 7, it becomes evident that the UFSAG method secures the topmost position in both ACC and NMI metrics. This suggests that UFSAG demonstrates the utmost effectiveness when juxtaposed with alternative methodologies. Moreover, to provide a more comprehensive insight into the evident enhancement in the clustering outcomes achieved by UFSAG we conduct paired t-tests $\alpha = 0.05$ using 20 runs per algorithm. Results (Table 8) show $h = 1$ and small p-values indicate significant improvements over baselines in most cases.

Table 8 reveals that the paired t-test results indicate significant discrepancies in the ACC and NMI values between UFSAG and the other methods. Across most datasets, the paired t-tests yield $h = 1$ and very small p-values. Conversely, on a few datasets, $h = 0$, suggesting that the ACC and NMI values of UFSAG do not exhibit a noticeable improvement compared to other algorithms. Overall, UFSAG demonstrates a substantial enhancement in ACC across the majority of cases. The clustering results in Table 3 show a consistent and significant improvement with UFSAG over other algorithms, confirming its superiority.

4.6 Convergence behavior of the UFSAG in practice

Experiments in this section is dedicated to examining the convergence behavior of UFSAG technique. As anticipated from the UFSAG convergence analysis in Section 3.2, the UFSAG cost function demonstrates a consistent decrease over multiple iterations until convergence is reached. This observation is further

Table 8: Paired t-test results of UFSAG vs. comparison algorithms

Algorithm	Metric	COIL20		Isolet		ORL		YaleB		Lung-discrete		Glioma	
		p	h	p	h	p	h	p	h	p	h	p	h
MCFS	ACC	$7.91e^{-03}$	1	$1.28e^{-17}$	1	0.972	0	$3.58e^{-31}$	1	$6.81e^{-02}$	0	$6.75e^{-07}$	1
	NMI	$6.50e^{-03}$	1	$2.83e^{-10}$	1	$1.20e^{-02}$	1	$8.86e^{-34}$	1	$7.54e^{-01}$	0	$7.95e^{-09}$	1
UDFS	ACC	$1.62e^{-07}$	1	$6.42e^{-21}$	1	$3.67e^{-11}$	1	$2.97e^{-32}$	1	$1.34e^{-02}$	1	$2.06e^{-05}$	1
	NMI	$2.87e^{-14}$	1	$3.26e^{-21}$	1	$2.03e^{-20}$	1	$2.35e^{-34}$	1	$2.73e^{-04}$	1	$6.18e^{-11}$	1
NDFS	ACC	0.118	0	$2.84e^{-11}$	1	$7.34e^{-04}$	1	$6.33e^{-30}$	1	$6.47e^{-05}$	1	$1.99e^{-04}$	1
	NMI	$5.77e^{-09}$	1	$5.09e^{-07}$	1	$3.38e^{-06}$	1	$4.36e^{-39}$	1	$2.53e^{-01}$	0	$6.41e^{-03}$	1
EUFS	ACC	$1.54e^{-14}$	1	$9.83e^{-20}$	1	$1.97e^{-08}$	1	$2.65e^{-36}$	1	$2.77e^{-04}$	1	$1.67e^{-02}$	1
	NMI	$4.28e^{-12}$	1	$1.92e^{-12}$	1	$1.12e^{-17}$	1	$2.07e^{-35}$	1	$5.96e^{-04}$	1	$4.05e^{-02}$	1
RSR	ACC	$1.53e^{-03}$	1	$9.03e^{-13}$	1	$7.43e^{-06}$	1	$1.20e^{-32}$	1	$1.23e^{-03}$	1	$6.38e^{-05}$	1
	NMI	$1.16e^{-04}$	1	$3.84e^{-05}$	1	$3.09e^{-05}$	1	$1.41e^{-23}$	1	$4.95e^{-03}$	1	$1.41e^{-05}$	1
DISR	ACC	$5.76e^{-07}$	1	$1.31e^{-14}$	1	$2.60e^{-12}$	1	$2.33e^{-35}$	1	$8.67e^{-07}$	1	$1.83e^{-03}$	1
	NMI	0.101	0	$4.05e^{-10}$	1	$3.28e^{-22}$	1	$9.56e^{-36}$	1	$1.69e^{-09}$	1	$4.41e^{-04}$	1
IUFS	ACC	$5.50e^{-04}$	1	$1.33e^{-10}$	1	$7.66e^{-03}$	1	$1.25e^{-19}$	1	$1.84e^{-01}$	0	$1.02e^{-04}$	1
	NMI	$1.11e^{-10}$	1	$1.71e^{-23}$	1	$3.16e^{-03}$	1	$4.08e^{-18}$	1	$7.21e^{-01}$	0	$1.16e^{-04}$	1
DUFS	ACC	0.254	0	$1.27e^{-07}$	1	0.913	0	$1.03e^{-30}$	1	$1.98e^{-02}$	1	$3.52e^{-03}$	1
	NMI	0.412	0	$5.19e^{-04}$	1	0.427	0	$1.21e^{-36}$	1	$2.42e^{-01}$	0	$1.76e^{-07}$	1
MFFS	ACC	$5.65e^{-03}$	1	$5.09e^{-08}$	1	$2.71e^{-11}$	1	$2.18e^{-35}$	1	$7.02e^{-01}$	0	$1.72e^{-03}$	1
	NMI	$5.66e^{-12}$	1	$1.02e^{-03}$	1	$4.69e^{-16}$	1	$8.11e^{-38}$	1	$5.35e^{-01}$	0	$9.08e^{-04}$	1
RRQR	ACC	$7.12e^{-05}$	1	$7.04e^{-02}$	0	$2.03e^{-02}$	1	$4.66e^{-09}$	1	$8.24e^{-02}$	0	$4.13e^{-11}$	1
	NMI	$5.22e^{-05}$	1	$9.31e^{-03}$	1	$8.91e^{-03}$	1	$7.53e^{-09}$	1	$1.42e^{-01}$	0	$7.94e^{-11}$	1
SDFS	ACC	$2.52e^{-04}$	1	$3.75e^{-05}$	1	$6.36e^{-02}$	0	$7.22e^{-07}$	1	$5.62e^{-04}$	1	$3.58e^{-09}$	1
	NMI	$1.88e^{-04}$	1	$6.52e^{-05}$	1	$7.14e^{-02}$	0	$8.65e^{-08}$	1	$7.07e^{-06}$	1	$8.43e^{-09}$	1
UFGOR	ACC	$1.48e^{-05}$	1	$4.91e^{-04}$	1	$5.07e^{-18}$	1	$6.18e^{-24}$	1	$6.26e^{-01}$	0	$9.91e^{-11}$	1
	NMI	$1.53e^{-06}$	1	0.073	0	$6.75e^{-22}$	1	$3.54e^{-13}$	1	$1.38e^{-01}$	0	$5.79e^{-07}$	1

illustrated in Figure 5, where we observe a similar trend across the datasets listed in Table 2, with the cost function steadily decreasing over iterations until it converges. These results demonstrate the effectiveness of the UFSAG optimization algorithm.

4.7 Runtime analysis

To comprehensively evaluate the computational efficiency of the UFSAG method, we compare its runtime against several representative unsupervised feature selection algorithms. The theoretical complexity of UFSAG is $\mathscr{O}(m^2)$ per iteration, as derived in Section 3.1. This efficiency stems from leveraging adaptive graph learning and subspace optimization with matrix factorization. Unlike methods such as RSR or DISR which involve cubic or higher-order complexities, UFSAG remains scalable for high-dimensional datasets.

To empirically validate this analysis, we measured the actual runtime of UFSAG and competing methods over 100 iterations using MATLAB R2023a on a workstation with an Intel Core i7 CPU and 32GB RAM. Table 9 presents the averaged runtimes (in seconds) on six benchmark datasets.

As seen in Table 9, UFSAG achieves a balanced runtime across datasets, often outperforming high-cost methods like RSR and DISR, and remaining comparable to UFGOR. This demonstrates the method's scalability and practical applicability to medium- and high-dimensional datasets. In conclusion, UFSAG provides a favorable trade-off between computational efficiency and performance. Its quadratic runtime makes it suitable for real-world applications, offering moderate complexity while achieving robust and accurate feature selection.

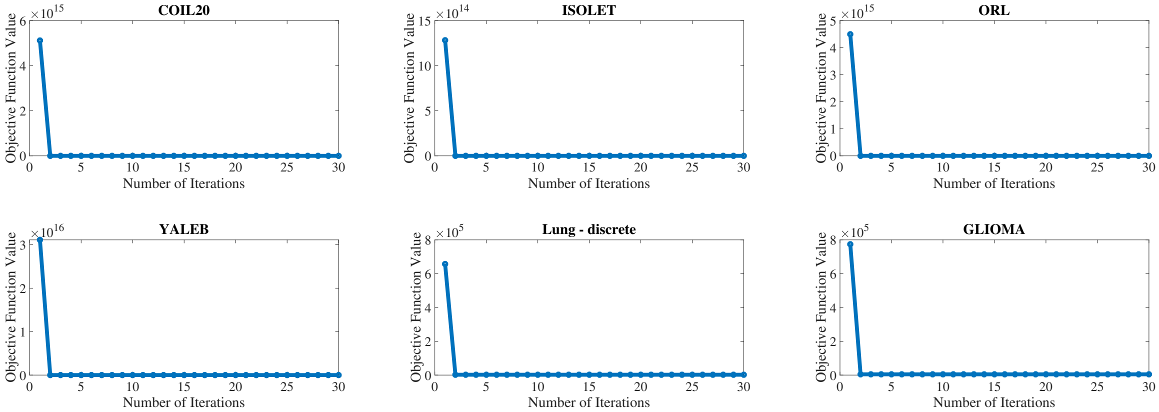

Figure 5: Convergence diagrams of the UFSAG on different datasets

Table 9: Runtime comparison (in seconds) of competing algorithms over 100 iterations

Dataset	RRQR	SDFS	MCFS	MFFS	UFGOR	UFSAG	DISR	RSR	UDFS	NDFS	DUFS	IUFS	EUFS
Isolet	10.7	1201.41	6.61	2.65	38.75	37.52	341.59	187.94	3.85	340.50	234.5	2.23	11.04
Lung-discrete	0.36	3.95	0.35	0.62	2.65	2.93	3.44	78.25	0.47	3.22	1.12	0.3	2.39
GLIOMA	4.6	6.03	0.61	42.91	230.1	212.8	6.56	47442.25	0.60	5.97	6.87	1.18	6.57
YaleB	36.1	1959.96	6.91	16.42	101.8	114.2	584.98	352.67	4.06	583.77	1740	2.25	12.18
ORL	10.3	169.88	2.72	3.00	29.5	30.4	123.69	1113.33	2.12	122.16	19.1	1.33	15.54
COIL20	21.2	1219.68	6.71	4.51	79.2	75.4	508.64	616.77	4.24	506.88	2117	2.37	17.75

4.8 UFSAG's resistance to noisy feature disruption

To evaluate the robustness of our proposed method against realistic noise, we conducted a controlled experiment by injecting partial Gaussian noise into the COIL20 dataset. Specifically, for each data sample (i.e., image), a certain percentage of randomly selected features (pixels) were corrupted by zero-mean Gaussian noise with fixed variance $\sigma^2 = 0.2$. We experimented with four levels of corruption, where $\rho \in \{0\%, 10\%, 20\%, 30\%\}$ of the features were randomly selected and corrupted in each image. Formally, the noisy data $\tilde{X}$ was generated as:

$$\tilde{X}_{ij} = \begin{cases} X_{ij} + \varepsilon_{ij}, & \text{if } j \in \mathscr{I}_{i,\rho},\ \varepsilon_{ij} \sim \mathscr{N}(0, 0.2), \\ X_{ij}, & \text{otherwise.} \end{cases}$$

where $\mathscr{I}_{i,\rho}$ denotes the set of $\rho\%$ randomly chosen features in the i-th sample to be corrupted.

We evaluated the performance of six representative unsupervised feature selection methods under increasing corruption ratios. For each method, we selected the top $k = 100$ features and evaluated clustering performance using k-means over 20 random initializations. The results in Figure 6 show the average ACC and NMI scores across corruption levels.

As observed in Figure 6, the performance of all methods degrades as the corruption ratio increases. However, our proposed method consistently achieves the best results, maintaining higher ACC and NMI

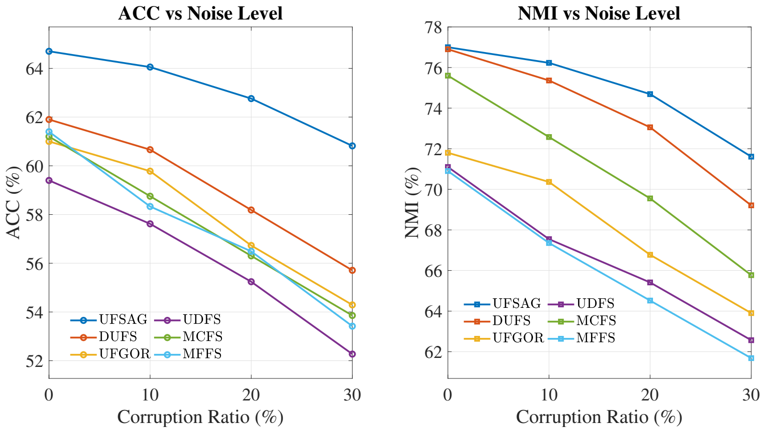

Figure 6: ACC and NMI of different methods on COIL20 under increasing partial feature corruption

across all noise levels. Among the baselines, UDFS and UFGOR perform relatively well, achieving the second and third best results, respectively, under most corruption scenarios. These findings confirm the robustness of UFSAG in preserving informative features and structural consistency, even when a portion of the input data is corrupted.

5 Conclusion

The UFSAG integrates subspace learning, local correlation analysis, and adaptive graph learning into a unified framework. Its matrix factorization preserves global structures, while the adaptive graph Laplacian enhances noise robustness, and the $\ell_{2,1}$-norm promotes feature sparsity.

The UFSAG opens several avenues for future research. Extending it to semi-supervised or supervised settings could improve its applicability in domains with limited labeled data. Integrating deep learning techniques may enhance its ability to capture complex, nonlinear relationships. Additionally, domain-specific adaptations, such as in bioinformatics or social network analysis, could further validate its practical utility.

It marks a significant advancement in unsupervised feature selection, providing a robust and adaptive solution for high-dimensional data analysis. By addressing key limitations of existing methods, it has the potential to impact diverse real-world applications, from healthcare to finance. We hope this work inspires further research into adaptive and interpretable feature selection techniques.

While UFSAG demonstrates significant advancements in unsupervised feature selection, it is important to acknowledge certain limitations. A key aspect is the selection of the parameter p, which controls the sparsity of the adaptive graph. Currently, this parameter is determined through empirical trial-and-error. Developing a theoretical framework for optimal parameter selection could further enhance the method's performance and robustness.

References

[1] V. Amarnadh, N.R. Moparthi, *Range control-based class imbalance and optimized granular elastic net regression feature selection for credit risk assessment*, Knowl. Inf. Syst. **66** (2024) 5281–5310.

[2] M.C. Barbieri, B.I. Grisci, M. Dorn, *Analysis and comparison of feature selection methods towards performance and stability*, Expert Syst. Appl. **16** (2024) 123667.

[3] R. Bellman, *A mathematical formulation of variational processes of adaptive type*, Proceedings of the fourth Berkeley symposium on mathematical statistics and probability (1961) 37–49.

[4] D.P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Massachusetts, 2016.

[5] D. Cai, X. He, J. Han, T.S. Huang, *Graph regularized nonnegative matrix factorization for data representation*, IEEE Trans. Pattern Anal. Mach. **33** (2010) 1548–1560.

[6] D. Cai, C. Zhang, X. He, *Unsupervised feature selection for multi-cluster data*, Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining (2010) 333–342.

[7] S. Chen, S. Ma, A. Man-Cho So, T. Zhang, *Nonsmooth optimization over the Stiefel manifold and beyond: Proximal gradient method and recent variants*, SIAM Rev. **66** (2024) 319–352.

[8] J.G. Dy, C.E. Brodley, *Feature selection for unsupervised learning*, J. Mach. Learn. Res. **5** (2004) 845–889.

[9] M.X. Hou, J.X. Li, G.M. Lu, *A supervised non-negative matrix factorization model for speech emotion recognition*, Speech Commun. **124** (2020) 13–20.

[10] A.J. Izenman, *Introduction to manifold learning*, Wiley Interdisciplinary Reviews: Comput. Statist. **4** (2012) 439–446.

[11] R. Kumar, R.S. Anand, *Bearing fault diagnosis using multiple feature selection algorithms with SVM*, Prog. Artif. Intell. **13** (2024) 119–133.

[12] J. Lee, W. Se , D.-W. Kim, *Efficient information-theoretic unsupervised feature selection*, Electron. Lett. **54** (2018) 76–77.

[13] D. Lee, H.S. Seung, *Algorithms for non-negative matrix factorization*, Proceedings of the 13th international conference on neural information processing systems (2000) 535–541.

[14] C. Li, H. Che, MF. Leung, C. Liu, Z. Yan, *Robust multi-view non-negative matrix factorization with adaptive graph and diversity constraints*, Inf. Sci. **634** (2023) 587–607.

[15] Z. Li, Y. Yang, J. Liu, X. Zhou, H. Lu, *Unsupervised feature selection using nonnegative spectral analysis*, Proceedings of the AAAI Conference on Artificial Intelligence (2021) 1026–1032.

[16] Y. Liu, K. Liu, C. Zhang, J. Wang, X. Wang, *Unsupervised feature selection via diversity-induced self-representation*, Neurocomputing **219** (2017) 350–363.

[17] H.R. Li, J.S. Zhang, G. Shi, J.M. Liu, *Graph-based discriminative nonnegative matrix factorization with label information*, Neurocomputing **266** (2017) 91–100.

[18] H. Lim, D.-W. Kim, *Pairwise dependence-based unsupervised feature selection*, Pattern Recognit. **111** (2021) 107663.

[19] H. Liu, H. Motoda, R. Setiono, Z. Zhao, *Feature selection: An ever evolving frontier in data mining*, [Feature Selection in Data Mining Logo] Proceedings of machine learning research (2010) 4–13.

[20] M. Meila, *Comparing clusterings–an information based distance*, J. Multivar. Anal. **98** (2007) 873–895.

[21] A. Moslemi, A. Ahmadian, *Subspace learning for feature selection via rank revealing QR factorization: Fast feature selection*, Expert Syst. Appl. **256** (2024) 124919.

[22] F. Nie, H. Huang, X. Cai, C. Ding, *Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization*, Adv. Neur. In. **23** (2010) 1–9.

[23] F. Nie, X. Wang, H. Huang, *Clustering and projected clustering with adaptive neighbors*, Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining (2014) 977–986.

[24] M. Noroozi, M. Salahi, S. Eskandari, *Feature selection via mixed-integer program and supervised infinite feature selection method*, J. Math. Model. **13** (2025) 341–356.

[25] O.N. Oyelade, E.F. Aminu, H. Wang, K. Rafferty, *An adaptation of hybrid binary optimization algorithms for medical image feature selection in neural network for classification of breast cancer*, Neurocomputing **617** (2025) 129018.

[26] M.G. Parsa, H. Zare, M. Ghatee, *Low-rank dictionary learning for unsupervised feature selection*, Expert Syst. Appl. **202** (2022) 117149.

[27] M. Qi, T. Wang, F. Liu, B. Zhang, J. Wang, Y. Yi, *Unsupervised feature selection by regularized matrix factorization*, Neurocomputing **273** (2018) 593–610.

[28] M. Samareh-Jahani, G. Aghamollaei, M. Eftekhari, F. Saberi-Movahed, *Unsupervised feature selection guided by orthogonal representation of feature space*, Neurocomputing **516** (2023) 61–76.

[29] M. Samareh-Jahani, F. Saberi-Movahed, M. Eftekhari, G. Aghamollaei, P. Tiwari, *Low-Redundant unsupervised feature selection based on data structure learning and feature orthogonalization*, Expert Syst. Appl. **240** (2024) 122556.

[30] H. Sohrabi, S. Esmaeili, P. Moradi, *Feature representation via graph regularized entropy weighted nonnegative matrix factorization*, AUT J. Math. Comput., **5** (2024) 289–304.

[31] J. Tang, H. Feng, *Robust local-coordinate non-negative matrix factorization with adaptive graph for robust clustering*, Inf. Sci. **610** (2022) 1058–1077.

[32] H. Wang, H. Huang, C. Ding, *Simultaneous clustering of multi-type relational data via symmetric nonnegative matrix tri-factorization*, Proceedings of the 20th ACM international conference on information and knowledge management (2011) 279–284.

[33] S. Wang, W. Pedrycz, Q. Zhu, W. Zhu, *Subspace learning for unsupervised feature selection via matrix factorization*, Pattern Recognit. **48** (2015) 10–19.

[34] S. Wang, J. Tang, H. Liu, *Embedded unsupervised feature selection*, Proceedings of the AAAI conference on artificial intelligence (2015) 470–476.

[35] Y. Wu, Y. Wang, L. Hu, J. Hu, *DCGNN: Adaptive deep graph convolution for heterophily graphs*, Inf. Sci. **666** (2024) 120–427.

[36] W.H. Wu, S. Kwong, J.H. Hou, Y.H. Jia, H.H.S. Ip, *Simultaneous dimensionality reduction and classification via dual embedding regularized nonnegative matrix factorization*, IEEE Trans. Image Process. **28** (2019) 3836–3847.

[37] W. Xu, Y. Gong, *Document clustering by concept factorization*, Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval, Sheffield, UK (2004) 202–209.

[38] Y. Yang, H.T. Shen, Z. Ma, Z. Huang, X. Zhou, *$\ell_{2,1}$-norm regularized discriminative feature selection for unsupervised learning*, Proceedings of international joint conference on artificial intelligence (2011) 1589–1594.

[39] Y. Yi, S. Lai, S. Li, J. Dai, W. Wang, J. Wang, *RRNMF-MAGL: Robust regularization non-negative matrix factorization with multi-constraint adaptive graph learning for dimensionality reduction*. Inf. Sci. **640** (2023) 119029.

[40] Z. Zhou, G. Si, H. Sun, K. Qu, W. Hou, *A robust clustering algorithm based on the identification of core points and k-NN kernel density estimation*, Expert Syst. Appl. **195** (2022) 116573.

[41] S. Zhou, P. Song, Y. Yu, W. Zheng, *Structural regularization based discriminative multi-view unsupervised feature selection*, Knowl. Based Syst. **272** (2023) 110601.

[42] Y. Zhao, S. Zheng, J. Pei, X. Yang, *Multiple discriminant preserving support subspace RBFNNs with graph similarity learning*, Inf. Sci. **619** (2023) 421–438.