Information Sciences 663 (2024) 120277

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Unsupervised feature selection using orthogonal encoder-decoder factorization

Maryam Mozafari, Seyed Amjad Seyedi, Rojiar Pir Mohammadiani, Fardin Akhlaghian Tab *

Department of Computer Engineering, University of Kurdistan, Sanandaj, Iran

ARTICLE INFO

Keywords:
Unsupervised feature selection
Encoder-decoder model
Self-representation learning
Pseudo-supervised learning
Nonnegative matrix factorization

ABSTRACT

Unsupervised feature selection (UFS) is a fundamental task in machine learning and data analysis, aimed at identifying a subset of non-redundant and relevant features from a high-dimensional dataset. Embedded methods seamlessly integrate feature selection into model training, resulting in more efficient and interpretable models. Current embedded UFS methods primarily rely on self-representation or pseudo-supervised feature selection approaches to address redundancy and irrelevant feature issues, respectively. Nevertheless, there is currently a lack of research showcasing the fusion of these two approaches. This paper proposes the Orthogonal Encoder-Decoder factorization for unsupervised Feature Selection (OEDFS) model, combining the strengths of self-representation and pseudo-supervised approaches. This method draws inspiration from the self-representation properties of autoencoder architectures and leverages encoder and decoder factorizations to simulate a pseudo-supervised feature selection approach. To further enhance the part-based characteristics of factorization, orthogonality constraints and local structure preservation restrictions are incorporated into the objective function. The optimization process is based on the multiplicative update rule, ensuring efficient convergence. To assess the effectiveness of the proposed method, comprehensive experiments are conducted on 14 datasets and compare the results with eight state-of-the-art methods. The experimental results demonstrate the superior performance of the proposed approach in terms of UFS efficiency.

1. Introduction

With the dramatic growth in information technology, high-dimensional data has become the subject of wide-ranging studies. The challenges that can occur when working with data in high-dimensional space are often referred to as the curse of dimensionality [1]. At present, with all this high-dimensional data being generated and consumed, Feature Selection (FS) holds researchers' attention for conducting more comprehensive studies [2]. Hence, feature selection has become a significant topic in machine learning [3], data mining [4], and bioinformatics [5]. There are commonly irrelevant and redundant features that can significantly affect data analysis or even lead to inappropriate models. The specific aim of FS is to select a limited number of features to overcome the effect of the curse of dimensionality problem on learning performance, while losing the least possible amount of information. Likewise,

* Corresponding author.
E-mail addresses: m.mozafari@uok.ac.ir (M. Mozafari), amjadseyedi@uok.ac.ir (S.A. Seyedi), r.pirmohamadiani@uok.ac.ir (R. Pir Mohammadiani), f.akhlaghian@uok.ac.ir (F. Akhlaghian Tab).

https://doi.org/10.1016/j.ins.2024.120277
Received 30 August 2023; Received in revised form 15 January 2024; Accepted 31 January 2024

Available online 5 February 2024

in machine learning, FS algorithms have a major impact on increasing learning efficiency, preventing overfitting, and reducing computational costs [6].

In conformity with whether label information is available or not, feature selection can be further categorized into three main groups: supervised, semi-supervised, and unsupervised feature selections [6]. Generally speaking, when data possess labels, the supervised approach is preferred. By contrast, when adequate labeled data are not obtainable, semi-supervised and unsupervised approaches are required. In practical use, labeled data can not be attained since data labeling by a human is a difficult and time-consuming task. Therefore, Unsupervised Feature Selection (UFS) is increasingly considered in machine learning research due to the difficulty and time-consuming nature of data labeling. UFS has been subdivided into three main methods: filter methods, wrapper methods, and embedding-based methods through different selection strategies. Filter-based methods select more significant features based on the intrinsic characteristics of data samples. They apply statistical criteria or information theory to evaluate the features and then choose the top-ranked features [7,8]. In wrapper methods, the optimal feature subset selection depends on the performance of a specific learning algorithm, but its major drawback is high computational costs [7]. The embedding methods try to combine the feature selection with a learning model. There is a general perception that feature selection in these methods has been an essential part of the model construction process. Practically speaking, embedded methods consider feature selection and evaluation in the same optimization process.

Embedded methods commonly make use of sparse learning models for the feature selection problem. Sparse learning models can eliminate a considerable number of redundant features to retain more relevant features. A well-known and successful sparse learning model for feature selection is implemented by minimizing an empirical error penalized by a sparse regularization term [2]. Self-representation and pseudo-supervised learning approaches in the field of UFS were further developed by recent advances in sparse learning methods. In the recent literature on UFS problem, self-representation-based and pseudo-supervised-based models exceeded all other views in importance [9]. In the generic self-representation model, each sample can be represented as a linear combination of other samples, indicating a sample-side self-representation. Self-representation has also been used for UFS, where one feature can be well reconstructed by the linear or non-linear combination of its relevant feature set. This method indicates a feature-side self-representation and provides some insights into UFS. For example, Zhu et al. [10] proposed a regularized self-representation (RSR) model for unsupervised feature selection.

From an alternative perspective, in supervised tasks, discrimination information is encoded in class labels. However, in unsupervised scenarios, a very common solution to find out the discriminative features is generating pseudo-label information. Since the cluster structure is one of the crucial underlying structures of data, numerous pseudo-supervised approaches have emerged to exploit these structures for unsupervised feature selection. These methods leverage cluster structures through various techniques, including spectral analysis [11–14], subspace clustering [15], and matrix factorization [11,16,17]. Similar to supervised feature selection methods, pseudo labels have also been used extensively to help select informative features in unsupervised feature selection methods [12,13,18,19].

Existing UFS methods usually consider the problem in terms of either self-representation or pseudo-supervised approaches. The first approach is more capable of detecting redundant features, while the second approach can detect the more informative features. Since both ignoring redundant features and selecting informative and discriminative features are essential for suitable feature selection, a fusion of the two approaches is essential. However, it would be challenging to design a UFS model that considers both self-representation and pseudo-supervised learning views simultaneously and takes into consideration the interaction between them. These two methods have different foundational principles and mathematical frameworks, making their integration difficult. Developing a UFS algorithm that blends these methodologies requires either creating a novel algorithm or adapting existing ones. Additionally, the computational requirements can increase, especially with high-dimensional data, impacting efficiency and scalability. The integration also complicates hyperparameter tuning due to extra parameters from both approaches. To our knowledge, no method has been developed to simultaneously incorporate both of these approaches.

Autoencoders are self-representation models frequently used in unsupervised machine learning. They embody the coexistence relationship between encoder and decoder models, making them a fundamental tool for feature learning. The interaction and integration of encoder and decoder models in feature learning involve how these two components work together to learn useful features from the input data [20]. On one hand, as it reduces dimensionality, the encoder learns to extract meaningful latent features from the input data. These features are typically distributed in the latent space and represent important characteristics of the input data. On the other hand, the decoder learns to reconstruct the input data from the encoded representation. This process helps in fine-tuning the features learned by the encoder, as it enforces the model to retain essential information in the reduced representation. This integration in feature learning is a crucial aspect of autoencoders and similar architectures. It facilitates the development of representations that capture the most salient aspects of the input data, making them valuable for tasks such as data compression, denoising, anomaly detection, and even generation [21].

In recent years, matrix factorization-based methods have been widely adopted for machine learning tasks such as feature selection [11,16,17,22]. Low-rank matrix factorization, as a data reconstruction method, is suitable for unsupervised tasks. The first mention of Nonnegative Matrix Factorization (NMF) [23] has proposed a part-based data representation method that decomposes the data matrix into a feature weight matrix and representation matrix, and it has been proven to inherently have a clustering characteristic [24]. The NMF model, as a decoder-only architecture, reconstructs the original data samples from its representation in a new space. On the other hand, the Encoder-Decoder architecture is a self-representation model that learns how to encode original data and reconstruct the data from the encoded representation. Encoder-decoder NMF, as a structure extension of NMF, integrates the decoder and encoder components that have been proposed for the community detection task [25].

Motivated by the mentioned properties, this paper proposes the Orthogonal Encoder-Decoder NMF model for the UFS problem (OEDFS). This model takes both self-representation and pseudo-supervised learning views into account, maintaining complexity comparable to other UFS methods [15,26,27]. More specifically, in a pseudo-supervised manner, the decoder factorization generates pseudo-labels due to its clustering characteristic. Simultaneously, the encoder uses these clusters as pseudo-labels to train a supervised learning model. Moreover, decoder and encoder factorization components refine and verify each other in a self-representation manner. These intrinsic properties of the proposed model ensure that the most representative and informative features can be determined. Besides, to preserve the local geometric of data in the representation space, a graph regularization expression is included in the model. By incorporating this regularization, the resulting data representations will respect the graph's similarity structure, ensuring that similar samples remain close in the reduced-dimensional space [28]. Additionally, to alleviate uncertain clustering results and form more distinct clusters, explicit orthogonality is imposed on the predicted pseudo-label matrix. This constraint enforces a form of hard clustering, where every sample is unambiguously assigned to a single cluster [24].

Based on all mentioned above, an unsupervised feature selection with an encoder-decoder structure (OEDFS) is formulated in this paper. The extensive empirical and theoretical validation has definitively established the superior performance of the proposed method. The major contributions are summarized as follows:

- The basic Encoder-Decoder model, named EDFS, is proposed for unsupervised feature selection. Encoder and decoder terms are integrated to benefit from the advantages of self-representation and pseudo-supervised learning views. This means that the decoder term exploits the cluster structure of the original data, and the encoder generates the relations between the projected data and the pseudo-labels.
- The proposed Encoder-Decoder feature selection structure instinctively imposes an implicit orthogonality constraint on the feature weight matrix. This constraint leads to a sparse feature representation, encouraging uncorrelated latent features and improving the model's performance.
- A graph regularization term is employed to ensure the consistency between the original feature points and the new feature points for every sample. The principle is that a closer correlation between two instances in the original data space indicates a closer correlation between their representations in the latent space, preserving the adjacency of instances optimally in both spaces.
- Finally, an orthogonal model named OEDFS is proposed, which increases clustering property by imposing an explicit orthogonality constraint on the pseudo-label matrix to achieve more distinctive clusters (hard clustering).

This paper is organized as follows. In Section 2, we review some related works for UFS. Section 3 clearly specifies the details of our method for UFS. The experimental results are shown and analyzed in Section 4. Finally, Section 5 provides a brief conclusion for this paper.

2. Related work

In recent years, there has been increased interest in Embedded feature selection methods. In the context of developing a learning model, unsupervised feature selection is typically achieved through embedded methods. These methods specifically focus on integrating feature selection into the learning algorithm (e.g., nonnegative matrix factorization [9], spectral regression [12], and discriminative analysis [29]). Most embedded methods reconstruct the original data by a self-representation model or deal with the UFS problem through a supervised feature selection model by generating pseudo-class labels. The remainder of this section provides an overview of the UFS methods, specifically, previous embedded methods that have a pseudo-supervised learning or self-representation basis and are important to clarify the proposed method in this study.

2.1. Pseudo-supervised approach

Learning the underlying structure of data is crucial in unsupervised tasks and has become the focus of many feature selection studies, ensuring that the selected features capture the underlying structure. The learned cluster structure in unsupervised feature selection is often compared to a pseudo-supervised regression model. Typically, traditional pseudo-supervised learning methods are based on spectral analysis and sparse constraints for feature subset selection [7]. For example, Multi-cluster FS (MCFS) [12], one of the earliest works, aims to preserve the multi-cluster structure information of the original data through spectral analysis and an L_1-regularized regression model. Zhao et al. [30] proposed a spectral-based feature selection model, adopting an embedding model (i.e., using sparse multi-output regression with $L_{2,1}$ constraint) to alleviate the inefficiency of its previous models due to redundant features. Unsupervised Discriminative Feature Selection (UDFS) [29] applied discriminative analysis and $L_{2,1}$ norm regularization to benefit from local discriminative information by considering the manifold structure and simultaneously exploiting feature correlations. Nonnegative Discriminative Feature Selection (NDFS) is a spectral clustering based method that combines cluster label learning with FS to explore discriminative information. Besides, nonnegativity is used to obtain more accurate cluster labels [11]. Robust Unsupervised Feature Selection (RUFS) [17] proposed a robust orthogonal nonnegative matrix factorization for learning cluster indicator matrix with local learning regularization. Robust Spectral Feature Selection (RSFS) [31] by jointly employing robust graph embedding and robust sparse spectral regression was proposed to be an effective way to deal with noises in the learned cluster labels. Unsupervised Feature Selection with Adaptive Structure Learning (FSASL) [19] simultaneously performs feature selection and local and global structure learning. The defined graph learns the data structure adaptively from the selected features, and chooses significant features to preserve the refined structure. In the mentioned traditional methods, the pseudo-label generation is performed

as a preprocessing step, while recent methods attempt to consider the generating pseudo-label process and feature selection in a unified model. In this approach, the generic pseudo-supervised UFS model can be formulated by:

$$\min_{\boldsymbol{W},\boldsymbol{T}} f(\boldsymbol{T}|\boldsymbol{X}) + g(\boldsymbol{W}|\boldsymbol{X},\boldsymbol{T}) + z(\boldsymbol{W}), \tag{1}$$

where $\boldsymbol{X}$ represents the input data matrix, the function $f(\boldsymbol{T}|\boldsymbol{X})$ generates pseudo-labels $\boldsymbol{T}$. The pseudo-supervised regression $g(\boldsymbol{W}|\boldsymbol{X},\boldsymbol{T})$ aims to derive the feature weight matrix $\boldsymbol{W}$ with pseudo-labels $\boldsymbol{T}$ as the regression target, and $z(\boldsymbol{W})$ serves as a sparse penalty term for feature selection.

Wang et al. [32] study directly embedded feature selection into a clustering algorithm via sparse learning. This method which is called EUFS applied non-negative orthogonality on the cluster indicator matrix. Robust joint graph sparse coding (RJGSC) [33] proposed a UFS model that preserves the local structure of data by considering subspace learning and joint sparse regression for feature selection. Parsa et al. [15] proposed a UFS method named Subspace Clustering Feature Selection (SCFS), which addresses UFS by employing subspace learning to capture sample similarities and using a regularized regression model to consider the latent structure of data. Zare et al. [34] proposed a UFS method in which the global and local similarity of samples was preserved by symmetric non-negative matrix factorization and spectral analysis, respectively. Zhao et al. [35] have considered a compact estimate of data reconstruction. It utilized a dual concept learning approach, including clustering data and features with lower-dimensional matrices, which has made the results more stable and reliable. In OCLSP [36], orthogonal basis clustering and feature selection are jointly performed, and adaptive graph regularization aids the model in preserving the local data structure. Shang et al. [26] proposed a UFS algorithm that utilizes the intrinsic structure of data space and feature space to preserve the consistency between the original data and the new representation of data. In UDPFS [37], a discriminative projection is learned by applying fuzziness to subspace learning. Specifically, data samples are represented as low-dimensional observations, while sparse membership is used to assign soft cluster pseudo-indicators. Yuan et al. [9] proposed a method that combines pseudo-label matrix learning with a sample-side self-expression module to explore data structure and mapping relationships between data and learned labels. Also, two regularization expressions (i.e., graph regularization) are used to consider the local structure of the original data. More recently, RULSP [27] utilizes matrix factorization to predict cluster labels and identify important features. Its orthogonal constraint ensures accurate class labeling, while a local preserving term enhances the projected data

2.2. Self-representation approach

Self-representation learning approaches assume each sample can be represented as a combination of the other samples, as $\boldsymbol{X} \approx \boldsymbol{X}\boldsymbol{S}_n$, or each feature can be represented as a combination of the other features, as $\boldsymbol{X} \approx \boldsymbol{S}_d\boldsymbol{X}$, indicating sample-side self-representation or feature-side self-representation, respectively. Motivated by this property, feature-side self-representation can effectively achieve a feature representation coefficient matrix to identify correlated features. According to [10], this feature selection approach can be formulated as follows:

$$\min_{\boldsymbol{S}_d} f(\boldsymbol{S}_d\boldsymbol{X}|\boldsymbol{X}) + z(\boldsymbol{S}_d), \tag{2}$$

where $f(\boldsymbol{S}_d\boldsymbol{X}|\boldsymbol{X})$ indicates the feature-side self-representation of the data matrix $\boldsymbol{X}$. To guide feature subset selection, sparse norm regularization $z(\boldsymbol{S}_d)$ is imposed on the coefficient matrix. In the following overview, we intend to introduce some self-representation UFS methods.

Regularized Self-Representation (RSR) [10] is one of the first efficient self-representation models for unsupervised feature selection. It proposes a feature-side self-representation model in which the $L_{2,1}$-norm is used to minimize the reconstruction error and regularize the representation coefficients. This formulation has been developed and is a core component of some extended UFS models, where RSR plays the most fundamental role. For instance, Lu et al. [38] introduced a UFS method based on the self-representation approach to explore relationships between features, incorporating an objective function that includes manifold learning to exploit the local structure of the data. To mitigate the impact of noisy data on feature selection, Tang et al. [39] extended the RSR model to a robust UFS method by using an $L_{2,1}$-norm to characterize the feature representation residual matrix and an L_1-norm-based graph regularization to preserve the local geometric structure. Unsupervised Feature Selection via Local Structure Learning and Sparse Learning (LSS-FS) [40] combines feature self-representation and manifold learning. The local structure of the samples is captured by a graph matrix learned from a low-dimensional space. Tang et al. [41] proposed robust unsupervised feature selection via dual self-representation and manifold regularization (DSRMR). This approach employs feature self-representation to learn the feature coefficient matrix and sample self-representation to preserve the local geometric structures of the original data by learning the similarity graph.

It is worth mentioning that most of these existing methods focus on one specific view, either self-representation or pseudo-supervised learning. However, a more holistic strategy appears to be most effective for UFS, considering both self-representation and pseudo-supervised learning simultaneously.

3. Proposed method

This section introduces the proposed UFS method, which utilizes matrix factorization within a pseudo-supervised framework characterized by its inherent self-representation property. The basic model is a self-representation Encoder-Decoder model that

simultaneously predicts pseudo-labels and learns pseudo-supervised regression to extract feature importance (subsection 3.2). Additionally, graph regularization is included in the model to capture geometric structure (subsection 3.3). We extend the proposed model to learn more discriminative latent features by imposing an explicit orthogonality constraint on the representation matrix (subsection 3.4). All these notions are taken into account in a unified objective function, and an effective optimization algorithm is provided to achieve a satisfactory solution (subsection 3.6). Furthermore, convergence analysis demonstrates the stability and effectiveness of the UFS method in reaching an optimal solution (subsection 3.7). Additionally, a thorough computational complexity analysis is conducted to assess the efficiency and scalability of the proposed model in handling large-scale datasets, affirming its practical utility (subsection 3.8). We will present a more elaborate explanation of our models in the following subsections.

3.1. Notations

In this section, we provide some notations and essential preliminaries for the problem formulation. Throughout this paper, bold capital letters represent matrices, bold lowercase letters represent vectors, and italic lowercase letters represent scalar values. For an arbitrary matrix $\boldsymbol{M} \in \mathbb{R}^{m\times n}$, the (i,j)-th entry of $\boldsymbol{M}$ is denoted by M_{ij}, and $|M_{ij}|$ denotes the absolute value of M_{ij}. The vectors $\boldsymbol{m}^{(i)}$ and $\boldsymbol{m}_j$ denote the i-th row and j-th column of $\boldsymbol{M}$, respectively. $\mathrm{Tr}(\boldsymbol{M})$ denotes the trace of the matrix $\boldsymbol{M}$ if $\boldsymbol{M}$ is square, and $\boldsymbol{M}^\top$ denotes the transpose of $\boldsymbol{M}$. The l_p-norm of vector $\boldsymbol{v} \in \mathbb{R}^n$ is defined as $\|\boldsymbol{v}\|_p = \left(\sum_{i=1}^n |v_i|^p\right)^{\frac{1}{p}}$. Additionally, the Frobenius norm of matrix $\boldsymbol{M}$ is defined as $\|\boldsymbol{M}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n M_{ij}^2} = \sqrt{\mathrm{Tr}(\boldsymbol{M}^\top \boldsymbol{M})}$. Specifically, the $L_{2,1}$-norm of matrix $\boldsymbol{M}$ is denoted by $\|\boldsymbol{M}\|_{2,1} = \sum_{i=1}^m \sqrt{\sum_{j=1}^n M_{ij}^2} = \sum_{i=1}^m \|\boldsymbol{m}^{(i)}\|$.

3.2. Basic model

Nonnegative matrix factorization (NMF) learns a low-dimensional data representation with interpretability properties. Given a nonnegative data matrix $\boldsymbol{X} \in \mathbb{R}_{\geq 0}^{d\times n}$, where each column represents an observation, NMF projects the original data into a latent space, expecting the feature representation to capture non-redundant and relevant features in the original space. Based on this, NMF factorizes the original data matrix $\boldsymbol{X}$ into a feature weight matrix $\boldsymbol{W}$ and a representation matrix $\boldsymbol{H}$, such that $\boldsymbol{X} \approx \boldsymbol{W}\boldsymbol{H}$. The columns of $\boldsymbol{W}$ span the new space, and the columns of $\boldsymbol{H}$ are responsible for representing samples in the latent space. The non-negativity constraint on NMF leads to a part-based representation that gives an intrinsic clustering property to the model, and its latent space can be considered a cluster space. In other words, one possible explanation is that NMF intrinsically exhibits soft clustering properties, generating $\boldsymbol{H}$ (i.e., cluster membership) as follows:

$$\min_{\boldsymbol{W},\boldsymbol{H}} \|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{H}\|_F^2, \quad \text{s.t.} \quad \boldsymbol{W}, \boldsymbol{H} \geq 0. \tag{3}$$

One strategy to address the UFS problem, employed in many FS methods, is selecting features that can effectively preserve the distribution of samples, often in the form of a cluster structure (pseudo-supervised learning), or reconstruct the data (self-representation learning). Cluster labels encode intrinsic discriminative information. Thus, generating pseudo-labels by reconstructing the original data is an effective approach for identifying discriminative features using label information. To establish our pseudo-supervised model, similar to (1), we introduce a pseudo-supervised regression term to (3). As the main contribution of this paper, this model utilizes the pseudo-labels generated by NMF as the target, where the data matrix $\boldsymbol{X}$ is projected into the pseudo-label matrix $\boldsymbol{H}$, as follows:

$$\min_{\boldsymbol{W},\boldsymbol{H}} \|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{H}\|_F^2 + \|\boldsymbol{H} - \boldsymbol{W}^\top \boldsymbol{X}\|_F^2, \quad \text{s.t.} \quad \boldsymbol{W}, \boldsymbol{H} \geq 0, \tag{4}$$

where $\boldsymbol{W} \in \mathbb{R}^{d\times r}$ is the feature weight matrix. Each row of $\boldsymbol{W}$ represents a weight vector used to project a sample into a cluster. Therefore, the magnitude of the weight vector indicates the significance of the feature for that cluster. By solving (4) based on pseudo-supervised learning, the feature weight matrix $\boldsymbol{W}$ is obtained to carry out feature selection. Incorporating this regression term as an Encoder NMF into the basic NMF model creates an Encoder-Decoder NMF with interesting characteristics. The integration and interaction of encoder and decoder factorizations facilitate the development of representations that capture the most hidden aspects of the input. Jointly training the encoder and decoder in factorization models to optimize a reconstruction loss enables the model to learn complementary representations. In this Autoencoder structure, both modules verify and refine each other to extract optimal pseudo-labels $\boldsymbol{H}$ and feature weights $\boldsymbol{W}$. Additionally, the implicit self-representation characteristic arises from the integration of the Encoder and Decoder modules without introducing additional components or hyperparameters. To be more specific, the decoder and encoder terms share the same projection matrix $\boldsymbol{W}$; therefore, the following approximation equations are attained:

$$\boldsymbol{X} \approx \boldsymbol{W}\boldsymbol{H}, \tag{5}$$

$$\boldsymbol{H} \approx \boldsymbol{W}^\top \boldsymbol{X}. \tag{6}$$

Now by substitution (6) in (5), we have

$$\boldsymbol{X} \approx \boldsymbol{W}\boldsymbol{W}^\top \boldsymbol{X}, \tag{7}$$

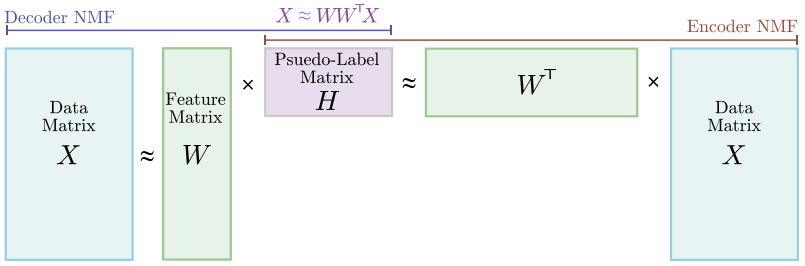

Fig. 1. The illustration of the Encoder-Decoder Factorization model for Unsupervised Feature Selection (EDFS). The Decoder module decomposes the data matrix $\boldsymbol{X}$ to generate the representation matrix $\boldsymbol{H}$ as $\boldsymbol{X} \approx \boldsymbol{W}\boldsymbol{H}$, and the Encoder module fine-tunes this representation by $\boldsymbol{H} \approx \boldsymbol{W}^{\top}\boldsymbol{X}$. The implicit self-representation $\boldsymbol{X} \approx \boldsymbol{W}\boldsymbol{W}^{\top}\boldsymbol{X}$ arises from the integration of the Encoder and Decoder modules.

which is equivalent to the projective NMF [42], assuming that the basis lies in the subspace spanned by the original samples. Additionally, similar to (2), equation (7) implies a feature-side self-representation, where $\boldsymbol{S} = \boldsymbol{W}\boldsymbol{W}^{\top}$. This self-representation property highlights the reconstruction of features, identifying redundant features based on the assumption that each feature can be represented by a combination of its relevant features. Moreover, the Encoder-Decoder model achieves a sparser feature weight matrix $\boldsymbol{W}$ due to its implicit orthogonality. Fig. 1 provides a schematic representation of the OEDFS model.

Consequently, to assess the importance of features, the feature weight matrix $\boldsymbol{W}$ is subjected to row sparsity, formulated as an $L_{2,1}$ norm minimization term. The sparse regularization term induces numerous feature coefficients in the $\boldsymbol{W}$ matrix to become small or exactly zero, primarily eliminating corresponding features (redundant features). Another supporting factor for the efficacy of the $L_{2,1}$ norm group sparsity is the substantial reduction of redundancy in feature selection, contributing to high performance. By incorporating the $L_{2,1}$ regularization into the objective function (4), the basic model can be expressed as follows:

$$\min_{\boldsymbol{W},\boldsymbol{H}} \|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{H}\|_F^2 + \|\boldsymbol{H} - \boldsymbol{W}^{\top}\boldsymbol{X}\|_F^2 + \gamma\|\boldsymbol{W}\|_{2,1}, \quad \text{s.t.} \quad \boldsymbol{W}, \boldsymbol{H} \geq 0, \tag{8}$$

where γ is the sparsity regularization hyperparameter. The Encoder-Decoder (ED) model in (8) aims to leverage both self-representation and pseudo-supervised learning methods to select non-redundant and more relevant features in an unsupervised learning paradigm.

3.3. Locality preservation

In representation learning methods, it is imperative to preserve the local geometric structure of data in the representation space [28]. Therefore, manifold regularization techniques are utilized to maintain the local structure in various learning tasks, such as feature selection. The local geometric structure can be effectively modeled by the nearest neighbor graph over the distribution of samples. Hence, adding a graph regularization term is becoming increasingly common, taking into account the intrinsic sample-level information of the data. More specifically, the graph regularization term is defined with the assumption that if some samples, e.g., $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, are similar in the high-dimensional space, then their corresponding representations (i.e., $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$) should have a similar relation. This underlying assumption can be captured by the following embedding function:

$$\min_{\boldsymbol{H}} \mathcal{R} = \sum_{i=1}^{n}\sum_{j=1}^{n}\|\boldsymbol{h}_i - \boldsymbol{h}_j\|^2 A_{ij}, \tag{9}$$

where A_{ij} indicates the similarity between two samples i and j in the nearest neighbor graph. One of the most common options for calculating this similarity is using the heat kernel function, defined as follows:

$$A_{ij} = \begin{cases} e^{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{\sigma}} & \text{if } \boldsymbol{x}_i \in \mathcal{N}_k(\boldsymbol{x}_j) \text{ or } \boldsymbol{x}_j \in \mathcal{N}_k(\boldsymbol{x}_i), \\ 0 & \text{otherwise}, \end{cases} \tag{10}$$

where $\mathcal{N}_k(\boldsymbol{x}_i)$ represents the set of k nearest neighbors of $\boldsymbol{x}_i$, and σ is the heat kernel width. To simplify (9) mathematically, we can write:

$$\min_{\boldsymbol{H}} \mathcal{R} = \sum_{i=1}^{n}\sum_{j=1}^{n}\|\boldsymbol{h}_i - \boldsymbol{h}_j\|^2 A_{ij} = \mathrm{Tr}(\boldsymbol{H}\boldsymbol{B}\boldsymbol{H}^{\top}) - \mathrm{Tr}(\boldsymbol{H}\boldsymbol{A}\boldsymbol{H}^{\top}) = \mathrm{Tr}(\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^{\top}), \tag{11}$$

where $\boldsymbol{B}$ is a diagonal matrix with elements defined as $B_{ii} = \sum_{j=1}^{n} A_{ij}$, and $\boldsymbol{L} \in \mathbb{R}^{n\times n}$ is the Laplacian matrix given by $\boldsymbol{L} = \boldsymbol{B} - \boldsymbol{A}$.

In alignment with typical representation models, such as embedded-based UFS approaches [15,26,27], we integrate the widely-used manifold regularization term (11) into the basic model (8). Leveraging insights gained from the two previous sections, we present the EDFS objective function as follows:

$$\min_{\boldsymbol{W},\boldsymbol{H}} \|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{H}\|_F^2 + \|\boldsymbol{H} - \boldsymbol{W}^{\top}\boldsymbol{X}\|_F^2 + \lambda\mathrm{Tr}(\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^{\top}) + \gamma\|\boldsymbol{W}\|_{2,1}, \quad \text{s.t.} \quad \boldsymbol{W}, \boldsymbol{H} \geq 0, \tag{12}$$

where the hyperparameter λ controls the preservation of locality. The graph-regularized Encoder-Decoder factorization model (12) learns a more informative representation by preserving the local structure.

3.4. Orthogonal encoder-decoder feature selection

It is noticeable that conventional NMF models have some major shortcomings, including their inability to generate a unique decomposition and to support coherent part interpretation. The non-negativity and orthogonality characteristics seem to make matrix factorization-based models more applicable to real-world problems [43–50]. Therefore, more attention should be focused on orthogonality constraints to obtain more precise latent cluster information in addition to the intrinsic non-negativity of NMF-based models. As the final contribution, this work aims to present the most effective solution by imposing an explicit orthogonality constraint on the cluster indicator or representation matrix $\boldsymbol{H}$ in the proposed model. Imposing orthogonal constraints on the cluster indicator matrix (i.e., $\boldsymbol{H}\boldsymbol{H}^\top = \boldsymbol{I}$) leads to hard clustering by forcing each column of $\boldsymbol{H}$ to have only one non-zero element [24]. As a result, an orthogonal nonnegative model can generate unique solutions and enhanced separation. Specifically, the orthogonal constraint $\boldsymbol{H}\boldsymbol{H}^\top = \boldsymbol{I}$ is included in the EDFS model (12), resulting in the final OEDFS model as follows:

$$\min_{\boldsymbol{W},\boldsymbol{H}} \|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{H}\|_F^2 + \|\boldsymbol{H} - \boldsymbol{W}^\top\boldsymbol{X}\|_F^2 + \lambda \mathrm{Tr}(\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^\top) + \gamma\|\boldsymbol{W}\|_{2,1}, \quad \text{s.t.} \quad \boldsymbol{W},\boldsymbol{H} \geq 0, \boldsymbol{H}\boldsymbol{H}^\top = \boldsymbol{I}. \tag{13}$$

The Orthogonal Encoder-Decoder Feature Selection (OEDFS) is assumed to employ pseudo-supervised and self-representation learning methods. Similarly, discriminative information is embedded in a projection matrix named the feature weight matrix $\boldsymbol{W}$, which is obtained through learning regularized regression between the projected data matrix $\boldsymbol{W}^\top\boldsymbol{X}$ and the hard pseudo-label matrix $\boldsymbol{H}$. In comparison, $\boldsymbol{H}$ is utilized for subspace learning, preserving both local and global structures through manifold regularization and the orthogonal constraint, respectively.

3.5. Feature ranking

The feature weight matrix $\boldsymbol{W}$ obtained from our optimization model is well-suited for the task of feature selection, as it reveals the distinctive features in the data. Here, the i-th row of $\boldsymbol{W}$, denoted as $\boldsymbol{w}^{(i)} \in \mathbb{R}^r$, corresponds to the importance of the i-th feature. More significantly, $\|\boldsymbol{w}^{(i)}\|$ is used to rank features based on their discrimination ability. Finally, we obtain the new data matrix $\boldsymbol{X}_{\text{subset}} \in \mathbb{R}^{m\times n}$ by selecting the first m ranked features.

3.6. Model optimization

The OEDFS model (13) involves two variables, namely $\boldsymbol{W}$ and $\boldsymbol{H}$, which need to be solved. We employ an iterative optimization method to address this problem. In this approach, one optimization variable is treated as a fixed variable while the optimum value for the other is calculated. To further simplify, we convert the sparsity regularization term $\|\boldsymbol{W}\|_{2,1}$ in (13) into a trace form $\mathrm{Tr}(\boldsymbol{W}^\top\boldsymbol{D}\boldsymbol{W})$, where $\boldsymbol{D} \in \mathbb{R}^{d\times d}$ is a diagonal matrix whose diagonal elements are

$$D_{ii} = \frac{1}{2\|\boldsymbol{w}^{(i)}\|}, \ \forall i \in \{1, 2, \ldots, d\}. \tag{14}$$

Therefore, an equivalent formulation of the OEDFS optimization problem (13) is presented as follows:

$$\min_{\boldsymbol{W},\boldsymbol{H}} \|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{H}\|_F^2 + \|\boldsymbol{H} - \boldsymbol{W}^\top\boldsymbol{X}\|_F^2 + \lambda \mathrm{Tr}(\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^\top) + \gamma \mathrm{Tr}(\boldsymbol{W}^\top\boldsymbol{D}\boldsymbol{W}), \quad \text{s.t.} \quad \boldsymbol{W},\boldsymbol{H} \geq 0, \quad \boldsymbol{H}\boldsymbol{H}^\top = \boldsymbol{I}. \tag{15}$$

The majority of scenarios involving optimization problems that incorporate orthogonal constraints exhibit NP-hard characteristics. This complexity arises primarily due to the nonlinear and non-convex nature of orthogonal constraints [24]. For the constraint optimization problem (15), we incorporate Lagrange multipliers $\boldsymbol{\Xi}$ into our model for orthogonal constraints. Therefore, a sensible approach to minimizing (15) involves minimizing the Lagrangian function. Additionally, the Lagrange coefficients are introduced to enforce the non-negativity constraints.

- *Updating rule for the feature weight matrix $\boldsymbol{W}$.*
  We fix $\boldsymbol{H}$ in (15), which assists in formulating the model's objective function as follows:

$$\min_{\boldsymbol{W}} \mathcal{L}(\boldsymbol{W}, \boldsymbol{\Theta}) = \|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{H}\|_F^2 + \|\boldsymbol{H} - \boldsymbol{W}^\top\boldsymbol{X}\|_F^2 + \gamma \mathrm{Tr}(\boldsymbol{W}^\top\boldsymbol{D}\boldsymbol{W}) - \mathrm{Tr}(\boldsymbol{\Theta}\boldsymbol{W}^\top), \tag{16}$$

  where $\boldsymbol{\Theta}$ is a Lagrangian coefficient for imposing a nonnegativity constraint on $\boldsymbol{W}$. To solve (16) using the gradient-based approach, we need to simplify it into the trace form, as follows:

$$\min_{\boldsymbol{W}} \mathcal{L}(\boldsymbol{W}, \boldsymbol{\Theta}) = \mathrm{Tr}[(\boldsymbol{X} - \boldsymbol{W}\boldsymbol{H})^\top(\boldsymbol{X} - \boldsymbol{W}\boldsymbol{H})] + \mathrm{Tr}[(\boldsymbol{H} - \boldsymbol{W}^\top\boldsymbol{X})^\top(\boldsymbol{H} - \boldsymbol{W}^\top\boldsymbol{X})] + \gamma \mathrm{Tr}(\boldsymbol{W}^\top\boldsymbol{D}\boldsymbol{W}) - \mathrm{Tr}(\boldsymbol{\Theta}\boldsymbol{W}^\top). \tag{17}$$

  The partial derivative of the above expression with respect to $\boldsymbol{W}$ is found as follows:

$$\frac{\partial \mathcal{L}(\boldsymbol{W}, \boldsymbol{\Theta})}{\partial \boldsymbol{W}} = -4\boldsymbol{X}\boldsymbol{H}^\top + 2\boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^\top + 2\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{W} + 2\gamma\boldsymbol{D}\boldsymbol{W} - \boldsymbol{\Theta}. \tag{18}$$

Algorithm 1 Orthogonal Encoder-Decoder unsupervised Feature Selection (OEDFS).

Input: Feature matrix $\boldsymbol{X}\in\mathbb{R}^{d\times n}$, regularization parameters λ and γ, and latent factor r;
Output: Feature score $s_i = \|\boldsymbol{w}^{(i)}\|, \forall i \in \{1,2,\ldots,d\}$.

```
Initialize W, H randomly; t = 0;
Construct the similarity matrix A according to (10);
while t < MaxIteration do
    Update D by (14);
    Update W by (21);
    Update H by (28);
    t = t + 1;
end while
Return W;
Evaluate the feature score by s_i = ||w^(i)||.
```

Let $\frac{\partial \mathcal{L}(\boldsymbol{W},\boldsymbol{\Theta})}{\partial \boldsymbol{W}} = \boldsymbol{0}$; then, we get

$$\boldsymbol{\Theta} = -4\boldsymbol{X}\boldsymbol{H}^\top + 2\boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^\top + 2\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{W} + 2\gamma\boldsymbol{D}\boldsymbol{W}. \tag{19}$$

Principally, the Karush-Kuhn-Tucker (KKT) condition is employed, and we can achieve the following:

$$\boldsymbol{W} \odot \boldsymbol{\Theta} = \boldsymbol{0}, \tag{20}$$

where $\odot$ denotes an element-wise product. Solving (20) yields the update rule for $\boldsymbol{W}$ as follows:

$$\boldsymbol{W} \leftarrow \boldsymbol{W} \odot \frac{2\boldsymbol{X}\boldsymbol{H}^\top}{\boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^\top + \boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{W} + \gamma\boldsymbol{D}\boldsymbol{W}}. \tag{21}$$

- *Updating rule for the pseudo-label matrix* $\boldsymbol{H}$.
  Fixing $\boldsymbol{W}$ in (15), we obtain the following optimization objective function:

$$\min_{\boldsymbol{H}} \mathcal{L}(\boldsymbol{H},\boldsymbol{\Xi}) = \|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{H}\|_F^2 + \|\boldsymbol{H} - \boldsymbol{W}^\top\boldsymbol{X}\|_F^2 + \lambda\mathrm{Tr}(\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^\top) - \mathrm{Tr}(\boldsymbol{\Xi}\boldsymbol{H}\boldsymbol{H}^\top - \boldsymbol{\Xi}). \tag{22}$$

Transforming (22) into the expression below is a computational convenience; thus, we have:

$$\min_{\boldsymbol{H}} \mathcal{L}(\boldsymbol{H},\boldsymbol{\Xi}) = \mathrm{Tr}[(\boldsymbol{X} - \boldsymbol{W}\boldsymbol{H})^\top(\boldsymbol{X} - \boldsymbol{W}\boldsymbol{H})] + \mathrm{Tr}[(\boldsymbol{H} - \boldsymbol{W}^\top\boldsymbol{X})^\top(\boldsymbol{H} - \boldsymbol{W}^\top\boldsymbol{X})] + \lambda\mathrm{Tr}(\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^\top) - \mathrm{Tr}(\boldsymbol{\Xi}\boldsymbol{H}\boldsymbol{H}^\top - \boldsymbol{\Xi}). \tag{23}$$

As the gradient descent method is employed to update the matrix $\boldsymbol{H}$, the next step involves computing the partial derivative of problem (23) with respect to $\boldsymbol{H}$, as follows:

$$\frac{\partial \mathcal{L}(\boldsymbol{H},\boldsymbol{\Xi})}{\partial \boldsymbol{H}} = -4\boldsymbol{W}^\top\boldsymbol{X} + 2\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{H} + 2\boldsymbol{H} + 2\lambda\boldsymbol{H}\boldsymbol{L} - 2\boldsymbol{\Xi}\boldsymbol{H}. \tag{24}$$

Let $\frac{\partial \mathcal{L}(\boldsymbol{H},\boldsymbol{\Xi})}{\partial \boldsymbol{H}} = \boldsymbol{0}$. Then, we can obtain the following equation:

$$\boldsymbol{\Xi}\boldsymbol{H} = -2\boldsymbol{W}^\top\boldsymbol{X} + \boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{H} + \boldsymbol{H} + \lambda\boldsymbol{H}\boldsymbol{L}. \tag{25}$$

According to $\boldsymbol{H}\boldsymbol{H}^\top = \boldsymbol{I}$, we have

$$\boldsymbol{\Xi} = -2\boldsymbol{W}^\top\boldsymbol{X}\boldsymbol{H}^\top + \boldsymbol{W}^\top\boldsymbol{W} + \boldsymbol{I} + \lambda\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^\top. \tag{26}$$

Substituting (26) into (24), we can obtain

$$\frac{\partial \mathcal{L}(\boldsymbol{H},\boldsymbol{\Xi})}{\partial \boldsymbol{H}} = -4\boldsymbol{W}^\top\boldsymbol{X} - 2\lambda\boldsymbol{H}\boldsymbol{A} + 4\boldsymbol{W}^\top\boldsymbol{X}\boldsymbol{H}^\top\boldsymbol{H} + 2\lambda\boldsymbol{H}\boldsymbol{A}\boldsymbol{H}^\top\boldsymbol{H}. \tag{27}$$

According to the KKT conditions and similar to the update rule for $\boldsymbol{W}$, we obtain the update rule for $\boldsymbol{H}$ as follows:

$$\boldsymbol{H} \leftarrow \boldsymbol{H} \odot \sqrt{\frac{2\boldsymbol{W}^\top\boldsymbol{X} + \lambda\boldsymbol{H}\boldsymbol{A}}{(2\boldsymbol{W}^\top\boldsymbol{X} + \lambda\boldsymbol{H}\boldsymbol{A})\boldsymbol{H}^\top\boldsymbol{H}}} \tag{28}$$

We have now completed the derivation of the updating rules necessary for the optimization process of OEDFS. Algorithm 1 provides an overview of the entire optimization process.

3.7. Convergence analysis

In this section, we explore the convergence properties of the proposed algorithm. To establish its convergence, we rely on several lemmas.

Lemma 1. *For arbitrary matrices* $\boldsymbol{A} \in \mathbb{R}_{+}^{n\times n}$, $\boldsymbol{B} \in \mathbb{R}_{+}^{k\times k}$, $\boldsymbol{S} \in \mathbb{R}_{+}^{n\times k}$, $\boldsymbol{S}' \in \mathbb{R}_{+}^{n\times k}$, *if both* $\boldsymbol{A}$ *and* $\boldsymbol{B}$ *are symmetric matrices, then the following inequality holds:*

$$\sum_{i=1}^{n}\sum_{p=1}^{k}(\boldsymbol{A}\boldsymbol{S}'\boldsymbol{B})_{ip}\frac{S_{ip}^2}{S'_{ip}} \geq \mathrm{Tr}[\boldsymbol{S}^\top \boldsymbol{A}\boldsymbol{S}\boldsymbol{B}].$$

We employ an auxiliary function to demonstrate the convergence of our algorithm. The definition of this auxiliary function is presented below.

Definition 1. $\mathcal{G}(\boldsymbol{H}, \boldsymbol{H}')$ is an auxiliary function of $\mathcal{F}(\boldsymbol{H})$, if it satisfies the following conditions:

$$\mathcal{G}(\boldsymbol{H}, \boldsymbol{H}') \geq \mathcal{F}(\boldsymbol{H}), \quad \mathcal{G}(\boldsymbol{H}, \boldsymbol{H}) = \mathcal{F}(\boldsymbol{H}).$$

Lemma 2. *If* $\mathcal{G}$ *is an auxiliary function of* $\mathcal{F}$, *then* $\mathcal{F}(H)$ *decreases under the following updating rule:*

$$\boldsymbol{H}^{t+1} = \arg\min_{\boldsymbol{H}} \mathcal{G}(\boldsymbol{H}, \boldsymbol{H}^t).$$

In the following, we introduce a theorem that characterizes the convergence behavior of the proposed algorithm.

Theorem 1. *The objective function* (13) *decreases under the update rules described in update rules* (21) *and* (28).

Proof 1. To demonstrate this, we adopt the alternating minimization technique to tackle the objective function (13). We first fix $\boldsymbol{W}$ and show that (13) decreases under the update rule of (21).

In (23), ignoring those constants independent of the optimization variable $\boldsymbol{H}$, we have

$$\hat{\mathcal{L}}(\boldsymbol{H}) = \mathrm{Tr}[-4\boldsymbol{H}^\top \boldsymbol{W}^\top \boldsymbol{X} + \boldsymbol{H}^\top \boldsymbol{W}^\top \boldsymbol{W}\boldsymbol{H} + \boldsymbol{H}^\top \boldsymbol{H} + \lambda \boldsymbol{H}\boldsymbol{L}\boldsymbol{H}^\top - \boldsymbol{\Xi}\boldsymbol{H}\boldsymbol{H}^\top] \tag{29}$$

Subsequently, we create an auxiliary function for $\hat{\mathcal{L}}(\boldsymbol{H})$ in the following manner:

$$\mathcal{G}(\boldsymbol{H}, \boldsymbol{H}') = -4\sum_{jk}(H^\top W^\top X)_{jk} + \sum_{jk}(H'^\top W^\top W)_{jk}\frac{(H^2)_{jk}}{(H')_{jk}}\sum_{jk}(H')_{jk}\frac{(H^2)_{jk}}{(H')_{jk}} + \sum_{jk}\lambda(H'L)_{jk}\frac{(H^\top)_{jk}^2}{(H')_{jk}} - \sum_{jk}(\Xi)_{jk}(H^2)_{jk} \tag{30}$$

Now we verify that (30) satisfies the expected condition of an auxiliary function as stated in Definition 1 of (29).

(1) It is evident that $\mathcal{G}(\boldsymbol{H}, \boldsymbol{H})$ is equal to $\hat{\mathcal{L}}(\boldsymbol{H})$.

(2) In accordance with Lemma 1, we observe that the second, third and fourth terms of $\mathcal{G}(\boldsymbol{H}, \boldsymbol{H}')$ are larger than the corresponding terms of $\hat{\mathcal{L}}(\boldsymbol{H})$, while the first and fifth terms of both $\mathcal{G}(\boldsymbol{H}, \boldsymbol{H}')$ and $\hat{\mathcal{L}}(\boldsymbol{H})$ are equal. Therefore, $\mathcal{G}(\boldsymbol{H}, \boldsymbol{H}') \geq \hat{\mathcal{L}}(\boldsymbol{H})$ is true.

Based on Lemma 2, in order to reduce the objective function (13) using the update rule (28), our focus should solely be on optimizing the following auxiliary function:

$$\boldsymbol{H}^{(t+1)} = \arg\min_{\boldsymbol{H}} \mathcal{G}(\boldsymbol{H}, \boldsymbol{H}'^{(t)}) \tag{31}$$

In equation (31), we fix $\boldsymbol{H}'^{(t)}$, and the global optimal solution of $\boldsymbol{H}$ is given by:

$$\frac{\partial \mathcal{G}(\boldsymbol{H}, \boldsymbol{H}'^{(t)})}{\partial \boldsymbol{H}} = -2\boldsymbol{W}^\top \boldsymbol{X} + \boldsymbol{W}^\top \boldsymbol{W}\boldsymbol{H} + \boldsymbol{H} + \lambda \boldsymbol{H}\boldsymbol{L} - \boldsymbol{\Xi}\boldsymbol{H} = 0 \tag{32}$$

Utilizing the KKT complementary relaxation condition, Eq. (32), and Lemma 2, we obtain:

$$\boldsymbol{H} \leftarrow \boldsymbol{H} \odot \sqrt{\frac{2\boldsymbol{W}^\top \boldsymbol{X} + \lambda \boldsymbol{H}\boldsymbol{A}}{(2\boldsymbol{W}^\top \boldsymbol{X} + \lambda \boldsymbol{H}\boldsymbol{A})\boldsymbol{H}^\top \boldsymbol{H}}} \tag{33}$$

Therefore, the objective function $\hat{\mathcal{L}}(\boldsymbol{H})$ decreases under the update rule for $\boldsymbol{H}$.

The proof of the convergence of the objective function under the updating rule of $\boldsymbol{W}$ is similar to that of $\boldsymbol{H}$. When $\boldsymbol{H}$ is fixed, the terms containing $\boldsymbol{W}$ in the objective function (13) are

$$\hat{\mathcal{L}}(\boldsymbol{W}) = \mathrm{Tr}[-4\boldsymbol{X}^\top \boldsymbol{W}\boldsymbol{H} + \boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^\top \boldsymbol{W}^\top + \boldsymbol{W}^\top \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{W}] + \gamma \mathrm{Tr}(\boldsymbol{W}^\top \boldsymbol{D}\boldsymbol{W}). \tag{34}$$

We can construct the following auxiliary function for $\hat{\mathcal{L}}(\boldsymbol{W})$:

$$\mathcal{G}(\boldsymbol{W}, \boldsymbol{W}') = -4\sum_{jk}(H^\top W^\top X)_{jk} + \sum_{jk}(W'HH^\top)_{jk}\frac{(W^\top)_{jk}^2}{(W')_{jk}} + \sum_{jk}(W'^\top XX^\top)_{jk}\frac{(W^2)_{jk}}{(W')_{jk}} + \sum_{jk}\lambda(DW')_{jk}\frac{(W^\top)_{jk}^2}{(W')_{jk}} \tag{35}$$

Table 1
The detailed information of the real-world datasets.

Dataset	#Feature	#Instance	#Class	Application
TUANDROMD	241	4464	2	Malware detection
lung-discrete	325	73	7	Medical science
Isolet	617	1560	26	Speech recognition
jaffe	676	213	10	Face recognition
Flowers17-ResNet18	1000	1360	17	DL-based plant classification
COIL20	1024	1440	20	Object recognition
ORL	1024	400	40	Face recognition
colon	2000	62	2	Gene expression
lung	3312	203	5	Medical science
RELATHE	4322	1427	2	Text recognition
GLIOMA	4434	50	4	Gene expression
TOX171	5748	171	4	Gene expression
ALLAML	7129	72	2	Gene expression
CLLSUB111	11340	111	3	Gene expression

It can be verified that $\mathcal{G}(\boldsymbol{W},\boldsymbol{W}) = \hat{\mathcal{L}}(\boldsymbol{W})$ and $\mathcal{G}(\boldsymbol{W},\boldsymbol{W}') \geq \hat{\mathcal{L}}(\boldsymbol{W})$ according to Lemma 1. To minimize $\mathcal{G}(\boldsymbol{W},\boldsymbol{W}')$, we take the derivative w.r.t. $\boldsymbol{W}$ and set it to zero:

$$\frac{\partial \mathcal{G}(\boldsymbol{W},\boldsymbol{W}'^{(t)})}{\partial \boldsymbol{W}} = -4\boldsymbol{X}\boldsymbol{H}^{\top} + 2\boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^{\top} + 2\boldsymbol{X}\boldsymbol{X}^{\top}\boldsymbol{W} + \gamma\boldsymbol{D}\boldsymbol{W} = 0 \tag{36}$$

By the KKT complementary relaxation condition, the update rule for $\boldsymbol{W}$ is:

$$\boldsymbol{W} \leftarrow \boldsymbol{W} \odot \frac{2\boldsymbol{X}\boldsymbol{H}^{\top}}{\boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^{\top} + \boldsymbol{X}\boldsymbol{X}^{\top}\boldsymbol{W} + \gamma\boldsymbol{D}\boldsymbol{W}}. \tag{37}$$

Therefore, the objective function $\hat{\mathcal{L}}(\boldsymbol{W})$ decreases under the update rule for $\boldsymbol{W}$. In summary, the objective function of OEDFS converges under the updating rules (33) and (37).

3.8. Computational complexity analysis

In the Orthogonal Encoder-Decoder Unsupervised Feature Selection (OEDFS) Algorithm 1, the computational complexity is divided into two stages. In the pre-optimizing stage, constructing the similarity matrix $\boldsymbol{A}$ takes $O(dn\log n)$. In the iterative stage, involving t iterations, updating the feature weight matrix $\boldsymbol{D}$ takes $O(drt)$ time. Additionally, updating $\boldsymbol{W}$, which includes a constant calculation of $\boldsymbol{X}\boldsymbol{X}^{\top}$, takes $O(d^2rt + dnrt + d^2n)$, while updating $\boldsymbol{H}$ takes $O(dnrt + n^2rt)$. Considering the number of features d, the overall computational complexity of Algorithm 1 depends on the interplay between d and the number of samples n. In scenarios where d is significantly smaller or comparable to n, the dominant term is n^2rt, resulting in an overall complexity of $O(n^2rt)$. Conversely, for larger d compared to n, the dominant term becomes d^2n, leading to an overall complexity of $O(d^2n)$.

4. Experiments

In this section, the performance of the OEDFS method is evaluated on 14 benchmark datasets.[1] The method is compared with several state-of-the-art methods in terms of clustering tasks on these datasets. The extensive experiments will be elaborated in the following subsections. Experimental results confirm that the proposed algorithm is the most effective ways for selecting subset features in most cases.

4.1. Datasets

The comparative evaluation of OEDFS and other works is conducted on 14 diverse benchmark and high-dimensional datasets. These datasets span various applications, including malware detection, medical science, speech recognition, face recognition, deep learning-based plant classification, object recognition, gene expression analysis, and text recognition. Each dataset is carefully selected to represent a specific application domain, ensuring a comprehensive evaluation of our proposed approach across various challenges and scenarios. All of these datasets are available on the repository [4]. Details of these datasets are summarized in Table 1, with the number of samples ranging from 50 to 4464 and the number of features varying from 241 to 11340.

4.2. Compared methods

To compare the experimental results, we selected some representative existing algorithms, including:

[1] The feature selection repository in Python (scikit-feature).

- **Baseline**: The baseline performs clustering using all original features.
- **LapS** [8]: This method uses the locality-preserving capability of each feature to evaluate the importance of the feature.
- **MCFS** [12]: MCFS aims to solve an L_1-regularized regression model with spectral analysis in order to select more effective subset features.
- **UDFS** [29]: UDFS proposed a joint framework that applied discriminative analysis and solved the $L_{2,1}$ norm regularized minimization problem to exploit local discriminative information of data, consequently select the most discriminative features.
- **NDFS** [11]: NDFS selects the features by devising a model which is a combination of nonnegative spectral analysis and $L_{2,1}$ norm regularization.
- **SCFS** [15]: SCFS uses subspace clustering to preserve the sample similarities by representation learning of low dimensional subspaces, and also considers the latent structure of data based on a regularized regression model.
- **DSLRL** [26]: proposed a model of learning from latent representation based on features correlation and samples correlation (i.e., using the intrinsic structure of data space and feature space).
- **RUSLP** [27]: proposed the robust feature selection model which imposed a low rank and orthogonality constraint to improve the performance of their matrix factorization based model.

4.3. Evaluation metrics

To evaluate the performance of the OEDFS two commonly used evaluation metrics, i.e., accuracy (ACC) and normalized mutual information (NMI) are employed. ACC is defined as follows:

$$\mathrm{ACC}(\boldsymbol{y},\boldsymbol{c}) = \frac{1}{n}\sum_{i=1}^{n}\delta(y_i, \mathrm{map}(c_i)), \tag{38}$$

where y_i is the true label of x_i, and c_i is the clustering result of x_i, n is the number of samples, and $\delta(x,y)=1$ if $x=y$, otherwise $\delta(x,y)=0$. Given two variables y and c, NMI is defined as follows:

$$\mathrm{NMI}(\boldsymbol{y},\boldsymbol{c}) = \frac{\mathcal{I}(\boldsymbol{y},\boldsymbol{c})}{\max(\mathcal{H}(\boldsymbol{y}),\mathcal{H}(\boldsymbol{c}))} \tag{39}$$

where $\boldsymbol{y}$ and $\boldsymbol{c}$ are the true labels and clustering results, respectively, $\max(a,b)$ returns the largest of the input values, and $\mathcal{H}(\boldsymbol{y})$ and $\mathcal{H}(\boldsymbol{c})$ are the entropy of $\boldsymbol{y}$ and $\boldsymbol{c}$ random variables.

4.4. Experimental settings

In our experiments, we employ the grid search method within the range of $\{0, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}\}$ to determine the values of parameters λ and γ, achieving the best results. The projection dimension or latent factor r in the proposed methods is set to the optimal value chosen from the range $\{20, 30, 40, \ldots, 100\}$. Additionally, the optimization algorithm runs for 500 iterations, and the local structure parameters are set to $k=5$ and $\sigma=1000$. It is worth noting that the parameters for other feature selection comparative algorithms are set to their values for optimal performance. For most datasets, the tested feature dimensions are {50, 100, 150, 200, 250, 300}, but for the TUANDROMD dataset, the selected feature dimensions are {20, 40, 60, 80, 100, 120}. We apply the K-means clustering algorithm, record the best performance for the given numbers of features, and note that the number of clusters precisely corresponds to the number of true classes for all datasets. Due to the sensitivity of different initializations, the experiment is repeated 10 times with random initial values. Finally, we report the average value and standard deviation of the two applied performance measures separately for all the algorithms.

4.5. Experimental results

In this section, we present the results obtained from the proposed model on 14 datasets. A comprehensive and comparative experimental study is provided in terms of the NMI and ACC values for eight different methods, as shown in Tables 2 and 3, respectively. Values in bold represent the best results, while underlined ones represent the second-best performance. Table 2 indicates that our OEDFS outperforms 11 out of 14 cases in terms of NMI, and in three other datasets, our model is ranked as the second-best. In Table 3, our method performs best in terms of ACC, except on four datasets where it is the second-best. Despite being the best-performing method in three datasets, the DSLRL method exhibits certain limitations. It relies on calculated feature correlation, suitable for certain non-image datasets, but faces limitations with image datasets due to their high dimensionality and complex spatial dependencies. Additionally, DSLRL's complexity, with four hyperparameters, may contribute to variations in performance across datasets. In summary, concerning the top recent methods (i.e., SCFS, RUSLP, and DSLRL), there was an approximate 7%, 7.7%, and 10.6% growth in average terms of NMI, and 5.2%, 7.1%, and 7.2% in terms of ACC, respectively. This suggests that our method produced a noticeable increase in results across a wide range of image and numerical datasets.

Furthermore, we present the feature selection results for different feature subsets. Figs. 2 and 3 illustrate the curves of ACC and NMI values for all the methods. The vertical axis represents NMI and ACC values, while the horizontal axis denotes the number of selected features. The proposed OEDFS method consistently outperforms other UFS methods in NMI and ACC across most cases, even as the number of features varies, as evident in six datasets. The curves in Figs. 2 and 3 demonstrate that the proposed model is the best in the majority of datasets and second-best in a minority. Specifically, on Jaffe and COIL20 datasets, OEDFS produces the best

Table 2
The experimental results of NMI (%) ± std measure on real-world datasets.

Dataset		Methods								
		Baseline	LapS	MCFS	UDFS	NDFS	SCFS	RUSLP	DSLRL	OEDFS
TUANDROMD	mean	4.47	2.53	4.23	1.17	6.33	13.99	19.36	16.05	**29.27**
	std	±0.9	±2.65	±0.95	±2.62	±0.58	±6.22	±8.91	±7.82	±6.7
Lung-discrete	mean	70.42	71.56	71.17	74.36	71.06	75.51	68.15	**76.86**	76.81
	std	±5.9	±3.74	±4.24	±4.39	±4.57	±4.4	±5.22	±3.55	±5.81
Isolet	mean	77.62	74.96	77.02	73.37	77.04	79.95	74.67	**81.98**	79.95
	std	±1.06	±1.37	±1.2	±1.22	±1.02	±0.61	±1.73	±1.34	±0.97
jaffe	mean	88.52	79.72	82.71	77.49	91.88	89.83	83.89	79.63	**95.26**
	std	±3.44	±1.55	±2.89	±1.4	±2.22	±3.37	±4.99	±2.16	±1.36
Flowers17 ResNet18	mean	60.98	60.81	60.21	59.53	57.92	63.5	59.13	63.73	**79.28**
	std	±1.55	±1.52	±1.6	±1.3	±1.28	±1.11	±2.13	±1.24	±1.43
COIL20	mean	78.8	71.16	75.34	71.66	77.99	78.19	72.12	74.79	**79.43**
	std	±1.13	±1.02	±1.25	±0.81	±1.24	±1.34	±0.02	±0.9	±1.4
ORL	mean	77.77	75.92	75.48	72.43	77.21	77.33	74.33	73.04	**78.51**
	std	±0.8	±0.84	±0.8	±1.49	±1.05	±0.92	±1.83	±1.37	±1.04
colon	mean	0.58	0.89	0.62	0.95	0.35	1.19	5.04	1.73	**11.68**
	std	±0.13	±0.24	±0.0	±0.34	±0.11	±0.58	±2.46	±1.18	±1.77
lung	mean	63.24	60.42	63.64	58.76	58.49	67.87	57.26	58.67	**70.76**
	std	±2.54	±1.99	±2.26	±1.12	±1.79	±1.34	±7.94	±1.76	±2.49
RELATHE	mean	0.13	0.16	0.1	0.16	0.34	0.7	2.92	0.39	**9.40**
	std	±0.05	±0.91	±0.02	±0.00	±0.42	±0.04	±0.99	±0.18	±0.42
GLIOMA	mean	50.61	50.12	50.1	51.22	50.46	53.54	51.57	**59.8**	55.13
	std	±1.72	±1.91	±2.8	±3.73	±1.65	±1.4	±1.57	±3.74	±2.42
TOX171	mean	12.97	13.46	17.1	10.64	24.75	19.6	28.11	12.87	**35.94**
	std	±2.21	±1.97	±6.59	±5.52	±2.52	±7.44	±3.56	±1.56	±2.19
ALLAML	mean	12.11	16.1	15.23	8.63	14.25	15.47	18.34	13.94	**20.01**
	std	±1.49	±2.1	±4.56	±4.38	±2.09	±0.0	±1.48	±3.41	±1.31
CLLSUB111	mean	18.73	18.73	21.27	12.64	23.6	24.26	35.50	2.66	**37.7**
	std	±0.0	±0.0	±0.0	±0.03	±0.0	±1.98	±0.41	±0.25	±1.28

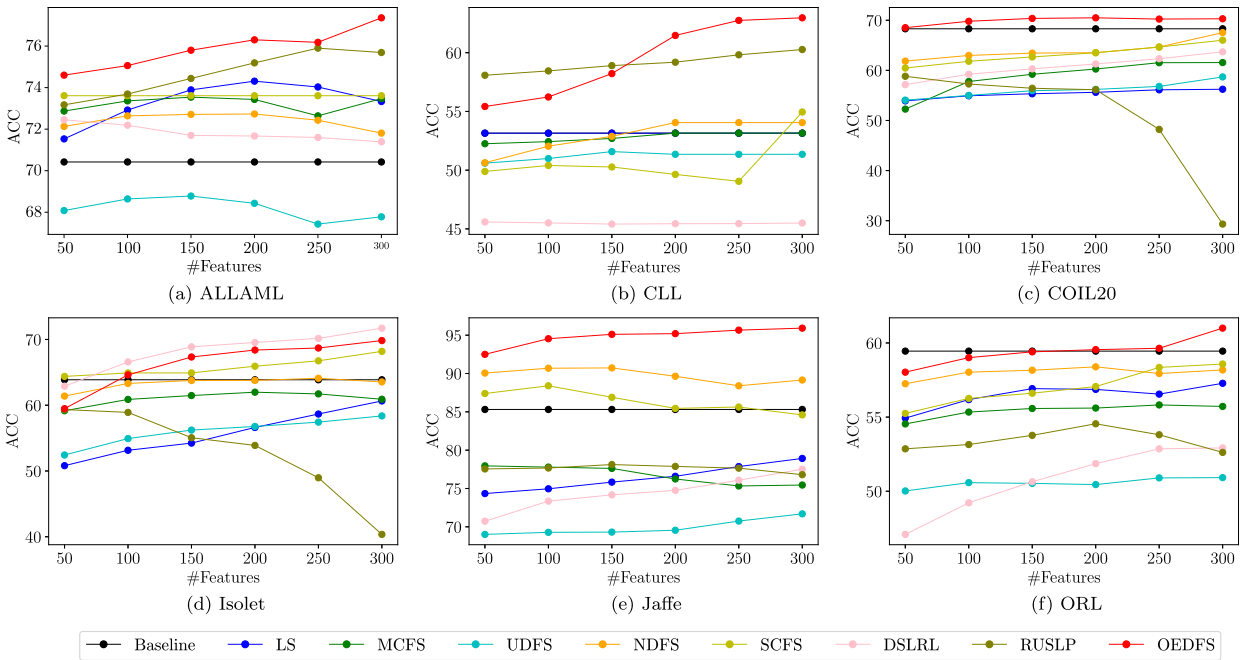

Fig. 2. ACC of all the methods on different datasets.

results across the entire subset of selected features. Overall, these pieces of evidence underscore the superiority of OEDFS compared to others. Furthermore, with an increase in the number of selected features, the model's performance initially tends to improve and then stabilizes. This indicates the stability and capability of the proposed method.

Table 3
The experimental results of ACC (%) ± std measure on real-world datasets.

Dataset		Methods								
		Baseline	LapS	MCFS	UDFS	NDFS	SCFS	RUSLP	DSLRL	OEDFS
TUANDROMD	mean	66.49	65.62	72.86	77.56	74.32	66.43	79.51	76.47	**84.4**
	std	±1.05	±1.94	±1.08	±1.73	±0.29	±2.13	±5.53	±6.97	±5.87
Lung-discrete	mean	74.52	77.81	76.58	80.41	76.1	80.82	72.39	**84.66**	82.05
	std	±9.95	±5.04	±5.94	±6.59	±6.65	±3.6	±5.73	±2.65	±7.6
Isolet	mean	63.87	60.65	61.98	58.37	64.1	68.19	59.34	**71.72**	69.83
	std	±1.99	±2.99	±2.33	±2.66	±2.83	±3.16	±3.62	±4.15	±3.18
jaffe	mean	85.31	78.92	77.79	71.69	90.73	88.4	78.12	77.51	**95.92**
	std	±5.93	±3.99	±5.29	±4.15	±4.43	±5.62	±9.04	±3.15	±0.99
Flowers17 ResNet18	mean	60.08	59.23	59.12	58.74	57.82	62.51	57.84	62.52	**70.49**
	std	±2.9	±2.1	±2.79	±2.12	±1.97	±1.79	±2.8	±1.74	±2.75
COIL20	mean	68.3	56.23	61.56	58.69	67.54	66.02	58.82	63.7	**70.24**
	std	±2.52	±2.27	±3.43	±1.83	±2.85	±2.44	±0.03	±1.59	±2.58
ORL	mean	59.45	57.28	55.72	50.92	58.39	58.58	54.55	52.92	**61.0**
	std	±1.85	±1.79	±2.01	±2.7	±2.08	±1.79	±3.17	±2.89	±3.05
colon	mean	54.84	56.26	54.84	56.35	54.6	55.91	**64.41**	55.55	61.45
	std	±0.0	±0.77	±0.0	±1.19	±0.79	±0.77	±3.31	±3.45	±1.19
lung	mean	68.42	71.43	70.88	68.37	62.3	85.52	76.42	70.99	**87.29**
	std	±7.58	±3.06	±6.99	±2.32	±2.64	±0.48	±12.11	±3.37	±2.73
RELATHE	mean	54.63	54.66	54.78	54.66	54.0	54.71	54.14	54.34	**55.14**
	std	±0.07	±0.0	±0.05	±0.0	±1.49	±0.42	±2.05	±0.06	±0.88
GLIOMA	mean	60.4	56.33	60.73	61.6	60.7	67.2	63.40	**75.8**	71.0
	std	±1.26	±2.29	±3.97	±4.4	±2.27	±5.27	±2.26	±7.31	±4.45
TOX171	mean	40.53	40.64	42.49	38.03	47.54	44.74	48.88	44.33	**51.58**
	std	±1.66	±2.91	±4.03	±4.58	±1.22	±4.8	±2.84	±2.25	±0.91
ALLAML	mean	70.42	74.31	73.54	68.78	72.71	73.61	75.90	72.45	**77.15**
	std	±1.86	±1.85	±4.05	±3.0	±1.82	±0.0	±1.13	±2.45	±0.84
CLLSUB111	mean	53.15	53.15	53.15	51.58	54.05	54.95	59.82	45.5	**62.97**
	std	±0.0	±0.0	±0.0	±0.4	±0.0	±1.27	±1.44	±0.64	±4.13

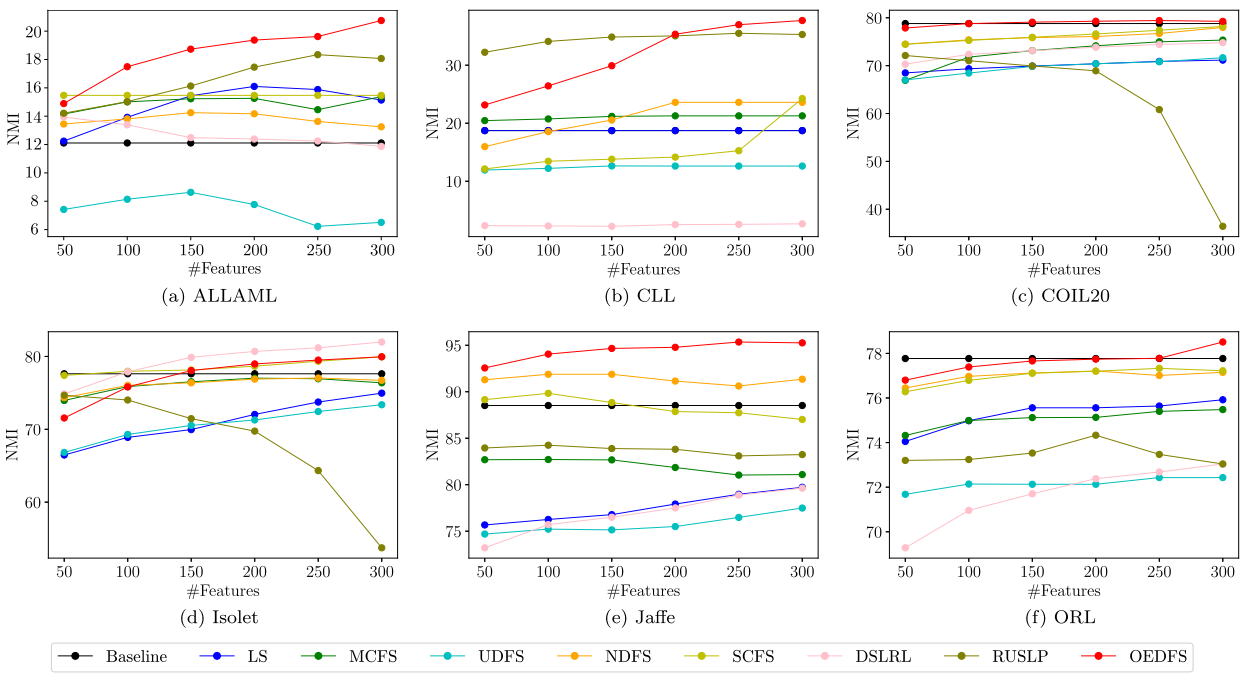

Fig. 3. NMI of all the methods on different datasets.

4.6. Parameter analysis

In this section, we investigate the impact of parameters on the OEDFS method. Specifically, our proposed objective function (13) has two tuning parameters, namely λ and γ, which are designed to preserve the local geometric structure of data and control the

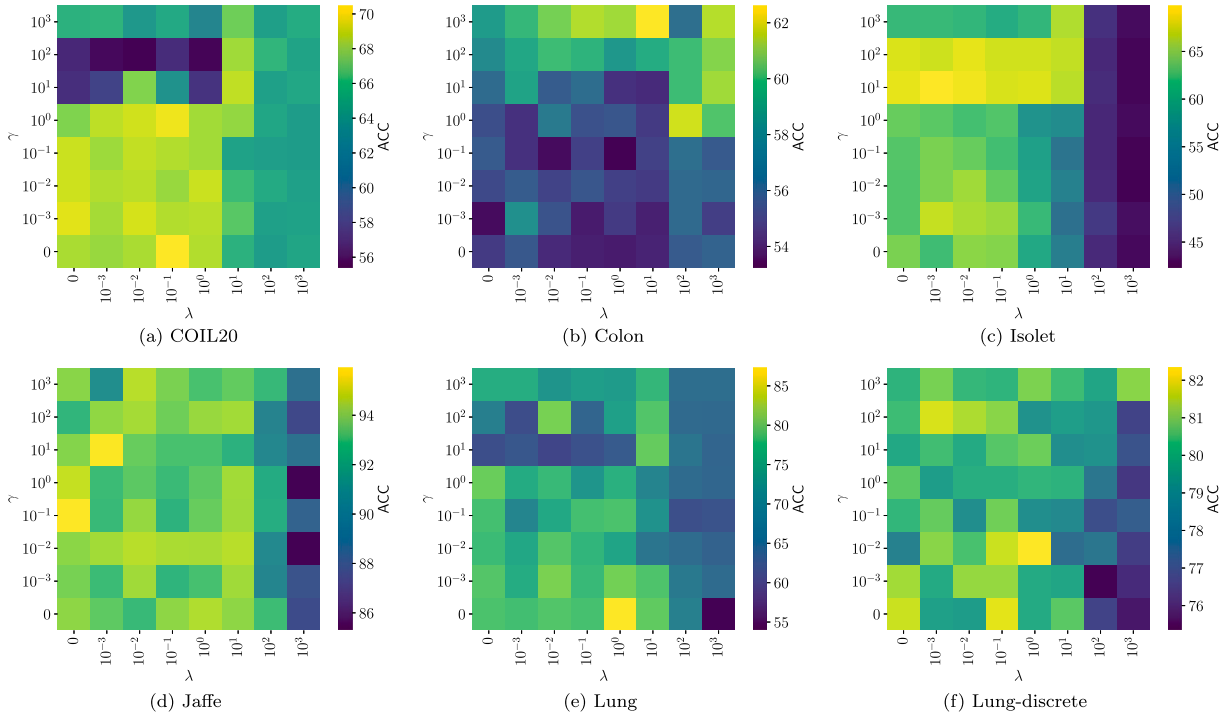

Fig. 4. The variation of ACC with respect to λ and γ on six datasets.

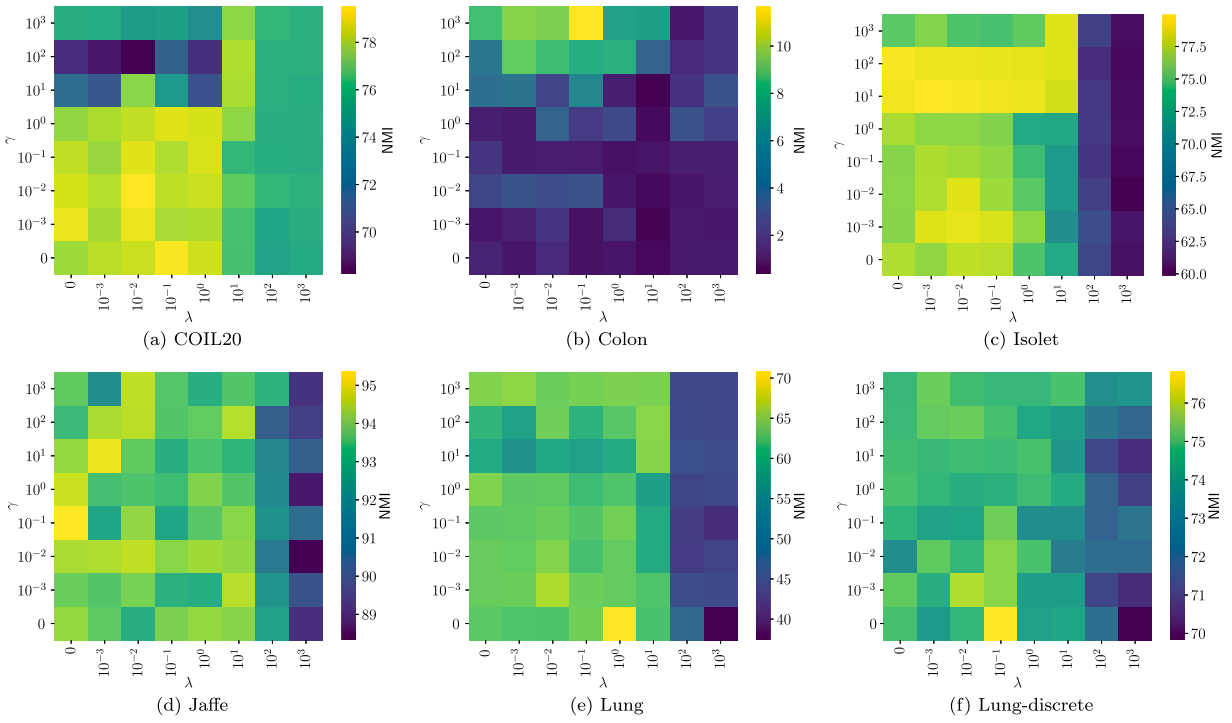

Fig. 5. The variation of NMI with respect to λ and γ on six datasets.

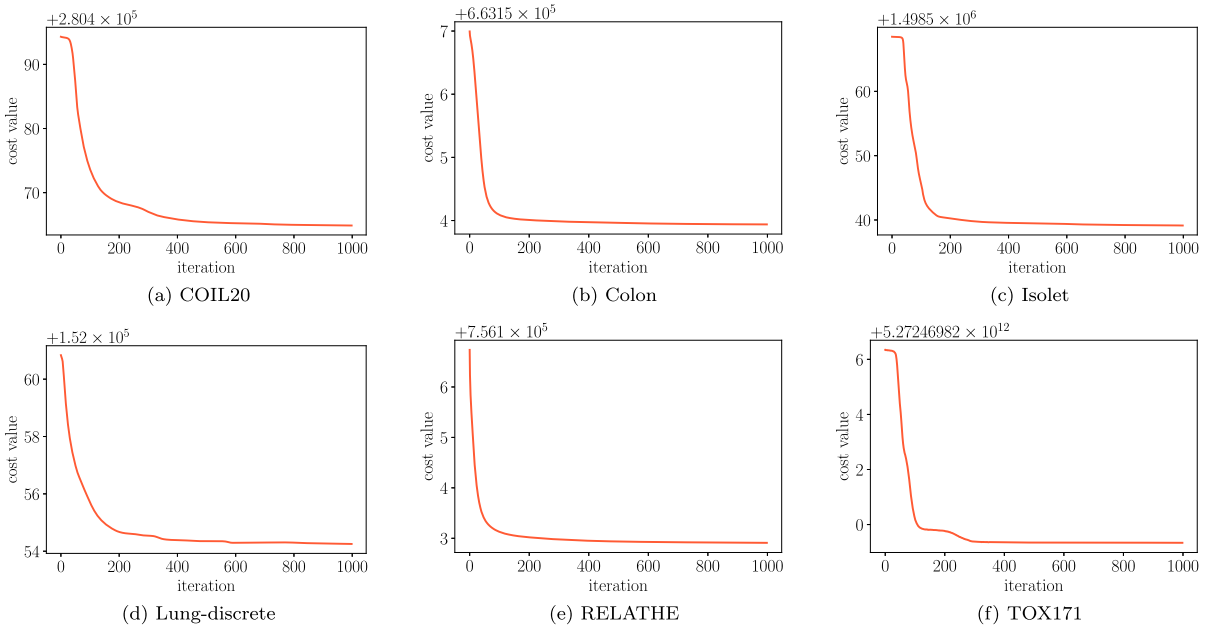

Fig. 6. The convergence curves of proposed model on the different datasets.

row sparsity of the feature weight matrix. We evaluate the method with eight different values for these parameters, selected from $\{0, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}\}$. The parameter sensitivity is assessed with respect to λ and γ for the Lung-discrete, Isolet, Jaffe, COIL20, Colon, and Lung datasets, in terms of NMI and ACC, as illustrated in Figs. 4 and 5, respectively. To demonstrate the impact of parameters λ and γ on the model's performance, we present the results obtained for each parameter pair. In these figures, the horizontal and vertical axes indicate the variation range of λ and γ, while the color represents the performance metric (NMI or ACC).

From Figs. 4 and 5, we can deduce that tuning the parameter λ is relatively insensitive, and setting it to very large values results in low performance. In most cases, it is preferable for the λ parameter to be set to values less than 10^2, typically around an intermediate value. To achieve the best ACC or NMI results, the γ parameter should be tuned for each dataset based on its specific characteristics. This is grounded in the assumption that different datasets require different levels of feature weight sparsity. To elaborate further, in Figs. 4 and 5, the best NMI and ACC values are achieved on COIL20, Lung, and Lung-discrete datasets when the γ parameter is set to a small or relatively small value. Conversely, setting large or relatively large values results in higher performances on the Jaffe, Colon, and Isolet datasets. It is evident that choosing parameter values from a reasonable range results the acceptable performance which indicate the OEDFS proposed model has the stability and is not too sensitive to regularization parameters. To sum up, it can be inferred that both the λ and γ parameters significantly contribute to the effectiveness of the proposed model.

4.7. Convergence analysis

In this section, we conduct an experimental study to analyze the convergence of Algorithm 1, which iteratively solves our objective function (13). The convergence curves of the objective value on the Colon, Isolet, TOX171, RELATHE, COIL20, and Lung-discrete datasets are provided in Fig. 6. As shown, there is no need for extensive iterations until the model stabilizes. The process reaches convergence during the initial iterations, eliminating the necessity for additional iterations. It is evident that Algorithm 1 converges quickly, typically within one-fifth of the total iterations on all datasets. Overall, this behavior is deemed acceptable, ensuring the efficiency of the entire OEDFS algorithm.

5. Conclusion

This paper proposed an innovative approach to unsupervised feature selection using an Orthogonal Encoder-Decoder Nonnegative Matrix Factorization framework. The method integrates self-representation and pseudo-supervised approaches to achieve effective feature selection without the need for labeled data. The proposed approach incorporates an autoencoder NMF model to perform feature-side self-representation. By leveraging the encoder factorization, pseudo-labels are generated through clustering, while the decoder factorization aims to learn feature weights resembling a supervised regression model. This process facilitates the identification of non-redundant and relevant features based on their contributions to the overall reconstruction of the input data. To enhance the clustering properties of NMF, an orthogonality constraint is imposed on the proposed objective function. Additionally, a graph regularization term is included in the model to promote the preservation of the intrinsic structure of the data. The optimization of the proposed method is achieved through the utilization of the Multiplicative Update Rule. Extensive experimental evaluations were

conducted, comparing the proposed method with eight existing feature selection methods on 14 diverse datasets. The results clearly demonstrate the efficiency and effectiveness of the proposed OEDFS method for the unsupervised feature selection task. It outperformed the competing methods in terms of feature selection accuracy and exhibited superior performance in identifying relevant features for different datasets.

In terms of future research, there are several promising directions to explore. One such direction involves the integration of deep autoencoder models to extract informative insights from complex data. This expansion could enable the method to capture intricate patterns and structures in high-dimensional data. Another potential avenue is extending the model to incorporate semi-supervised feature selection by leveraging partial information or limited labels. This integration would combine unsupervised learning with supervised information, enhancing feature selection performance. Additionally, the utilization of a feature-wise robust loss function could improve the model's resistance to noisy data.

CRediT authorship contribution statement

Maryam Mozafari: Writing – original draft, Visualization, Software, Methodology. **Seyed Amjad Seyedi:** Writing – review & editing, Writing – original draft, Methodology, Conceptualization. **Rojiar Pir Mohammadiani:** Writing – review & editing, Validation, Conceptualization. **Fardin Akhlaghian Tab:** Writing – review & editing, Validation, Supervision, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, Feature Selection for High-Dimensional Data, Springer, 2015.
[2] X. Li, Y. Wang, R. Ruiz, A survey on sparse learning models for feature selection, IEEE Trans. Cybern. 52 (3) (2022) 1642–1660.
[3] J. Cai, J. Luo, S. Wang, S. Yang, Feature selection in machine learning: a new perspective, Neurocomputing 300 (2018) 70–79.
[4] J. Li, K. Cheng, S. Wang, F. Morstatter, R.P. Trevino, J. Tang, H. Liu, Feature selection: a data perspective, ACM Comput. Surv. 50 (6) (2017) 1–45.
[5] L. Wang, Y. Wang, Q. Chang, Feature selection methods for big data bioinformatics: a survey from the search perspective, Methods 111 (2016) 21–31.
[6] G. Chandrashekar, F. Sahin, A survey on feature selection methods, in: 40th-Year Commemorative Issue, Comput. Electr. Eng. 40 (1) (2014) 16–28.
[7] S. Solorio-Fernández, J.A. Carrasco-Ochoa, J.F. Martínez-Trinidad, A review of unsupervised feature selection methods, Artif. Intell. Rev. 53 (2) (2020) 907–948.
[8] X. He, D. Cai, P. Niyogi, Laplacian Score for Feature Selection, Advances in Neural Information Processing Systems, vol. 18, MIT Press, 2005.
[9] A. Yuan, M. You, D. He, X. Li, Convex non-negative matrix factorization with adaptive graph for unsupervised feature selection, IEEE Trans. Cybern. 52 (6) (2022) 5522–5534.
[10] P. Zhu, W. Zuo, L. Zhang, Q. Hu, S.C. Shiu, Unsupervised feature selection by regularized self-representation, Pattern Recognit. 48 (2) (2015) 438–446.
[11] Z. Li, Y. Yang, J. Liu, X. Zhou, H. Lu, Unsupervised feature selection using nonnegative spectral analysis, in: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, 2012, pp. 1026–1032.
[12] D. Cai, C. Zhang, X. He, Unsupervised feature selection for multi-cluster data, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2010, pp. 333–342.
[13] M. Fan, X. Chang, X. Zhang, D. Wang, L. Du, Top-k supervise feature selection via admm for integer programming, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, 2017, pp. 1646–1653.
[14] C. Hou, F. Nie, X. Li, D. Yi, Y. Wu, Joint embedding learning and sparse regression: a framework for unsupervised feature selection, IEEE Trans. Cybern. 44 (6) (2014) 793–804.
[15] M.G. Parsa, H. Zare, M. Ghatee, Unsupervised feature selection based on adaptive similarity learning and subspace clustering, Eng. Appl. Artif. Intell. 95 (2020) 103855.
[16] Z. Li, J. Tang, Unsupervised feature selection via nonnegative spectral analysis and redundancy control, IEEE Trans. Image Process. 24 (12) (2015) 5343–5355.
[17] M. Qian, C. Zhai, Robust unsupervised feature selection, in: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, 2013, pp. 1621–1627.
[18] C. Hou, F. Nie, D. Yi, Y. Wu, Feature selection via joint embedding learning and sparse regression, in: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Two, 2011, pp. 1324–1329.
[19] L. Du, Y.-D. Shen, Unsupervised feature selection with adaptive structure learning, in: Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 209–218.
[20] Y. Wang, H. Yao, S. Zhao, Auto-encoder based dimensionality reduction, Neurocomputing 184 (2016) 232–242.
[21] P. Baldi, Autoencoders, unsupervised learning, and deep architectures, in: Proceedings of ICML Workshop on Unsupervised and Transfer Learning, vol. 27, 2012, pp. 37–49.
[22] S. Wang, W. Pedrycz, Q. Zhu, W. Zhu, Subspace learning for unsupervised feature selection via matrix factorization, Pattern Recognit. 48 (1) (2015) 10–19.
[23] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (6755) (1999) 788–791.
[24] C. Ding, T. Li, W. Peng, H. Park, Orthogonal nonnegative matrix t-factorizations for clustering, in: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006, pp. 126–135.
[25] B.-J. Sun, H. Shen, J. Gao, W. Ouyang, X. Cheng, A non-negative symmetric encoder-decoder approach for community detection, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017, pp. 597–606.
[26] R. Shang, L. Wang, F. Shang, L. Jiao, Y. Li, Dual space latent representation learning for unsupervised feature selection, Pattern Recognit. 114 (2021) 107873.
[27] C. Luo, J. Zheng, T. Li, H. Chen, Y. Huang, X. Peng, Orthogonally constrained matrix factorization for robust unsupervised feature selection with local preserving, Inf. Sci. 586 (2022) 662–675.

[28] D. Cai, X. He, J. Han, T.S. Huang, Graph regularized nonnegative matrix factorization for data representation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1548–1560.
[29] Y. Yang, H.T. Shen, Z. Ma, Z. Huang, X. Zhou, L2, 1-norm regularized discriminative feature selection for unsupervised learning, in: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, vol. Volume Two, 2011, pp. 1589–1594.
[30] Z. Zhao, L. Wang, H. Liu, Efficient spectral feature selection with minimum redundancy, in: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, 2010, pp. 673–678.
[31] L. Shi, L. Du, Y.-D. Shen, Robust spectral learning for unsupervised feature selection, in: 2014 IEEE International Conference on Data Mining, 2014, pp. 977–982.
[32] S. Wang, J. Tang, H. Liu, Embedded unsupervised feature selection, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI'15, AAAI Press, 2015, pp. 470–476.
[33] X. Zhu, X. Li, S. Zhang, C. Ju, X. Wu, Robust joint graph sparse coding for unsupervised spectral feature selection, IEEE Trans. Neural Netw. Learn. Syst. 28 (6) (2017) 1263–1275.
[34] H. Zare, M.G. Parsa, M. Ghatee, S.H. Alizadeh, Similarity preserving unsupervised feature selection based on sparse learning, in: 2020 10th International Symposium on Telecommunications (IST), 2020, pp. 50–55.
[35] H. Zhao, L. Du, J. Wei, Y. Fan, Local sensitive dual concept factorization for unsupervised feature selection, IEEE Access 8 (2020) 133128–133143.
[36] X. Lin, J. Guan, B. Chen, Y. Zeng, Unsupervised feature selection via orthogonal basis clustering and local structure preserving, IEEE Trans. Neural Netw. Learn. Syst. 33 (11) (2022) 6881–6892.
[37] R. Wang, J. Bian, F. Nie, X. Li, Unsupervised discriminative projection for feature selection, IEEE Trans. Knowl. Data Eng. 34 (2) (2022) 942–953.
[38] Q. Lu, X. Li, Y. Dong, Structure preserving unsupervised feature selection, Neurocomputing 301 (2018) 36–45.
[39] C. Tang, X. Zhu, J. Chen, P. Wang, X. Liu, J. Tian, Robust graph regularized unsupervised feature selection, Expert Syst. Appl. 96 (2018) 64–76.
[40] C. Lei, X. Zhu, Unsupervised feature selection via local structure learning and sparse learning, Multimed. Tools Appl. 77 (22) (2018) 29605–29622.
[41] C. Tang, X. Liu, M. Li, P. Wang, J. Chen, L. Wang, W. Li, Robust unsupervised feature selection via dual self-representation and manifold regularization, Knowl.-Based Syst. 145 (2018) 109–120.
[42] Z. Yuan, E. Oja, Projective nonnegative matrix factorization for image compression and feature extraction, in: Image Analysis, Springer, 2005, pp. 333–342.
[43] S.A. Seyedi, F. Akhlaghian Tab, A. Lotfi, N. Salahian, J. Chavoshinejad, Elastic adversarial deep nonnegative matrix factorization for matrix completion, Inf. Sci. 621 (2023) 562–579.
[44] S.A. Seyedi, S.S. Ghodsi, F. Akhlaghian, M. Jalili, P. Moradi, Self-paced multi-label learning with diversity, in: Proceedings of the Eleventh Asian Conference on Machine Learning, vol. 101, 2019, pp. 790–805.
[45] R. Abdollahi, S. Amjad Seyedi, M. Reza Noorimehr, Asymmetric semi-nonnegative matrix factorization for directed graph clustering, in: 2020 10th International Conference on Computer and Knowledge Engineering (ICCKE), 2020, pp. 323–328.
[46] S.A. Seyedi, P. Moradi, F.A. Tab, A weakly-supervised factorization method with dynamic graph embedding, in: 2017 Artificial Intelligence and Signal Processing Conference (AISP), 2017, pp. 213–218.
[47] Z. Shajarian, S.A. Seyedi, P. Moradi, A clustering-based matrix factorization method to improve the accuracy of recommendation systems, in: 2017 Iranian Conference on Electrical Engineering (ICEE), 2017, pp. 2241–2246.
[48] R. Mahmoodi, S.A. Seyedi, F. Akhlaghian Tab, A. Abdollahpouri, Link prediction by adversarial nonnegative matrix factorization, Knowl.-Based Syst. 280 (2023) 110998.
[49] A. Hajiveiseh, S.A. Seyedi, F. Akhlaghian Tab, Deep asymmetric nonnegative matrix factorization for graph clustering, Pattern Recognit. 148 (2024) 110179.
[50] M. Faraji, S.A. Seyedi, F. Akhlaghian Tab, R. Mahmoodi, Multi-label feature selection with global and local label correlation, Expert Syst. Appl. 246 (2024) 123198.