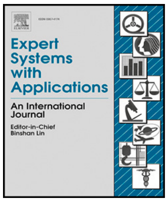
Contents lists available at ScienceDirect

Expert Systems With Applications

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



Deep Autoencoder-like NMF with Contrastive Regularization and Feature Relationship Preservation

Navid Salahian, Fardin Akhlaghian Tab [*], Seyed Amjad Seyedi, Jovan Chavoshinejad

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

ARTICLE INFO

Keywords:
Deep learning
Autoencoder structure
Nonnegative matrix factorization
Contrastive regularization
Data representation

ABSTRACT

Nonnegative Matrix Factorization is a data analysis method to discover parts-based, linear representations of data. It has been successfully used in a great variety of applications. Deep Nonnegative Matrix Factorization (deep NMF) was recently established to cope with the extraction of hierarchical latent feature representation, and it has been demonstrated to achieve outstanding results in unsupervised representation learning. However, defining a suitable regularization for the deep models is a key challenge, and the existing Deep NMF approaches lack a well-suited regularization. In this paper, we propose the Deep Autoencoder-like NMF with Contrastive Regularization and Feature Relationship preservation (DANMF-CRFR) to address the above problem. Inspired by contrastive learning, this deep model is able to learn discriminative and instructive deep features while adequately enforcing the local and global structures of the data to its decoder and encoder components. Meanwhile, DANMF-CRFR also imposes feature correlations on the basis matrices during feature learning to improve part-based learning capabilities. Multiplicative updating rules and convergence guarantees are also provided. Extensive experimental results demonstrate the advantages of the proposed model. The source code for reproducing our results can be found at https://github.com/NavidSalahian/DANMF_CRFR.

1. Introduction

Representation learning is one of the most important foundations in data mining, pattern recognition, and machine learning. Its goal is to discover the inherent characteristics hidden in high-dimensional data and map it into a much lower-dimensional space, which greatly reduces the computational overheads for the succeeding tasks. It has been used in a variety of classification and clustering applications (Bengio, Courville, & Vincent, 2013). Numerous data representation techniques have been developed to deal with high-dimensional data, including principal component analysis (PCA) (Turk & Pentland, 1991), linear discriminant analysis (LDA) (Belhumeur, Hespanha, & Kriegman, 1997), manifold learning (Belkin & Niyogi, 2003; Roweis & Saul, 2000; Tenenbaum, de Silva, & Langford, 2000), nonnegative matrix factorization (NMF) (Lee & Seung, 1999, 2000), concept factorization (Xu & Gong, 2004), sparse coding (SC) (Wright, Yang, Ganesh, Sastry, & Ma, 2009), low-rank representation (LRR) (Liu et al., 2013), and deep learning (LeCun, Bengio, & Hinton, 2015). These approaches have been effectively applied to various real-world tasks, including face recognition, graph clustering, recommendation, natural language processing, multi-task, and transfer learning.

Nonnegative matrix factorization (NMF) is one of the most widely used methods for learning compact representations of high-dimensional data, due to its superior performance in low-dimensional nonnegative representation. Lee and Seung (1999, 2000) developed the NMF method, which represents the original sample by a collection of basis vectors. In the NMF algorithm, the original samples may be reconstructed using a linear combination of basis vectors, and the reconstruction coefficient is a new representation of the original data. Such a representation based on a weighted summation of the basic vectors has a pretty straightforward and intuitive interpretation, in line with the concept of "parts form a whole" in human perception. This concept significantly increases the interpretations for various practical applications. NMF has been applied to various problem domains including pattern representation and recognition (Seyedi, Moradi, & Tab, 2017), feature extraction and dimensionality reduction (Tsuge, Shishibori, Kuroiwa, & Kita, 2001), face recognition (Wang, Jia, Hu, & Turk, 2005), bioinformatics (Kim & Park, 2007), document clustering (Xu & Gong, 2004), graph clustering (Abdollahi, Amjad Seyedi, & Reza Noorimehr, 2020), visual tracking (Wu, Shen, & Ling, 2014) and topic modeling (Kuang, Choo, & Park, 2015).

[*] Corresponding author.
E-mail addresses: n.salahian@uok.ac.ir (N. Salahian), f.akhlaghian@uok.ac.ir (F.A. Tab), amjadseyedi@uok.ac.ir (S.A. Seyedi), j.chavoshinejad@uok.ac.ir (J. Chavoshinejad).

https://doi.org/10.1016/j.eswa.2022.119051
Received 8 June 2022; Received in revised form 8 October 2022; Accepted 13 October 2022
Available online 20 October 2022

Several researchers have suggested NMF-based algorithms by including different constraints (e.g., orthogonality and sparseness) in the NMF framework to improve the effectiveness of the algorithm. These variants have been effectively used for a wide range of learning tasks, with outstanding experimental performance. Orthogonal NMF (ONMF) methods can successfully mitigate model complexity and result in unique solutions. Since the cluster indicator matrix has the same form of an orthogonal matrix, ONMF has an inherent connection to clustering (Ding, Li, Peng and Park, 2006). Additionally, a local-based representation can be enforced by using sparseness constraints, which augments the uniqueness of the decomposition. In constrained NMF problems, ONMF and sparse NMF are the most commonly and extensively used, and they have almost become a requirement in practice (Hoyer, 2004).

In some circumstances, data of many applications in a high-dimensional Euclidean space can be represented from a nonlinear low-dimensional manifold embedded in a high-dimensional ambient space that is locally flat. It has been shown that by identifying and preserving the underlying geometrical data structure, learning performance may be greatly improved (He, Yan, Hu, Niyogi, & Zhang, 2005). There are numerous manifold learning methods, such as Locally Linear Embedding (LLE) (Roweis & Saul, 2000), Laplacian Eigenmap (LE) (Belkin & Niyogi, 2003), Locality Preserving Projections (LPP) (He & Niyogi, 2003), and Stochastic Neighbor Embedding (SNE) (Hinton & Roweis, 2002). They are distinguished by the local topological attribute under consideration and the local connection between a point and its neighbors. For example, SNE can concurrently model global and local structures and cope with various manifolds. The SNE algorithm turns pairwise Euclidean distances to probabilities to simulate pairwise similarities between chosen neighbors (Hinton & Roweis, 2002). LE aims to discover low-dimensional embeddings of the original data while retaining the similarity links between local points defined by the k-nearest neighbors' or super-ball criteria (Belkin & Niyogi, 2003). The Elastic Embedding (EE) (Carreira-Perpiñan, 2010) is a dimensionality reduction approach that focuses on the relationship between SNE and LE to maintain both local and global structures underlying the data. EE can to preserve local neighborhood structure while preventing data points from crowding together.

Graph regularized nonnegative matrix factorization (GNMF) (Cai, He, Han, & Huang, 2011) considers the inherent geometrical features of data on a manifold by incorporating a Laplacian regularization to learn the latent nonlinear structures of the data. GNMF considers linear and nonlinear data correlations in the original data space by modeling it as a manifold embedded in ambient space and applying NMF on this manifold. As a result, it is more discriminating than basic NMF, which only considers the Euclidean structure. However, conventional works have a shallow structure, which means that the matrix decomposition carries out only once on the data throughout the feature learning process, which is insufficient to learn the intrinsic features of complex data. In recent years, deep learning (Goodfellow, Bengio, & Courville, 2016) has once again become attracted scholarly attention. Numerous academic topics use deep learning techniques to learn and represent highly dimensional complex data. Hence some research, motivated by deep learning (LeCun et al., 2015), attempts to transform the shallow NMF structure into a deep NMF structure that updates all of the factors iteratively to hierarchically represent the data so that the NMF can learn the hierarchical structure of complex data more effectively (De Handschutter, Gillis, & Siebert, 2021). For instance, Trigeorgis, Bousmalis, Zafeiriou, and Schuller (2014) introduced a Deep Semi-Nonnegative Matrix Factorization (Deep Semi-NMF) method that can build a deep structure by repeatedly decomposing data to learn the relation between each layer. Therefore, it can cause extraction of the intrinsic features of the data.

Regularization is one of the most important aspects of machine learning; especially deep learning (Goodfellow et al., 2016); however, the regularization term has many interpretations, and regularization techniques are mostly investigated separately. Analyzing a deep learning model with highly distributed representation is difficult and makes regularization and parameter interpretation challenging. In practical deep learning methods, it has been found that the best fitting model is a big model that has been regularized suitably. Additionally, deep matrix factorization without imposing additional restrictions on the factors (such as regularizations), allows for extremely non-unique decompositions. The uniqueness of the solution is crucial to make certain reproducibility and interpretability. Numerous constraints and regularizations might be employed depending on the application (De Handschutter et al., 2021).

Representation learning refers to the learning a mapping from the raw data domain to a feature vector or matrix, in the hope of capturing and exploiting more abstract and useful concepts that improve performance on a range of downstream tasks (Bengio et al., 2013). The motivation of this paper is to extract a more abstract and discriminative representation in a task-agnostic way. In this direction, we introduce a model for hierarchically extracting complicated data distribution underlying an input matrix using an unsupervised deep autoencoder-like structure (Ye, Chen, & Zheng, 2018). Unlike most existing deep NMF models, which only consider a decoder loss, the encoder part of the proposed model can also be used to factorize any matrix directly. This structure removes the burden of retraining a model's underlying factorization for every new data sample (unseen along the training stage) as existing non-autoencoder deep-layered models are required. In contrastive representation learning, the quality of the representation is approximated by how well the representation separates similar and dissimilar samples (Le-Khac, Healy, & Smeaton, 2020). This learning has become one of the hottest topic in unsupervised learning, due to its impressive performance in many tasks. Its motivation is to maximize the similarity of positive pairs, and the distance (dissimilarity) of negative pairs in the representation space (Hadsell, Chopra, & LeCun, 2006). Inspired by contrastive learning (Chen, Kornblith, Norouzi, & Hinton, 2020), we propose a new contrastive regularization (CR) including two local and global graph regularizers on the representation matrix to preserve data manifold and extract useful concepts. It aims to learn a representation to pull similar samples in some metric space and push apart the dissimilar samples. Most deep NMF models lack a proper regularization on basis (feature) matrices for preserving feature manifold and making more interpretable features. In this work, a deep regularization term is introduced that involves the data matrix and the basis matrices such that the pairwise relationship of the represented features is preserved as a scale form of the pairwise relationship of the original data features.

Based on the recent deep matrix factorization models, we proposed Deep Autoencoder-like Nonnegative Matrix Factorization with Contrastive Regularization and Feature Relationship preservation (DANMF-CRFR) by incorporating the appropriate penalties on each layer. DANMF-CRFR can learn the higher-level data representations and achieve remarkable clustering performance in both numerical and image datasets. Compared with previous Deep NMF methods, the main contributions of this work are summarized as follows:

- The proposed model takes into account both the encoder and decoder loss terms. After the training process, this deep model can represent out-of-sample data without requiring a retraining process. As a result, it provides significant benefits in model efficiency and reusability compared to conventional Deep NMF models.
- We introduce a new contrastive regularization of manifold learning to be embedded in the factorization process and preserve both the local and global structures of the original data in the new low-dimensional space.
- To maintain the relationship between the distribution of pair features in the deep representation space, we incorporate a customized regularization in the deep model. In this loss term, the

feature correlation in the representation space is scaled with the original space, which is equal to imposing an implicit orthogonality constraint on the representation matrix.

- We propose a Deep Autoencoder-like NMF model with Contrastive Regularization and Feature Relationship preservation (DANMF-CRFR). This model can learn feature representation while maintaining the data manifolds and geometric structures of the feature sufficiently to explore important information underlying the input data.

The remainder of the paper is organized as follows. In Section 2, we present related works and Preliminaries about shallow and deep NMF variants and explain their characteristics. Section 3 describes the proposed DANMF-CRFR model in detail. Extensive experimental results and analyze the time complexity are shown in Section 4. Finally, Section 5 wraps up the paper and gives some future works.

2. Background

In this section, we present the related works on variants of NMF algorithms in the first Section 2.1. Then the preliminaries of basic NMF and encoder–decoder NMF models are presented in the following Section 2.2.

2.1. Related work

The basic nonnegative matrix factorization decomposes a nonnegative matrix $\boldsymbol{X}$ into (typically) two nonnegative matrices $\boldsymbol{W}$ and $\boldsymbol{H}$. In most cases, the factorization problem cannot be solved analytically, and is approximated numerically. Furthermore, since the dimensions of the factor matrices $\boldsymbol{W}$ and $\boldsymbol{H}$ may be substantially less than the dimensions of the product matrix $\boldsymbol{X}$, NMF can be regarded as a low-rank approximation approach for extracting any latent structure in the data. NMF with a sum of the squared error cost function is similar to soft k-means clustering (Ding, He, & Simon, 2005), and when using the I-divergence cost function, it is similar to the probabilistic latent semantic indexing (Ding, Li and Peng, 2006). Furthermore, Ding, Li, Peng and Park (2006) indicated that Spectral clustering is theoretically closely related to NMF. The key characteristics typically used to combine with NMF are sparsity and orthogonality. Existing NMF algorithms fall into four categories: (1) basic NMF (BNMF), which imposes only the non-negativity constraint; (2) constrained NMF (CNMF), which imposes additional constraints such as regularization; (3) generalized NMF, which transcends prevalent data types or factorization modes more broadly; and (4) structured NMF (SNMF), which modifies standard factorization formulations (Wang & Zhang, 2013).

As previously stated, under the only non-negativity constraint, Basic NMF will not get the unique solution. Thus, to overcome the ill-posedness, it is necessary to introduce extra auxiliary constraints on the feature matrix and/or coefficient matrix as regularization terms. These regularization terms can include side information and more properly represent the characteristics of the problems. NMF with orthogonality constraint on either the factor $\boldsymbol{W}$ or $\boldsymbol{H}$ is known as orthogonal NMF. Li, Hou, Zhang, and Cheng (2001) used the orthogonality principle to reduce redundancy across various bases at first; subsequently, Ding, Li, Peng and Park (2006) formally introduced the notion of Orthogonal NMF. The sparseness constraint helps in increasing the uniqueness of the factorization as well as imposing a local-based representation. Hoyer proposed an NMF technique with sparse constraints for promoting the part-based representation of the basis matrix (Hoyer, 2004). In this approach, a sparse metric based on the relationship between the L_1 and L_2 norms was developed to determine the sparsity of the vector after factorization. Cai et al. (2011) demonstrate that data is generally sampled from low-dimensional manifolds embedded in high-dimensional space, and they suggest the graph regularized nonnegative matrix factorization (GNMF) method. Hedjam, Abdesselam, and Melgani (2021) incorporate a regularization term in the NMF cost function so that the pairwise relationships between the features of samples and the features of the cluster centroids are preserved after transformation. This causes the scatter of the cluster centroids to match proportionately with the scatter of the samples.

In a broad sense, generalized NMF methods may be regarded as extensions of NMF. In contrast to Constrained NMF, which introduces certain extra constraints like penalty terms, it breaks the intrinsic non-negativity constraint to some extent, alters the data types, or changes the decomposition pattern, and so on. Ding, Li, and Jordan (2010) relax the constraints on factorizations to make the approach relevant to the analysis of both positive and negative data. They restrict the coefficient matrix to be non-negative and offer the technique for semi-NMFs. For interpretability reasons, Convex NMF (CNMF) restricts each column of the basis matrix to be a convex combination of data points. The basis matrix in CNMF is represented by $\boldsymbol{W} = \boldsymbol{X}\boldsymbol{G}$. The benefit of this convex constraint is that each column of $\boldsymbol{W}$ may be interpreted as a weighted sum of specific data samples, and the final coefficient matrix $\boldsymbol{H}$ becomes more sparse and orthogonal. Some other works (Peng, Zhang, Kang, Chen, & Cheng, 2021; Tolić, Antulov-Fantulin, & Kopriva, 2018) propose the extended CNMF methods, which exploit local structures of the data to construct basis vectors and a coefficient matrix. The learned basis and coefficients preserve the intrinsic geometrical structures of the data. Also, these approaches imply an orthogonal constraint on the coefficient and consider nonlinear data structures in kernel space.

In the solution to the NMF learning problem, Structured NMF imposes other characteristics or structures. Rather than introducing new restrictions as penalty terms, it alters the standard factorization formulation directly. Ding, Li, Peng and Park (2006) present a Tri-factor NMF or NMTF ($\boldsymbol{X} \approx \boldsymbol{W}\boldsymbol{S}\boldsymbol{H}$), which is a useful tool for clustering row and column samples at the same time. Both W and H are forced to be orthogonal by the model. Consequently, $\boldsymbol{W}$ and $\boldsymbol{H}$ relate to the row and column clusters, respectively. The matrix $\boldsymbol{S}$ is provided to add more degrees of freedom to the approximation, ensuring that it stays accurate. Peng, Ser, Chen, and Lin (2020) proposed the CNMTF algorithm, which is a Correntropy-based Orthogonal Nonnegative Matrix Tri-Factorization. This method is a robust NMTF algorithm that considers double graph regularization and orthogonality conditions. Previous latent factor methods are mostly built on reconstructing high-dimensional data from its sample representation, especially decoder, requiring the representation to have certain desired qualities. However, these approaches lack an encoder part that explicitly converts samples to their representation. Sun, Shen, Gao, Ouyang, and Cheng (2017) introduced a nonnegative symmetric encoder–decoder method for community detection. By incorporating a decoder and an encoder directly into an integrated loss function.

Due to the fact that the original data matrix in the real-world problems often includes complicated information, it is difficult to extract the high-level characteristics of the data using the NMF approach based on a shallow structure. Trigeorgis et al. (2014) introduced a Deep Semi-NMF algorithm that may establish a deep structure by factorizing the data hierarchically, allowing the relationships between the different layers to be exploited to reveal the inherent high-level features of the original data. The basic principle behind Deep semi-NMF is to decompose the coefficient matrix several times using the semi-NMF algorithm, then use the decomposed outputs as the input for the next layer and, construct a deep model structure. Unfortunately, Deep semi-NMF fails to extract a part-based representation of input data due to containing negative values in the basis matrix. Yu, Zhou, Cichocki, and Xie (2018) introduced a deep NMF method that extends non-smooth NMF with a deep framework that effectively explores part-based and hidden features of complex data. Guo and Zhang (2020) proposed a deep architecture of the Sparse Nonnegative Matrix Factorization called (SDNMF), which can effectively extract the sparse structure of the data and applies the sparse restrictions to each layer basis matrix and representation matrix. Guo (2021) proposed a deep NMF model that extends Sparse Dual Graph Regularization NMF with a deep structure that can preserve the inherent geometrical structures of the data in the sample and feature latent spaces.

Shu, Wu, Hu, You, and Fan (2021) introduced the Deep Semi-NMF-EP approach for learning a hidden representation of high-dimensional data with Elastic Preserving. It creates two graphs in each layer to discover data elasticity. However, its regularization term is not in line with its nonnegative framework; therefore, this method failed to define proper update rules. Shu, Sun, Tang, and You (2022) introduced Adaptive Graph Regularized Deep Semi-Nonnegative Matrix Factorization called AGRDSNMF, which maintains the inherent local structure of data by learning a dynamic graph in each layer in contrast to a predetermined fixed graph. Zhao, Wang, and Pei (2021) developed a deep NMF structure based on the underlying basis image layers (DNBMF), which are used to solve face recognition tasks. This deep NMF architecture conducts deep factorization on the basis matrix to produce the underlying basis matrix that may be used to represent the local characteristics of data. Additionally, a regularized variant called RDNBMF is also proposed as extend of DNBMF. This extension incorporates a regularization term to the objective function and constrains the basis images matrix directly to produce a significant difference between the factorized basis in each layer which makes a better sparse representation. Even though these methods can achieve more accurate sample representation, DNBMF has not a proper regularization term, and both structures are hierarchical multi-layers rather than deep ones. Ye et al. (2018) introduced a deep autoencoder-like NMF (DANMF) model for the community detection problem. DANMF is made up of two deep-structured encoders and decoders. Like the deep autoencoder network, the encoder component aims to map the graph network into the community membership space, and the decoder component tries to transform low-dimensional representations into reconstruction space. Lyu, Xie, and Sun (2017) presented a deep NMF model for learning the hierarchical features of data for Orthogonal Nonnegative Matrix Factorization. Yang and Xu (2021) proposed a deep autoencoder-like network for ONMF, namely DAutoED-ONMF, which extends the DANMF method by applying explicit orthogonality constraints on the representation matrix. The drawback of this method is the lack of a proper regularization term for the deep structure. It only focuses on preserving the local structure and ignores the global geometric structure.

Although these deep NMF algorithms enhance complex data analysis, they all lack proper regularization to efficiently conform to the deep structure models. In the following, we introduce a holistic data representation method that combines and develops ideas and components from the current dominant paradigms for Deep structure.

2.2. Preliminaries

Given a nonnegative observation data matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n] \in \mathbb{R}_+^{d\times n}$, where each column of $\boldsymbol{X}$ denotes a sample vector, d represents the number of features, and n represents the number of samples. The NMF algorithm aims to seek two nonnegative matrices $\boldsymbol{W} \in \mathbb{R}_+^{d\times k}$ and $\boldsymbol{H} \in \mathbb{R}_+^{k\times n}$, which can well reconstruct data matrix $\boldsymbol{X}$ as follow:

$$\boldsymbol{X} \approx \boldsymbol{W}\boldsymbol{H}. \tag{1}$$

The squared Euclidean distance is the commonly used cost function to measure the quality of the approximation which can be written as follow:

$$\min_{\boldsymbol{W},\boldsymbol{H}} \mathcal{L}_{\text{NMF}} = \|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{H}\|_F^2 = \sum_{i=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{W}\boldsymbol{h}_i\|^2, \text{ s.t.} \quad \boldsymbol{W} \geq 0, \boldsymbol{H} \geq 0, \tag{2}$$

where $\|\cdot\|_F$ stands for the matrix Frobenius norm. By adopting multiplicative update rules (Lee & Seung, 2000) for nonnegative optimization, the updating rules of the (2) can be obtained as follow:

$$\boldsymbol{W} \leftarrow \boldsymbol{W} \odot \frac{\boldsymbol{X}\boldsymbol{H}^\top}{\boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^\top}, \quad \boldsymbol{H} \leftarrow \boldsymbol{H} \odot \frac{\boldsymbol{W}^\top \boldsymbol{X}}{\boldsymbol{W}^\top \boldsymbol{W}\boldsymbol{H}}. \tag{3}$$

where $\odot$ shows the Hadamard product, and $\boldsymbol{A}^\top$ denotes the transpose of the matrix $\boldsymbol{A}$.

From the above model described in (2), we can see that this model only relies on the loss function of the decoder that approximates the original input $\boldsymbol{X}$ from the representation $\boldsymbol{H}$ in latent space $\boldsymbol{W}$. Moreover, considering a decoder-only structure requires executing the whole of procedure for every out of samples as input for the factorization process. To address the aforementioned issues and better inherit the representation learning capability of autoencoders, it makes sense and is essential to incorporate its encoder stage in the loss function. By unifying the loss function of the encoder and the decoder stage, the proposed factorization model does not need to be retrained to work with data that was not seen during the training process. The loss function of the encoder can be expressed as follows:

$$\min_{\boldsymbol{W},\boldsymbol{H}\geq 0} \|\boldsymbol{H} - \boldsymbol{W}^\top \boldsymbol{X}\|_F^2 = \sum_{i=1}^{n} \|\boldsymbol{h}_i - \boldsymbol{W}^\top \boldsymbol{x}_i\|^2, \text{ s.t.} \quad \boldsymbol{W} \geq 0, \boldsymbol{H} \geq 0. \tag{4}$$

By combining the decoder and encoder stage, respectively, into a unified loss function, the objective function of autoencoder-like is:

$$\min_{\boldsymbol{W},\boldsymbol{H}} \mathcal{L}_{\text{ANMF}} = \|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{H}\|_F^2 + \|\boldsymbol{H} - \boldsymbol{W}^\top \boldsymbol{X}\|_F^2, \text{ s.t.} \quad \boldsymbol{W} \geq 0, \boldsymbol{H} \geq 0. \tag{5}$$

3. Proposed model

In this section, based on the Deep Autoencoder-like NMF model, we propose a new representation method that employs manifold structure and feature correlation to include Contrastive Regularization and Feature Relationship preservation (DANMF-CRFR). It can be applied to different kinds of real-world datasets. Its success is mainly due to four factors: (1) It uses the deep autoencoder structure to obtain a more abstract and discriminative representation, which also provides the capability to handle unseen data (Section 3.1); (2) By exploiting both global and local manifolds of data, it efficiently preserves the geometrical structure of data, and so improves the capability of the deep model for complex data representation (Section 3.2); (3) The feature relationship preservation biases the representation method to learn more discriminative latent feature, without the need to impose an explicit orthogonal constraint on the proposed model (Section 3.3); (4) The model integrates the above notions into one joint learning problem and adopts an efficient multiplicative update rules strategy for optimization (Section 3.4). The structure of the DANMF-CRFR model is shown in Fig. 1.

3.1. Model formulation

Despite the commendable success of the autoencoder-like in data representation, there is still considerable room to improve the representation performance further. As we mentioned before, in many practical applications, datasets might have highly complex features and hidden feature relationships such as lighting conditions, pose, and resolution. Shallow autoencoder-like NMF models with two factors only consider the information in a single layer, ignoring the fact that the input data involve hierarchical features with hidden information. To cover this deficiency, recently, researchers have proposed several deep matrix factorization models (Shu et al., 2021; Trigeorgis et al., 2014; Yang & Xu, 2021; Ye et al., 2018; Zhao et al., 2021) to understand such complex data better and explore hierarchical features and latent information. Therefore, it is meaningful to develop the shallow autoencoder-like model by a deep structure. A deep autoencoder-like model decomposes the nonnegative data matrix $\boldsymbol{X}$ into $L+1$ nonnegative factor matrices as follows:

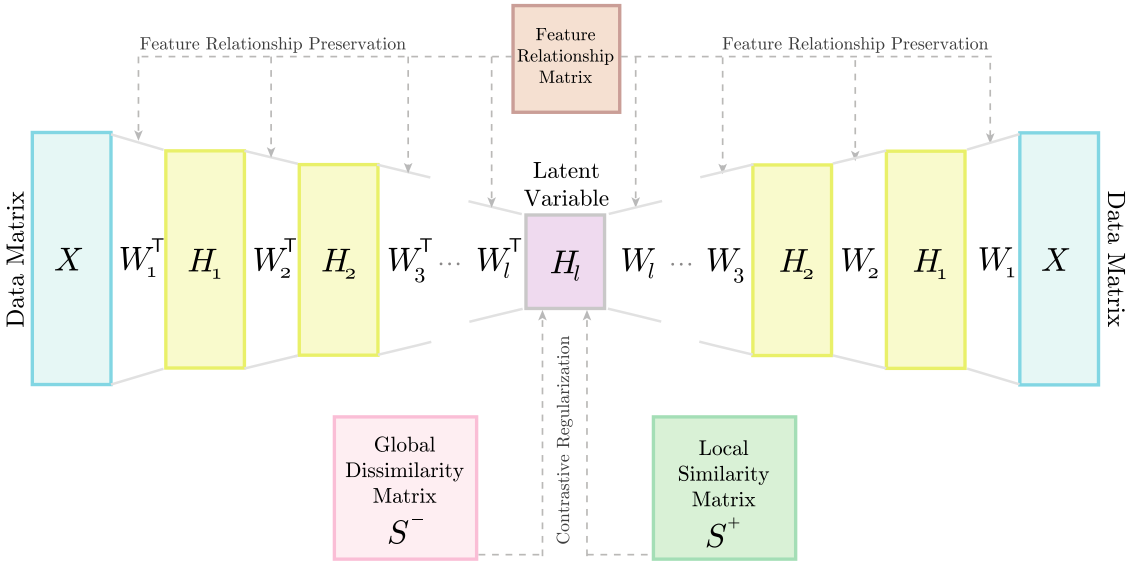

Feature Relationship Preservation
Feature Relationship Matrix
Feature Relationship Preservation
Data Matrix
X
$W_1^\top$
H_1
$W_2^\top$
H_2
$W_3^\top$
$\cdots$
$W_l^\top$
Latent Variable
H_l
W_l
$\cdots$
W_3
H_2
W_2
H_1
W_1
X
Data Matrix
Contrastive Regularization
Global Dissimilarity Matrix S^-
Local Similarity Matrix S^+

Fig. 1. The illustration of the deep autoencoder-like NMF with Contrastive Regularization and Feature Relationship preservation (DANMF-CRFR).

$$\begin{array}{ll}
\boldsymbol{X} \approx \boldsymbol{W}_1\boldsymbol{H}_1 & \boldsymbol{H}_1 \approx \boldsymbol{W}_1{}^\top\boldsymbol{X} \\
\boldsymbol{H}_1 \approx \boldsymbol{W}_2\boldsymbol{H}_2 & \boldsymbol{H}_2 \approx \boldsymbol{W}_2{}^\top\boldsymbol{H}_1 \\
\vdots & \vdots \\
\boldsymbol{H}_{l-1} \approx \boldsymbol{W}_l\boldsymbol{H}_l & \boldsymbol{H}_l \approx \boldsymbol{W}_l{}^\top\boldsymbol{H}_{l-1} \\
\downarrow & \downarrow \\
\boldsymbol{X} \approx \boldsymbol{W}_1\ldots\boldsymbol{W}_l\boldsymbol{H}_l & \boldsymbol{H}_l \approx \boldsymbol{W}_l{}^\top\ldots\boldsymbol{W}_1^\top\boldsymbol{X},
\end{array} \tag{6}$$

where $\boldsymbol{H}_l \in \mathbb{R}_+^{k\times n}, \boldsymbol{W}_i \in \mathbb{R}_+^{r_{i-1}\times r_i}(1 \leq i \leq l)$, and we set $d = r_0 \geq r_1 \geq \cdots \geq r_{l-1} \geq r_l = k$.

After that, each layer is fine-tuned by alternating minimization of (7).

$$\min_{\boldsymbol{W}_i,\boldsymbol{H}_l} \mathcal{L}_{DANMF} = \|\boldsymbol{X} - \boldsymbol{W}_1\ldots\boldsymbol{W}_l\boldsymbol{H}_l\|_F^2 + \|\boldsymbol{H}_l - \boldsymbol{W}_l^\top\ldots\boldsymbol{W}_1^\top\boldsymbol{X}\|_F^2 \tag{7}$$
$$\text{s.t.} \quad \boldsymbol{H}_l \geq 0, \boldsymbol{W}_i \geq 0, \forall i = 1, 2, \ldots, l.$$

3.2. Contrastive regularization

Recently, contrastive learning has been applied to both supervised and unsupervised data, and its outstanding performance on a variety of vision and language tasks is significant. Inspired by this learning paradigm (Chen et al., 2020; Hadsell et al., 2006), we introduce a contrastive regularization based on the local and global data structures. The main idea of contrastive regularization is to induce representation such that similar instances stay close to each other, while dissimilar ones are far apart. Intuitively, for an efficient unsupervised representation learning method, if two instances are close to each other in the local space (resp. far from each other in the global space), their representations learned by the model should be highly similar (resp. dissimilar) to each other based on some distance measures. Thus, the structures can be naturally used to constrain the similarity and dissimilarity relationships among the representations.

In this method, the attractive regularization is incorporated into the deep model to preserve the local structure of the data. In a straightforward definition, the local structure captures the form of cohesive parts (potential classes). This regularization forces similar samples in the high-dimensional space to attract each other in the representation space. On the other hand, the modeling based on neighborhood relationships might be risky statistically in high-dimensional space (sparsely populated due to the curse of dimensionality). For example, some euclidean nearest neighbors may be semantically dissimilar (Bengio et al., 2013). To address this shortcoming, we propose repulsive regularization, which has a more comprehensive view and tries to increase the between-class separation. This regularization allows a moderate distance in the high-dimensional space to be faithfully modeled by a much larger distance in the representation space. As a result, it eliminates the unwanted attractive forces between represented samples. It could be claimed that, to capture the local and global structures, these contrary forces refine each other in a competitive process.

3.2.1. Modeling of the repulsive term

We utilized the inner product of the low-dimensional representations to measure their similarities, i.e., $\boldsymbol{H}^\top\boldsymbol{H} \in \mathbb{R}_+^{n\times n}$, and introduced the following repulsive term,

$$\min_{\boldsymbol{H}_l} \|\boldsymbol{S}^- \odot \boldsymbol{H}_l{}^\top\boldsymbol{H}_l\|_1, \tag{8}$$

where $\|\cdot\|_1$ is an L_1 matrix norm, the symbol $\odot$ shows Hadamard product, and the dissimilarity matrix $\boldsymbol{S}^-$ is constructed based on euclidean distance between each pair instances in the high-dimensional space, that is

$$S_{ij}^- = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2. \tag{9}$$

Apparently, the product of the low-dimensional representations commensurate with the dissimilarity degree minimizes the repulsive regularization in (8), such that the low-dimensional representations of dissimilar instances are pushed away from each other.

3.2.2. Modeling of the attractive term

We can see that the regularization in (8) ignores the similarity between the low-dimensional representations of these instances. Based on the idea that if two instances are close to each other with high confidence, it is rational to assume that their representations are also similar. To this end, we propose the following attractive regularization to compensate for the repulsive regularization in (8).

$$\min_{\boldsymbol{H}_l} \|\boldsymbol{S}^+ \odot \tilde{\boldsymbol{H}}_l\|_1, \tag{10}$$

where $\tilde{H}_{i,j} = \left\|\boldsymbol{h}_i - \boldsymbol{h}_j\right\|^2$. Therefore, we further employ this term to regularize the model. We use the Gaussian similarity function to transform the distance to similarity measurement and construct the similarity graph based on the p-nearest neighbor graph as follows,

$$S^+{}_{i,j} = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{\sigma^2}}, & \text{if } \boldsymbol{x}_j \in \mathcal{N}_p(\boldsymbol{x}_i) \\ 0, & \text{otherwise,} \end{cases} \tag{11}$$

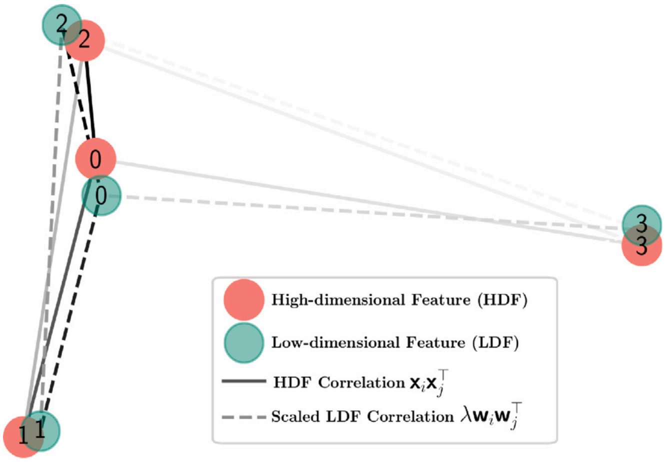

2
2
0
0
3
3
1
1
High-dimensional Feature (HDF)
Low-dimensional Feature (LDF)
HDF Correlation $\mathbf{x}_i\mathbf{x}_j^\top$
Scaled LDF Correlation $\lambda\mathbf{w}_i\mathbf{w}_j^\top$

Fig. 2. Graph-based illustration of the relationship between pairs of features in the high-dimensional space and in the low-dimensional space.

where σ is the bandwidth parameter, and $\mathcal{N}_p(\boldsymbol{x}_i)$ indicates that instance i belongs to the p-nearest neighbors of instance j. By combining the aforementioned terms (8) and (10), our contrastive regularization is formed as follows,

$$\min_{\boldsymbol{H}_l} \mathcal{R}_{CR}(\boldsymbol{H}_l) = \lambda_1 \|\boldsymbol{S}^- \odot \boldsymbol{H}_l{}^\top \boldsymbol{H}_l\|_1 + \lambda_2 \|\boldsymbol{S}^+ \odot \tilde{\boldsymbol{H}}_l\|_1, \tag{12}$$

where λ_1 and λ_2 are repulsive and attractive parameters, respectively.

3.3. Feature relationship preservation

The feature manifold problems arise when the recognition depends on the correlation of features (Dietterich, Jain, Lathrop, & Lozano-Pérez, 1993) (such as a correlation between eye width and height, sepal length and width, and vertical and horizontal extents of a character). Accordingly, in addition to preserving the data manifold, preserving the proportionality of the represented samples (feature manifold) can be helpful. Recently, some regularized NMF methods (Hedjam et al., 2021; Shang, Jiao, & Wang, 2012) are proposed to preserve the pairwise relationship between the high-dimensional features and their corresponding latent features. This relationship leads to forcing the scatter of the representations to coincide proportionally with the scatter of the data points. This regularization utilizes the feature correlation $\boldsymbol{X}\boldsymbol{X}^\top \in \mathbb{R}_+^{d\times d}$ as an outline (basis contour); therefore, the model tries to match the deep representation feature correlation $\boldsymbol{W}_1\ldots\boldsymbol{W}_l\boldsymbol{W}_l^\top\ldots\boldsymbol{W}_1^\top \in \mathbb{R}_+^{d\times d}$ to the form of this outline.

In this subsection, we want to add a tailored regularization for the proposed deep model. This regularization can be useful because it implies implicit orthogonality for the representation matrix $\boldsymbol{H}_l$. Therefore, it becomes possible to learn a deep representation with more discriminative power (for more details, see Appendix). To achieve this goal, we propose a deep regularization term as

$$\min_{\boldsymbol{W}_i} \mathcal{R}_{FR}(\boldsymbol{W}_i) = \|\boldsymbol{X}\boldsymbol{X}^\top - \lambda_3 \boldsymbol{W}_1\ldots\boldsymbol{W}_l\boldsymbol{W}_l^\top\ldots\boldsymbol{W}_1^\top\|_F^2 = \|\boldsymbol{X}\boldsymbol{X}^\top - \lambda_3 \boldsymbol{\Psi}_l\boldsymbol{\Psi}_l^\top\|_F^2, \tag{13}$$

where $\boldsymbol{\Psi}_l = \prod_{i=1}^{l} \boldsymbol{W}_i = \boldsymbol{W}_1\ldots\boldsymbol{W}_l \in \mathbb{R}_+^{d\times k}$ is hierarchical latent features, and λ_3 is a scalar that controls the scale relationship between the two spaces. The Feature Relationship Preservation (FRP) regularization considers the global feature correlations to induce the feature structure in the representation space. In this regularization, $[\boldsymbol{X}\boldsymbol{X}^\top]_{i,j}$ indicates the correlation degree between feature $\boldsymbol{x}^{(i)}$ and feature $\boldsymbol{x}^{(j)}$, and $[\boldsymbol{\Psi}_l\boldsymbol{\Psi}_l^\top]_{i,j}$ measures the correlation degree between latent feature $\boldsymbol{\Psi}_l^{(i)}$ and $\boldsymbol{\Psi}_l^{(j)}$. Therefore, (13) forces the deep factorization model to preserve these pairwise proportionalities as much as possible.

Fig. 2 is an example of this feature relationship preservation on the *Iris* dataset. Solid lines demonstrate the feature correlation or basis contour, and dashed lines show the representation of feature correlation or mapped contour. The color intensity of edges indicates the feature relationship degree. For example, there is a strong correlation between f_0 and f_2 features (sepal and petal lengths), and this correlation degree is preserved in the representation space as well. The lack of correlation between f_1 and f_3 features (sepal and petal widths) in both contours indicates the ability to memorize the relationships in this regularization.

By integrating predefined terms (i.e., (7), (12), and (13)), the unified cost function of the DANMF-CRFR model $\mathcal{L} = \mathcal{L}_{\text{DANMF}} + \lambda_{1,2}\mathcal{R}_{CR} + \lambda_3\mathcal{R}_{FR}$ is then given as

$$\begin{aligned}\min_{\boldsymbol{W}_i,\boldsymbol{H}_l} \mathcal{L}(\boldsymbol{W}_i, \boldsymbol{H}_i) = &\|\boldsymbol{X} - \boldsymbol{W}_1\ldots\boldsymbol{W}_l\boldsymbol{H}_l\|_F^2 + \|\boldsymbol{H}_l - \boldsymbol{W}_l^\top\ldots\boldsymbol{W}_1^\top\boldsymbol{X}\|_F^2 \\ &+ \lambda_1\|\boldsymbol{S}^- \odot \boldsymbol{H}_l^\top\boldsymbol{H}_l\|_1 + \lambda_2\|\boldsymbol{S}^+ \odot \tilde{\boldsymbol{H}}_l\|_1 \\ &+ \|\boldsymbol{X}\boldsymbol{X}^\top - \lambda_3\boldsymbol{W}_1\ldots\boldsymbol{W}_l\boldsymbol{W}_l^\top\ldots\boldsymbol{W}_1^\top\|_F^2 \\ &\text{s.t.} \quad \boldsymbol{W}_i \geq 0, \boldsymbol{H}_i \geq 0, \forall i = 1, 2, \ldots, l.\end{aligned} \tag{14}$$

This model is defined based on a deep autoencoder network. Different from existing deep NMF methods, it memorizes feature correlations in the learning process. Furthermore, it takes both the local and global geometrical structure of the data into consideration in a contrastive manner. Therefore, the DANMF-CRFR derives valuable information from complex data structures.

3.4. Numerical solution

To accelerate factor reconstruction in our model, we pre-train each layer to have an initial approximation of the matrices $\boldsymbol{W}_i$, $\boldsymbol{H}_i$, since this significantly reduces the model's training time. Pre-training on deep autoencoder networks has already been shown to be beneficial (Hinton & Salakhutdinov, 2006). For carrying out the pre-training, initially, we factorize input data matrix $\boldsymbol{X} \approx \boldsymbol{W}_1\boldsymbol{H}_1$ by minimizing $\|\boldsymbol{X} - \boldsymbol{W}_1\boldsymbol{H}_1\|_F^2 + \|\boldsymbol{H}_1 - \boldsymbol{W}_1^\top\boldsymbol{X}\|_F^2$, where $\boldsymbol{W}_1 \in \mathbb{R}_+^{d\times r_1}$ and $\boldsymbol{H}_1 \in \mathbb{R}_+^{r_1\times n}$. Moreover, we factorize the coefficient matrix $\boldsymbol{H}_1 \approx \boldsymbol{W}_2\boldsymbol{H}_2$ by minimizing $\|\boldsymbol{H}_1 - \boldsymbol{W}_2\boldsymbol{H}_2\|_F^2 + \|\boldsymbol{H}_2 - \boldsymbol{W}_2^\top\boldsymbol{H}_1\|_F^2$, where $\boldsymbol{W}_2 \in \mathbb{R}_+^{r_1\times r_2}$ and $\boldsymbol{H}_2 \in \mathbb{R}_+^{r_2\times n}$. This process will be repeated until all layers have been pre-trained. Afterward, each layer is fine-tuned by alternating minimization of the cost function in (14). In the following, we present the updating rules.

Algorithm 1 Deep Autoencoder-like NMF with Contrastive Regularization and Feature Relationship Preservation (DANMF-CRFR)

Input: The data matrix $\boldsymbol{X}$; the size of each layer r_i; the regularization parameters λ_1, λ_2, and λ_3;
Output: $\boldsymbol{W}_i$ $(1 \le i \le l)$, $\boldsymbol{H}_i$ $(1 \le i \le l)$

```
Calculate Dissimilarity matrix $\boldsymbol{S}^-$
Calculate Similarity matrix $\boldsymbol{S}^+$
▷ Pre-training process:
$\boldsymbol{W}_1, \boldsymbol{H}_1 \leftarrow$ShallowANMF($\boldsymbol{X}, r_1$);
for $i = 2$ to $l$ do
    $\boldsymbol{W}_i, \boldsymbol{H}_i \leftarrow$ ShallowANMF($\boldsymbol{H}_{i-1}, r_i$);
end for
▷ Fine-tuning process:
while convergence not reached do
    for $i = 1$ to $l$ do
        $\boldsymbol{\Psi}_{i-1} \leftarrow \prod_{\tau=1}^{i-1} \boldsymbol{W}_\tau (\boldsymbol{\Psi}_0 \leftarrow \mathbf{I})$;
        $\boldsymbol{\Phi}_{i+1} \leftarrow \prod_{\tau=i+1}^{l} \boldsymbol{W}_\tau (\boldsymbol{\Phi}_{p+1} \leftarrow \mathbf{I})$;
        Update $\boldsymbol{W}_i$ according to (20);
        $\boldsymbol{\Psi}_i \leftarrow \boldsymbol{\Psi}_{i-1} \boldsymbol{W}_i$;
        Update $\boldsymbol{H}_i$ according to (29) ($i < l$, optional) or according to
        (27) ($i = l$);
    end for
end while
return $\boldsymbol{W}_i, \boldsymbol{H}_i, \forall i = 1, 2, \dots, l$;
```

3.4.1. Updating rule of the basis matrix $\boldsymbol{W}_i$ $(1 \le i \le p)$

By fixing all the factors except for $\boldsymbol{W}_i$, the cost function in (14) is reduced to:

$$\begin{aligned} \min_{\boldsymbol{W}_i} \mathcal{L}(\boldsymbol{W}_i) =& \|\boldsymbol{X} - \boldsymbol{\Psi}_{i-1}\boldsymbol{W}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{H}_l\|_F^2 + \|\boldsymbol{H}_l - \boldsymbol{\Phi}_{i+1}^\top \boldsymbol{W}_i^\top \boldsymbol{\Psi}_{i-1}^\top \boldsymbol{X}\|_F^2 \\ &+ \|\boldsymbol{X}\boldsymbol{X}^\top - \lambda_3 \boldsymbol{\Psi}_{i-1}\boldsymbol{W}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{\Phi}_{i+1}^\top \boldsymbol{W}_i^\top \boldsymbol{\Psi}_{i-1}^\top\|_F^2 \\ &\text{s.t.} \quad \boldsymbol{W}_i \ge 0, \end{aligned} \tag{15}$$

where $\boldsymbol{\Psi}_{i-1} = \boldsymbol{W}_1 \ldots \boldsymbol{W}_{i-1}$ and $\boldsymbol{\Phi}_{i+1} = \boldsymbol{W}_{i+1} \ldots \boldsymbol{W}_l$. When $i = 1$, we set $\boldsymbol{\Psi}_0 = \boldsymbol{I}$. Similarly, when $i = p$, we set $\boldsymbol{\Phi}_{l+1} = \boldsymbol{I}$. To solve (15), we define a Lagrangian multiplier $\boldsymbol{\Theta}_i$ to enforce the nonnegative constraints on $\boldsymbol{W}_i$, resulting in the following equivalent cost function:

$$\begin{aligned} \min_{\boldsymbol{W}_i, \boldsymbol{\Theta}_i} \mathcal{L}(\boldsymbol{W}_i, \boldsymbol{\Theta}_i) =& \|\boldsymbol{X} - \boldsymbol{\Psi}_{i-1}\boldsymbol{W}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{H}_l\|_F^2 + \|\boldsymbol{H}_l - \boldsymbol{\Phi}_{i+1}^\top \boldsymbol{W}_i^\top \boldsymbol{\Psi}_{i-1}^\top \boldsymbol{X}\|_F^2 \\ &+ \|\boldsymbol{X}\boldsymbol{X}^\top - \lambda_3 \boldsymbol{\Psi}_{i-1}\boldsymbol{W}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{\Phi}_{i+1}^\top \boldsymbol{W}_i^\top \boldsymbol{\Psi}_{i-1}^\top\|_F^2 \\ &- \mathrm{Tr}(\boldsymbol{\Theta}_i \boldsymbol{W}_i^\top), \end{aligned} \tag{16}$$

where Tr($\boldsymbol{A}$) it the sum of the entries on the main diagonal of $\boldsymbol{A}$. (16) can be further rewritten as follows:

$$\begin{aligned} \min_{\boldsymbol{W}_i, \boldsymbol{\Theta}_i} \mathcal{L}(\boldsymbol{W}_i, \boldsymbol{\Theta}_i) = & \mathrm{Tr}(-2\boldsymbol{X}^\top \boldsymbol{\Psi}_{i-1}\boldsymbol{W}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{H}_l \\ &+ \boldsymbol{H}_l^\top \boldsymbol{\Phi}_{i+1}^\top \boldsymbol{W}_i^\top \boldsymbol{\Psi}_{i-1}^\top \boldsymbol{\Psi}_{i-1}\boldsymbol{W}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{H}_l \\ &+ \boldsymbol{H}_l^\top \boldsymbol{H}_l - 2\boldsymbol{H}_l^\top \boldsymbol{\Phi}_{i+1}^\top \boldsymbol{W}_i^\top \boldsymbol{\Psi}_{i-1}^\top \boldsymbol{X} + \boldsymbol{X}^\top \boldsymbol{\Psi}_{i-1}\boldsymbol{W}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{\Phi}_{i+1}^\top \boldsymbol{W}_i^\top \boldsymbol{\Psi}_{i-1}^\top \boldsymbol{X}) \\ &+ \mathrm{Tr}(-2\lambda_3 \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{\Psi}_{i-1}\boldsymbol{W}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{\Phi}_{i+1}^\top \boldsymbol{W}_i^\top \boldsymbol{\Psi}_{i-1}^\top \\ &+ \lambda_3^2 \boldsymbol{\Psi}_{i-1}\boldsymbol{W}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{\Phi}_{i+1}^\top \boldsymbol{W}_i^\top \boldsymbol{\Psi}_{i-1}^\top \boldsymbol{\Psi}_{i-1}\boldsymbol{W}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{\Phi}_{i+1}^\top \boldsymbol{W}_i^\top \boldsymbol{\Psi}_{i-1}^\top) \\ &- \mathrm{Tr}(\boldsymbol{\Theta}_i \boldsymbol{W}_i^\top). \end{aligned} \tag{17}$$

By setting the partial derivative of $\mathcal{L}(\boldsymbol{W}_i, \boldsymbol{\Theta}_i)$ with respect to $\boldsymbol{W}_i$ to $\mathbf{0}$, we obtain:

$$\begin{aligned} \boldsymbol{\Theta}_i =& -4\boldsymbol{\Psi}_{i-1}^\top \boldsymbol{X}\boldsymbol{H}_l^\top \boldsymbol{\Phi}_{i+1}^\top + 2\boldsymbol{\Psi}_{i-1}^\top \boldsymbol{\Psi}_{i-1}\boldsymbol{W}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{H}_l\boldsymbol{H}_l^\top \boldsymbol{\Phi}_{i+1}^\top \\ &+ 2\boldsymbol{\Psi}_{i-1}^\top \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{\Psi}_{i-1}\boldsymbol{W}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{\Phi}_{i+1}^\top - 4\lambda_3 \boldsymbol{\Psi}_{i-1}^\top \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{\Psi}_{i-1}\boldsymbol{W}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{\Phi}_{i+1}^\top \\ &+ 4\lambda_3^2 \boldsymbol{\Psi}_{i-1}^\top \boldsymbol{\Psi}_{i-1}\boldsymbol{W}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{\Phi}_{i+1}^\top \boldsymbol{W}_i^\top \boldsymbol{\Psi}_{i-1}^\top \boldsymbol{\Psi}_{i-1}\boldsymbol{W}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{\Phi}_{i+1}^\top. \end{aligned} \tag{18}$$

From the complementary slackness condition of the Karush–Kuhn–Tucker (KKT), we obtain:

$$\boldsymbol{\Theta}_i \odot \boldsymbol{W}_i = \mathbf{0}, \tag{19}$$

Formula (19) is the fixed point equation that the solution must satisfy at convergence. By solving this equation, we have the following updating rule for $\boldsymbol{W}_i$:

$$\boldsymbol{W}_i \leftarrow \boldsymbol{W}_i \odot \frac{2\boldsymbol{\Psi}_{i-1}^\top \boldsymbol{X}\boldsymbol{H}_l^\top \boldsymbol{\Phi}_{i+1}^\top + 2\lambda_3 \boldsymbol{\Psi}_{i-1}^\top \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{\Psi}_{i-1}\boldsymbol{W}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{\Phi}_{i+1}^\top}{\boldsymbol{\Omega}_i}, \tag{20}$$

$$\begin{aligned} \boldsymbol{\Omega}_i =& \boldsymbol{\Psi}_{i-1}^\top \boldsymbol{\Psi}_{i-1}\boldsymbol{W}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{H}_l\boldsymbol{H}_l^\top \boldsymbol{\Phi}_{i+1}^\top + \boldsymbol{\Psi}_{i-1}^\top \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{\Psi}_{i-1}\boldsymbol{W}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{\Phi}_{i+1}^\top \\ &+ 2\lambda_3^2 \boldsymbol{\Psi}_{i-1}^\top \boldsymbol{\Psi}_{i-1}\boldsymbol{W}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{\Phi}_{i+1}^\top \boldsymbol{W}_i^\top \boldsymbol{\Psi}_{i-1}^\top \boldsymbol{\Psi}_{i-1}\boldsymbol{W}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{\Phi}_{i+1}^\top. \end{aligned} \tag{21}$$

3.4.2. Updating rule of the final representation matrix $\boldsymbol{H}_l$

By fixing all the factors except for $\boldsymbol{H}_l$, the cost function in (14) is reduced to:

$$\begin{aligned} \min_{\boldsymbol{H}_l} \mathcal{L}(\boldsymbol{H}_l) =& \|\boldsymbol{X} - \boldsymbol{\Psi}_l\boldsymbol{H}_l\|_F^2 + \|\boldsymbol{H}_l - \boldsymbol{\Psi}_l^\top \boldsymbol{X}\|_F^2 \\ &+ \lambda_1 \|\boldsymbol{S}^- \odot \boldsymbol{H}_l\boldsymbol{H}_l^\top\|_1 + \lambda_2 \|\boldsymbol{S}^+ \odot \tilde{\boldsymbol{H}}_l\|_1 \quad \text{s.t.} \quad \boldsymbol{H}_l \ge 0. \end{aligned} \tag{22}$$

To solve (22), we introduce a Lagrangian multiplier $\boldsymbol{\Xi}_i$ to enforce the nonnegative constraints on $\boldsymbol{H}_l$, resulting in the following equivalent cost function:

$$\begin{aligned} \min_{\boldsymbol{H}_l, \boldsymbol{\Xi}_l} \mathcal{L}(\boldsymbol{H}_l, \boldsymbol{\Xi}_l) =& \|\boldsymbol{X} - \boldsymbol{\Psi}_l\boldsymbol{H}_l\|_F^2 + \|\boldsymbol{H}_l - \boldsymbol{\Psi}_l^\top \boldsymbol{X}\|_F^2 \\ &+ \lambda_1 \|\boldsymbol{S}^- \odot \boldsymbol{H}_l\boldsymbol{H}_l^\top\|_1 + \lambda_2 \|\boldsymbol{S}^+ \odot \tilde{\boldsymbol{H}}_l\|_1 - \mathrm{Tr}(\boldsymbol{\Xi}_l \boldsymbol{H}_l^\top), \end{aligned} \tag{23}$$

(23) can be further rewritten as follows,

$$\begin{aligned} \min_{\boldsymbol{H}_l, \boldsymbol{\Xi}_l} \mathcal{L}(\boldsymbol{H}_l, \boldsymbol{\Xi}_l) =& \mathrm{Tr}(-2\boldsymbol{X}^\top \boldsymbol{\Psi}_l\boldsymbol{H}_l + \boldsymbol{H}_l^\top \boldsymbol{\Psi}_l^\top \boldsymbol{\Psi}_l\boldsymbol{H}_l + \boldsymbol{H}_l^\top \boldsymbol{H}_l - 2\boldsymbol{H}_l^\top \boldsymbol{\Psi}_l^\top \boldsymbol{X}) \\ &+ \mathrm{Tr}(\lambda_1 \boldsymbol{H}_l\boldsymbol{S}^- \boldsymbol{H}_l^\top) + 2\mathrm{Tr}(\lambda_2 \boldsymbol{H}_l\boldsymbol{D}^+ \boldsymbol{H}_l^\top) \\ &- \mathrm{Tr}(2\lambda_2 \boldsymbol{H}_l\boldsymbol{S}^+ \boldsymbol{H}_l^\top) - \mathrm{Tr}(\boldsymbol{\Xi}_l \boldsymbol{H}_l^\top) \end{aligned} \tag{24}$$

where $\boldsymbol{D}^+$ is a diagonal matrix with diagonal entry D_{ii}^+, which is the sum of ith row of $\boldsymbol{S}^+$ ($D_{ii}^+ = \sum_j S_{ij}^+$). By setting the partial derivative of $\mathcal{L}(\boldsymbol{H}_l, \boldsymbol{\Xi}_l)$ with respect to $\boldsymbol{H}_l$ to $\mathbf{0}$, we obtain:

$$\boldsymbol{\Xi}_l = -4\boldsymbol{\Psi}_l^\top \boldsymbol{X} + 2\boldsymbol{\Psi}_l^\top \boldsymbol{\Psi}_l\boldsymbol{H}_l + 2\boldsymbol{H}_l + 2\lambda_1 \boldsymbol{H}_l\boldsymbol{S}^- + 4\lambda_2 \boldsymbol{H}_l\boldsymbol{D}^+ - 4\lambda_2 \boldsymbol{H}_l\boldsymbol{S}^+. \tag{25}$$

From the complementary slackness condition of the Karush–Kuhn–Tucker (KKT), we have:

$$\boldsymbol{\Xi}_l \odot \boldsymbol{H}_l = \mathbf{0}. \tag{26}$$

Following similar derivation process of the updating rule for $\boldsymbol{W}_i$, the updating rule for $\boldsymbol{H}_l$ is obtained as follows:

$$\boldsymbol{H}_l \leftarrow \boldsymbol{H}_l \odot \frac{2\boldsymbol{\Psi}_l^\top \boldsymbol{X} + 2\lambda_2 \boldsymbol{H}_l\boldsymbol{S}^+}{\boldsymbol{\Psi}_l^\top \boldsymbol{\Psi}_l\boldsymbol{H}_l + \boldsymbol{H}_l + \lambda_1 \boldsymbol{H}_l\boldsymbol{S}^- + 2\lambda_2 \boldsymbol{H}_l\boldsymbol{D}^+}. \tag{27}$$

3.4.3. Updating rule of the representation matrix $\boldsymbol{H}_i$ $(1 \le i < l)$

The updating of $\boldsymbol{H}_i$ is optional, since it does not significant affect the cost function in (14). However, we want to extract the latent features in each layer. To optimize $\boldsymbol{H}_i$, we search to minimize the following cost function:

$$\min_{\boldsymbol{H}_i} \mathcal{L}(\boldsymbol{H}_i) = \|\boldsymbol{X} - \boldsymbol{\Psi}_i\boldsymbol{H}_i\|_F^2 + \|\boldsymbol{H}_i - \boldsymbol{\Psi}_i^\top \boldsymbol{X}\|_F^2, \text{ s.t.} \quad \boldsymbol{H}_i \ge 0. \tag{28}$$

Similar to $\boldsymbol{H}_l$, $\boldsymbol{H}_i$ can be updated by

$$\boldsymbol{H}_i \leftarrow \boldsymbol{H}_i \odot \frac{2\boldsymbol{\Psi}_i^\top \boldsymbol{X}}{\boldsymbol{\Psi}_i^\top \boldsymbol{\Psi}_i\boldsymbol{H}_i + \boldsymbol{H}_i}. \tag{29}$$

Until now, we have all the updating rules done. The overall optimization process of DANMF-CRFR is outlined in Algorithm 1, where the *ShallowANMF* procedure runs the pre-training as described in the beginning of this section.

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

Dataset	#sample	#feature	#class	Layers	Application
Iris	150	4	3	60–50–40	Biology
Glass	214	9	6	60–50–40	Criminological investigation
Wine	178	13	3	60–50–40	Chemistry
Zoo	101	16	17	60–50–40	Animal science
Libras Movement	360	90	15	60–50–40	Hand movement recognition
MNIST	1000	784	10	120–100–80	Handwriting recognition
FashionMNIST	1000	784	10	120–100–80	Cloth recognition
Yale	165	1024	15	120–100–80	Face recognition
ORL	400	1024	40	120–100-80	Face recognition
Coil20	1440	1024	20	120–100–80	Object recognition

4. Experimental results

In this section, the experimental setup, analysis of the results, and parameters setting are discussed in detail. Extensive experiments are conducted to validate the effectiveness of the proposed model in comparison to twelve baseline and state-of-the-art models on ten benchmark datasets.

To obtain the best performance of the proposed and competing methods, the best values of corresponding parameters are determined through the grid search technique. Some of the parameters related to the proposed method are, the repulsive regularization parameter λ_1, the attractive regularization parameter λ_2, and the feature relationship parameter λ_3. The number of deep model layers for different datasets usually is set to three; however, the size of the layers is different for each dataset. For the sake of clarity and simplicity, the layer size configuration of DANMF-CRFR for all datasets is shown in Table 1. The shallow autoencoder-like method is used to pre-training each layer of our proposed method hierarchically. Then, the obtained initial matrices are used in the fine-tuning process to achieve the final optimal matrices. In addition, we set the number of pre-training iterations to 500 and the number of fine-tuning iterations to 1000. After satisfying the convergence condition, we utilize the obtained feature representation at the top-layer $\boldsymbol{H}_l$ of the autoencoder model and employ the standard k-means clustering algorithm to obtain the final result of DANMF-CRFR. For simplicity and to reduce the sensitivity of the bandwidth parameter σ, we limited the number of nearest neighbors to $p = Log(n) + 1$, where n is the number of samples. Also, we set the bandwidth parameter to 1000 in our experiments.

The proposed method is compared with some baseline and state-of-the-art methods that have applied a clustering algorithm on their extracted top-level hidden latent, to achieve the best clustering performance considering NMI, ACC, and ARI measures. For each approach, we execute it ten times with various initial values and report the average performance and standard deviation. The suggested method is implemented in *Python* and tested on a workstation equipped with an *Intel(R) i5-5500* CPU running at 2.20 GHz with two cores and 8 Gigabytes of memory.

4.1. Datasets

To analyze and investigate the performance of the suggested clustering model, we conduct experiments on five numerical and five image datasets, which are widely used to evaluate the performance of matrix factorization models and their modifications. The detailed information of the datasets is described in the following, and summarized in Table 1. Also, Fig. 3 shows sample images from MNIST, Fashion-MNIST, Yale, ORL, and Coil20 datasets.

Yale: The Yale face dataset (Belhumeur et al., 1997) consists of 15 different individuals, each of which is taken with 11 images and captured under different conditions, including center-light, with glasses, happy, left-light, without glasses, normal, right-light, sad, sleepy, surprised, and wink. Each image is scaled to 32 × 32 in our experiments.

Coil20: This dataset (Nene, Nayar, Murase, et al., 1996) is created by Columbia University and consists of a 1440 gray image of 20 objects, and each object has 72 images with 32 × 32 dimensions. Images are taken from five degrees apart and are rotated on a turntable.

ORL: There are 400 faces corresponding to 40 people's faces. Each person has 10 images in the ORL dataset. The images change in pose and expression (open or closed eyes, smiling or not). The dimension of each image is 30 × 25 (Samaria & Harter, 1994).

MNIST: The MNIST dataset (LeCun, 1998) contains handwritten digit images which were taken from American high school students, and images are formatted in 28 × 28 pixels value with a grayscale format. It includes 70,000 images from numbers 0 to 9. We choose 100 random samples from each class to create a new dataset with 1000 digit images.

Fashion-MNIST: The Fashion-MNIST (Xiao, Rasul, & Vollgraf, 2017) is a dataset that consists of 70,000 samples that form a collection of shirt, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, and ankle boot images. Each sample is a 28 × 28 grayscale image, associated with a label from 10 classes. We choose 100 random samples from each class to create a new dataset with 1000 images.

Wine: The Wine dataset (Asuncion & Newman, 2007) contains data from wine chemical analysis. It includes 178 instances of 13 features each that are classified into 3 classes.

Iris: The Iris dataset (Asuncion & Newman, 2007) includes 3 classes of 50 samples, where each class refers to a kind of Iris plant. Each sample has 4 attributes. This popular dataset has been used in several clustering problems.

Glass: The Glass dataset (Asuncion & Newman, 2007) is about several types of glass that has totally 214 samples in 6 classes with 9 features.

Libras Movement: The Libras Movement dataset (Asuncion & Newman, 2007) contains 15 classes with 90 features and 24 instances per class (total 360) where each class represents a hand movement type.

Zoo: The zoo dataset (Asuncion & Newman, 2007) consists of 101 samples with 16 Boolean features, which is a collection of statistical information about animals.

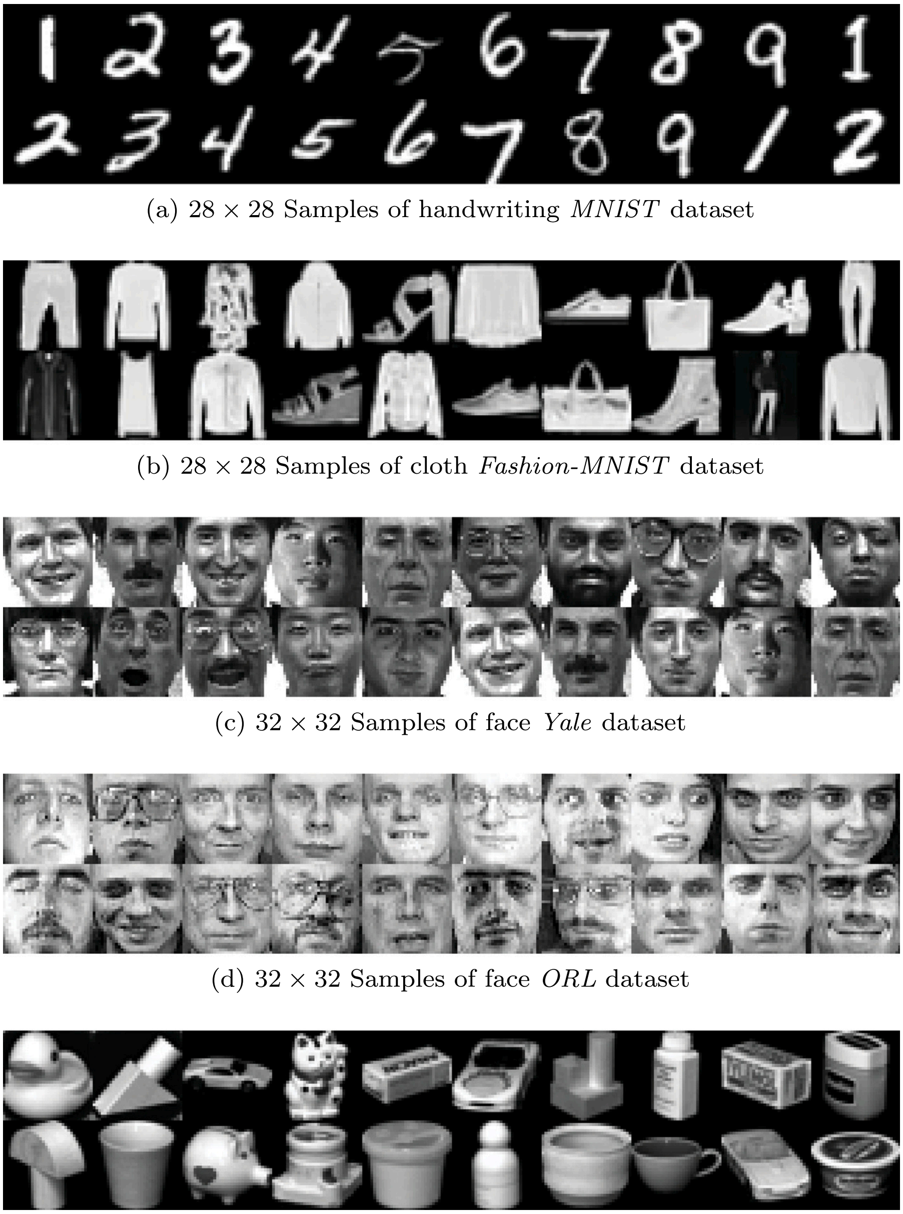
4.2. Evaluation metrics

In this section, we introduce three widely used quantitative metrics, Normalized Mutual Information (NMI), Accuracy (ACC), and Adjust Rand Index (ARI), to evaluate the clustering performance. The NMI measures the shared information between two statistical distributions, which is defined as

$$NMI(c, y) = \frac{MI(c, y)}{max(H(c), H(y))}. \tag{30}$$

This formula can provide a degree of agreement for two clustering results where c and y indicate the predicted labels and their ground truth clusters, respectively. H is the entropy function, and MI is the mutual information. The NMI is equal to 1 if the ground truth clusters and corresponding predicted labels are similar, and is close to 0 if they are mostly different. We used the implementation provided by the Python library *scikit-learn* (Pedregosa et al., 2011).

The ACC criterion indicates the percentage of data points for which the produced clusters can be successfully mapped to ground-truth



(a) 28 × 28 Samples of handwriting *MNIST* dataset

(b) 28 × 28 Samples of cloth *Fashion-MNIST* dataset

(c) 32 × 32 Samples of face *Yale* dataset

(d) 32 × 32 Samples of face *ORL* dataset

(e) 32 × 32 Samples of object *Coil20* dataset

Fig. 3. Five grayscale image datasets.

classes using the Hungarian algorithm (Kuhn, 1955). It is specifically defined as follows:

$$ACC(c,y)=\frac{\sum_{i=1}^{n}\delta(map(c_i),y_i)}{n}, \tag{31}$$

where n is the number of all data samples, y_i is a ground truth label, $\bar{y}_i = map(c_i)$ is the optimum matching function that can permute all clustering results in order to get the best mapping between clustering labels and actual labels, and $\delta(\cdot,\cdot)$ is the delta function that equals 1 if $y_i = \bar{y}_i$ and equals to 0 otherwise.

The Adjusted Rand Index (ARI) is a measure that evaluates the similarity between two data clusters (Hubert & Arabie, 1985). It is equal to 0 if the correspondence between two classes is less than what would be predicted by chance, and it would be 1 if the clusters are identical. ARI score may be negative when the relationship is less than what would be predicted by chance. Its equation is defined as follows:

$$ARI(c,y)=\frac{\sum_{i,j}\binom{n_{ij}}{2}-\sum_i\binom{n_{i.}}{2}\sum_j\binom{n_{.j}}{2}/\binom{n}{2}}{1/2[\sum_i\binom{n_{i.}}{2}+\sum_j\binom{n_{.j}}{2}]-\sum_i\binom{n_{i.}}{2}\sum_j\binom{n_{.j}}{2}/\binom{n}{2}}, \tag{32}$$

where c stands for the clustering results and y stands for the ground-truth clustering labels. n_{ij} is the number of identical samples in both cluster c_i and cluster y_j, and $n_{i.}$ and $n_{.j}$ are the number of identical samples in the cluster c_i and cluster y_j, respectively. Notably, the ARI is a chance-corrected form of the RI that is employed as an external criterion for comparing clustering results, has a range of $[0 \quad 1]$, while ARI has a range of $[-1 \quad 1]$ (Hubert & Arabie, 1985).

4.3. Compared methods

To verify the superior performance of the DANMF-CRFR method for data representation, we consider the other 12 algorithms consisting of baselines, state-of-the-art shallow, and deep NMF methods as the comparison methods, which are listed as follows:

NMF: NMF (Lee & Seung, 1999) seeks to find two nonnegative matrices, which their product is an approximation of the observation data matrix.

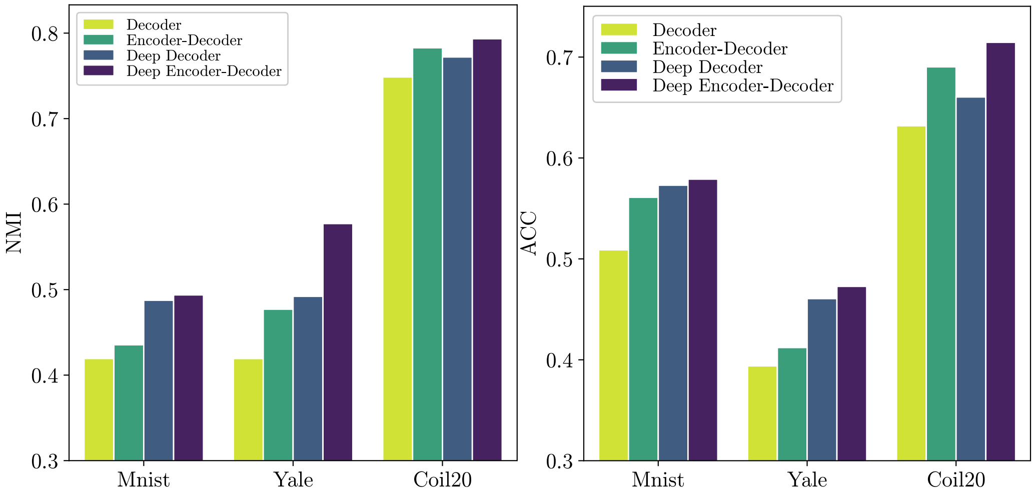


Fig. 4. Data representation results of the underlying models on three high-dimensional datasets in terms of NMI and ACC.

ONMF-W and **ONMF-H**: By exploring the gradient relation between the Euclidean space and Stiefel manifold, Choi (2008) defines a multiplicative update rule to optimize the ONMF-W model, which includes an orthogonal constraint on the basis matrix, and similarly ONMF-H model imposes orthogonality constraint on the coefficient matrix.

ONMTF: Orthogonal Nonnegative Matrix Tri-Factorization (ONMTF) reconstructs the input matrix by three factors and imposes Bi-orthogonal constraints on the basis and coefficient factors (Ding, Li, Peng and Park, 2006).

Semi-NMF: It is an extended version of the NMF which maintains the same concepts of NMF (Ding et al., 2010). Its representation matrix is still restricted to be nonnegative while placing no restriction on the signs of the basis matrix. It lends itself to a convenient clustering interpretation.

GNMF: Graph regularized NMF (Cai et al., 2011) as a variation of NMF aims to involve the geometrical structure of the data space by utilizing a Laplacian graph regularization.

NMF-FRP: This method is a constrained NMF method (Hedjam et al., 2021), which adds a regularization term to basic NMF to preserve the scaling scatter of data and its centroids

Deep Semi-NMF: As an extension of the shallow-layered Semi-NMF model, Trigeorgis et al. (2014) presented a deep structure of the semi-nonnegative matrix factorization model. It aims to explore hidden representations of data with complex structure.

DANMF: By integrating the encoder and decoder component with the NMF framework, Ye et al. (2018) introduced a DANMF model, for which the efficiency of the model is shown to deal with community detection tasks.

DNBMF and **RDNBMF**: DNBMF multi-layer approach is built hierarchically by further factorizing basis matrix $\boldsymbol{W}$ instead of coefficient matrix $\boldsymbol{H}$ (Zhao et al., 2021). Moreover, a regularized variant called RDNBMF is also proposed in this paper, which includes sparseness constraint on each basis vector which reflects distinct parts-based features.

DAutoED-ONMF: This method is an extension of the DANMF and introduces a deep autoencoder-like structure for ONMF (Yang & Xu, 2021). It applied explicit orthogonality constraints on the representation matrix.

4.4. Comparing underlying models

In this subsection, the performance of the underlying model of the proposed method (i.e., Deep Encoder–Decoder NMF) is compared with the three other underlying models. The methods are the standard (Decoder) NMF, Encoder–Decoder NMF, deep Decoder NMF, and deep Decoder–Encoder NMF. Standard NMF considers only the loss function of the decoder component, and the Encoder–Decoder NMF structure integrates the decoder and encoder into a unified loss function. The third structure is a deep version of decoder NMF, and finally, the Deep Encoder–Decoder structure deeps the shallow encoder–decoder NMF by a hierarchical structure, which can extract complex hierarchical and structural information with implicit lower-level hidden features. As shown in Fig. 4, two evaluation metrics (i.e., ACC and NMI) are employed to compare four underlying structures for analyzing their performance in three datasets. From the obtained results, the shallow Encoder–Decoder structure method performs better than the shallow decoder or standard NMF method, and it shows the effectiveness of considering the encoder architecture. Moreover, the Deep decoder framework, in comparison with shallow models, almost performs better due to its capabilities to explore and discover informative and hierarchical features in complex data. However, the Deep Decoder–Encoder framework always performs superior to other frameworks, because this structure, in addition to benefits characteristics of Deep structure, can capture the representation better. Since the encoder verifies and refines the decoder by attempting to retrain the representation.

4.5. Results

Tables 2–4 demonstrate the results of clustering on five image datasets and five numerical datasets for all algorithms in terms of NMI, ACC, and ARI measures. The results show that the proposed method achieves the best results in the majority of datasets. According to these tables, DANMF-CRFR is ranked first in 23 cases out of 30, and for the rest of the cases, it is placed as the second-best. It can be derived that, in comparison with other methods, our method is applicable to a wide range of datasets. We listed the characteristics of the proposed model based on the experimental data as follows:

- From the results, we find that the proposed DANMF-CRFR method, multi-layers (Zhao et al., 2021), and deep methods (Trigeorgis et al., 2014; Yang & Xu, 2021; Ye et al., 2018) almost performed better than single-layer methods. This advantage can be attributed to the better ability of deep methods to discover and extract hidden and complex features of data such as images.

Table 2
Experimental results (mean NMI ± standard deviation) on the 10 datasets for data representation. The best results are highlighted in bold and the second-best results are highlighted in italic.

Method	Iris	Wine	Glass	Zoo	Movement	ORL	MNIST	Yale	Coil20	Fashion
NMF	0.582 ±0.040	0.638 ±0.010	0.355 ±0.014	0.768 ±0.028	0.439 ±0.027	0.626 ±0.024	0.365 ±0.012	0.402 ±0.013	0.616 ±0.015	0.523 ±0.015
ONMF-W	0.714 ±0.030	0.413 ±0.020	0.378 ±0.031	0.725 ±0.028	0.336 ±0.011	0.328 ±0.024	0.251 ±0.024	0.265 ±0.022	0.452 ±0.023	0.266 ±0.015
ONMF-H	0.763 ±0.060	0.721 ±0.010	0.363 ±0.034	0.841 ±0.036	0.573 ±0.023	0.728 ±0.012	0.452 ±0.013	0.470 ±0.034	0.713 ±0.011	0.592 ±0.026
ONMTF	0.636 ±0.054	0.680 ±0.040	0.321 ±0.031	0.630 ±0.035	0.517 ±0.016	0.423 ±0.021	0.410 ±0.024	0.420 ±0.025	0.650 ±0.014	0.514 ±0.009
Semi-NMF	0.736 ±0.042	0.662 ±0.090	0.275 ±0.060	0.748 ±0.032	0.527 ±0.009	0.730 ±0.007	0.410 ±0.017	0.462 ±0.018	0.665 ±0.024	0.525 ±0.010
GNMF	0.664 ±0.055	0.631 ±0.040	0.393 ±0.022	**0.908** ±0.000	0.592 ±0.016	0.781 ±0.010	0.456 ±0.017	0.501 ±0.000	0.740 ±0.012	0.540 ±0.021
NMF-FRP	0.702 ±0.051	0.643 ±0.020	0.362 ±0.026	0.730 ±0.036	0.464 ±0.017	0.667 ±0.000	0.392 ±0.026	0.420 ±0.026	0.608 ±0.017	0.484 ±0.016
Deep Semi-NMF	0.741 ±0.015	0.716 ±0.000	0.355 ±0.011	0.765 ±0.000	0.532 ±0.020	0.705 ±0.021	0.413 ±0.012	0.481 ±0.021	0.682 ±0.012	0.534 ±0.006
DANMF	0.740 ±0.000	0.816 ±0.012	0.365 ±0.021	0.815 ±0.026	0.601 ±0.016	0.770 ±0.000	0.460 ±0.000	0.522 ±0.025	0.801 ±0.014	0.540 ±0.015
DNBMF	0.753 ±0.061	0.756 ±0.030	0.355 ±0.033	0.786 ±0.031	0.592 ±0.005	0.782 ±0.013	0.477 ±0.018	0.503 ±0.030	0.803 ±0.016	0.531 ±0.023
RDNBMF	0.751 ±0.053	0.823 ±0.020	0.340 ±0.002	0.793 ±0.000	0.605 ±0.006	0.791 ±0.011	0.490 ±0.014	0.512 ±0.023	0.790 ±0.019	0.542 ±0.000
DAutoED-ONMF	0.833 ±0.006	**0.853** ±0.020	0.342 ±0.046	0.824 ±0.024	0.587 ±0.022	0.728 ±0.007	0.462 ±0.017	0.495 ±0.020	0.742 ±0.023	0.544 ±0.008
DANMF-CRFR	**0.949** ±0.001	0.840 ±0.028	**0.407** ±0.003	**0.908** ±0.001	**0.633** ±0.006	**0.806** ±0.022	**0.532** ±0.013	**0.549** ±0.012	**0.831** ±0.013	**0.612** ±0.008

Table 3
Experimental results (mean ACC ± standard deviation) on the 10 datasets for data representation. The best results are highlighted in bold and the second-best results are highlighted in italic.

Method	Iris	Wine	Glass	Zoo	Movement	ORL	MNIST	Yale	Coil20	Fashion
NMF	0.716 ±0.036	0.632 ±0.021	0.457 ±0.027	0.803 ±0.003	0.352 ±0.027	0.405 ±0.024	0.464 ±0.034	0.357 ±0.030	0.471 ±0.017	0.550 ±0.021
ONMF-W	0.675 ±0.005	0.595 ±0.007	0.509 ±0.013	0.780 ±0.014	0.267 ±0.002	0.122 ±0.017	0.357 ±0.013	0.220 ±0.017	0.241 ±0.033	0.295 ±0.019
ONMF-H	0.846 ±0.007	0.904 ±0.005	0.433 ±0.029	0.895 ±0.021	0.465 ±0.018	0.580 ±0.033	0.561 ±0.020	0.409 ±0.350	0.590 ±0.027	**0.623** ±0.020
ONMTF	0.736 ±0.046	0.821 ±0.113	0.368 ±0.012	0.766 ±0.034	0.424 ±0.025	0.144 ±0.010	0.493 ±0.017	0.364 ±0.120	0.547 ±0.033	0.526 ±0.019
Semi-NMF	0.848 ±0.090	0.843 ±0.115	0.472 ±0.031	0.827 ±0.027	0.444 ±0.005	0.563 ±0.016	0.514 ±0.035	0.420 ±0.030	0.538 ±0.029	0.562 ±0.018
GNMF	0.821 ±0.067	0.830 ±0.024	0.506 ±0.014	**0.929** ±0.000	0.463 ±0.011	0.672 ±0.009	0.543 ±0.220	0.448 ±0.189	0.645 ±0.015	0.600 ±0.048
NMF-FRP	0.832 ±0.053	0.852 ±0.041	0.498 ±0.005	0.821 ±0.043	0.368 ±0.012	0.342 ±0.014	0.449 ±0.027	0.367 ±0.025	0.455 ±0.019	0.501 ±0.021
Deep Semi-NMF	0.913 ±0.000	0.903 ±0.002	0.521 ±0.007	0.913 ±0.000	0.465 ±0.012	0.560 ±0.018	0.499 ±0.017	0.481 ±0.015	0.637 ±0.019	0.557 ±0.005
DANMF	0.886 ±0.000	0.941 ±0.008	0.529 ±0.003	0.906 ±0.011	0.487 ±0.012	0.620 ±0.010	0.542 ±0.015	0.476 ±0.024	0.722 ±0.012	0.611 ±0.015
DNBMF	0.892 ±0.036	0.924 ±0.016	0.530 ±0.021	0.889 ±0.017	0.476 ±0.007	0.605 ±0.022	0.570 ±0.025	0.455 ±0.045	0.702 ±0.019	0.567 ±0.032
RDNBMF	0.890 ±0.015	0.942 ±0.007	0.534 ±0.007	0.891 ±0.000	0.480 ±0.008	0.643 ±0.023	0.577 ±0.027	0.458 ±0.019	0.719 ±0.023	0.602 ±0.013
DAutoED-ONMF	0.947 ±0.002	**0.957** ±0.007	0.541 ±0.014	0.897 ±0.022	0.480 ±0.022	0.572 ±0.015	0.557 ±0.022	0.472 ±0.025	0.677 ±0.029	0.581 ±0.016
DANMF-CRFR	**0.986** ±0.001	0.949 ±0.011	**0.570** ±0.002	0.920 ±0.002	**0.494** ±0.008	**0.675** ±0.023	**0.595** ±0.018	**0.490** ±0.023	**0.733** ±0.018	0.619 ±0.011

Table 4
Experimental results (mean ARI ± standard deviation) on the 10 datasets for data representation. The best results are highlighted in bold and the second-best results are highlighted in italic..

Method	Iris	Wine	Glass	Zoo	Movement	ORL	MNIST	Yale	Coil20	Fashion
NMF	0.550 ±0.070	0.490 ±0.202	0.223 ±0.022	0.778 ±0.025	0.180 ±0.035	0.216 ±0.019	0.232 ±0.190	0.137 ±0.020	0.398 ±0.250	0.379 ±0.018
ONMF-W	0.679 ±0.015	0.578 ±0.007	0.252 ±0.015	0.683 ±0.091	0.115 ±0.011	0.071 ±0.003	0.135 ±0.018	0.093 ±0.011	0.169 ±0.032	0.132 ±0.013
ONMF-H	0.640 ±0.014	0.734 ±0.006	0.227 ±0.032	0.645 ±0.061	0.284 ±0.027	0.329 ±0.027	0.326 ±0.020	0.170 ±0.018	0.504 ±0.018	0.403 ±0.025
ONMTF	0.540 ±0.046	0.640 ±0.090	0.188 ±0.023	0.427 ±0.042	0.247 ±0.017	0.059 ±0.008	0.294 ±0.021	0.142 ±0.016	0.361 ±0.028	0.375 ±0.018
Semi-NMF	0.674 ±0.112	0.649 ±0.138	0.138 ±0.064	0.691 ±0.087	0.253 ±0.012	0.356 ±0.014	0.269 ±0.029	0.181 ±0.014	0.429 ±0.039	0.380 ±0.014
GNMF	0.540 ±0.020	0.646 ±0.080	0.235 ±0.000	0.950 ±0.000	0.304 ±0.020	**0.500** ±0.020	0.345 ±0.033	0.234 ±0.025	0.543 ±0.015	0.394 ±0.028
NMF-FRP	0.615 ±0.071	0.650 ±0.072	0.235 ±0.029	0.652 ±0.035	0.203 ±0.010	0.259 ±0.002	0.271 ±0.027	0.140 ±0.010	0.345 ±0.025	0.314 ±0.017
Deep Semi-NMF	0.770 ±0.000	0.721 ±0.006	0.191 ±0.013	0.770 ±0.000	0.237 ±0.024	0.309 ±0.036	0.302 ±0.014	0.203 ±0.027	0.486 ±0.020	0.350 ±0.007
DANMF	0.716 ±0.000	0.807 ±0.018	0.235 ±0.001	0.673 ±0.052	0.316 ±0.020	0.408 ±0.014	0.288 ±0.015	**0.287** ±0.027	0.625 ±0.024	0.404 ±0.003
DNBMF	0.732 ±0.081	0.782 ±0.042	0.203 ±0.030	0.648 ±0.066	0.304 ±0.012	0.457 ±0.038	0.323 ±0.025	0.203 ±0.021	0.628 ±0.027	0.372 ±0.033
RDNBMF	0.725 ±0.032	0.834 ±0.036	0.178 ±0.007	0.602 ±0.000	0.318 ±0.003	0.468 ±0.019	0.334 ±0.032	0.243 ±0.048	0.631 ±0.027	0.394 ±0.010
DAutoED-ONMF	0.854 ±0.006	**0.872** ±0.022	0.200 ±0.041	0.750 ±0.078	0.317 ±0.028	0.350 ±0.019	0.327 ±0.020	0.212 ±0.020	0.540 ±0.031	0.368 ±0.012
DANMF-CRFR	**0.960** ±0.000	0.852 ±0.031	**0.253** ±0.002	**0.951** ±0.003	**0.333** ±0.004	0.475 ±0.011	**0.382** ±0.022	0.269 ±0.020	**0.653** ±0.023	**0.412** ±0.020

- According to the obtained results, compared to other deep methods that only consider the error of the decoder part, including Deep Semi-NMF, DNBMF, and RDNBMF, the deep encoder–decoder-based methods present a better performance in terms of clustering. This performance is because of consideration of both encoder and decoder structure into an integrated loss function with better inheriting the learning capability of deep autoencoder structure. This structure leads to learning the valuable feature of our method for input data in an unsupervised way.
- Our method, compared to other deep autoencoder-like methods such as DANMF and Dauto-ONMF, almost outperforms across different evaluation clustering metrics. The reason for achieving such a good result, in addition to considering the local structure of the data and orthogonality constraint, is considering the global structure and maintaining the feature relationship on each layer.
- According to the results on the small-scale datasets such as Glass and Zoo, we can see that Deep and Multi-layer methods have not always performed better in comparison to some shallow methods such as GNMF and ONMF-H. Nevertheless, our proposed method almost outperforms in comparison with all deep, multi-layer, and shallow methods, which indicates that our well-regularized method gives more flexibility in controlling the capacity of the model.

4.6. *Parameters selection*

In this subsection, we will evaluate the effects of the parameters in the proposed algorithm. However, the tuning parameter for the unsupervised learning method is usually a difficult task. Our proposed method has three main parameters, the repulsive graph regularization parameter λ_1, the attractive graph regularization parameter λ_2, and the deep feature relationship preservation parameter λ_3. We conduct the results of NMI and ACC measures on the four evaluated datasets to demonstrate the clustering performance with respect to the variation of parameter values. Fig. 5 demonstrates the NMI and ACC values of our proposed model with different λ_1, λ_2, and λ_3 upon the Iris, Wine, Yale, and MNIST datasets. Note that, This figure is 4D, i.e., three axes correspond to λ_1, λ_2, and λ_3, and the fourth dimension is reflected by the color of the points.

4.6.1. *Selection of parameter λ_1*

This parameter determines the repulsion rate for our model, and our selected values for analyzing this parameter on the four datasets are $\{0, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}\}$. According to the results, it can be concluded that the optimal value for this parameter is usually 10^{-4} or 10^{-3}. For some image datasets like MNIST optimal value for λ_1 is 0.

4.6.2. *Selection of parameter λ_2*

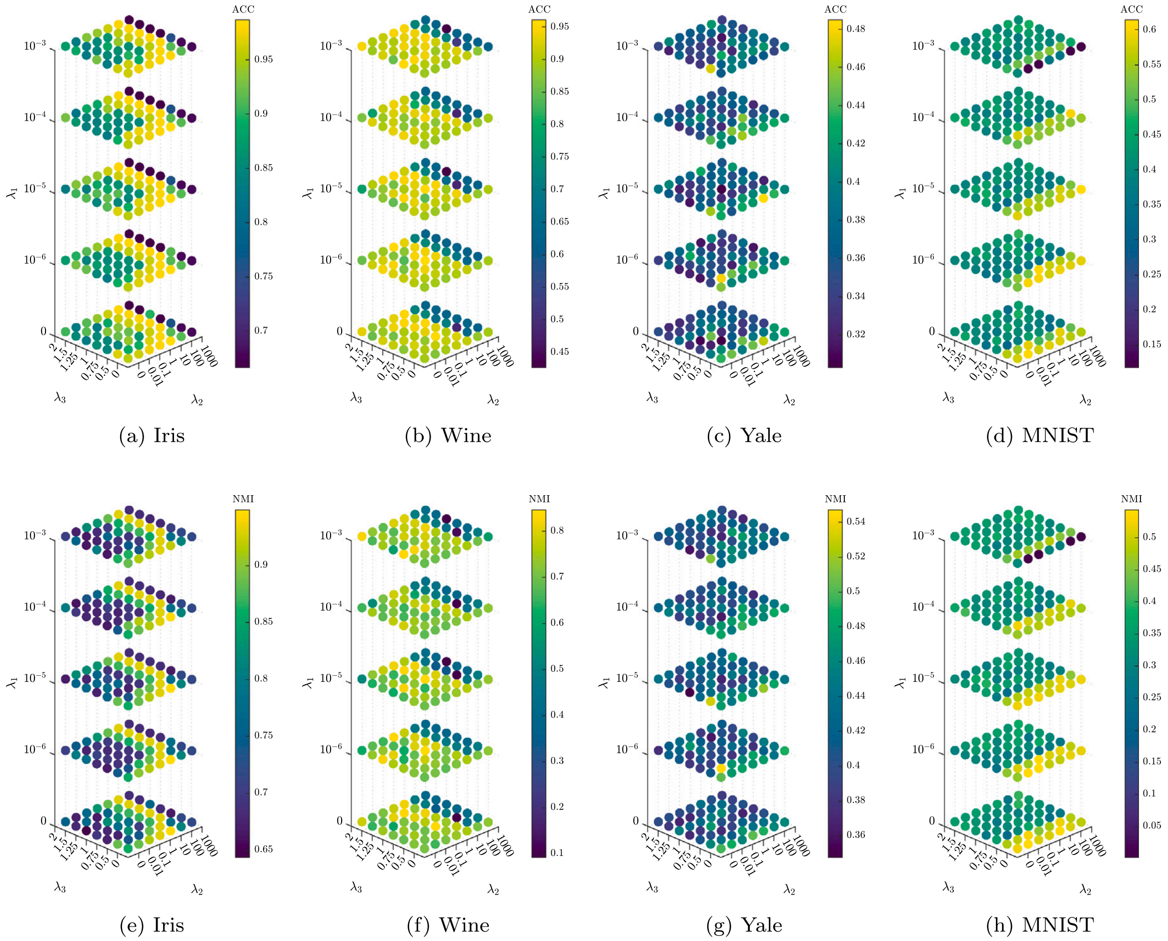
This parameter controls attractive regularization, and the values set for λ_2 parameter are $\{0, 0.01, 0.1, 1, 10, 100, 1000\}$. As we can observe in Fig. 5, λ_2 with large values usually has a better performance in terms of NMI and ACC measures on the four datasets. Values close to zero or very large ones for this parameter may not perform relatively well.

4.6.3. *Selection of parameter λ_3*

In our method, λ_3 controls the feature relationships between original and latent spaces. Feature relationship parameter λ_3 is tuned in the range of $\{0, 0.5, 0.75, 1, 1.25, 1.5, 2\}$. According to the obtained results, λ_3 is a sensitive parameter that usually has to be adjusted carefully. The optimal range is between 0.5 and 1.5, and the proposed method can achieve good clustering performance on the four datasets by selecting λ_3 in this range.

4.7. *Ablation study*

To analyze the influence of repulsive, attractive, and Feature Relationship Preservation (FRP) regularization terms of DANMF-CRFR, we further conduct ablation experiments on all datasets based on NMI, ACC, and ARI evaluation measures. In this analysis, we define five evaluation cases to show the effectiveness of each part of our model.



(a) Iris (b) Wine (c) Yale (d) MNIST

(e) Iris (f) Wine (g) Yale (h) MNIST

Fig. 5. Parameter Analysis of the proposed algorithm with respect to the parameters λ_1, λ_2 and λ_3 on four real-world datasets.

Table 5
Ablation study of the effect of regularization terms on proposed method, while **bold** show the best performance and underline style indicates the equal performance.

	Case	Dataset									
		Iris	Wine	Glass	Zoo	Movement	ORL	MNIST	Yale	Coil20	Fashion-MNIST
NMI	Basic	0.886	0.707	0.270	0.870	0.596	0.693	0.374	0.458	0.631	0.588
	+Repulsive	0.914	0.720	0.334	0.891	0.610	0.713	0.413	0.510	0.663	0.598
	+Attractive	0.914	0.752	0.393	0.893	0.621	0.793	0.497	0.504	**0.831**	0.592
	+FRP	0.899	0.828	0.309	0.888	0.610	0.711	0.424	0.477	0.653	0.596
	Final	**0.949**	**0.840**	**0.407**	**0.908**	**0.633**	**0.806**	**0.532**	**0.549**	**0.831**	**0.612**
ACC	Basic	0.960	0.893	0.542	0.900	0.459	0.515	0.452	0.375	0.457	0.585
	+Repulsive	0.973	0.895	0.557	0.912	0.481	0.540	0.477	0.454	0.491	0.605
	+Attractive	0.973	0.910	0.517	0.920	0.476	0.640	0.538	0.448	**0.733**	0.592
	+FRP	0.966	0.952	0.548	0.912	0.490	0.536	0.499	0.406	0.461	0.594
	Final	**0.986**	**0.957**	**0.570**	**0.920**	**0.494**	**0.675**	**0.595**	**0.490**	**0.733**	**0.619**
ARI	Basic	0.885	0.697	0.166	0.908	0.300	0.281	0.201	0.190	0.382	0.365
	+Repulsive	0.922	0.703	0.214	0.936	0.320	0.318	0.245	0.252	0.439	0.384
	+Attractive	0.922	0.741	0.244	0.945	0.311	0.414	0.313	0.242	**0.653**	0.366
	+FRP	0.903	0.852	0.195	0.935	0.314	0.320	0.259	0.201	0.416	0.369
	Final	**0.960**	**0.870**	**0.253**	**0.951**	**0.333**	**0.475**	**0.382**	**0.269**	**0.653**	**0.412**

In Table 5, the Basic case means DANMF-CRFR without any regularization, +Repulsive case means the Basic model with the repulsive term, +Attractive case means the Basic model with the attractive term, and +FRP case means the Basic model with FRP term. The Final case means the Basic model with three regularization terms (i.e., DANMF-CRFR). From this table, we can see that +Repulsive, +Attractive, and +FRP cases produce better performances than the Basic case on all datasets, demonstrating each regularization's effectiveness. On Yale and Fashion-MNIST, the +Repulsive case has higher performance than the +Attractive and +FRP cases, while on Glass, Zoo, Movement, ORL, MNIST, and Coil20, +Attractive produces higher performance, and +FRP has better results only on the Wine dataset. Eventually, based on the Final case, it is evident that combining all regularizations achieved the best performance. We can conclude that all three regularizers are important to our model and complement each other.

4.8. Convergence curves

In this section, the convergence of our proposed method for three numerical and image datasets is represented. In Fig. 6, it is shown that

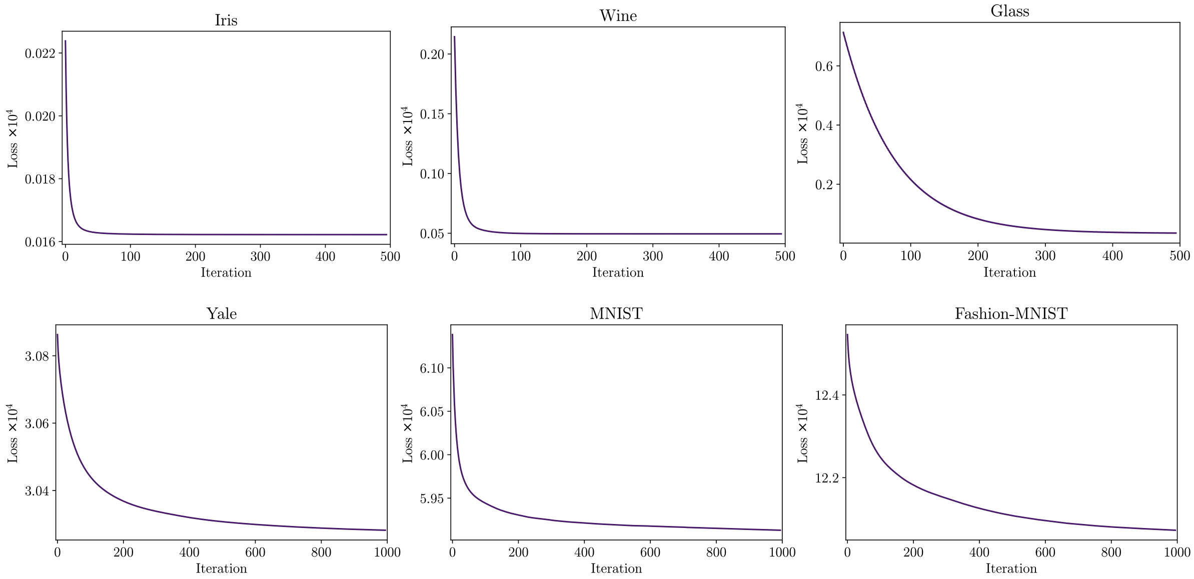


Fig. 6. Convergence of DANMF-CRFR on the numerical and image datasets.

the objective function of this method is reduced uniformly under the update rules represented in Section 3.4. The X-axis and the Y-axis in the diagrams drawn in this figure show the number of iterations and the objective value of the function, respectively. Therefore, from the results obtained for the convergence of the proposed method on different datasets, it can be concluded that this method is convergent. From Fig. 6, we can derive that convergence of model on the small-scale datasets with a simple structure (such as Iris and Wine) occurred faster than on high-dimensional datasets.

5. Conclusion

In this paper, a novel data representation model, called Deep Autoencoder-like Nonnegative Matrix Factorization with Contrastive Regularization and Feature Relationship preservation (DANMF-CRFR), is introduced to tackle some of the shortcomings of previous Deep NMF methods. DANMF-CRFR learns the instructive hierarchical features by performing double graph regularization in a contrastive manner, preserving the intrinsic local and global geometrical structures of the input data while mining the data information in the representation space. Meanwhile, during the encoder–decoder learning, we introduce a deep feature relationship preservation regularization that reconciles the low-dimensional feature correlation to the high-dimensional feature correlation to improve the part-based learning capabilities of this model. Numerous experimental results on ten image and numerical datasets show that the DANMF-CRFR model outperforms other state-of-the-art methods. For future works, we plan to utilize other loss functions robust to data contaminated by noise and outliers, e.g., the $L_{2,1}$, correntropy, and elastic losses. The next obvious step is to explore the suitability of such fusion-based encoder–decoder models for performing manifold-based multi-view clustering.

CRediT authorship contribution statement

Navid Salahian: Methodology, Implementation, Writing – original draft, Visualization. **Fardin Akhlaghian Tab:** Conceptualization, Methodology, Writing – review & editing, Supervision. **Seyed Amjad Seyedi:** Conceptualization, Methodology, Writing – review & editing, Visualization. **Jovan Chavoshinejad:** Implementation.

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.

Appendix. Implicit orthogonality

Theorem 1. *Assume that* $\boldsymbol{X} \in \mathbb{R}_+^{d\times n}$, $\boldsymbol{\Psi}_l \in \mathbb{R}_+^{d\times k}$, *and* $\boldsymbol{H} \in \mathbb{R}_+^{k\times n}$ *are such that* $\boldsymbol{X} = \boldsymbol{\Psi}_l \boldsymbol{H}_l$. *By assuming* $\boldsymbol{\Psi}_l$ *has a left inverse, the formulas* $\boldsymbol{X}\boldsymbol{X}^\top \approx \lambda_3 \boldsymbol{\Psi}_l \boldsymbol{\Psi}_l^\top$ *and* $\boldsymbol{H}\boldsymbol{H}^\top \approx \lambda_3 \boldsymbol{I}$ *are equal (*λ_3 *is a scalar).*

Proof. The characteristics of one-sided inverse matrices (Hedjam et al., 2021; Radhakrishna & Mitra, 1972) are as follows:

1. Let $\boldsymbol{A}$ be a $d \times k$ matrix and rank $\rho(\boldsymbol{A}) = k$. $\boldsymbol{A}$ is left invertible if there exists an $k \times d$ matrix $\boldsymbol{A}^{-\mathrm{L}}$ such that $\boldsymbol{A}^{-\mathrm{L}}\boldsymbol{A} = \boldsymbol{I}_k$, where $\boldsymbol{I}_k$ is the identity matrix with k rows and k columns.
2. Let $\boldsymbol{A}$ be an $k \times n$ matrix and rank of $\rho(\boldsymbol{A}) = k$. $\boldsymbol{A}$ is right invertible if there exists a $n \times k$ matrix $\boldsymbol{A}^{-\mathrm{R}}$ such that $\boldsymbol{A}\boldsymbol{A}^{-\mathrm{R}} = \boldsymbol{I}_k$, where $\boldsymbol{I}_k$ stands for identity matrix with k rows and k columns.

Thus, if $\boldsymbol{\Psi}_l$ has rank $\rho(\boldsymbol{\Psi}_l) = k$, then its transposition $\boldsymbol{\Psi}_l^\top$ has rank $\rho(\boldsymbol{\Psi}_l^\top) = k$ as well, and by applying the assumptions (1) and (2) above, we can deduce that $\boldsymbol{\Psi}_l$ has a left inverse $\boldsymbol{\Psi}_l^{-\mathrm{L}}$ and $\boldsymbol{\Psi}_l^\top$ has a right inverse $\boldsymbol{\Psi}_l^{-\mathrm{R}}$. The following proof is concluded:

$$
\begin{aligned}
\boldsymbol{X}\boldsymbol{X}^\top &\approx \lambda_3 \boldsymbol{\Psi}_l \boldsymbol{\Psi}_l^\top && \text{(A.1)}\\
\boldsymbol{\Psi}_l \boldsymbol{H}_l \boldsymbol{H}_l^\top \boldsymbol{\Psi}_l^\top &\approx \boldsymbol{\Psi}_l \lambda_3 \boldsymbol{I} \boldsymbol{\Psi}_l^\top && \text{(since } \boldsymbol{X} = \boldsymbol{\Psi}_l \boldsymbol{H}_l\text{)}\\
\boldsymbol{\Psi}_l^{-\mathrm{L}}[\boldsymbol{\Psi}_l \boldsymbol{H}_l \boldsymbol{H}_l^\top \boldsymbol{\Psi}_l^\top] &\approx \boldsymbol{\Psi}_l^{-\mathrm{L}}[\boldsymbol{\Psi}_l \lambda_3 \boldsymbol{I} \boldsymbol{\Psi}_l^\top]\\
\boldsymbol{H}_l \boldsymbol{H}_l^\top \boldsymbol{\Psi}_l^\top &\approx \lambda_3 \boldsymbol{I} \boldsymbol{\Psi}_l^\top\\
[\boldsymbol{H}_l \boldsymbol{H}_l^\top \boldsymbol{\Psi}_l^\top]\boldsymbol{\Psi}_l^{-\mathrm{R}} &\approx [\lambda_3 \boldsymbol{I} \boldsymbol{\Psi}_l^\top]\boldsymbol{\Psi}_l^{-\mathrm{R}}\\
\boldsymbol{H}_l \boldsymbol{H}_l^\top &\approx \lambda_3 \boldsymbol{I}
\end{aligned}
$$

References

Abdollahi, R., Amjad Seyedi, S., & Reza Noorimehr, M. (2020). Asymmetric semi-nonnegative matrix factorization for directed graph clustering. In *International conference on computer and knowledge engineering* ICCKE, (pp. 323–328).

Asuncion, A., & Newman, D. (2007). UCI machine learning repository.

Belhumeur, P., Hespanha, J., & Kriegman, D. (1997). Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(7), 711–720.

Belkin, M., & Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, *15*(6), 1373–1396.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1798–1828.

Cai, D., He, X., Han, J., & Huang, T. S. (2011). Graph regularized nonnegative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(8), 1548–1560.

Carreira-Perpiñan, M. Á. (2010). The elastic embedding algorithm for dimensionality reduction. In *Proceedings of the 27th international conference on international conference on machine learning* (pp. 167—174).

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th international conference on machine learning, Vol. 119* (pp. 1597–1607).

Choi, S. (2008). Algorithms for orthogonal nonnegative matrix factorization. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)* (pp. 1828–1832).

De Handschutter, P., Gillis, N., & Siebert, X. (2021). A survey on deep matrix factorizations. *Computer Science Review*, *42*, Article 100423.

Dietterich, T., Jain, A., Lathrop, R., & Lozano-Pérez, T. (1993). A comparison of dynamic reposing and tangent distance for drug activity prediction. In J. Cowan, G. Tesauro, & J. Alspector (Eds.), *Advances in neural information processing systems, Vol. 6*. Morgan-Kaufmann.

Ding, C., He, X., & Simon, H. D. (2005). On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proceedings of the 2005 SIAM international conference on data mining* SDM, (pp. 606–610).

Ding, C. H., Li, T., & Jordan, M. I. (2010). Convex and semi-nonnegative matrix factorizations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(1), 45–55.

Ding, C., Li, T., & Peng, W. (2006). Nonnegative matrix factorization and probabilistic latent semantic indexing: Equivalence, chi-square statistic, and a hybrid method. In *Proceedings of the 21st national conference on artificial intelligence, Vol. 1* (pp. 342—347).

Ding, C., Li, T., Peng, W., & Park, H. (2006). Orthogonal nonnegative matrix T-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 126—135).

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Guo, W. (2021). Sparse dual graph-regularized deep nonnegative matrix factorization for image clustering. *IEEE Access*, *9*, 39926–39938.

Guo, Z., & Zhang, S. (2020). Sparse deep nonnegative matrix factorization. *Big Data Mining and Analytics*, *3*(1), 13–28.

Hadsell, R., Chopra, S., & LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition, Vol. 2* (pp. 1735–1742).

He, X., & Niyogi, P. (2003). Locality preserving projections. In *Proceedings of the 16th international conference on neural information processing systems* (pp. 153—160).

He, X., Yan, S., Hu, Y., Niyogi, P., & Zhang, H.-J. (2005). Face recognition using Laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(3), 328–340.

Hedjam, R., Abdesselam, A., & Melgani, F. (2021). NMF with feature relationship preservation penalty term for clustering problems. *Pattern Recognition*, *112*, Article 107814.

Hinton, G. E., & Roweis, S. (2002). Stochastic neighbor embedding. In S. Becker, S. Thrun, & K. Obermayer (Eds.), *Advances in neural information processing systems, Vol. 15*. MIT Press.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), 504–507.

Hoyer, P. O. (2004). Non-negative matrix factorization with sparseness constraints.. *Journal of Machine Learning Research*, *5*(9).

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, *2*(1), 193–218.

Kim, H., & Park, H. (2007). Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, *23*(12), 1495–1502.

Kuang, D., Choo, J., & Park, H. (2015). Nonnegative matrix factorization for interactive topic modeling and document clustering. In *Partitional clustering algorithms* (pp. 215–243).

Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, *2*(1–2), 83–97.

Le-Khac, P. H., Healy, G., & Smeaton, A. F. (2020). Contrastive representation learning: A framework and review. *IEEE Access*, *8*, 193907–193934.

LeCun, Y. (1998). The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*(6755), 788–791.

Lee, D., & Seung, H. S. (2000). Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems, Vol. 13*.

Li, S., Hou, X. W., Zhang, H. J., & Cheng, Q. S. (2001). Learning spatially localized, parts-based representation. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition, Vol. 1* (pp. I207–I212).

Liu, G., Lin, Z., Yan, S., Sun, J., Yu, Y., & Ma, Y. (2013). Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(1), 171–184.

Lyu, B., Xie, K., & Sun, W. (2017). A deep orthogonal non-negative matrix factorization method for learning attribute representations. In *Neural information processing* (pp. 443–452).

Nene, S. A., Nayar, S. K., Murase, H., et al. (1996). Columbia object image library (coil-20).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Peng, S., Ser, W., Chen, B., & Lin, Z. (2020). Robust orthogonal nonnegative matrix tri-factorization for data representation. *Knowledge-Based Systems*, *201–202*, Article 106054.

Peng, C., Zhang, Z., Kang, Z., Chen, C., & Cheng, Q. (2021). Nonnegative matrix factorization with local similarity learning. *Information Sciences*, *562*, 325–346.

Radhakrishna, R. C., & Mitra, S. (1972). Generalized inverse of a matrix and its applications. In *Proc. sixth Berkeley symp. on math. statist. and prob, Vol. 1* (pp. 601–620).

Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*(5500), 2323–2326.

Samaria, F., & Harter, A. (1994). Parameterisation of a stochastic model for human face identification. In *Proceedings of 1994 IEEE workshop on applications of computer vision* (pp. 138–142).

Seyedi, S. A., Moradi, P., & Tab, F. A. (2017). A weakly-supervised factorization method with dynamic graph embedding. In *2017 artificial intelligence and signal processing conference* AISP, (pp. 213–218).

Shang, F., Jiao, L., & Wang, F. (2012). Graph dual regularization non-negative matrix factorization for co-clustering. *Pattern Recognition*, *45*(6), 2237–2250.

Shu, Z., Sun, Y., Tang, J., & You, C. (2022). Adaptive graph regularized deep semi-nonnegative matrix factorization for data representation. *Neural Processing Letters*, 1–19.

Shu, Z.-q., Wu, X.-j., Hu, C., You, C.-z., & Fan, H.-h. (2021). Deep semi-nonnegative matrix factorization with elastic preserving for data representation. *Multimedia Tools and Applications*, *80*(2), 1707–1724.

Sun, B.-J., Shen, H., Gao, J., Ouyang, W., & Cheng, X. (2017). A non-negative symmetric encoder-decoder approach for community detection. In *Proceedings of the 2017 ACM on conference on information and knowledge management* (pp. 597–606).

Tenenbaum, J. B., de Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, *290*(5500), 2319–2323.

Tolić, D., Antulov-Fantulin, N., & Kopriva, I. (2018). A nonlinear orthogonal non-negative matrix factorization approach to subspace clustering. *Pattern Recognition*, *82*, 40–55.

Trigeorgis, G., Bousmalis, K., Zafeiriou, S., & Schuller, B. (2014). A deep semi-NMF model for learning hidden representations. In E. P. Xing, & T. Jebara (Eds.), *Proceedings of machine learning research*: *vol. 32*, *Proceedings of the 31st international conference on machine learning* (pp. 1692–1700). Bejing, China.

Tsuge, S., Shishibori, M., Kuroiwa, S., & Kita, K. (2001). Dimensionality reduction using non-negative matrix factorization for information retrieval. In *2001 IEEE international conference on systems, man and cybernetics. E-systems and e-man for cybernetics in cyberspace, Vol. 2* (pp. 960–965 vol.2).

Turk, M., & Pentland, A. (1991). Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, *3*(1), 71–86.

Wang, Y., Jia, Y., Hu, C., & Turk, M. (2005). NON-NEGATIVE MATRIX factorization FRAMEWORK FOR FACE recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, *19*(04), 495–511.

Wang, Y.-X., & Zhang, Y.-J. (2013). Nonnegative matrix factorization: A comprehensive review. *IEEE Transactions on Knowledge and Data Engineering*, *25*(6), 1336–1353.

Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., & Ma, Y. (2009). Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(2), 210–227.

Wu, Y., Shen, B., & Ling, H. (2014). Visual tracking via online nonnegative matrix factorization. *IEEE Transactions on Circuits and Systems for Video Technology*, *24*(3), 374–383.

Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747.

Xu, W., & Gong, Y. (2004). Document clustering by concept factorization. In *Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 202—209).

Yang, M., & Xu, S. (2021). Orthogonal nonnegative matrix factorization using a novel deep autoencoder network. *Knowledge-Based Systems*, *227*, Article 107236.

Ye, F., Chen, C., & Zheng, Z. (2018). Deep autoencoder-like nonnegative matrix factorization for community detection. In *Proceedings of the 27th ACM international conference on information and knowledge management* (pp. 1393—1402).

Yu, J., Zhou, G., Cichocki, A., & Xie, S. (2018). Learning the hierarchical parts of objects by deep non-smooth nonnegative matrix factorization. *IEEE Access*, *6*, 58096–58105.

Zhao, Y., Wang, H., & Pei, J. (2021). Deep non-negative matrix factorization architecture based on underlying basis images learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(6), 1897–1913.