Pattern Recognition 137 (2023) 109282

Contents lists available at ScienceDirect

Pattern Recognition

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

Self-supervised semi-supervised nonnegative matrix factorization for data clustering

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

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

ARTICLE INFO

Article history:
Received 31 August 2022
Revised 3 December 2022
Accepted 27 December 2022
Available online 28 December 2022

Keywords:
Nonnegative matrix factorization
Semi-supervised learning
Self-supervised learning
Ensemble clustering

ABSTRACT

Semi-supervised nonnegative matrix factorization exploits the strengths of matrix factorization in successfully learning part-based representation and is also able to achieve high learning performance when facing a scarcity of labeled data and a large amount of unlabeled data. Its major challenge lies in how to learn more discriminative representations from limited labeled data. Furthermore, self-supervised learning has been proven very effective at learning representations from unlabeled data in various learning tasks. Recent research works focus on utilizing the capacity of self-supervised learning to enhance semi-supervised learning. In this paper, we design an effective Self-Supervised Semi-Supervised Nonnegative Matrix Factorization (S^4NMF) in a semi-supervised clustering setting. The S^4NMF directly extracts a consensus result from ensembled NMFs with similarity and dissimilarity regularizations. In an iterative process, this self-supervisory information will be fed back to the proposed model to boost semi-supervised learning and form more distinct clusters. The proposed iterative algorithm is used to solve the given problem, which is defined as an optimization problem with a well-formulated objective function. In addition, the theoretical and empirical analyses investigate the convergence of the proposed optimization algorithm. To demonstrate the effectiveness of the proposed model in semi-supervised clustering, we conduct extensive experiments on standard benchmark datasets. The source code for reproducing our results can be found at https://github.com/ChavoshiNejad/S4NMF.

1. Introduction

The last two decades have witnessed an explosion in the amount of digitally stored information and a concurrent increase in demand for large-scale data. This trend results from increasing interest in researching and developing novel techniques for data mining and big data processing. The appropriate low-dimensional representation must generally be adopted when dealing with large amounts of high-dimensional data [1]. The latent structural information in high-dimensional data can be efficiently revealed by a suitable low-dimensional representation, which is also used to improve learning and simplify computations.

Among the most widely used dimensionality reduction methods, matrix factorization is one that is capable of learning a low-dimensional representation from high-dimensional data. The other popular examples of canonical techniques are nonnegative matrix factorization (NMF) [2,3], singular value decomposition (SVD) [4], QR decomposition (QRD) [5], deterministic column-based matrix decomposition [6], Linear Discriminant Analysis (LDA) [7], regularized LDA [8], Principal Component Analysis (PCA) [9], and Independent Component Analysis (ICA) [10].

NMF was first proposed as a matrix factorization technique in foundational studies [2,3]. The nonnegative constraints in matrix factorization lead to extracting localized features that can be used to construct a semantically interpretable representation of parts. This representation is feasible since only additive (not subtractive) combinations are allowed. NMF has shown outstanding performance in a wide range of applications, including pattern recognition, machine learning, computer vision, image processing, and biomedical engineering [11]. In technical terms, it decomposes a nonnegative matrix into two smaller nonnegative matrices known as the basis matrix and the coefficient matrix. Besides its representation capability, there are a wide variety of clustering problems that NMF can solve. Ding et al. [12] discovered the link between NMF and k-means, and proved that NMF can be utilized as a clustering method.

NMF methods that have been used so far are mostly unsupervised, which means they do not pay much attention to any supervised information that could be hidden in the data. Recently,

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

https://doi.org/10.1016/j.patcog.2022.109282
0031-3203/© 2022 Elsevier Ltd. All rights reserved.

numerous semi-supervised NMF methods have been developed to incorporate supervised information. There are two main categories of these methods that can be defined. In the first, the available label information (i.e., the pointwise constraints) is directly included as hard constraints in the objective function [13–15]. However, exploiting the local geometrical structure of data is ignored in such semi-supervised algorithms. The second approach implicitly utilizes pairwise constraints into the weight matrix of the data graph, including Must-Link (ML) constraint (indicating that two samples belong to the same class) and Cannot-Link (CL) constraint (indicating that two samples belong to different classes) [16].

Unfortunately, it is not practicable to construct massively labeled datasets for supervised scenarios. As a result, developing a learning algorithm that can effectively learn to identify new concepts while utilizing just a modest quantity of labeled instances is a fundamental research challenge. The fact that humans can grasp new concepts after observing a few (labeled) instances indicates that this aim is theoretically feasible. On the other hand, unlabeled data is significantly easier to access in many real-world applications, which is why a considerable amount of research is being done on using such data to train models. Recently, self-supervised learning methods offer a lot of potential in this endeavor for enhancing representations when the number of labeled data is limited. Although solely self-supervised approaches have shown remarkable results [17], the representations learned by these techniques are much inferior to those obtained by supervised methods. Therefore, their practical functionality is restricted, and self-supervision alone has yet to be shown to be adequate. Recently self-supervision algorithms have been considered state-of-the-art in the semi-supervised learning manner [18]. Even a modest number of labeled examples would significantly impact self-supervised learning techniques.

This paper bridges self-supervised and semi-supervised learning paradigms to provide a framework for the semi-supervised clustering problem. Our model leverages (i) self-supervised learning to generate supervisory signals and (ii) semi-supervised learning that forces representations to be aligned with pairwise constraints. More specifically, in addition to similarity and dissimilarity constraints for learning information from each labeled sample, we propose a perspective of self-supervision to exploit the pseudo-labels, which can boost the training process. It is generally believed that ensemble models are usually better at generalization than a single model. According to Dietterich [19], an ensemble of models succeeded in better accuracy if and only if its members are diverse and accurate. Based on ensemble clustering [20], we present a diverse and accurate NMF technique that may gradually boost semi-supervised clustering performance by leveraging the factorization sensitivity to initialization characteristics and additional supervisory information to guide it.

Specifically, in an iterative process, we perform multiple semi-supervised NMFs (by using the supervision of ML and CL constraints) with random initializations on the affinity matrix, leading to multiple decomposed matrices in each iteration. In this process, by adaptively ranking the quality of those matrices, a new affinity matrix is generated from the one-hot coding of those matrices, which is called the clustering partition. The refined affinity matrix with incorporated supervisory information clustering partition is often more discriminative than the relationships between samples represented in the unsupervised affinity matrix; because a sparse affinity matrix could produce a perfect partition [21]. As a result, the updated self-supervisory affinity matrix is expected to be better than the one prior to learning a new set of clustering partitions. Mathematically, we explicitly define Self-Supervised Semi-Supervised NMF (S^{4}NMF) as a constrained optimization problem and present an effective and efficient algorithm to solve it with the theoretical convergence guaranteed.

This paper is organized as follows. In Section 2, we briefly introduce the NMF and related semi-supervised NMF methods. Section 3 presents the mathematical model and solution technique of the proposed algorithm. Moreover, the theoretical convergence of the optimization is analyzed. In Section 4, we provide extensive experimental results and comprehensive analysis to demonstrate the efficiency of our algorithm. Finally, Section 5 provides the conclusion and future works.

2. Related work

This section introduces the preliminaries of the baseline methods, including the unsupervised NMF, semi-supervised NMF, and their respective variants.

2.1. Unsupervised NMFs

NMF is a commonly used dimensionality reduction approach in data mining and machine learning. NMF was initially suggested as positive matrix factorization in the work of *Paatero and Tapper* [22]. Nonnegative matrix factorization (NMF) learns a part-based data representation as a linear model. The original data matrix $\boldsymbol{X} \in \mathbb{R}^{d\times n}$ is to be approximated by a pair of nonnegative matrices $\boldsymbol{U} \in \mathbb{R}^{d\times k}$ and $\boldsymbol{V} \in \mathbb{R}^{n\times k}$.

$$\boldsymbol{X} \approx \boldsymbol{U}\boldsymbol{V}^\top, \tag{1}$$

where $\boldsymbol{U}$ represents a set of basis vectors, and $\boldsymbol{V}$ can be considered as the representation of each sample with regard to these basis vectors in the preceding representation. The objective function based on Euclidean distance is stated as follows to evaluate the decomposition quality,

$$\min_{\boldsymbol{U},\boldsymbol{V}} \|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V}^\top\|_F^2, \quad \text{s.t.} \quad \boldsymbol{U}, \boldsymbol{V} \geq 0, \tag{2}$$

where $\|\cdot\|_F$ indicates the Frobenius norm of the matrix.

Obtaining the global minimum of the objective function is difficult since it is not convex in both $\boldsymbol{U}$ and $\boldsymbol{V}$. To get a local minimum for NMF, Lee and Seung [3] developed a simple but effective multiplicative update method for NMF and demonstrated its convergence. The approximation error is reduced by continually executing the iterative update method. The final values of $\boldsymbol{U}$ and $\boldsymbol{V}$ are acquired when the specified terminal condition is satisfied. Over the last decades, several enhanced variants have been presented for various diverse purposes [11]. Sparse low-dimensional representations have been established in [23] to take advantage of the fact that sparsity may lead to a better part-based representation, which implies that the low-dimensional representation only contains a limited number of non-zero coefficients. According to recent NMF research, NMF does not necessarily result in local part-based representations. Therefore, the sparseness constraint on the squared error-based objective function was applied by Hoyer [23] to ensure that part-based characteristics were preserved. Also, Ding et al. [24] enforce the orthogonality of low-dimensional representations in orthogonal NMF. A hard clustering interpretation is achieved by combining the orthogonality constraint with the nonnegative constraint in the NMF.

Graph regularized Nonnegative Matrix Factorization (GNMF) incorporates the inherent geometrical structures of the data on a manifold by a Laplacian regularization [25] to learn the latent nonlinear structures of the data. GNMF involves both linear and nonlinear relations between the data points in the original data space by modeling the data space as a manifold embedded in ambient space and conducting NMF on this manifold. Most datasets are assumed to be embedded in a nonlinear manifold that is defined by a graph. Symmetric NMF (SymNMF) was presented [26] to expand NMF to a nonlinear clustering approach by decomposition of the similarity matrix of the graph into the product of a nonnegative matrix and its transpose ($\boldsymbol{A} = \boldsymbol{V}\boldsymbol{V}^\top$), so the final clustering assignments can be derived directly. As a result, SymNMF is more successful at dealing with nonlinear data and eliminates the influence of additional clustering approaches (such as k-means) that NMF-based methods impose. The SymNMF is stated as follows,

$$\min_{\boldsymbol{V}} \|\boldsymbol{A} - \boldsymbol{V}\boldsymbol{V}^\top\|_F^2, \quad \text{s.t.} \quad \boldsymbol{V} \geq 0, \tag{3}$$

which is strongly associated with the objective function of the graph clustering [26]. SymNMF can be solved using the Multiplicative Update Rule (MUR) [3], similar to basic NMF. At each iteration, MUR is used to determine the update rule for $\boldsymbol{V}$.

2.2. Semi-supervised NMFs

The basic NMF cannot use the label information since, inherently, it is an unsupervised model. For this reason, semi-supervised NMF is suggested, which uses supervisory information to guide the NMF process. The use of unlabeled data in combination with limited amounts of labeled data has been reported by numerous machine learning researchers to boost learning accuracy significantly [27]. A completely labeled training set may be unfeasible due to the expense of labeling, but acquiring a limited quantity of labeled data is quite affordable and more practical. Semi-supervised learning can be instrumental in such scenarios. Therefore, extending the use of NMF to a semi-supervised setting would be beneficial. As a pioneer work, Lee et al. [28] decomposed both the data matrix and the label matrix into the same representation matrix using different basis matrices concurrently.

Liu et al. [14] presented the constrained NMF (CNMF), which extended NMF to a semi-supervised manner. As long as data with the same label are forced to have the same representation, the new representations may have better discriminating power since the label information is used as a hard constraint. CNMF is defined as follows,

$$\min_{\boldsymbol{U},\boldsymbol{Z}} \|\boldsymbol{X} - \boldsymbol{U}(\boldsymbol{C}\boldsymbol{Z})^\top\|_F^2, \quad \text{s.t.} \quad \boldsymbol{U}, \boldsymbol{Z} \geq 0, \tag{4}$$

where $\boldsymbol{C}\boldsymbol{Z}$ performs the same function as $\boldsymbol{V}$ in (2). CNMF is a free parameter model that can handle both labeled and unlabeled data. However, it still has one important drawback: the assumption that data samples from the same class have precisely the same representation may not be a practical assumption in many applications. Discriminative NMF (DNMF) is a semi-supervised NMF proposed by [29] that uses the label information of a percentage of data as a discriminative constraint. In DNMF, points with the same labels are all assigned to the same axis in the new representation space. However, in those methods, there is no assurance that labeled data points with different labels will not be clustered together in the new representation space. According to [30], a modified CNMF was proposed by Xing et al. that uses the label information of a subset of the data similar to the regularization term of the DNMF. The discriminative characteristic of data points in the new representation space is improved due to this regularizer.

SEMINMF [31], another version of CNMF, uses the graph Laplacian to incorporate the local structure of the data and utilizes label information as fitting constraints to learn. Although these semi-supervised NMF models include prior supervisory information, they ignore sparsity and noise robustness. To deal with noisy data, Wang et al. [32] improved the CNMF by including a robust loss function based on $\ell_{2,1}$ norm. Semi-supervised NMF via constraint propagation (CPSNMF) [16] propagates supervisory information (pairwise constraints) to the whole dataset first and then obtains a refined Laplacian graph to regularize NMF decomposition. Based on CNMF, Peng et al. [15] presented CSNMF, which uses the entropy-based loss function instead of the Squared Euclidean Distance (SED) to mitigate the impact of non-Gaussian noise or outlier samples. To provide a more robust discriminative data representation, it incorporates two types of supervisory information, including pointwise and pairwise constraints. Li et al. [13] presented a robust structured NMF learning framework that leverages the block-diagonal structure and the $\ell_{2,p}$ norm loss function to learn a robust discriminative representation. The $\ell_{2,p}$ norm loss function efficiently handles noise and outliers.

The complicated hierarchical and structural information of the original high-dimensional data is hard to be extracted by employing a single-layer clustering algorithm through the low-dimensional representation learned by shallow NMF with semi-supervised guidance. Therefore, Trigeorgis et al. [33] proposed deep weakly supervised factorization (Deep WSF), a deep learning method for NMF, and suggested a new multi-layer structure. Deep WSF employs some prior knowledge to construct a graph for incorporating partially labeled samples in its hierarchical representation model. Meng et al. [34] proposed a semi-supervised deep NMF model (SGDNMF) with dual hypergraph regularizations and bi-orthogonal constraints on hidden layers for learning data representation. By constructing data and feature hypergraphs, SGDNMF extracts the high-order relationship between samples and preserves the local structure in the low-dimensional space.

The semi-supervised SymNMF model introduced by Jia et al. in [35] for clustering task. Instead of relying on a predetermined similarity graph in existing SymNMF, the suggested model learns an adaptive similarity graph and concurrently conducts SymNMF. Jointly using the graph construction and SymNMF clustering steps may take advantage of their mutual improvement described above. Moreover, to better use the pairwise constraints cannot-links, they introduced a formulation in the form of a cannot-link propagation problem, in which they used the propagated dissimilarity relations to restrict the solution space of the similarity graph. NMF model proposed by Jia et al. [36] uses two complementing dissimilarity and similarity regularizations on representations based on must-link and cannot-link constraints to describe the connection between labeled data samples and a limited number of unlabeled ones. Recently, [37] introduced a new self-supervised SymNMF (S^{3}NMF) for clustering by ensembling multiple random nonnegative matrix factorization without any prior knowledge. This approach is accomplished by using generated pseudo-supervisory signals and taking advantage of the fact that SymNMF is sensitive to initialization.

It has been proved that self-supervised learning can also be a remarkably strong foundation for semi-supervised learning in natural language processing and computer vision with a dramatically improved performance compared to the state-of-the-art [18]. For example, Chen et al. [38] established a three-step methodology for semi-supervised ImageNet classification: pretraining based on contrastive learning, fine-tuning by partial labels, and distillation using a self-training method. The problem of semi-supervised learning for image classifiers was addressed by Zhai et al. [39]. Their main idea is that a fast-growing field of self-supervised visual representation learning can boost the field of semi-supervised learning. They have filled the gap between self-supervision methods and semi-supervised learning and combined these two methods by suggesting a framework (S^{4}L) that can be used to turn any self-supervision method into a semi-supervised learning algorithm and combining these two methods.

3. Proposed method

We start this section by briefly reviewing the Symmetric NMF method for unsupervised and semi-supervised clustering. Then, we present our self-supervised semi-supervised learning and the corresponding optimization algorithm. The convergence of the optimization scheme and computational complexity analysis are provided in the last subsection.

3.1. Unsupervised symmetric NMF

Consider data matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{d\times n}$, where each column represents a single sample vector. We identify the p nearest neighbors of each data point $\boldsymbol{x}_j$ and connect $\boldsymbol{x}_j$ to those neighbors with edges. The weight matrix $\boldsymbol{A}$ on the graph can be defined in different ways. Typically, the matrix $\boldsymbol{A}$ indicates the similarity between each pair of elements in a set of n. The following is an example of the most common ones,

$$A_{i,j} = \begin{cases} e^{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{\sigma^2}}, & \text{if } \boldsymbol{x}_j \in \mathcal{N}_p(\boldsymbol{x}_i) \text{ or } \boldsymbol{x}_i \in \mathcal{N}_p(\boldsymbol{x}_j) \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Considering a nonnegative matrix $\boldsymbol{X} \in \mathbb{R}_+^{d\times n}$ and a positive integer $k < \min(d, n)$, NMF seeks to identify two nonnegative matrices $\boldsymbol{U} \in \mathbb{R}_+^{d\times k}$ and $\boldsymbol{V} \in \mathbb{R}_+^{n\times k}$ such that the low-rank matrix $\boldsymbol{U}\boldsymbol{V}^\top$ approximates the input matrix $\boldsymbol{X}$, which implies that $X_{i,j} \approx (\boldsymbol{U}\boldsymbol{V}^\top)_{i,j}$ for $i = 1, \ldots, d$ and $j = 1, \ldots, n$. When the input matrix $\boldsymbol{A} \in \mathbb{R}_+^{n\times n}$ is symmetric, it is rational to search for a symmetric low-rank approximation. Symmetric nonnegative matrix factorization (SymNMF) looks for a matrix $\boldsymbol{V} \in \mathbb{R}_+^{n\times k}$ in which $\boldsymbol{V}\boldsymbol{V}^\top$ reconstructs $\boldsymbol{A}$, that is $A_{i,j} \approx (\boldsymbol{V}\boldsymbol{V}^\top)_{i,j}$ for $1 \le i, j \le n$. SymNMF is mostly applied to a clustering algorithm. Decomposing $\boldsymbol{A}$ into r rank-one factors is equivalent to performing the SymNMF $\boldsymbol{V}\boldsymbol{V}^\top$ on $\boldsymbol{A}$ [26].

$$\min_{\boldsymbol{V}} \|\boldsymbol{A} - \boldsymbol{V}\boldsymbol{V}^\top\|_F^2, \quad \text{s.t.} \quad \boldsymbol{V} \ge 0. \tag{6}$$

3.2. Basic semi-supervised model

For a suitable representation learning method, if two instances are from the same cluster (respectively, different clusters), they should have similar (respectively, dissimilar) representations in the low-dimensional space. Similarity and dissimilarity relations can constrain low-dimensional representations of labeled data samples to achieve semi-supervision.

3.2.1. Modeling of the dissimilarity of labeled samples

The inner product of the low-dimensional representations, i.e., $\boldsymbol{V}\boldsymbol{V}^\top \in \mathbb{R}^{n\times n}$ is used to quantify their similarity, and the following dissimilarity regularizer is proposed,

$$\min_{\boldsymbol{V}} \|\boldsymbol{D} \odot \boldsymbol{V}\boldsymbol{V}^\top\|_1, \tag{7}$$

where $\|\cdot\|_1$ indicates the ℓ_1 norm of a matrix and matrix $\boldsymbol{D}$ is generated based on the label information.

$$D_{i,j} = \begin{cases} 1, & \text{if } \boldsymbol{x}_i, \boldsymbol{x}_j \in \boldsymbol{X}_l \text{ and } y_i \neq y_j \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

where $\boldsymbol{X}_l \in \mathbb{R}^{d\times l}$ is the labeled set, l is number of labeled samples, and y_i indicates the label of sample $\boldsymbol{x}_i$.

As the dissimilarity regularizer in (7) is minimized, the product of the low-dimensional representations of labeled data samples with different labels will decrease, causing the low-dimensional representations to move further apart.

3.2.2. Modeling of the similarity of labeled samples

According to (7), we can see that the element in $\boldsymbol{D}$ that corresponds to labeled samples within the same class is zero. In other words, this regularization pays no attention to the similarity between the low-dimensional representations of these samples. To balance out the dissimilarity regularizer, we suggest the following similarity regularizer,

$$\min_{\boldsymbol{V}} \|\boldsymbol{S} \odot \widetilde{\boldsymbol{V}}\|_1, \tag{9}$$

where $\widetilde{V}_{i,j} = \|\boldsymbol{v}_i - \boldsymbol{v}_j\|^2$ denotes the pairwise Euclidean distance matrix and similarity matrix $\boldsymbol{S}$ is obtained from the label information.

$$S_{i,j} = \begin{cases} 1, & \text{if } \boldsymbol{x}_i, \boldsymbol{x}_j \in \boldsymbol{X}_l \text{ and } y_i = y_j \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

The Euclidean distance between the low-dimensional representations of labeled data samples belonging to the same class will decrease as a result of the minimization of the similarity regularizer in (9). This will cause the low-dimensional representations to be near to one other. Using the previously mentioned similarity and dissimilarity regularizers, the basic formulation of the proposed model is as follows,

$$\min_{\boldsymbol{V}} \|\boldsymbol{A} - \boldsymbol{V}\boldsymbol{V}^\top\|_F^2 + \lambda_1 \|\boldsymbol{D} \odot \left(\boldsymbol{V}\boldsymbol{V}^\top\right)\|_1 + \lambda_2 \|\boldsymbol{S} \odot \widetilde{\boldsymbol{V}}\|_1, \quad \text{s.t.} \quad \boldsymbol{V} \ge 0. \tag{11}$$

where λ_1 and λ_2 control the impact of dissimilarity and similarity regularizations, respectively. The supervisory information in our method is modeled as labels; however, our model can be readily adapted to deal with problems where the supervisory information is given as pairwise constraints.

3.3. Self-supervised semi-supervised NMF

In general, semi-supervised clustering algorithms enhance clustering performance by using pairwise constraints produced by expert knowledge. Nevertheless, the performance of semi-supervised algorithms is less stable and robust since it relies on the parameter values and the order of considered pairwise constraints [15]. In addition to imposing these constraints, the generation of pseudo-supervisory constraints is also fruitful for this work and is our main focus. Recent developments in self-supervised learning show that the use of unlabeled data effectively can be achieved by preserving consistency across various views on samples [18]. Therefore, we employ the cluster ensemble approach to generate these different views for our self-supervised framework. Cluster ensemble approaches [40] are proposed to address the limitations of single clustering approaches, which are less sensitive to parameter values. An ensemble clustering method can incorporate multiple clustering solutions into a framework and generate more stable and robust results.

Since NMF needs to solve a non-convex optimization problem, the initialization of variables is an important step in the process. By utilizing such a sensitivity characteristic of NMF, we propose self-supervised semi-supervised NMF (S^{4}NMF), which can significantly boost clustering performance by using additional information. More specifically, S^{4}NMF is motivated by the successes of ensemble clustering [40] and uses the coherence among the results of multiple semi-supervised models to achieve a consensus and generate some synthetic constraints. We can interpret the latent matrix of SNMF with different random initialization as varied clustering results since it is sensitive to the initialization of variables. Thus, the main goal of our S^{4}NMF is to enhance the semi-supervised clustering capabilities of SNMF using various initializations.

To accomplish this objective, a collection of homogeneous semi-supervised NMFs with varied decompositions is created. In contrast to the NMF in Eq. (11), where the representation is started with a single random nonnegative matrix $\boldsymbol{V}_0$, we first produce a set of random nonnegative matrices indicated by $\{\boldsymbol{V}_{0\boldsymbol{m}} \in \mathbb{R}^{n\times c}\}|_{m=1}^{b}$, where b is the size of the set. These semi-supervised models can be easily utilized by stacking them up to a new one, i.e., by defining the ensemble model as $\mathcal{L} = \sum_{m=1}^{b} \ell_m$ to obtain a fusion framework. However, this approach assigns equal weight to each factorization. Obviously, it fails to account for the significance of different factorizations and may be negatively affected

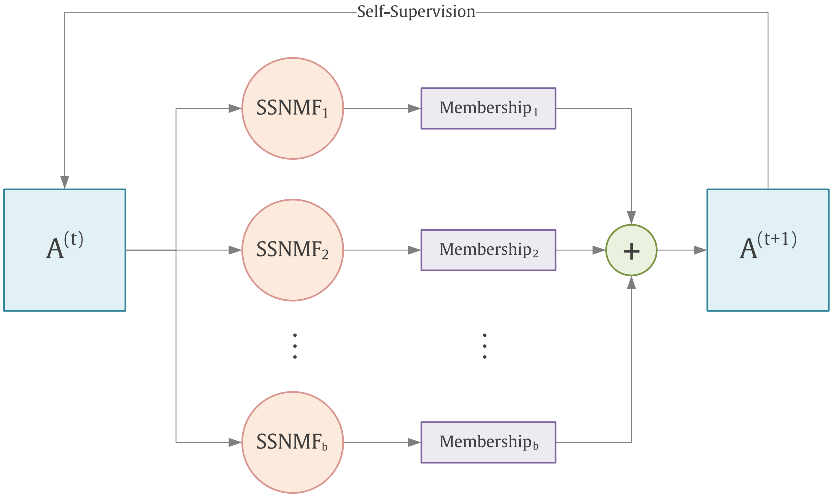


Fig. 1. The architecture of Self-Supervised Semi-Supervised NMF (S^{4}NMF) model. $A^{(t)}$ indicates the affinity matrix for t^{th} iteration, each red circle is a Semi-supervised NMF model with a random initialization, and the green circle means a weighted combination. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

by including an inaccurate one. Integrating these models linearly while using appropriate weights $\alpha_m (m = 1, \dots, b)$ and preserving the smoothness of the weights distribution by adding a parameter τ is a more practical approach. We can obtain the semi-supervised ensemble model in an ensemble setting with the introduction of an appropriate weight parameter. The formulation is as follows,

$$\min_{\boldsymbol{\alpha}, \boldsymbol{V}_{\mathbf{m}}} \sum_{\mathbf{m}=1}^{b} \alpha_{\mathbf{m}}^{\tau} \Big(||\boldsymbol{A} - \boldsymbol{V}_{\mathbf{m}} \boldsymbol{V}_{\mathbf{m}}^{\top}||_F^2 + \lambda_1 \|\boldsymbol{D} \odot \big(\boldsymbol{V}_{\mathbf{m}} \boldsymbol{V}_{\mathbf{m}}^{\top}\big)\|_1 + \lambda_2 ||\boldsymbol{S} \odot \widetilde{\boldsymbol{V}}_{\mathbf{m}}||_1 \Big),$$
$$\text{s.t.} \quad \boldsymbol{\alpha} \mathbf{1} = \mathbf{1}, \ \boldsymbol{\alpha}, \boldsymbol{V}_{\mathbf{m}} \geq 0 \ \forall \mathbf{m}, \tag{12}$$

where $\alpha_{\mathbf{m}}$ is the m-th entry of $\boldsymbol{\alpha} \in \mathbb{R}^{b \times 1}$, the weight vector that balances the contribution of each ensemble whose derivation will be discussed later. $\mathbf{1} \in \mathbb{R}^{b \times 1}$ denotes the all one vector, the constraint $\alpha \mathbf{1} = 1$ avoids the trivial solution of $\boldsymbol{\alpha}$ (i.e., $\boldsymbol{\alpha} = 0$), and $\boldsymbol{\alpha} \geq 0$ guarantees that each $\boldsymbol{\alpha}_{\mathbf{m}}$ is a valid weight.

As mentioned before, Symmetric NMF can be used as a general approach to graph clustering. More research focused on modifying the original model by incorporating constraints into Symmetric NMF to increase the interpretability of the factorized matrix, i.e., the position of the greatest value of $\boldsymbol{v}_i$ can reveal the cluster assignment of $\boldsymbol{x}_i$. Specifically, the partition matrix (or clustering membership matrix) $\boldsymbol{M}$ is generated for symmetric NMF by

$$M_{\mathbf{m}_{ij}} = \begin{cases} 1, & \text{if } V_{\mathbf{m}_{ij}} = \max_j V_{\mathbf{m}_{ij}} \\ 0, & \text{others,} \end{cases} \tag{13}$$

where $M_{i,j}$ and $V_{i,j}$ represent the (i, j)-th elements of $\boldsymbol{M}$ and $\boldsymbol{V}$ matrices, respectively. Therefore, b clustering partitions can be obtained and represented as $\{\boldsymbol{M}_{\mathbf{m}}\}_{\mathbf{m}=1}^{b}$. Taking into consideration the fact that each $\boldsymbol{M}_{\mathbf{m}}$ is, more discriminative than the original similarity matrix $\boldsymbol{A}$, we could be able to generate an enhanced affinity matrix $\boldsymbol{A}$ as,

$$\boldsymbol{A} = \sum_{\mathbf{m}=1}^{b} \alpha_{\mathbf{m}} \boldsymbol{M}_{\mathbf{m}} \boldsymbol{M}_{\mathbf{m}}^{\top}. \tag{14}$$

A set of new and superior semi-supervised clustering ensembles can be generated using the updated affinity $\boldsymbol{A}$ under multiple initializations. Until the maximum number of iterations is reached, this procedure is repeated. Refer to Algorithm 1 for a detailed description of our method.

Algorithm 1 Self-Supervised Semi-Supervised NMF (S^{4}NMF).

```
Input: Data matrix $\boldsymbol{X}$, partial labels $\boldsymbol{y}$, number of clusters $c$,
number of ensembles $b$, hyper-parameters $\lambda_1, \lambda_2$, $\tau = 2$;
Initialize: iter=t=1, OuterIter=10, InnerIter=500,
$\boldsymbol{V}_{\mathbf{m}} = rand^{+}(n, c)$ $\forall \mathbf{m}$, initialize $\boldsymbol{A}$ by (5), and ensemble weight
$\alpha_{\mathbf{m}} = 1/b$;
Output: a set of clustering results $\{\boldsymbol{M}_{\mathbf{m}}\}_{m=1}^{b}$.

Construct dissimilarity and similarity matrices by (8), (10), respectively.
while iter <OuterIter do
    while t <InnerIter do
        for m = 1 to $b$ do
            Update $\boldsymbol{V}_{\mathbf{m}}$ according to (20);
        end for
        Update $\boldsymbol{\alpha}$ by (26)
    end while
    Generate clustering membership matrices $\{\boldsymbol{M}_{\mathbf{m}}\}_{m=1}^{b}$ by (13).
    Update $\boldsymbol{A}$ by (14).
    iter=iter+1; t=t+1;
end while
```

3.4. Numerical solution

Eq. (12) is non-convex, and thus quite challenging to solve. To tackle this challenge, we propose to optimize $\boldsymbol{V}_{\mathbf{m}}$, $\forall \mathbf{m}$ and $\boldsymbol{\alpha}$ alternatively and iteratively, i.e., update $\boldsymbol{V}_{\mathbf{m}}$, $\forall \mathbf{m}$ with a fixed $\boldsymbol{\alpha}$, and then update $\boldsymbol{\alpha}$ with a fixed $\boldsymbol{V}_{\mathbf{m}}$, $\forall \mathbf{m}$.

- Update $\boldsymbol{V}_{\mathbf{m}}$: With a fixed $\boldsymbol{\alpha}$, the $\boldsymbol{V}$-subproblem is expressed as

$$\min_{\boldsymbol{V}_{\mathbf{m}}} \|\boldsymbol{A} - \boldsymbol{V}_{\mathbf{m}} \boldsymbol{V}_{\mathbf{m}}^{\top}\|_F^2 + \lambda_1 \|\boldsymbol{D} \odot \big(\boldsymbol{V}_{\mathbf{m}} \boldsymbol{V}_{\mathbf{m}}^{\top}\big)\|_1 + \lambda_2 \|\boldsymbol{S} \odot \widetilde{\boldsymbol{V}}_{\mathbf{m}}\|_1,$$
$$\text{s.t. } \boldsymbol{V}_{\mathbf{m}} \geq 0. \tag{15}$$

Based on the conditions, $\|\boldsymbol{P}\|_F^2 = \text{Tr}(\boldsymbol{P}\boldsymbol{P}^{\top})$, and $\text{Tr}(\boldsymbol{PQR}) = \text{Tr}(\boldsymbol{RPQ}) = \text{Tr}(\boldsymbol{QRP})$, and to get the minima of eq. (15), we introduce Lagrangian multiplier $\boldsymbol{\Theta}$, and construct a Lagrangian function $\Phi(\boldsymbol{V}_{\mathbf{m}}, \boldsymbol{\Theta}_{\mathbf{m}})$, we derive

$$\begin{aligned}\Phi(\boldsymbol{V}_{\mathbf{m}},\boldsymbol{\Theta}_{\mathbf{m}}) = & Tr\Big(-2\boldsymbol{A}\boldsymbol{V}_{\mathbf{m}}\boldsymbol{V}_{\mathbf{m}}^{\top}+\boldsymbol{V}_{\mathbf{m}}\boldsymbol{V}_{\mathbf{m}}^{\top}\boldsymbol{V}_{\mathbf{m}}\boldsymbol{V}_{\mathbf{m}}^{\top}\Big)\\ & +\lambda_1 Tr\Big(\boldsymbol{V}_{\mathbf{m}}^{\top}\boldsymbol{D}\boldsymbol{V}_{\mathbf{m}}\Big)+2\lambda_2 Tr\Big(\boldsymbol{V}_{\mathbf{m}}^{\top}\boldsymbol{B}\boldsymbol{V}_{\mathbf{m}}\Big)\\ & -2\lambda_2 Tr\Big(\boldsymbol{V}_{\mathbf{m}}^{\top}\boldsymbol{S}\boldsymbol{V}_{\mathbf{m}}\Big)-Tr\Big(\boldsymbol{V}_{\mathbf{m}}^{\top}\boldsymbol{\Theta}_{\mathbf{m}}\Big)\end{aligned} \tag{16}$$

where $\boldsymbol{B}$ is a diagonal matrix, and $B_{ii}=\sum_j S_{ij}$ and $\boldsymbol{\Theta}$ enforces the nonnegative constraint $\boldsymbol{V}_{\mathbf{m}}\geq 0$.
The partial derivatives of $\Phi(\boldsymbol{V}_{\mathbf{m}},\boldsymbol{\Theta}_{\mathbf{m}})$ with respect to $\boldsymbol{V}_{\mathbf{m}}$ is

$$\begin{aligned}\frac{\partial\Phi(\boldsymbol{V}_{\mathbf{m}},\boldsymbol{\Theta}_{\mathbf{m}})}{\partial\boldsymbol{V}_{\mathbf{m}}} = & -4\boldsymbol{A}\boldsymbol{V}_{\mathbf{m}}+4\boldsymbol{V}_{\mathbf{m}}\boldsymbol{V}_{\mathbf{m}}^{\top}\boldsymbol{V}_{\mathbf{m}}+2\lambda_1\boldsymbol{D}\boldsymbol{V}_{\mathbf{m}}+4\lambda_2\boldsymbol{B}\boldsymbol{V}_{\mathbf{m}}\\ & -4\lambda_2\boldsymbol{S}\boldsymbol{V}_{\mathbf{m}}-\boldsymbol{\Theta}_{\mathbf{m}}\end{aligned} \tag{17}$$

By setting the partial derivative of $\Phi(\boldsymbol{V}_{\mathbf{m}},\boldsymbol{\Theta}_{\mathbf{m}})$ with respect to $\boldsymbol{V}_{\mathbf{m}}$ to 0, we have:

$$\Theta_{\mathbf{m}} = -\,4\boldsymbol{A}\boldsymbol{V}_{\mathbf{m}}+4\boldsymbol{V}_{\mathbf{m}}\boldsymbol{V}_{\mathbf{m}}^{\top}\boldsymbol{V}_{\mathbf{m}}+2\lambda_1\boldsymbol{D}\boldsymbol{V}_{\mathbf{m}}+4\lambda_2\boldsymbol{B}\boldsymbol{V}_{\mathbf{m}}-4\lambda_2\boldsymbol{S}\boldsymbol{V}_{\mathbf{m}} \tag{18}$$

the complementary slackness condition of the Karush-Kuhn-Tucker (KKT) conditions [3], we obtain:

$$\boldsymbol{\Theta}_{\mathbf{m}}\odot\boldsymbol{V}_{\mathbf{m}}^{(t)}=0 \tag{19}$$

where $\odot$ denotes the element-wise product. Eq. (19) is the fixed point equation that the solution must satisfy at convergence. By solving this equation, we derive the following updating rule for $\boldsymbol{V}_{\mathbf{m}}$:

$$\boldsymbol{V}_{\mathbf{m}}^{(t+1)}\leftarrow\boldsymbol{V}_{\mathbf{m}}^{(t)}\odot\left(\frac{\boldsymbol{A}\boldsymbol{V}_{\mathbf{m}}^{(t)}+\lambda_2\boldsymbol{S}\boldsymbol{V}_{\mathbf{m}}^{(t)}}{\boldsymbol{V}_{\mathbf{m}}^{(t)}\boldsymbol{V}_{\mathbf{m}}^{(t)\top}\boldsymbol{V}_{\mathbf{m}}^{(t)}+\frac{\lambda_1}{2}\boldsymbol{D}\boldsymbol{V}_{\mathbf{m}}^{(t)}+\lambda_2\boldsymbol{B}\boldsymbol{V}_{\mathbf{m}}^{(t)}}\right)^{\frac{1}{4}} \tag{20}$$

- Solve $\boldsymbol{\alpha}$: With the fixed $\boldsymbol{V}_{\mathbf{m}}$, $\forall\mathbf{m}$, the $\boldsymbol{\alpha}$-subproblem is rewritten as

$$\min_{\boldsymbol{\alpha}}\sum_{m=1}^{b}(\alpha_{\mathbf{m}})^{\tau}e_{\mathbf{m}},\quad \text{s.t.}\quad \boldsymbol{\alpha}\mathbf{1}=\mathbf{1},\ \boldsymbol{\alpha},\geq 0\ . \tag{21}$$

where

$$\begin{aligned}e_{\mathbf{m}} = & \left\|\boldsymbol{S}-\boldsymbol{V}_{\mathbf{m}}^{(t+1)}\boldsymbol{V}_{\mathbf{m}}^{(t+1)\top}\right\|_F^2\\ & +\lambda_1\left\|\boldsymbol{D}\odot\left(\boldsymbol{V}_{\mathbf{m}}^{(t+1)}\boldsymbol{V}_{\mathbf{m}}^{(t+1)\top}\right)\right\|_1+\lambda_2\left\|\boldsymbol{S}\odot\widetilde{\boldsymbol{V}}_{\mathbf{m}}^{(t+1)}\right\|_1\end{aligned} \tag{22}$$

The Lagrange function of Eq. (21) is

$$\mathcal{L}=\sum_{m}^{b}(\alpha_m)^{\tau}e_m-\mu\left(\sum_{m}^{b}\alpha_m-1\right)\text{s.t.}\quad\boldsymbol{\alpha}\geq 0. \tag{23}$$

Taking the first order derivative of Eq. (23) $\frac{\partial\mathcal{L}}{\partial\alpha_{\mathbf{m}}}=\tau(\alpha_m)^{\tau-1}e_m-\mu$, and setting it to zero, we have,

$$\alpha_{\mathbf{m}}=\left(\frac{\mu}{\tau e_{\mathbf{m}}}\right)^{\frac{1}{\tau-1}}\quad,\forall\mathbf{m}. \tag{24}$$

Based on the constraint $\sum_m^b\alpha_m=1$, μ can be obtained as

$$\mu=\left(\frac{1}{\sum_{\mathbf{m}}^{b}(\tau e_{\mathbf{m}})^{\frac{1}{1-\tau}}}\right)^{\tau-1} \tag{25}$$

Substituting μ into Eq. (24), α_m is obtained:

$$\alpha_{\mathbf{m}}\leftarrow\frac{(\tau e_{\mathbf{m}})^{\frac{1}{1-\tau}}}{\sum_{\mathbf{m}}^{b}(\tau e_{\mathbf{m}})^{\frac{1}{1-\tau}}} \tag{26}$$

Since both the numerator and denominator of Eq. (26) are larger than 0, we have $\alpha_m>0,\forall\mathbf{m}$, and the nonnegative constraint for $\boldsymbol{\alpha}$ is satisfied. The solution in Eq. (26) satisfies the Karush-Kuhn-Tucker (KKT) conditions of Eq. (21), and thus it is a local optimum. Moreover, as Eq. (21) is a convex problem, Eq. (26) is the global optimum of Eq. (21).

3.5. Convergence

In this section, we theoretically prove the convergence of Algorithm 1.

Theorem 1: Algorithm 1 monotonically decreases the objective function in (12) and finally converges to a limiting point.

Proof: In order to prove Theorem 1, we first need to give the definition of upper bound auxiliary function.

Definition 1: Function $G(\boldsymbol{V},\boldsymbol{V}^t)$ is the upper bound auxiliary function of $\boldsymbol{\Phi}(\boldsymbol{V})$, if the following two conditions are satisfied:

$$G(\boldsymbol{V},\boldsymbol{V}^t)\geq\boldsymbol{\Phi}(\boldsymbol{V}),\,G(\boldsymbol{V},\boldsymbol{V})=\boldsymbol{\Phi}(\boldsymbol{V}^{\boldsymbol{t}}) \tag{27}$$

Lemma 1: Given G is the upper bound auxiliary function, then $\boldsymbol{\Phi}(\boldsymbol{V})$ is non-increasing under the updating:

$$\boldsymbol{V}^{t+1}=\arg\min_{\boldsymbol{V}}G(\boldsymbol{V},\boldsymbol{V}^t) \tag{28}$$

Based on Lemma 1, we can prove Theorem 1 by first finding an appropriate upper bound auxiliary function for (15) with respect to $\boldsymbol{V}$ with fixed $\boldsymbol{A}$ and $\boldsymbol{\alpha}$, and then showing that (20) minimizes that upper bound auxiliary function.

$$\begin{aligned}\Phi(\boldsymbol{V}_{\mathbf{m}}) = & -2\mathrm{Tr}\Big(\boldsymbol{A}\boldsymbol{V}_{\mathbf{m}}\boldsymbol{V}_{\mathbf{m}}^{\top}\Big)+\mathrm{Tr}\Big(\boldsymbol{V}_{\mathbf{m}}\boldsymbol{V}_{\mathbf{m}}^{\top}\boldsymbol{V}_{\mathbf{m}}\boldsymbol{V}_{\mathbf{m}}^{\top}\Big)+\lambda_1\mathrm{Tr}\Big(\boldsymbol{V}_{\mathbf{m}}^{\top}\boldsymbol{D}\boldsymbol{V}_{\mathbf{m}}\Big)\\ & +2\lambda_2\mathrm{Tr}\Big(\boldsymbol{V}_{\mathbf{m}}^{\top}\boldsymbol{B}\boldsymbol{V}_{\mathbf{m}}\Big)-2\lambda_2\mathrm{Tr}\Big(\boldsymbol{V}_{\mathbf{m}}^{\top}\boldsymbol{S}\boldsymbol{V}_{\mathbf{m}}\Big)\end{aligned} \tag{29}$$

We define the auxiliary function $G(\boldsymbol{V}_{\mathbf{m}},\boldsymbol{V}_{\mathbf{m}}^t)$ as in the following:

$$\begin{aligned}G(\boldsymbol{V}_{\mathbf{m}},\boldsymbol{V}_{\mathbf{m}}^t) = & -2\sum_{i,j=1}^{n}\sum_{k=1}^{c}A_{ij}V_{\mathbf{m}_{ik}}^t V_{\mathbf{m}_{jk}}^t\left(1+\log\frac{V_{\mathbf{m}_{ik}}V_{\mathbf{m}_{jk}}}{V_{\mathbf{m}_{ik}}^t V_{\mathbf{m}_{jk}}^t}\right)\\ & +\sum_{i,j=1}^{n}\sum_{k=1}^{c}(\boldsymbol{V}_{\mathbf{m}}^t\boldsymbol{V}_{\mathbf{m}}^{t\,\top})_{ij}V_{\mathbf{m}_{ik}}^t\frac{(V_{\mathbf{m}_{jk}})^4}{(V_{\mathbf{m}_{jk}}^t)^3}\\ & +\lambda_1\sum_{i,j=1}^{n}\sum_{k=1}^{c}D_{ij}V_{\mathbf{m}_{ik}}^t\frac{(V_{\mathbf{m}_{jk}})^4}{(V_{\mathbf{m}_{jk}}^t)^3}\\ & +2\lambda_2\sum_{i,j=1}^{n}\sum_{k=1}^{c}B_{ij}V_{\mathbf{m}_{ik}}^t\frac{(V_{\mathbf{m}_{jk}})^4}{(V_{\mathbf{m}_{jk}}^t)^3}\\ & -2\lambda_2\sum_{i,j=1}^{n}\sum_{k=1}^{c}S_{ij}V_{\mathbf{m}_{ik}}^t V_{\mathbf{m}_{jk}}^t\left(1+\log\frac{V_{\mathbf{m}_{ik}}V_{\mathbf{m}_{jk}}}{V_{\mathbf{m}_{ik}}^t V_{\mathbf{m}_{jk}}^t}\right)\end{aligned} \tag{30}$$

It is easy to find that if $\boldsymbol{V}_{\mathbf{m}}=\boldsymbol{V}_{\mathbf{m}}^t$, then $G(\boldsymbol{V}_{\mathbf{m}}^t,\boldsymbol{V}_{\mathbf{m}}^t)=\Phi(\boldsymbol{V}_{\mathbf{m}}^t)$. For the first part of $G(\boldsymbol{V}_{\mathbf{m}},\boldsymbol{V}_{\mathbf{m}}^t)$,

$$\Gamma=-2\sum_{i,j=1}^{n}\sum_{k=1}^{c}A_{ij}V_{\mathbf{m}_{ik}}^t V_{\mathbf{m}_{jk}}^t\left(1+\log(\frac{V_{\mathbf{m}_{ik}}V_{\mathbf{m}_{jk}}}{V_{\mathbf{m}_{ik}}^t V_{\mathbf{m}_{jk}}^t})\right), \tag{31}$$

based on the condition $1+\log x\leq x$, it can written as

$$\begin{aligned}\Gamma & \geq -2\sum_{i,j=1}^{n}\sum_{k=1}^{c}A_{ij}V_{\mathbf{m}_{ik}}^t V_{\mathbf{m}_{jk}}^t\frac{V_{\mathbf{m}_{ik}}V_{\mathbf{m}_{jk}}}{V_{\mathbf{m}_{ik}}^t V_{\mathbf{m}_{jk}}^t}\\ & = -2\sum_{i,j=1}^{n}\sum_{k=1}^{c}A_{ij}V_{\mathbf{m}_{ik}}V_{\mathbf{m}_{jk}}=-2\mathrm{Tr}(\boldsymbol{A}\boldsymbol{V}_{\mathbf{m}}\boldsymbol{V}_{\mathbf{m}}^{\top})\end{aligned} \tag{32}$$

The fifth part in $G(\boldsymbol{V}_{\mathbf{m}},\boldsymbol{V}_{\mathbf{m}}^t)$ is similar to the first part. Setting $V_{\mathbf{m}_{jk}}=\mu_{\mathbf{m}_{jk}}V_{\mathbf{m}_{jk}}^t$, the second part can be transformed into $\Delta=\sum_{i,j=1}^{n}\sum_{k,h=1}^{c}V_{\mathbf{m}_{ih}}^t V_{\mathbf{m}_{jh}}^t V_{\mathbf{m}_{ik}}^t V_{\mathbf{m}_{jk}}^t\mu_{\mathbf{m}_{jk}}^4$. Because i and j are symmetric, k and h are symmetric, and $a^4+b^4+c^4+d^4\geq 2(a^2b^2+c^2d^2)\geq 4abcd$, so

$$\mu^4_{\mathbf{m}_{jk}} = \frac{\mu^4_{\mathbf{m}_{jk}} + \mu^4_{\mathbf{m}_{jh}} + \mu^4_{\mathbf{m}_{ik}} + \mu^4_{\mathbf{m}_{ih}}}{4} \geq \mu_{\mathbf{m}_{jk}}\mu_{\mathbf{m}_{jh}}\mu_{\mathbf{m}_{ik}}\mu_{\mathbf{m}_{ih}}$$

$$\begin{aligned} \Delta &\geq \sum_{i,j=1}^{n}\sum_{k,h=1}^{c} V^t_{\mathbf{m}_{ih}}V^t_{\mathbf{m}_{jh}}V^t_{\mathbf{m}_{ik}}V^t_{\mathbf{m}_{jk}}\mu_{\mathbf{m}_{jk}}\mu_{\mathbf{m}_{jh}}\mu_{\mathbf{m}_{ik}}\mu_{\mathbf{m}_{ih}} \\ &= \sum_{i,j=1}^{n}\sum_{k,h=1}^{c} V_{\mathbf{m}_{ih}}V_{\mathbf{m}_{jh}}V_{\mathbf{m}_{ik}}V_{\mathbf{m}_{jk}} = \mathrm{Tr}(\boldsymbol{V}_m\boldsymbol{V}_m^\top\boldsymbol{V}_m\boldsymbol{V}_m^\top) \end{aligned} \tag{33}$$

The third and fourth parts are similar to the second part.

Now, we can see $G(\boldsymbol{V}_\mathbf{m}, \boldsymbol{V}^t_\mathbf{m}) \geq \Phi(\boldsymbol{V}_\mathbf{m})$ is established. Because $\boldsymbol{V}^{t+1}_\mathbf{m} = \arg\min_{\boldsymbol{V}_\mathbf{m}} G(\boldsymbol{V}_\mathbf{m}, \boldsymbol{V}^t_\mathbf{m})$, we calculate the derivative of $G(\boldsymbol{V}_\mathbf{m}, \boldsymbol{V}^t_\mathbf{m})$,

$$\begin{aligned} \frac{\partial G(\boldsymbol{V}_\mathbf{m}, \boldsymbol{V}^t_\mathbf{m})}{\partial V_{\mathbf{m}_{ik}}} =& -4\sum_{i=1}^{n} A_{ij}V^t_{\mathbf{m}_{ik}}\frac{(V_{\mathbf{m}_{jk}})^3}{(V^t_{\mathbf{m}_{jk}})^3} \\ &+4\sum_{i=1}^{n}(\boldsymbol{V}^t_\mathbf{m}\boldsymbol{V}^{t\,\top}_\mathbf{m})_{ij}V^t_{\mathbf{m}_{ik}}\frac{(V_{\mathbf{m}_{jk}})^3}{(V^t_{\mathbf{m}_{jk}})^3} \\ &+2\lambda_1\sum_{i=1}^{n} D_{ij}V^t_{\mathbf{m}_{ik}}\frac{(V_{\mathbf{m}_{jk}})^3}{(V^t_{\mathbf{m}_{jk}})^3} \\ &+4\lambda_2\sum_{i=1}^{n} B_{ij}V^t_{\mathbf{m}_{ik}}\frac{(V_{\mathbf{m}_{jk}})^3}{(V^t_{\mathbf{m}_{jk}})^3} \\ &-4\lambda_2\sum_{i=1}^{n} S_{ij}V^t_{\mathbf{m}_{ik}}\frac{V^t_{\mathbf{m}_{jk}}}{V_{\mathbf{m}_{jk}}} \\ =& -4(\boldsymbol{A}\boldsymbol{V}^t_\mathbf{m})_{jk}\frac{(V_{\mathbf{m}_{jk}})^3}{(V^t_{\mathbf{m}_{jk}})^3} \\ &+4(\boldsymbol{V}^t_\mathbf{m}\boldsymbol{V}^{t\,\top}_\mathbf{m}\boldsymbol{V}^t_\mathbf{m})_{jk}\frac{(V_{\mathbf{m}_{jk}})^3}{(V^t_{\mathbf{m}_{jk}})^3} \\ &+2\lambda_1(\boldsymbol{D}\boldsymbol{V}^t_\mathbf{m})_{jk}\frac{(V_{\mathbf{m}_{jk}})^3}{(V^t_{\mathbf{m}_{jk}})^3} \\ &+4\lambda 2(\boldsymbol{B}\boldsymbol{V}^t_\mathbf{m})_{jk}\frac{(V_{\mathbf{m}_{jk}})^3}{(V^t_{\mathbf{m}_{jk}})^3} \\ &-4\lambda 2(\boldsymbol{S}\boldsymbol{V}^t_\mathbf{m})_{jk}\frac{V^t_{\mathbf{m}_{jk}}}{V_{\mathbf{m}_{jk}}} = 0 \end{aligned} \tag{34}$$

Then we derive the following equation that corresponds to (20):

$$\boldsymbol{V}_{\mathbf{m}_{jk}} \leftarrow \boldsymbol{V}^t_{\mathbf{m}_{jk}}\left(\frac{(\boldsymbol{A}\boldsymbol{V}^t_\mathbf{m} + \lambda_2\boldsymbol{S}\boldsymbol{V}^t_\mathbf{m})_{jk}}{(\boldsymbol{V}^t_\mathbf{m}\boldsymbol{V}^{t\,\top}_\mathbf{m}\boldsymbol{V}^t_\mathbf{m} + \frac{\lambda_1}{2}\boldsymbol{D}\boldsymbol{V}^t_\mathbf{m} + \lambda_2\boldsymbol{B}\boldsymbol{V}^t_\mathbf{m})_{jk}}\right)^{\frac{1}{4}} \tag{35}$$

The empirical convergence analysis are presented in Section 4.7.

3.6. Computational complexity analysis

We first analyze the computational complexity of $\boldsymbol{V}_\mathbf{m}$ and the $\boldsymbol{\alpha}$ update rules (i.e., (20) and (26)) alternatively and iteratively. For the $\boldsymbol{V}_\mathbf{m}$ update rule, the computational complexity is $O(n^2cb)$ and the $\boldsymbol{\alpha}$ update rule has a complexity of $O(n)$. Therefore, the complexity of each inner iteration of Algorithm 1 (steps 3–11) is $O(n^2cb)$. Algorithm 1 involves repeatedly updating the self-supervision matrix with the computational complexity of $O(n^2cbt)$, where t is the maximum outer iteration number of Algorithm 1, and constructing $\boldsymbol{D}$ and $\boldsymbol{S}$ with the computational complexity of $O(l^2)$, where l is the number of labeled samples. Therefore, the complexity of each iteration of Algorithm 1 is $O(n^2cbt)$.

4. Experimental results

In this section, we evaluate the performance of the suggested model by conducting extensive experiments and comparing it with several relevant state-of-the-art algorithms using three clustering evaluation metrics on twelve datasets. The empirical convergence analysis of the S^{4}NMF is provided in the last subsection.

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

Dataset	#sample	#feature	#class	Application
Chart	600	60	6	control chart time
Iris	150	4	3	biology
Seeds	210	7	3	agrophysics
Yale	165	1024	15	face recognition
ORL	400	1024	40	face recognition
UMIST	575	644	20	face recognition
MNIST	1000	784	10	handwriting recognition
Glass	214	9	6	criminological investigation
Zoo	101	16	17	animal science
Wine	178	13	3	chemistry
BC	569	30	2	breast tumor diagnosis
Coil20	1440	1024	20	object recognition

4.1. Datasets

We conduct extensive experiments on twelve datasets, which include three face image datasets (i.e., ORL, Yale, and UMIST), one digit handwriting dataset (i.e., MNIST), one object image dataset (i.e., COIL20), and seven numerical datasets (i.e., Chart, Iris, Seeds, Glass, Zoo, Wine, and BC). Table 1 shows an overview of the detailed characteristics of the datasets that were used in the clustering. We generate a data matrix for each image dataset, which includes the samples and number of dimensions, and we normalize each sample, i.e., each column of the data matrix, to have unit Euclidean length.

- Chart: The Chart dataset includes 600 samples of control charts that were generated synthetically. The six categories of control charts are as follows: 1. Normal 2. Cyclic 3. Increasing trend 4. Decreasing trend 5. Upward shift 6. Downward shift.
- Iris: The Iris dataset includes three classes of 50 samples, each class corresponding to a different type of Iris plant. Each sample contains four attributes. Several clustering problems have used this popular dataset.
- Seeds: The Seeds dataset includes 210 data samples, seven real attributes, and three classes (class labels are 1: Kama, 2: Rosa, 3: Canadian) with 70 data examples in each class.
- Yale (Faces dataset): The Yale dataset includes 165 grayscale images of 15 individuals. There are 11 images per subject, one per facial expression or configuration, such as with/without glasses, center-lighting, left-lighting, right-lighting, normal, sad, sleepy, and so on. The original images were normalized in the experiments and cropped into 32 × 32 pixels for clustering.
- ORL (Faces dataset): The AT&T ORL dataset includes 40 distinct subjects, each with ten different images. For some subjects, the images were taken at different lighting, times, and facial expressions. Here, the original images were normalized and cropped into 32 × 32 pixels for clustering.
- UMIST: The UMIST dataset includes 575 face images of 20 people. Gray images sized to 28 × 23 are used to represent each face image.
- MNIST: The MNIST dataset includes 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.
- Glass: The Glass dataset is about several types of glass that has totally 214 samples in 6 classes with 9 features.

- Zoo: The Zoo dataset includes 101 samples with 16 Boolean features, which is a collection of statistical information about animals.
- Wine: The Wine dataset includes data from wine chemical analysis. It contains 178 examples of 13 attributes, each of that are classified into three classes.
- BC: Breast Cancer Wisconsin dataset deals with the classification of the intensity of breast cancer. Class ‘M’ refers to a malignant level of infection, and class ‘B’ refers to a benign level of infection. It contains 569 data instances and 31 attributes such as radius, perimeter, texture, smoothness, etc., for each cell nucleus.
- Coil20: This dataset 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.

4.2. Compared methods

Here, we provide an overview of twelve comparing algorithms,

- NMF (unsupervised) is a basic NMF model [3].
- SymNMF (unsupervised) provided the clustering result directly by performing the NMF on a predetermined similarity matrix [26].
- ONMF-W (unsupervised) imposed orthogonality constraints on the basis matrix [24].
- ONMF-H (unsupervised) imposed orthogonality constraints on the representation matrix [24].
- CNMF (semi-supervised) incorporated the label information as extra, hard constraints to ensure that data points from the same class have identical latent representations [14].
- RCNMF (semi-supervised) leveraged the dual supervision information to develop a more discriminative data representation while concurrently adopting the correntropy similarity measure to lessen the impact of non-Gaussian noise and outliers [32].
- SemiNMF-CP (semi-supervised) applied pairwise constraint propagation to construct an acceptable graph from the supervised information and then adopted GNMF using the learned graph [16].
- RSNMF (semi-supervised) utilized a block-diagonal structure to embed the label information into NMF [13].
- DSNMF (semi-supervised) accomplished precise community detection by incorporating similarity and dissimilarity regularizations [36].
- DSSNMF (semi-supervised) achieved a more discriminative representation space by incorporating the label information of fraction of data to the objective function of NMF and using it as the regularization term [30].
- RCNMF-CP (semi-supervised) proposed a robust CSNMF algorithm that was robust to non-Gaussian noise by concurrently employing pointwise and pairwise constraints of the data points [15].
- S^3NMF (self-supervised) boosted clustering efficiency by capitalizing on the sensitivity of SNMF to initialization, without depending on any extra information [36].

4.3. Evaluation metrics

In this section, we introduce three widely used quantitative metrics, Normalized Mutual Information (NMI), Accuracy (ACC), Adjust Rand Index (ARI), F-measure (F1), and Purity to evaluate the clustering performance. The NMI measures the shared information between two statistical distributions, which is defined as

$$NMI(c,y) = \frac{MI(c,y)}{max(H(c),H(y))}. \tag{36}$$

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 are close to 0 if they are mostly different.

The ACC criterion indicates the percentage of data points for which the produced clusters can be successfully mapped to ground-truth classes. It is specifically defined as follows:

$$ACC(c,y) = \frac{\sum_{i=1}^{n} \delta(map(c_i), y_i)}{n}, \tag{37}$$

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 true 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. It is equal to 0 if the correspondence between two classes is less than what would be predicted by chance, and it would be one 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,

$$ARI(c,y) = \frac{\sum_{i,j}\binom{n_{ij}}{2} - \sum_i \binom{n_{i.}}{2}\sum_j\binom{n_{.j}}{2}/\binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{n_{i.}}{2} + \sum_j \binom{n_{.j}}{2}\right] - \sum_i \binom{n_{i.}}{2}\sum_j \binom{n_{.j}}{2}/\binom{n}{2}}, \tag{38}$$

where c stands for the clustering results and y stands for the ground true 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 $[-1 \quad 1]$.

The efficiency of clustering and classification algorithms is frequently evaluated using the F-Measure. It utilizes a combination of the concepts of precision and recall from information retrieval. Let $C = C_1, C_2, \ldots, C_k$. Let $C = C_1, C_2, \ldots, C_k$ represent a clustering of dataset D, and $C^* = \{C_1^*, C_2^*, \ldots, C_l^*\}$ represent the correct class set of D. Therefore, $Rec(i,j)$, which represents the recall of cluster j with regard to class i is defined as $|C_j^* \cap C_i^*|/|C_i^*|$. $Prec(i,j)$ is stated as $lvertCj*capCi*rvert/lvertCirvert$ and denotes the precision of the cluster j with regard to the class i. The two values in the F-Measure are combined using the formula below:

$$\mathrm{F1}_{i,j} = 2 \times \frac{Prec(i,j) \times Rec(i,j)}{Prec(i,j) + Rec(i,j)}, \tag{39}$$

The purity is used to quantify the proportion of data samples from a single class that are contained in each cluster. Purity is stated as follows:

$$Purity(c,y) = \frac{1}{n}\sum_{j=1}^{n} \max_j (n_i^j), \tag{40}$$

where n_i^j is the number of samples in cluster i belonging to original class j.

4.4. Experimental settings

The samples from each dataset were divided into two groups in the experiments: the labeled data, which are used to provide supervised information, and the unlabeled data for evaluating the performance of the model. We chose 10% of the samples for each

Table 2
NMI results on real-world datasets: The best result is highlighted in **bold** style, while underline style indicates the second-best.

Method	Chart	Iris	Seeds	Yale	ORL	UMIST	MNIST	Glass	Zoo	Wine	BC	coil20
NMF	0.633	0.582	0.605	0.402	0.626	0.531	0.365	0.355	0.768	0.638	0.254	0.616
SymNMF	0.653	0.699	0.551	0.505	0.723	0.622	0.494	0.306	0.705	0.582	0.630	0.691
ONMF-W	0.440	0.714	0.399	0.265	0.328	0.495	0.251	0.378	0.725	0.413	0.253	0.452
ONMF-H	0.612	0.763	0.458	0.470	0.728	0.637	0.452	0.363	0.841	0.721	0.272	0.716
CNMF	0.765	0.740	0.585	0.492	**0.857**	0.626	0.508	0.306	0.724	0.807	0.677	0.776
RCNMF	0.860	0.880	0.685	0.551	0.850	0.678	0.549	0.280	0.806	0.816	0.708	0.778
CPSNMF	0.795	0.477	0.705	0.493	0.784	0.640	0.336	0.286	0.840	0.782	0.560	0.778
RSNMF	0.815	0.803	0.678	0.497	0.739	0.672	0.506	0.263	0.814	0.774	0.577	0.777
DSNMF	0.800	0.850	0.740	0.544	0.821	0.648	0.460	**0.443**	0.815	0.820	0.647	0.799
DSSNMF	0.607	0.638	0.671	0.491	0.855	0.642	0.597	0.395	0.740	0.790	0.383	0.826
RCNMF-CP	0.800	0.871	0.695	0.511	0.824	**0.877**	0.629	0.411	0.863	0.834	0.558	0.847
S^{3}NMF	0.761	0.804	0.679	0.558	0.804	0.744	0.522	0.360	0.831	0.847	0.675	0.740
S^{4}NMF	**0.918**	**0.898**	**0.776**	**0.595**	0.855	0.850	**0.698**	0.436	**0.891**	**0.893**	**0.764**	**0.858**

Table 3
ACC results on real-world datasets: The best result is highlighted in **bold** style, while underline style indicates the second-best.

Method	Chart	Iris	Seeds	Yale	ORL	UMIST	MNIST	Glass	Zoo	Wine	BC	coil20
NMF	0.500	0.716	0.857	0.357	0.405	0.416	0.464	0.457	0.803	0.632	0.801	0.471
SymNMF	0.673	0.673	0.762	0.479	0.540	0.471	0.604	0.607	0.822	0.753	0.931	0.537
ONMF-W	0.440	0.675	0.571	0.220	0.122	0.395	0.357	0.509	0.780	0.595	0.801	0.241
ONMF-H	0.600	0.846	0.733	0.409	0.580	0.506	0.561	0.433	0.895	0.904	0.787	0.596
CNMF	0.665	0.833	0.843	0.448	0.758	0.560	0.623	0.537	0.861	0.944	0.935	0.707
RCNMF	0.760	0.967	0.895	0.497	0.773	0.557	0.644	0.556	0.861	0.949	0.942	0.717
CPSNMF	0.665	0.627	0.910	0.461	0.650	0.579	0.385	0.551	0.891	0.933	0.893	0.720
RSNMF	0.703	0.920	0.900	0.436	0.643	0.600	0.606	0.509	0.881	0.927	0.917	0.697
DSNMF	0.667	0.953	0.914	0.515	0.720	0.569	0.555	0.645	0.871	0.944	0.930	0.723
DSSNMF	0.745	0.807	0.895	0.455	**0.778**	0.602	0.715	0.603	0.871	0.938	0.856	0.725
RCNMF-CP	0.667	0.960	0.900	0.485	0.713	**0.805**	0.686	0.570	0.871	0.955	0.912	0.764
S^{3}NMF	0.800	0.933	0.876	0.527	0.680	0.657	0.634	0.631	0.921	0.955	0.944	0.640
S^{4}NMF	**0.952**	**0.973**	**0.933**	**0.594**	0.773	0.803	**0.816**	**0.668**	**0.941**	**0.972**	**0.963**	**0.781**

Table 4
ARI results on real-world datasets: The best result is highlighted in **bold** style, while underline style indicates the second-best.

Method	Chart	Iris	Seeds	Yale	ORL	UMIST	MNIST	Glass	Zoo	Wine	BC	coil20
NMF	0.391	0.550	0.634	0.137	0.216	0.259	0.232	0.223	0.778	0.490	0.358	0.398
SymNMF	0.479	0.561	0.503	0.235	0.357	0.311	0.365	0.180	0.485	0.460	0.743	0.402
ONMF-W	0.233	0.679	0.269	0.093	0.071	0.211	0.135	0.252	0.683	0.578	0.358	0.169
ONMF-H	0.452	0.640	0.412	0.170	0.329	0.331	0.326	0.227	0.645	0.734	0.329	0.506
CNMF	0.595	0.636	0.597	0.232	0.622	0.327	0.392	0.126	0.509	0.834	0.754	0.583
RCNMF	0.617	0.904	0.720	0.273	0.610	0.385	0.417	0.132	0.727	0.847	0.779	0.572
CPSNMF	0.612	0.387	0.748	0.221	0.462	0.355	0.149	0.159	0.829	0.803	0.611	0.612
RSNMF	0.487	0.787	0.725	0.233	0.220	0.380	0.365	0.143	0.742	0.790	0.694	0.552
DSNMF	0.617	0.866	0.788	0.282	0.541	0.349	0.303	0.310	0.804	0.830	0.736	0.625
DSSNMF	0.512	0.584	0.715	0.195	0.623	0.397	0.493	0.236	0.570	0.815	0.502	0.548
RCNMF-CP	0.615	0.886	0.718	0.272	0.537	**0.750**	0.523	0.277	0.849	0.864	0.678	0.691
S^{3}NMF	0.657	0.818	0.673	0.305	0.519	0.503	0.410	0.211	0.678	0.865	0.786	0.502
S^{4}NMF	**0.893**	**0.922**	**0.809**	**0.360**	**0.652**	0.738	**0.634**	**0.330**	**0.945**	**0.915**	**0.857**	**0.714**

class at random to be the labeled data, and the rest were utilized as the unlabeled data. (Remember that the ORL has only ten images in each class; based on the work of [13], we picked 20% of the ORL samples as the labeled data). For the proposed model and the state-of-the-art models that were compared, all hyper-parameters were selected from {0, 0.001, 0.01, 0.1, 1, 10, 100, 1000}. For a fair comparison, the p-nearest neighbor graph with $p = 5$ was adopted for all models having a Laplacian graph structure. For all methods, the parameter k for low-dimensional representation was set to the number of classes. Eventually, we set the parameters for the proposed method to $\tau = 2$ and $b = 20$. Based on Algorithm 1, when $b = 20$, the S^{4}NMF model could provide 20 distinct clustering solutions, and we reported the average of its performance.

4.5. Results and analysis

Tables 2–6 illustrate the clustering performance of all compared methods, and the proposed method on the twelve datasets, with the best performance for each metric highlighted in bold and the second-best, underlined. It is clear from Tables 2–6 that:

- As can be observed from these clustering results, it is clear that the proposed method obtains the highest clustering performance on almost all datasets, as shown by the boldface in the tables, demonstrating that S^{4}NMF can find more appropriate clusters than other approaches.
- In comparison to the second-best methods, S^{4}NMF achieves an average improvement of 0.028, 0.034, and 0.06 in terms of NMI, ACC, ARI, F1, and Purity respectively. More specifically, compared to the best unsupervised, self-supervised, and semi-supervised clustering algorithms on the Chart and MNIST datasets, our approach significantly increases the ACC value from 0.8 to 0.952 and from 0.715 to 0.816, respectively.
- The proposed method achieves the best performance on all evaluation metrics under 48 out of 60 cases and the second

Table 5
F1 results on real-world datasets: The best result is highlighted in **bold** style, while underline style indicates the second-best.

Method	Chart	Iris	Seeds	Yale	ORL	UMIST	MNIST	Glass	Zoo	Wine	BC	coil20
NMF	0.438	0.704	0.853	0.272	0.384	0.310	0.336	0.292	0.556	0.631	0.784	0.385
SymNMF	0.649	0.667	0.769	0.411	0.515	0.360	0.533	0.316	0.593	0.747	0.925	0.426
ONMF-W	0.318	0.584	0.479	0.153	0.082	0.268	0.291	0.303	0.581	0.499	0.781	0.174
ONMF-H	0.534	0.838	0.740	0.319	0.494	0.381	0.483	0.368	0.726	0.908	0.774	0.539
CNMF	0.549	0.830	0.849	0.390	0.726	0.468	0.598	0.314	0.630	0.945	0.928	0.681
RCNMF	0.710	0.967	0.896	0.448	0.751	0.434	0.613	0.407	0.681	0.946	0.934	0.678
CPSNMF	0.553	0.546	0.905	0.375	0.559	0.453	0.281	0.322	**0.831**	0.941	0.875	0.664
RSNMF	0.644	0.920	0.900	0.361	0.677	0.506	0.580	0.275	0.761	0.930	0.904	0.682
DSNMF	0.554	0.953	0.918	0.508	0.698	0.480	0.489	0.429	0.743	0.944	0.913	0.700
DSSNMF	0.639	0.802	0.896	0.410	**0.783**	0.559	0.712	0.437	0.723	0.940	0.845	0.708
RCNMF-CP	0.555	0.960	0.897	0.454	0.670	0.730	0.642	0.425	0.702	0.955	0.907	0.734
S^{3}NMF	0.793	0.933	0.877	0.482	0.646	0.590	0.595	0.366	0.760	0.956	0.940	0.562
S^{4}NMF	**0.951**	**0.973**	**0.928**	**0.531**	0.741	**0.744**	**0.793**	**0.447**	0.762	**0.973**	**0.958**	**0.738**

Table 6
Purity results on real-world datasets: The best result is highlighted in **bold** style, while underline style indicates the second-best.

Method	Chart	Iris	Seeds	Yale	ORL	UMIST	MNIST	Glass	Zoo	Wine	BC	coil20
NMF	0.747	0.807	0.852	0.388	0.498	0.428	0.499	**0.850**	0.842	0.697	0.801	0.589
SymNMF	0.717	0.860	0.762	0.509	0.583	0.457	0.632	0.486	0.673	0.787	0.930	0.604
ONMF-W	0.733	**0.987**	0.900	0.497	0.583	0.362	0.383	0.846	0.842	0.747	0.801	0.760
ONMF-H	0.737	0.847	0.738	0.545	0.585	0.534	0.566	0.654	0.891	0.904	0.779	0.662
CNMF	0.880	0.830	0.848	0.515	**0.785**	0.550	0.653	0.491	0.772	0.944	0.935	0.735
RCNMF	0.900	0.967	0.895	0.558	0.773	0.554	0.652	0.411	0.861	0.949	0.940	0.753
CPSNMF	0.897	0.780	0.905	0.418	0.645	0.527	0.398	0.551	0.901	0.933	0.893	0.707
RSNMF	0.805	0.920	0.900	0.564	0.723	0.631	0.594	0.780	0.881	0.927	0.917	0.759
DSNMF	0.897	0.953	0.919	0.545	0.725	0.546	0.540	0.696	0.891	0.944	0.923	0.736
DSSNMF	0.732	0.807	0.895	0.467	0.778	0.577	0.711	0.598	0.762	0.938	0.856	0.734
RCNMF-CP	0.898	0.960	0.900	0.552	0.733	**0.868**	0.721	0.664	0.901	0.955	0.912	0.827
S^{3}NMF	0.818	0.933	0.876	0.552	0.683	0.649	0.650	0.514	0.762	0.955	0.944	0.664
S^{4}NMF	**0.952**	0.973	**0.929**	**0.612**	0.778	0.803	**0.796**	0.650	**0.941**	**0.972**	**0.961**	**0.839**

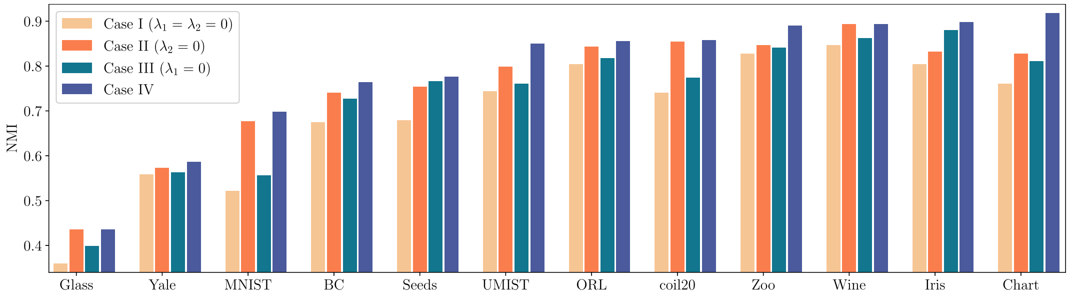


Fig. 2. Ablation study on the effect of regularization terms of proposed method based on NMI measure.

best performance under 10 out of the remaining 12 cases regarding all compared methods, indicating that our method performs very well compared to all other methods.

- Generally, through experiments, we find that the performance of the compared approaches is not stable across different datasets. For example, RCNMF-CP performs well in partitioning UMIST but not BC. DSNMF favors seeds and Glass over UMIST and MNIST. The performance of DSSNMF on the ORL is much higher compared to Yale and BC. In contrast, the robustness of our method is demonstrated by the fact that it consistently generates the best or almost the best performance across these twelve datasets.

Two types of relations, namely the dissimilarity and the similarity between labeled data, are employed in the proposed model to regularize the factorization. Here, we examined their effect on the performance of clustering. The NMIs, ACCs, ARIs, F1s, and Puritys of the proposed model in different settings are shown in Table 7 and Figs. 2–6. Particularly, case I ($\lambda_1 = \lambda_2 = 0$) means the proposed method without regularization, Case II ($\lambda_2 = 0$) indicates that only the dissimilarity regularization is used, while case III ($\lambda_1 = 0$) indicates only the similarity is used, and finally Case IV is S^{4}NMF model with similarity and dissimilarity regularizations. From Table 7 and Figs. 2–6, it is possible to derive the following conclusions:

- On all 12 datasets, Case II outperforms Case I in terms of NMI, ACC, ARI, F1, and Purity proving the efficiency of the dissimilarity regularizer.
- Case III demonstrates the efficiency of the similarity regularizer by producing better results than Case I across all 12 datasets for NMI, ACC, ARI, F1, and purity.
- On most datasets, Case II produces better NMI, ACC, ARI, F1, and Purity than Case III; however, on Iris and Seeds, Case III provides higher NMI, ACC, ARI, F1, and Purity than Case II; which implies that the dissimilarity regularization in the proposed model is mostly more significant than the similarity regularizer.
- Finally, we can draw the conclusion based on Case IV that both regularizers contribute significantly to our model and complement one another.

Table 7
Ablation study on the effect of regularization terms of proposed method, while **bold** show the best performance and underline style indicates the second-best performance.

	Case		Dataset											
			Chart	Iris	Seeds	Yale	ORL	UMIST	MNIST	Glass	Zoo	Wine	BC	Coil20
NMI	I	$(\lambda_1 = \lambda_2 = 0)$	0.761	0.804	0.679	0.558	0.804	0.744	0.522	0.360	0.831	0.847	0.675	0.740
	II	$(\lambda_2 = 0)$	0.827	0.832	0.754	0.573	0.843	0.798	0.677	**0.436**	0.855	**0.893**	0.740	0.854
	III	$(\lambda_1 = 0)$	0.811	0.880	0.766	0.563	0.817	0.760	0.556	0.399	0.851	0.862	0.727	0.774
	IV	proposed	**0.918**	**0.898**	**0.776**	**0.595**	**0.855**	**0.850**	**0.698**	**0.436**	**0.891**	**0.893**	**0.764**	**0.858**
ACC	I	$(\lambda_1 = \lambda_2 = 0)$	0.800	0.933	0.876	0.527	0.680	0.657	0.634	0.631	0.921	0.955	0.944	0.640
	II	$(\lambda_2 = 0)$	0.837	0.947	0.919	0.570	0.753	0.730	0.790	**0.668**	**0.941**	**0.972**	0.958	0.774
	III	$(\lambda_1 = 0)$	0.867	0.967	0.929	0.558	0.710	0.701	0.671	0.654	0.931	0.961	0.954	0.710
	IV	proposed	**0.952**	**0.973**	**0.933**	**0.594**	**0.773**	**0.803**	**0.816**	**0.668**	**0.941**	**0.972**	**0.963**	**0.781**
ARI	I	$(\lambda_1 = \lambda_2 = 0)$	0.657	0.818	0.673	0.306	0.519	0.5032	0.410	0.211	0.678	0.865	0.786	0.502
	II	$(\lambda_2 = 0)$	0.734	0.851	0.776	0.325	0.617	0.625	0.599	**0.330**	0.840	**0.915**	0.837	0.704
	III	$(\lambda_1 = 0)$	0.739	0.904	0.800	0.318	0.555	0.549	0.462	0.264	0.842	0.882	0.824	0.590
	IV	proposed	**0.893**	**0.922**	**0.809**	**0.360**	**0.652**	**0.738**	**0.634**	**0.330**	**0.945**	**0.915**	**0.857**	**0.714**
F1	I	$(\lambda_1 = \lambda_2 = 0)$	0.793	0.933	0.877	0.482	0.646	0.590	0.595	0.366	0.760	0.956	0.940	0.562
	II	$(\lambda_2 = 0)$	0.832	0.947	0.914	0.531	0.731	0.701	0.789	**0.447**	0.762	**0.973**	0.954	0.717
	III	$(\lambda_1 = 0)$	0.860	0.967	**0.928**	0.517	0.683	0.641	0.641	0.371	0.778	0.962	0.951	0.679
	IV	proposed	**0.951**	**0.973**	**0.928**	**0.541**	**0.741**	**0.744**	**0.793**	**0.447**	**0.793**	**0.973**	**0.958**	**0.738**
Purity	I	$(\lambda_1 = \lambda_2 = 0)$	0.818	0.933	0.876	0.552	0.683	0.649	0.650	0.514	0.762	0.955	0.944	0.664
	II	$(\lambda_2 = 0)$	0.902	0.947	0.914	**0.612**	**0.778**	0.734	0.792	**0.650**	0.891	**0.972**	0.958	0.808
	III	$(\lambda_1 = 0)$	0.862	0.967	**0.929**	0.467	0.715	0.713	0.699	0.556	0.851	0.961	0.954	0.748
	IV	proposed	**0.952**	**0.973**	**0.929**	**0.612**	**0.778**	**0.803**	**0.796**	**0.650**	**0.941**	**0.972**	**0.961**	**0.839**

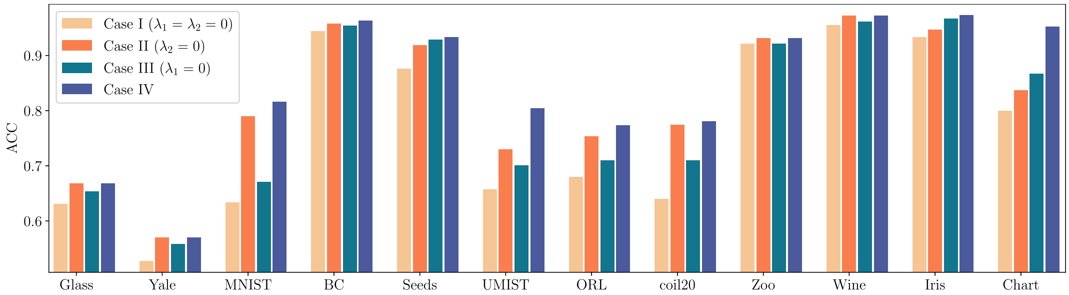


Fig. 3. Ablation study on the effect of regularization terms of proposed method based on ACC measure.

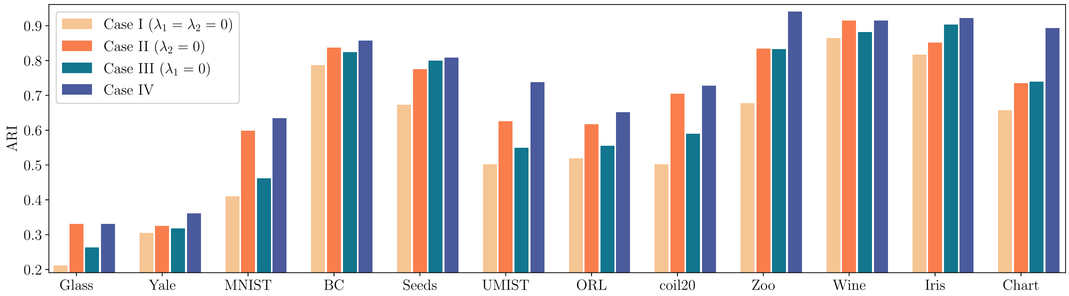


Fig. 4. Ablation study on the effect of regularization terms of proposed method based on ARI measure.

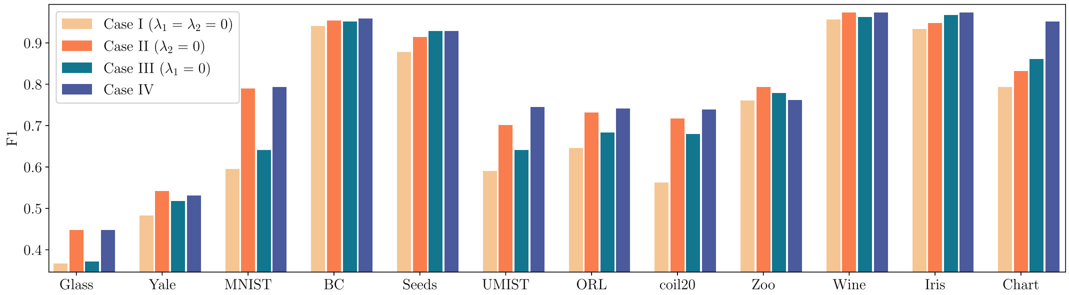


Fig. 5. Ablation study on the effect of regularization terms of proposed method based on F1 measure.

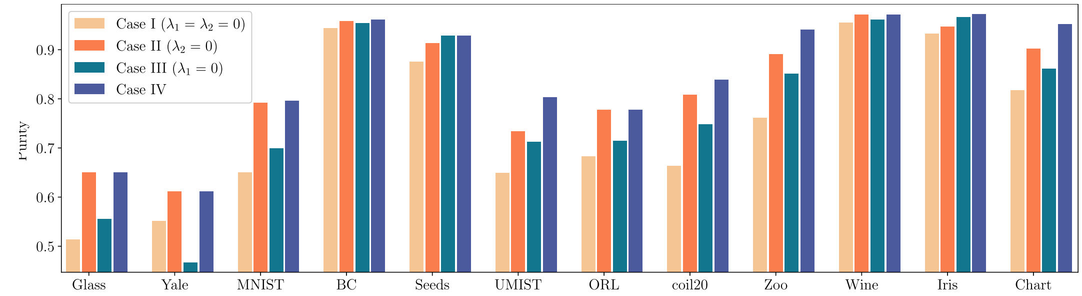
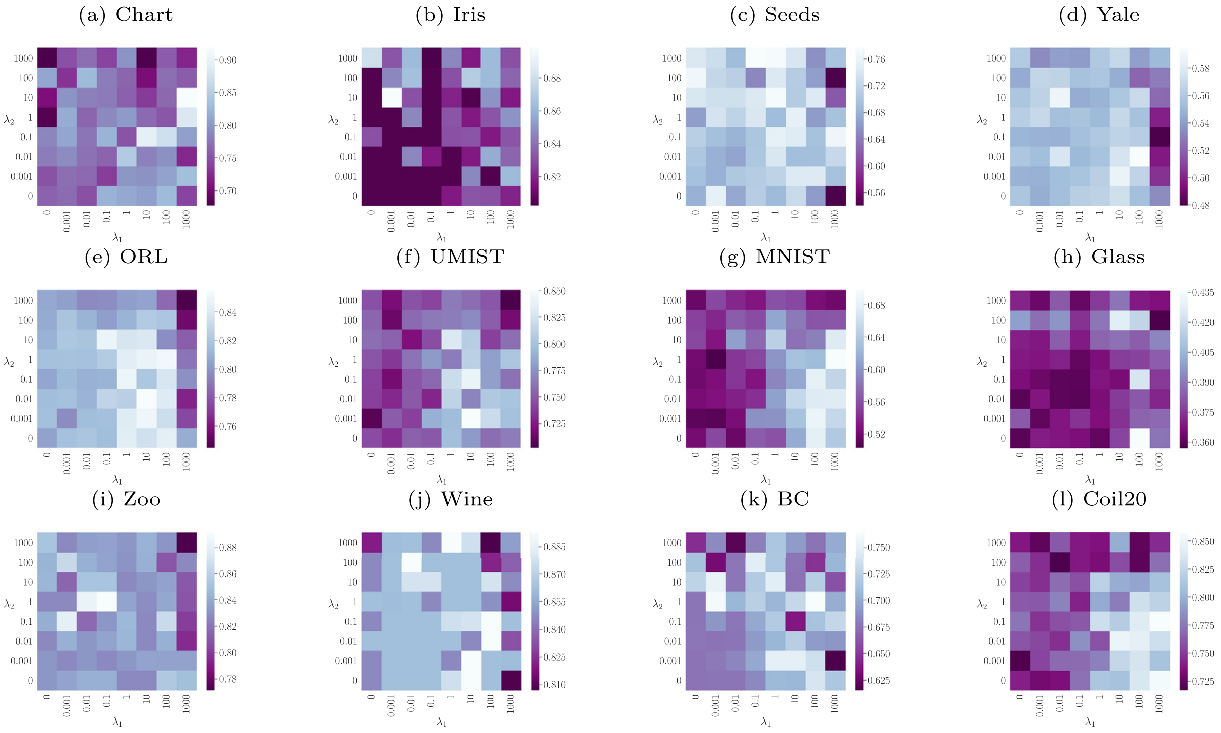


Fig. 6. Ablation study on the effect of regularization terms of proposed method based on Purity measure.


(a) Chart (b) Iris (c) Seeds (d) Yale

(e) ORL (f) UMIST (g) MNIST (h) Glass

(i) Zoo (j) Wine (k) BC (l) Coil20

Fig. 7. Parameter analysis (in terms of NMI) on the parameters λ_1 and λ_2, where the lighter color describes the higher NMI values.

4.6. Parameter analysis

In this section, we analyzed the influence of the hyper-parameters (i.e., λ_1 and λ_2) of the proposed method on semi-supervised clustering performance. Figs. 7–9 illustrate the NMI, ACC, and ARI of the proposed method with various λ_1 and λ_2 on all the twelve datasets. Note that these figures are heatmaps, where the measures are represented by color, and the two axes correspond to the parameters λ_1 and λ_2. The following conclusions are derived from Figs. 7–9.

Both λ_1 and λ_2 with relatively large values result in low performance. For UMIST, MNIST, and Coil20, a higher λ_1 and small λ_2 usually generate better performance, which indicates that in these datasets, the dissimilarity relation between the objects is more important. For ORL and Yale, we can see that a higher value for parameter λ_1 is still effective, and the results are not very sensitive to parameter λ_2. For Chart, Iris, and Glass, setting both parameters is essential in achieving good performance. Meanwhile, for Seeds, Zoo, Wine, and BC datasets, vice versa. As a result, in the suggested model, λ_1 and λ_2 are complementary.

4.7. Convergence analysis

In this section, we empirically analyze the convergence of the S^{4}NMF algorithm. Fig. 10 illustrates the objective value w.r.t. the number of iterations on the Iris dataset. The x-axis and the y-axis in the diagrams drawn in this figure show the iterations and the objective value of the function (12), respectively. As described in Section 3, the proposed algorithm is defined as an ensemble of SSNMF models introduced in (11). Therefore, the speed of convergence of this model has been investigated in Fig. 10(a). In this experiment, five independent ensemble constituents with different initializations have been run in parallel, and the loss value of each model has been reported in each iteration. It is shown that, under the update rule (20), the objective of each Semi-Supervised NMF (lines 3–8 of Algorithm 1) quickly converges as a monotonically decreasing sequence of nonnegative values. In the following, we have analyzed the convergence of the proposed algorithm with five ensemble constituents, 200 inner iterations, and five outer iterations. According to Fig. 10(b), we can see the fluctuations after every 200 Inner iterations. These fluctuations are due to the updating input affinity matrix (14) in each outer iteration, which is

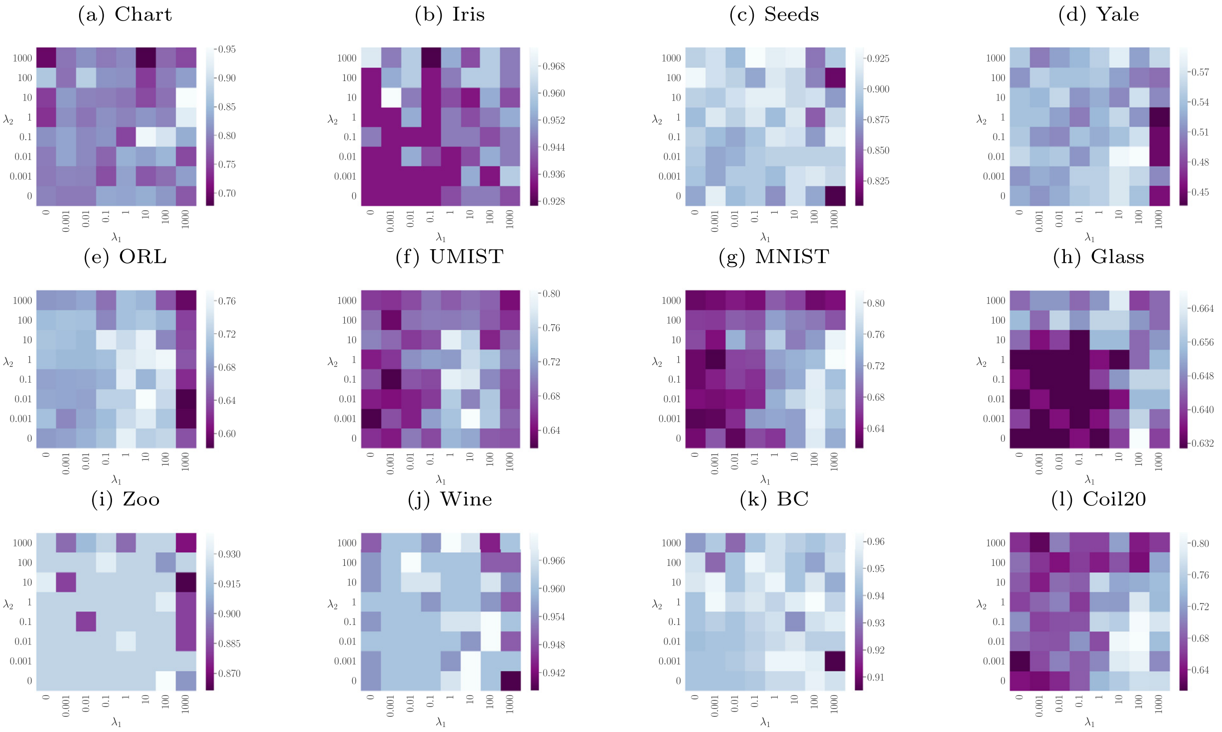


Fig. 8. Parameter analysis (in terms of ACC) on the parameters λ_1 and λ_2, where the lighter color describes the higher ACC values.

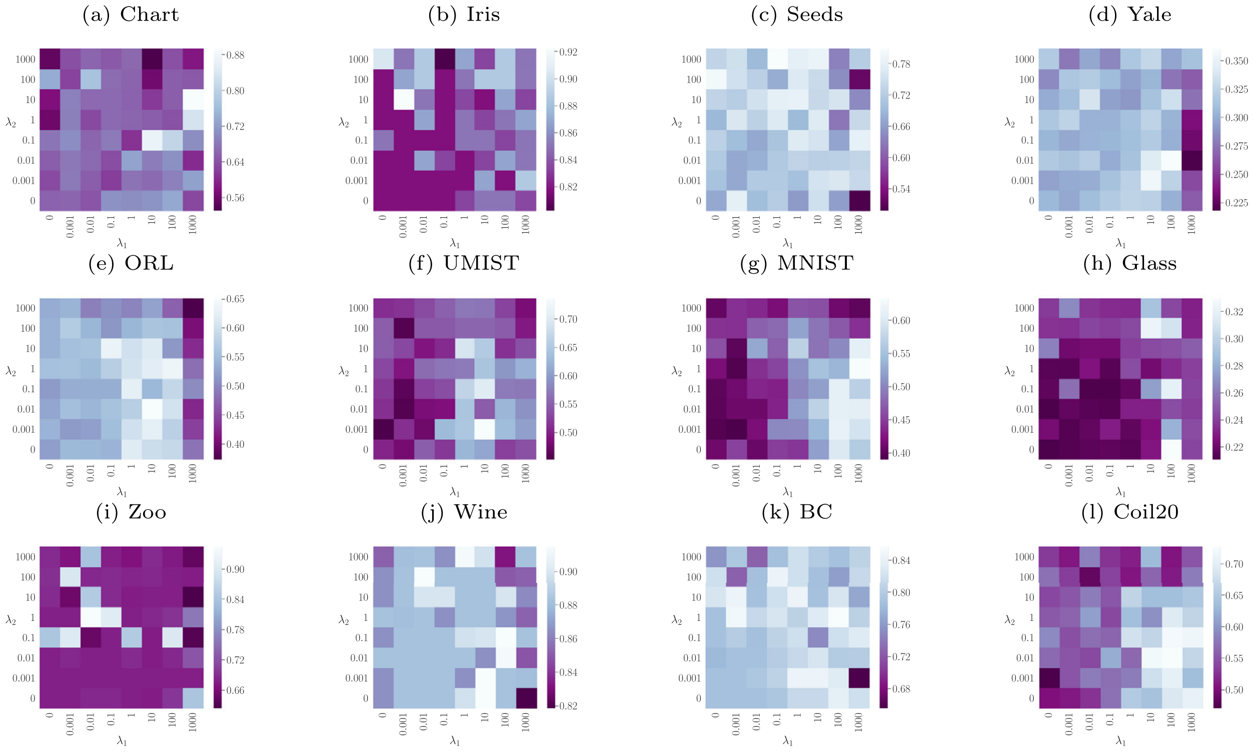
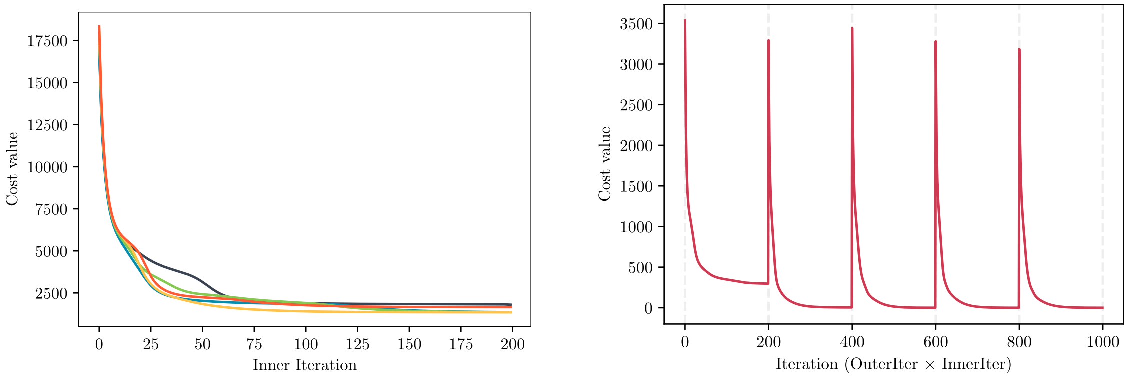


Fig. 9. Parameter analysis (in terms of ARI) on the parameters λ_1 and λ_2, where the lighter color describes the higher ARI values.



(a) (b)

Fig. 10. Convergence analysis of S^{4}NMF model on the Iris datasets. (a) Convergence curve of 5 SSNMF models (lines 3–8 of Algorithm 1) with different initialization. (b) Convergence curve of Algorithm 1. After every 200 Inner iterations, the input affinity matrix is updated.

in line with the self-supervised learning framework. Nevertheless, the proposed algorithm is still successful in decreasing the objective value. Therefore, from the results obtained for the convergence of the proposed method, it can be concluded that Algorithm 1 is convergent.

5. Conclusion

This paper introduced the S^{4}NMF, a semi-supervised learning method that integrates concepts and components from current self-supervised and semi-supervised paradigms in a unified Symmetric Nonnegative Matrix Factorization model. Our model benefits from ensemble learning to make a self-supervised model and utilizes powerful semi-supervised learning to form more distinct clusters. More specifically, in addition to similarity and dissimilarity constraints for learning information from each labeled sample, we introduced a perspective of self-supervision to exploit the pseudo-labels from unlabeled samples. In addition, we introduced an effective multiplicative update algorithm for optimizing S^{4}NMF and theoretically proved its convergence. Through extensive experiments on semi-supervised clustering, we found that the proposed model outperformed other methods in all evaluated settings. Nevertheless, the diversity of the ensemble model is low and the learning capacity can be improved by combining different factorization extensions. In future work, to improve the generalization capability, we can increase diversity in ensemble members by adding diverse regularization terms to the model. Separately, we want to incorporate other ideas from the contrastive self-supervised learning literature into semi-supervised methods. Finally, besides clustering applications, we are interested in investigating the efficiency of the proposed method in other learning tasks.

Declaration of Competing Interest

The authors certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

The authors report the following details of affiliation or involvement in an organization or entity with a financial or non-financial interest in the subject matter or materials discussed in this manuscript. Please specify the nature of the conflict on a separate sheet of paper if the space below is inadequate.

Data availability

Data will be made available on request.

References

[1] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828, doi:10.1109/TPAMI.2013.50.
[2] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (6755) (1999) 788–791, doi:10.1038/44565.
[3] D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, Adv. Neural Inf. Process. Syst. (NeurIPS) 13 (2000).
[4] R. Duda, P. Hart, D. Stork, Pattern classification, Wiley, 2012.
[5] I. Manohar, B. Bhikkaji, G. Ganesan, A qr decomposition approach to factor modelling, Signal Process. 132 (2017) 19–28, doi:10.1016/j.sigpro.2016.05.017.
[6] X. Li, Y. Pang, Deterministic column-based matrix decomposition, IEEE Trans. Knowl. Data Eng. 22 (1) (2009) 145–149, doi:10.1109/TKDE.2009.64.
[7] P. Belhumeur, J. Hespanha, D. Kriegman, Eigenfaces vs. fisherfaces: recognition using class specific linear projection, IEEE Trans. Pattern Anal. Mach. Intell. 19 (7) (1997) 711–720, doi:10.1109/34.598228.
[8] Y. Pang, S. Wang, Y. Yuan, Learning regularized lda by clustering, IEEE Trans. Neural Netw. Learn. Syst. 25 (12) (2014) 2191–2201, doi:10.1109/TNNLS.2014.2306844.
[9] S. Wold, K. Esbensen, P. Geladi, Principal component analysis, Chemometric. Intell. Lab. Syst. 2 (1) (1987) 37–52, doi:10.1016/0169-7439(87)80084-9.
[10] P. Comon, Independent component analysis, a new concept? Signal Process. 36 (3) (1994) 287–314, doi:10.1016/0165-1684(94)90029-9. Higher Order Statistics
[11] N. Gillis, Nonnegative matrix factorization, SIAM, 2020, doi:10.1137/1.9781611976410.bm.
[12] C. Ding, X. He, H.D. Simon, On the equivalence of nonnegative matrix factorization and spectral clustering, in: SIAM International Conference on Data Mining (SDM), 2005, pp. 606–610, doi:10.1137/1.9781611972757.70.
[13] Z. Li, J. Tang, X. He, Robust structured nonnegative matrix factorization for image representation, IEEE Trans. Neural Netw. Learn. Syst. 29 (5) (2017) 1947–1960, doi:10.1109/TNNLS.2017.2691725.
[14] H. Liu, Z. Wu, X. Li, D. Cai, T.S. Huang, Constrained nonnegative matrix factorization for image representation, IEEE Trans. Pattern Anal. Mach. Intell. 34 (7) (2011) 1299–1311, doi:10.1109/TPAMI.2011.217.
[15] S. Peng, W. Ser, B. Chen, Z. Lin, Robust semi-supervised nonnegative matrix factorization for image clustering, Pattern Recognit. 111 (2021) 107683, doi:10.1016/j.patcog.2020.107683.
[16] D. Wang, X. Gao, X. Wang, Semi-supervised nonnegative matrix factorization via constraint propagation, IEEE Trans. Cybern. 46 (1) (2015) 233–244, doi:10.1109/TCYB.2015.2399533.
[17] A. Kolesnikov, X. Zhai, L. Beyer, Revisiting self-supervised visual representation learning, in: IEEE/CVF conference on computer vision and pattern recognition (CVPR), 2019, pp. 1920–1929, doi:10.1109/CVPR.2019.00202.
[18] P.H. Le-Khac, G. Healy, A.F. Smeaton, Contrastive representation learning: a framework and review, IEEE Access 8 (2020) 193907–193934, doi:10.1109/ACCESS.2020.3031549.
[19] T.G. Dietterich, Ensemble methods in machine learning, in: International Workshop on Multiple Classifier Systems, Springer, 2000, pp. 1–15, doi:10.1007/3-540-45014-9_1.

[20] Y. Bian, H. Chen, When does diversity help generalization in classification ensembles? IEEE Trans. Cybern. (2021), doi:10.1109/TCYB.2021.3053165.
[21] C.-G. Li, C. You, R. Vidal, Structured sparse subspace clustering: a joint affinity learning and subspace clustering framework, IEEE Trans. Image Process. 26 (6) (2017) 2988–3001, doi:10.1109/TIP.2017.2691557.
[22] P. Paatero, U. Tapper, Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values, Environmetrics 5 (2) (1994) 111–126, doi:10.1002/env.3170050203.
[23] P.O. Hoyer, Non-negative matrix factorization with sparseness constraints, J. Mach. Learn. Res. 5 (9) (2004).
[24] C. Ding, T. Li, W. Peng, H. Park, Orthogonal nonnegative matrix t-factorizations for clustering, in: ACM SIGKDD International conference on Knowledge Discovery and Data mining (KDD), 2006, pp. 126–135, doi:10.1145/1150402.1150420.
[25] D. Cai, X. He, J. Han, T.S. Huang, Graph regularized nonnegative matrix factorization for data representation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2010) 1548–1560, doi:10.1109/TPAMI.2010.231.
[26] D. Kuang, C. Ding, H. Park, Symmetric nonnegative matrix factorization for graph clustering, in: SIAM International Conference on Data Mining (ICDM), SIAM, 2012, pp. 106–117, doi:10.1137/1.9781611972825.10.
[27] X. Zhu, Z. Ghahramani, J.D. Lafferty, Semi-supervised learning using gaussian fields and harmonic functions, in: International conference on Machine learning (ICML), 2003, pp. 912–919.
[28] H. Lee, J. Yoo, S. Choi, Semi-supervised nonnegative matrix factorization, IEEE Signal Process. Lett. 17 (1) (2009) 4–7, doi:10.1109/LSP.2009.2027163.
[29] M. Babaee, S. Tsoukalas, M. Babaee, G. Rigoll, M. Datcu, Discriminative nonnegative matrix factorization for dimensionality reduction, Neurocomputing 173 (2016) 212–223, doi:10.1016/j.neucom.2014.12.124.
[30] Z. Xing, M. Wen, J. Peng, J. Feng, Discriminative semi-supervised non-negative matrix factorization for data clustering, Eng. Appl. Artif. Intell. 103 (2021) 104289, doi:10.1016/j.engappai.2021.104289.
[31] Y. He, H. Lu, S. Xie, Semi-supervised non-negative matrix factorization for image clustering with graph laplacian, Multimed. Tools Appl. 72 (2) (2014) 1441–1463, doi:10.1007/s11042-013-1465-1.
[32] J. Wang, F. Tian, C.H. Liu, X. Wang, Robust semi-supervised nonnegative matrix factorization, in: International joint conference on neural networks (IJCNN), IEEE, 2015, pp. 1–8, doi:10.1109/IJCNN.2015.7280422.
[33] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, B.W. Schuller, A deep matrix factorization method for learning attribute representations, IEEE Trans. Pattern Anal. Mach. Intell. 39 (3) (2016) 417–429, doi:10.1109/TPAMI.2016.2554555.
[34] Y. Meng, R. Shang, F. Shang, L. Jiao, S. Yang, R. Stolkin, Semi-supervised graph regularized deep nmf with bi-orthogonal constraints for data representation, IEEE Trans. Neural Netw. Learn. Syst. 31 (9) (2019) 3245–3258, doi:10.1109/TNNLS.2019.2939637.
[35] Y. Jia, H. Liu, J. Hou, S. Kwong, Semisupervised adaptive symmetric non-negative matrix factorization, IEEE Trans. Cybern. 51 (5) (2020) 2550–2562, doi:10.1109/TCYB.2020.2969684.
[36] Y. Jia, S. Kwong, J. Hou, W. Wu, Semi-supervised non-negative matrix factorization with dissimilarity and similarity regularization, IEEE Trans. Neural Netw. Learn. Syst. 31 (7) (2020) 2510–2521, doi:10.1109/TNNLS.2019.2933223.
[37] Y. Jia, H. Liu, J. Hou, S. Kwong, Q. Zhang, Self-supervised symmetric non-negative matrix factorization, IEEE Trans. Circuits Syst. Video Technol. (2021), doi:10.1109/TCSVT.2021.3129365.
[38] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, G.E. Hinton, Big self-supervised models are strong semi-supervised learners, in: Advances in Neural Information Processing Systems (NeurIPS), volume 33, 2020, pp. 22243–22255.
[39] X. Zhai, A. Oliver, A. Kolesnikov, L. Beyer, S4l: Self-supervised semi-supervised learning, in: EEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 1476–1485, doi:10.1109/ICCV.2019.00156.
[40] D. Huang, C.-D. Wang, J.-H. Lai, Locally weighted ensemble clustering, IEEE Trans. Cybern. 48 (5) (2018) 1460–1473, doi:10.1109/TCYB.2017.2702343.

Jovan Chavoshinejad is currently working towards M.Sc. in the department of Computer Engineering, University of Kurdistan. Her research interests include machine learning, semi-supervised learning, self-supervised learning, adversarial training, and deep learning. She received her bachelor's degree in Software Engineering from the University of Kurdistan in 2017.

Seyed Amjad Seyedi is a graduate research assistant at the University of Kurdistan working on deep learning (from optimization and generalization aspects). He received his Master's in Artificial Intelligence from the Department of Computer Engineering at the University of Kurdistan in 2018. His work mainly focused on matrix factorization and low-rank approximation.

Fardin Akhlaghian Tab is the associate professor of Computer engineering at the University of Kurdistan, and his research focuses on machine learning and computer vision. He did his Ph.D. in Computer Vision at the University of Wollongong in 2005. He holds a master's degree from Tehran University of Tarbiat Modarres in 1992.

Navid Salahian is currently working towards M.Sc. in the department of Computer Engineering at the University of Kurdistan. His research interests include machine learning, representation learning, unsupervised learning, deep learning, and multi-view/multi-modal learning. He received a bachelor's degree in Software Engineering from the University of Kurdistan in 2020.