Pattern Recognition

Deep Asymmetric Nonnegative Matrix Factorization for Graph Clustering

--Manuscript Draft--

Manuscript Number:	PR-D-23-00613R1
Article Type:	Full Length Article
Section/Category:	Machine learning
Keywords:	Nonnegative matrix factorization; deep learning; Graph clustering; directed graph
Corresponding Author:	Fardin Akhlaghian Tab, Ph.D. University of Kurdistan Sanandaj, IRAN, ISLAMIC REPUBLIC OF
First Author:	Akram Hajiveiseh
Order of Authors:	Akram Hajiveiseh
	Seyed Amjad Seyedi
	Fardin Akhlaghian Tab, Ph.D.
Abstract:	Graph clustering is a fundamental technique in machine learning that has widespread applications in various fields. Deep Nonnegative Matrix Factorization (DNMF) was recently emerged to cope with the extraction of several layers of features, and it has been demonstrated to achieve remarkable results on unsupervised tasks. While DNMF has been applied for analyzing graphs, the effectiveness of the current DNMF approaches for graph clustering is generally unsatisfactory: these methods are intrinsically data representation models, and their objective functions do not capture cluster structures, also ignores direction which is crucial in the directed graph clustering problems. To overcome these downsides, this paper proposes a graph-specific DNMF model based on the Asymmetric NMF which can handle undirected and directed graphs. Inspired by hierarchical graph clustering and graph summarization approaches, the Deep Asymmetric Nonnegative Matrix Factorization (DAsNMF) is introduced for the directed graph clustering problem. In a pseudo-hierarchical clustering setting, DAsNMF decomposes the input graph to extract low-level to high-level node representations and graph representations (summarized graphs). In addition, the asymmetric cosine and PageRank-based similarities are imposed on the proposed model to preserve the local and global graph structures. The learning process is formulated as a unified optimization problem to jointly train representation learning model and clustering model. The extensive experimental studies validate the effectiveness of the proposed method on directed graphs.
Suggested Reviewers:	Jihong Pei, PhD Professor, Shenzhen University jhpei@szu.edu.cn Prof. Pei recently has published a related paper in the Pattern Recognition journal. https://doi.org/10.1016/j.patcog.2022.108984
	Richi Nayak, PhD Professor, Queensland University of Technology r.nayak@qut.edu.au Prof. Nayak recently has published a related paper in the Pattern Recognition journal. https://doi.org/10.1016/j.patcog.2022.108815

Deep Asymmetric Nonnegative Matrix Factorization for Graph Clustering

Akram Hajiveiseh, Seyed Amjad Seyedi, Fardin Akhlaghian Tab*

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

Abstract

Graph clustering is a fundamental technique in machine learning that has widespread applications in various fields. Deep Nonnegative Matrix Factorization (DNMF) was recently emerged to cope with the extraction of several layers of features, and it has been demonstrated to achieve remarkable results on unsupervised tasks. While DNMF has been applied for analyzing graphs, the effectiveness of the current DNMF approaches for graph clustering is generally unsatisfactory: these methods are intrinsically data representation models, and their objective functions do not capture cluster structures, also ignores direction which is crucial in the directed graph clustering problems. To overcome these downsides, this paper proposes a graph-specific DNMF model based on the Asymmetric NMF which can handle undirected and directed graphs. Inspired by hierarchical graph clustering and graph summarization approaches, the Deep Asymmetric Nonnegative Matrix Factorization (DAsNMF) is introduced for the directed graph clustering problem. In a pseudo-hierarchical clustering setting, DAsNMF decomposes the input graph to extract low-level to high-level node representations and graph representations (summarized graphs). In addition, the asymmetric cosine and PageRank-based similarities are imposed on the proposed model to preserve the local and global graph structures. The learning process is formulated as a unified optimization problem to jointly train representation learning model and clustering model. The extensive experimental studies validate the effectiveness of the proposed method on directed graphs.

Keywords: nonnegative matrix factorization, deep learning, graph clustering, directed graph

1. Introduction

Complex Network Analysis is an essential research field and application in recent years and is being widely rolled in many fields such as social sciences, recommendation, protein-protein analysis, public opinion analysis, metabolic networks. One of the fundamental elements of discrete mathematics is

*Corresponding author

Email addresses: akram.hajiveiseh@uok.ac.ir (Akram Hajiveiseh), amjadseyedi@uok.ac.ir (Seyed Amjad Seyedi), f.akhlaghian@uok.ac.ir (Fardin Akhlaghian Tab)

network analysis which is a representation of graph theory [1]. As networks have become immeasurable and large amounts of data can be modeled as a complex network, complex network analysis has attracted the attention of the scientific community and provides valuable explanations for complex network models, functions, and behaviors. Among several research topics in complex network analysis, graph clustering is a crucial task of separating entities into grouping items of components. Identifying clusters efficiently assists in identifying the characteristics of a given graph. Clusters are defined as a set of nodes that have similar characteristics, and grouping nodes in the context of graphs is based on their pairwise similarities. Researchers have developed a wide variety of strategies for clustering nodes [1].

There are many directed networks, including hyperlinked web structures, citation networks, and lateral gene transfer networks. The clustering issue in directed graphs, however, has received little interest in recent years. Clustering is much simpler in undirected graphs than in directed graphs. In contrast to undirected cases, directed graphs are characterized by utilizing asymmetrical matrices. As a consequence, spectral analysis in these graphs becomes much more complex [2]. It is common to ignore the edges' directedness and assume the graph is undirected. Taking the edges' directedness into consideration, on the contrary, can significantly increase the clustering performance, as it enables the handling of a considerable amount of relevant information. Furthermore, in some cases, ignoring edge-directedness might generate unexpected consequences. Designing algorithms for directed graphs is a challenging task. As an example, a directed network is characterized by asymmetrical matrices (adjacency matrix, Laplacian, etc.), which makes spectral analysis substantially more challenging. Few approaches are easily expandable from the undirected to a directed scenario. If not, the issue must be reconstructed from scratch [3].

Nonnegative matrix factorization (NMF) [4] is a strong data representation and interpretation technology that has lately gained popularity. It factorizes a nonnegative data matrix into two nonnegative matrices of lower rank. The first matrix represents the features or latent factors, and the second matrix represents the weights or coefficients of these features in the original data. NMF has been successfully applied in a wide range of research areas, document clustering [5], data clustering [6], graph clustering [7], link prediction [8], recommender systems [9], data representation [10], matrix completion [11], Imaging data analysis [12], and multi-view clustering [13]. Although basic NMF has been widely utilized by researchers for clustering and is generally said to have higher clustering quality than standard approaches such as k-means, it is not an intrinsic clustering method that can be applied to all situations. This is because capacities and limits of a clustering algorithm derive from its assumptions about cluster structure [14].

Numerous different extensions of NMF has been developed for clustering [15] because of its excellent explainability and relationship with data clustering and graph clustering [16]. Symmetric NMF

(SymNMF) [17] is a broad framework for graph clustering that inherits the advantages of NMF by requiring that the clustering assignment matrix be nonnegative. SymNMF, unlike NMF, is based on a matrix of data point similarity and factors a symmetric matrix of pairwise similarity values that do not have to be nonnegative. SymNMF captures the cluster structure implicit in graph representations more naturally than commonly used spectral clustering methods. A SymNMF model is created to show the symmetry of an undirected graph. This method operates by learning a special latent representation matrix $\boldsymbol{H}$ to create a low-rank approximation $\boldsymbol{H}\boldsymbol{H}^\top$ to the intended graph's adjacency matrix and accurately characterize its symmetry [17]. Since SymNMF is being used for undirected graphs, there is a sensible vacancy for applying NMF to directed graphs which are filled with asymmetric nonnegative matrix factorization (AsNMF) [18]. In particular, the AsNMF method utilizes the directed graph's plain adjacency matrix to facilitate clustering. To guarantee an asymmetrical reconstruction, AsNMF utilizes an additional asymmetric factor to the factorization ($\boldsymbol{A} \approx \boldsymbol{H}\boldsymbol{W}\boldsymbol{H}^\top$) which contains inter-cluster information.

Tosyali et al. [19] provided a regularized asymmetric nonnegative matrix factorization (RANMF) approach for clustering in directed networks. RANMF imposes the cosine node similarity information in form of an additional regularization term into the AsNMF. To improve the directed graph clustering and inspired by Semi-Nonnegative Matrix Factorization (Semi-NMF) [20], Regularized Asymmetric Semi-NMF (RAsNMF) is presented [21], which introduced a clustering approach by relaxing the nonnegativity constraints. Thus, the latent component produced is more suited for clustering. In addition, they utilized an asymmetric graph regularization to preserve the directness of the similarity information. Sun et al. [22] introduced the nonnegative symmetric encoder-decoder (NNSED) matrix factorization for undirected graph clustering. The proposed approach improves state-of-the-art latent factor models for graph clustering by explicitly combining an encoder and a decoder factorization terms into a unified cost function.

Real-world complex networks typically contain extremely complicated hierarchical characteristics, such as microscopic node similarities and macroscopic structure of clusters, which are extremely challenging to reveal using shallow approaches [23]. If we have a clear overview of the mapping among the original network and the cluster space, we may identify complex structures that cannot be translated by methods based on shallow NMF. Deep learning is well acknowledged in the field for its capability to hierarchically extract low-level to high-level semantic information from raw data [24]. Deep learning methods are highly effective in providing flexible solutions for achieving good performance in graph clustering to (1) learn nonlinear graph properties, (2) represent lower-dimensional graph embeddings preserving the complex graph structure, and (3) achieve better graph clustering from diverse information [25].

To take advantage of the deep learning properties in the matrix factorization model, and under

the influence of semi-NMF [20] on clustering, a novel deep framework called Deep semi-NMF [26] was developed, aiming to explore hidden representations of data. Deep semi-NMF provides a closer tie between deep matrix factorization and clustering [27]. Similarly, inspired by the Encoder-Decoder NMF (NNSED) [22] and the deep autoencoder's representation learning capabilities, Deep Autoencoder-like NMF (DANMF) [28] is developed for learning low-dimensional and nonlinear node representations in graph clustering task. Zhang et al. [29] developed DANMF and introduced a method for community discovery called Structural Deep Nonnegative Matrix Factorization (SDNMF). SDNMF utilizes both first-order and second-order similarities simultaneously to reduce the sparsity problem. This method extracts second order similarity by the pairwise cosine similarity of the nodes. Al-sharoa and Rahahleh [30] proposed a Deep Robust Autoencoder-like NMF (DRANMF) model to extract the community structure. DRANMF improved the robustness of the DANMF against noise and outliers by a $L_{2,1}$ loss function. He et al. [31] proposed the Deep Robust NMF (DRNMF) for network embedding, which utilizes the high-order proximity similarities of the network as the input matrix. To improve the robustness against noise, they used the $L_{2,1}$ norm for the objective function. Huang et al. [32] proposed the Modularized Deep Nonnegative Matrix Factorization (MDNMF), which preserves both the instinct community structure and the topology information. More precisely, MDNMF is a combination of deep and modularity-based NMFs. The deep NMF can extract the hidden features of the complex network, and the Modularized NMF can capture the topology. Deep Symmetric NMF (DSNMF) [33] is an extension of deep NMF. It incorporates multi-layer regularization techniques designed to enforce a symmetric penalty by constraining the relationship between matrices $\boldsymbol{W}$ and $\boldsymbol{H}$ at each layer, where $\boldsymbol{W}$ is forced to be equal to the transpose of $\boldsymbol{H}$.

The used deep NMF models in the existing graph clustering methods are based on the basic NMF which attempts to decompose an input data matrix hierarchically. These general data representation models cannot handle the links and local graph structure explicitly. Hence, most methods are proposed for graph embedding tasks, and cannot directly be considered as a graph clustering method. Therefore, all deep NMF-based methods, until this research, have not properties of a natural clustering method and lack capabilities in dealing with link directness. This paper proposes a specialized Deep NMF model which can extract complex structures of undirected and directed graphs. Inspired by the Asymmetric NMF and Deep NMF models, we propose the Deep Asymmetric NMF (DAsNMF) model which is analyzed in the directed graph clustering problem. The proposed model, similar to the hierarchical graph clustering method, has a multi-level latent graph summarization approach based on the deep factorization scheme which extracts a more abstract graph in each layer. In addition, to preserve the local and global structures of the directed graph, two tailored asymmetric similarities are adopted for our asymmetric model. Deep AsNMF inherits the direction preservation and hierarchical structure extraction capabilities from the Asymmetric and Deep NMF models, to be fitted to the directed graph

clustering problem. The most significant contributions of this paper are:

- The basic model is a novel deep NMF structure for learning directed and undirected graphs throughout extending the shallow AsNMF structure. The architecture of the Deep AsNMF model is designed in a hierarchical way like the deep NMF. As a result, this direction-aware model can hierarchically extract low-level to high-level graph structures.
- To cover the matrix sparsity and preserve the local structure in the proposed deep model, the first-order and second-order similarities are mixed as an input matrix. Asymmetric Cosine measure is adopted as the second-order similarity to maintain the graph directness in the input matrix.
- Since preserving the local and global structures plays an important role in the complex network analysis, this method extracts global topological information from the input graph by the PageRank centrality algorithm and imposes it in the form of a tailored graph regularization to the DAsNMF model.
- This paper proposes a Deep Asymmetric Nonnegative Matrix Factorization model with local and global structure preservation. In a unified optimization framework, this model can learn hierarchical representations while maintaining the graph structures to discover more accurate clusters in directed graphs.

The following is how this paper is structured: Section 2 provides an overview of the fundamental concepts and theoretical foundation. Section 3 presents the Deep Asymmetric NMF model and optimization algorithm. Some experiments are carried out in Section 4 to demonstrate the efficiency of our approach, and Section 5 presents the results as well as recommendations for further work.

2. Background

This section introduces some preliminaries necessary to understand the basic Nonnegative Matrix Factorization and its graph-specific extends, including Symmetric Nonnegative Matrix Factorization (SymNMF) and Asymmetric Nonnegative Matrix Factorization (AsNMF).

2.1. Nonnegative Matrix Factorization

The basic Nonnegative matrix factorization [4] method is a generic low-rank matrix decomposition method that focuses on the analysis of nonnegative data matrices. $\boldsymbol{X} \in \mathbb{R}_{+}^{m \times n}$ is a data matrix composed of n samples as columns, each with m features, which can be factorized into two matrices as $\boldsymbol{X} \approx \boldsymbol{W}\boldsymbol{H}$, where the basis matrix $\boldsymbol{W} \in \mathbb{R}_{+}^{m \times r}$ is the feature representation, and coefficient matrix $\boldsymbol{H} \in \mathbb{R}_{+}^{r \times n}$ is is the sample representation, (for a low-rank r) $r \leq min(m, n)$. In the basic NMF model,

each sample $\boldsymbol{X}_i$ can be represented as an additive linear combination of nonnegative basis vectors, which is $\boldsymbol{x}_i \approx \boldsymbol{W}\boldsymbol{h}_i$. Its objective function can be written as:

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

where $\|\boldsymbol{M}\|_F$ indicates the Frobenius Norm of matrix $\boldsymbol{M}$.

2.2. Symmetric Nonnegative Matrix Factorization

SymNMF has been widely adopted in various data analysis tasks [17]. This model factorizes a similarity matrix $\boldsymbol{A}$ and is based on the assumption that similar samples ($A_{i,j} > 0$) should have similar representations ($\boldsymbol{h}_i\boldsymbol{h}_j^\top > 0$) and dissimilar samples ($A_{i,j} = 0$) should have opposite representations ($\boldsymbol{h}_i\boldsymbol{h}_j^\top = 0$). Given a symmetric matrix $\boldsymbol{A} \in \mathbb{R}_+^{n\times n}$, a SymNMF model seeks for its low-rank approximation $\hat{\boldsymbol{A}}$ on the latent factor (LF) matrix $\boldsymbol{H} \in \mathbb{R}_+^{n\times r}$ with r denoting the dimension of the latent space, i.e., $\hat{\boldsymbol{A}} = \boldsymbol{H}\boldsymbol{H}^\top$. To obtain $\boldsymbol{H}$, an objective function describing the difference between $\boldsymbol{A}$ and $\hat{\boldsymbol{A}}$ is necessary, as the following function:

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

2.3. Asymmetric Nonnegative Matrix Factorization

Given the adjacency, it is possible to demonstrate directed graphs by using the formula $\boldsymbol{A} \in \mathbb{R}_+^{n\times n}$, where n is the total number of nodes and A_{ij} takes the value 1 if there is a directed edge connecting nodes i and j and 0 otherwise. To compute the AsNMF, which is used for clustering in undirected and directed graphs, the adjacency matrix $\boldsymbol{A}$ is required [18].

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

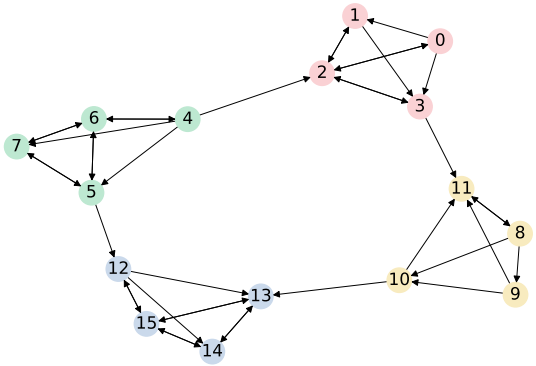
By utilizing this formulation, not only are we able to infer information regarding grouping from $\boldsymbol{H}$, but also information regarding cluster-level interactions from $\boldsymbol{W}$. To be more specific, when dealing with an undirected graph that is represented by a symmetric $\boldsymbol{A}$, the matrix $\boldsymbol{W}$ is obtained in the form of a symmetric matrix. The elements of the symmetric matrix show the connectivity between the clusters. On the other hand, in the scenario of a directed graph with an asymmetric $\boldsymbol{A}$, the elements in $\boldsymbol{W}$ demonstrate the directness of the connections between the clusters. For instance, if the pth cluster (and all of its nodes) is directed to the qth cluster (and its nodes), W_{pq} will have a value that is greater

than zero. The AsNMF method is proposed as an optimization strategy for the problem presented in (3) involving a directed graph. The multiplicative updating rules that are proposed are as follows:

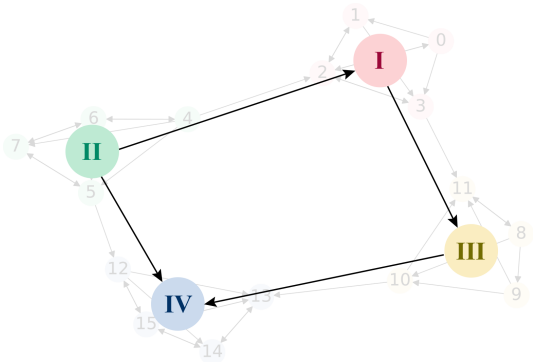
$$\boldsymbol{H} \leftarrow \boldsymbol{H} \odot \left[\frac{\boldsymbol{A}^\top \boldsymbol{H}\boldsymbol{W} + \boldsymbol{A}\boldsymbol{H}\boldsymbol{W}^\top}{\boldsymbol{H}\boldsymbol{W}^\top \boldsymbol{H}^\top \boldsymbol{H}\boldsymbol{W} + \boldsymbol{H}\boldsymbol{W}\boldsymbol{H}^\top \boldsymbol{H}\boldsymbol{W}^\top} \right]^{\frac{1}{4}} \tag{4}$$

$$\boldsymbol{W} \leftarrow \boldsymbol{W} \odot \frac{\boldsymbol{H}^\top \boldsymbol{A}\boldsymbol{H}}{\boldsymbol{H}^\top \boldsymbol{H}\boldsymbol{W}\boldsymbol{H}^\top \boldsymbol{H}} \tag{5}$$

where $\odot$ indicates the Hadamard product.



(a) Input Graph $\boldsymbol{A}$



(b) Clustering result and Inter-cluster Interactions

	I	II	III	IV
0	3e-01	3e-06	2e-06	5e-25
1	3e-01	1e-06	2e-05	2e-25
2	4e-01	3e-02	6e-04	9e-13
3	3e-01	3e-07	8e-02	5e-29
4	6e-02	3e-01	7e-18	2e-11
5	8e-14	4e-01	8e-07	4e-02
6	2e-12	4e-01	3e-08	2e-07
7	7e-17	4e-01	8e-08	2e-06
8	2e-17	0e+00	4e-01	1e-08
9	2e-19	0e+00	3e-01	9e-07
10	3e-18	1e-19	3e-01	6e-02
11	3e-02	2e-50	4e-01	4e-07
12	1e-14	6e-02	2e-08	3e-01
13	3e-30	9e-12	5e-02	4e-01
14	5e-12	1e-11	2e-08	4e-01
15	1e-12	4e-07	2e-08	4e-01

$\boldsymbol{H}$

	I	II	III	IV
I	5.51	6e-24	2e-01	5e-74
II	9e-02	4.17	6e-38	1e-01
III	1e-31	2e-51	2.92	1e-01
IV	2e-70	6e-44	1e-31	4.10

$\boldsymbol{W}$

(c) Optimal Matrices

Figure 1: An example of Asymmetric Nonnegative Matrix Factorization and its interpretability.

- *Interpretability of AsNMF*

This section provides a detailed example of how to use Asymmetric NMF to cluster and interpret directed graphs. Figure 1 (a) illustrates a directed graph consisting of 16 nodes and four clusters, where each cluster contains four nodes with strong intra-cluster connections. Additionally, there are inter-cluster connections such as a connection from node 4 to node 2. To cluster the graph, an asymmetric

matrix is generated and decomposed to the $\boldsymbol{H}$ membership matrix and the $\boldsymbol{W}$ inter-cluster interaction matrix, as shown in Figure 1 (c). The $\boldsymbol{H}$ matrix contains 16 rows corresponding to the nodes and four columns corresponding to the clusters, where the H_{ij} index represents the degree of node i's membership in cluster j. For instance, node 11 has strong ties to clusters III and I, a weak tie to cluster IV, and minimal ties to cluster II. Meanwhile, the $\boldsymbol{W}$ matrix shows the degree of intra-cluster and inter-cluster connections, where W_{ij} represents the connections between cluster i and cluster j. Notably, no connections exist from cluster IV to any cluster, but clusters II and III have connections to cluster IV, and the connection between cluster IV and cluster I is relatively weak. By interpreting these matrices, the final clustering and inter-cluster interactions can be obtained, as shown in Figure 1 (c).

2.4. Deep Nonnegative Matrix Factorization

The basic NMF (1) is a shallow learning process, which simultaneously learns the feature representation $\boldsymbol{W}$ and the sample representation $\boldsymbol{H}$ from the input matrix $\boldsymbol{X}$. To develop the knowledge of feature hierarchy in the datasets using deep NMF, the matrix $\boldsymbol{H}_1$ obtained over the single layer could be decomposed into $\boldsymbol{W}_2$ and $\boldsymbol{H}_2$. By extending the shallow NMF to a two-layer NMF structure, we transform the single-layer establishment into a deep structure. The Basic Deep Nonnegative Matrix Factorization model factorizes the data matrix $\boldsymbol{X}$ into $p+1$ factors, expressed by the following decomposition:

$$\boldsymbol{X} \approx \boldsymbol{W}_1\boldsymbol{W}_2 \ldots \boldsymbol{W}_p\boldsymbol{H}_p, \tag{6}$$

where $\boldsymbol{W} \in \mathbb{R}_+^{r_{i-1}\times r_i}$, $i \in 1, 2, \ldots, p$, and $r_0 = m$. The implicit representations of each layer can be provided by the following factorization:

$$\begin{aligned} \boldsymbol{H}_{p-1} &\approx \boldsymbol{W}_p\boldsymbol{H}_p \\ &\vdots \\ \boldsymbol{H}_2 &\approx \boldsymbol{W}_3 \ldots \boldsymbol{W}_p\boldsymbol{H}_p \\ \boldsymbol{H}_1 &\approx \boldsymbol{W}_2 \ldots \boldsymbol{W}_p\boldsymbol{H}_p \end{aligned} \tag{7}$$

After pre-training matrices by hierarchical factorizations, deep NMF methods fine-tune model to reduce the reconstruction error. Therefore, the Basic Deep NMF model is introduced in the following objective function:

$$\min_{\boldsymbol{W}_i, \boldsymbol{H}_p} \mathcal{L} = \|\boldsymbol{X} - \boldsymbol{W}_1 \ldots \boldsymbol{W}_p \boldsymbol{H}_p\|_F^2, \text{ s.t. } \quad \boldsymbol{H}_p \geq 0, \boldsymbol{W}_i \geq 0, \forall i = 1, 2, \ldots, p. \tag{8}$$

3. Proposed Method

In this section, we propose a deep method for directed graph clustering based on Asymmetric Nonnegative Matrix Factorization (AsNMF). Its success is mainly due to four factors: (1) It takes into account both first-order and second-order asymmetric similarities, to compensate for input sparsity and preserve local structure (Section 3.1); (2) It uses the deep structure of the input matrix to obtain a compact and abstract latent graph representation, which also provides a natural solution to hierarchical graph clustering (Section 3.2); (3) It imposes global topological information by a directed graph regularization to the DAsNMF model (Section 3.3); (4) It integrates the above into a unified loss function and an effective optimization procedure (Section 3.4).

3.1. Input Matrix

Given an unsigned graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, which $\mathcal{V} = \{v_1, \ldots, v_n\}$ describes n nodes, $\mathcal{E} = e_{ij}, i, j = \{1, \ldots, n\}$ and e_{ij} defines the link between node v_i and node v_j. Generally, graph $\mathcal{G}$ is represented by an adjacency matrix $\boldsymbol{A}$. As the most explicit representation of graphs, the element $[A]_{ij}$ of matrix $\boldsymbol{A}$ relates to the probability of connection between node v_i and node v_j. For unweighted graph, if there is an edge between node v_i and node v_j, A_{ij} is allocated with 1 and otherwise 0.

In this paper, to preserve structural information and to compensate for the input sparsity, a new similarity matrix is generated, which serves as the input of the model. The generated matrix takes into account both first-order and second-order similarity. The graph's first-order similarity is commonly represented by the adjacency matrix denoted by $\boldsymbol{A} \in \mathbb{R}^{n \times n}$. However, in real-world information networks, only a limited number of links can be observed, leaving many hidden relationships unrecognized. Consequently, although missed links may have zero first-order similarity, this does not necessarily mean that the nodes have zero similarity. To account for this, it becomes necessary to learn the higher-order similarity between nodes. One possible solution to this is to consider common neighbors, which serves as a complementary solution. Specifically, the second-order similarity, defined as $\boldsymbol{S} \in \mathbb{R}^{n \times n}$, is utilized to preserve the local graph structure and address the sparsity problem. This similarity assumes that two nodes are more likely to be similar if they share many common neighbors [34]. The cosine similarity is a normalized common neighbor measure as follows:

$$COS_{u,v} = \frac{\boldsymbol{a}_u . \boldsymbol{a}_v}{\|\boldsymbol{a}_u\| . \|\boldsymbol{a}_v\|} \tag{9}$$

where $\boldsymbol{a}_u$ and $\boldsymbol{a}_v$ are uth and vth row of matrix $\boldsymbol{A}$. Since in eq. (9) $COS(u,v) = COS(v,u)$, this measure generates a symmetric second-order similarity matrix that ignores the direction. To avoid this contradiction, we refine equation (9) by the proportion of common neighbors, normalized by the degree of node u to introduce an asymmetric similarity measure as follows:

$$\frac{\boldsymbol{a}_u.\boldsymbol{a}_v}{\|\boldsymbol{a}_u\|.\|\boldsymbol{a}_v\|} \cdot \frac{|\Gamma_u \cap \Gamma_v|}{|\Gamma_u|} \tag{10}$$

where Γ_u and Γ_v represent the set of neighbors of u and v nodes, respectively. Eq. (10) only considers the ratio of common neighbors that nodes have among all their neighbors and ignores the proportion of common neighbors in the total number of their neighbors. Hence, an additional parameter is needed to combine with Eq. (10). This second coefficient is referred to as the Sorensen index. Finally, the Asymmetric Cosine similarity ($ACOS$) is defined as follows:

$$S_{u,v} = ACOS_{u,v} = \frac{\boldsymbol{a}_u.\boldsymbol{a}_v}{\|\boldsymbol{a}_u\|.\|\boldsymbol{a}_v\|} \cdot \frac{|\Gamma_u \cap \Gamma_v|}{|\Gamma_u|} \cdot \frac{2|\Gamma_u \cap \Gamma_v|}{|\Gamma_u| + |\Gamma_v|} \tag{11}$$

Now, the problem is how to construct a proper input matrix by combining the first-order similarity $\boldsymbol{A}$ and second-order similarity $\boldsymbol{S}$ appropriately. As a simple combination, we use

$$\boldsymbol{A_S} = \boldsymbol{A} + \eta \boldsymbol{S} \tag{12}$$

where η is a hyperparameter that determines the contribution of the second-order similarity in the input matrix.

3.2. Deep Asymmetric NMF Model

As shown in (3), shallow Asymmetric NMF (AsNMF) extracts a one-layer cluster membership matrix $\boldsymbol{H}$, and an inter-cluster interaction matrix $\boldsymbol{W}$, directly. However, real-world graphs tend to exhibit complex and diverse hierarchical patterns. As a result, it is likely that the mapping between the input space and the cluster space contains intricate structural and hierarchical information, which may include implicit low-level to high-level hidden features [24]. Recent interest in deep learning has led to novel node representation methods that learn low-dimensional vectors instead of learning a compact graphical representation of the whole graph, which is important in graph analysis [25]. A promising new direction is graph summarization using deep graph representations learned automatically from the context encoded in the graph. Given the existence of graph summarization methods using latent

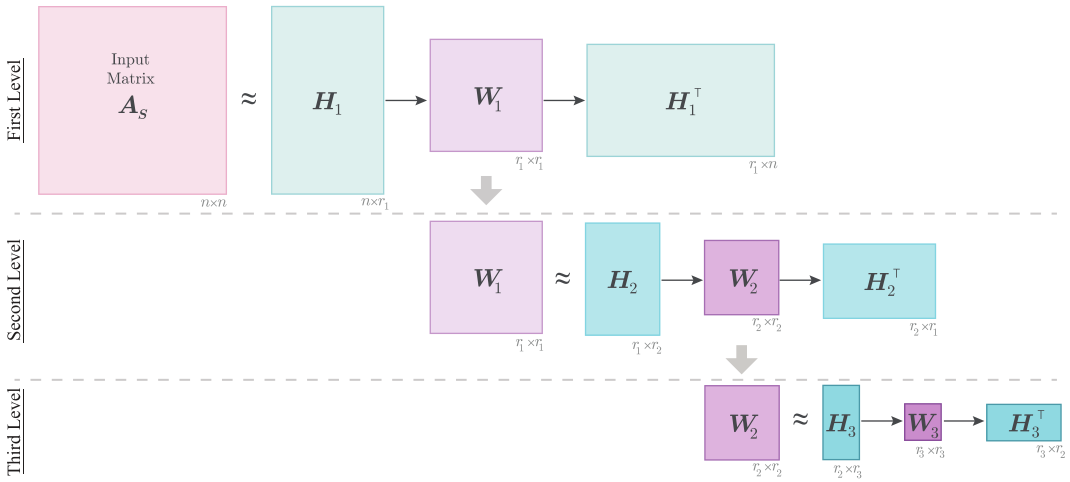


Figure 2: The architecture of Hierarchical AsNMF. For illustration purpose, the depth is fixed at 3.

graph representations [35], as well as the recent successes of deep learning, deep graph representations for clustering and summarization naturally seem promising. In this section, the purpose is to model the graph clustering problems in a multi-layer graph summarization approach. In a deep factorization framework, an asymmetric matrix factorization on the input graph has been applied to approximate the graph reconstruction and multi-layer node representation, and provide compact graph summarization for further layers. In other words, similar to widely used agglomerative hierarchical clustering approaches [1], DAsNMF starts with clustering nodes to subclusters, and then hierarchically merges subclusters by assuming each subcluster of the previous layer is a hypernode.

From a hierarchical factorization point of view, the following factorizations can be considered:

$$
\begin{aligned}
\boldsymbol{A_S} &\approx \boldsymbol{H}_1 \boldsymbol{W}_1 \boldsymbol{H}_1^\top \\
\boldsymbol{A_S} &\approx \boldsymbol{H}_1 \boldsymbol{H}_2 \boldsymbol{W}_2 \boldsymbol{H}_2^\top \boldsymbol{H}_1^\top \\
&\vdots \\
\boldsymbol{A_S} &\approx \boldsymbol{H}_1 \ldots \boldsymbol{H}_p \boldsymbol{W}_p \boldsymbol{H}_p^\top \ldots \boldsymbol{H}_1^\top
\end{aligned}
\tag{13}
$$

According to (13), in the first layer, for clustering the input graph to r_1 subclusters, the matrix $\boldsymbol{A_s}$ is factorized to the first-level node representation $H_1 \in \mathbb{R}_+^{n \times r_1}$ and mapping matrix $\boldsymbol{W}_1 \in \mathbb{R}_+^{r_1 \times r_1}$ which indicate the first-level cluster memberships and the inter-cluster interaction, respectively. In other words, similar nodes are assigned to a subcluster, and each subcluster is represented as a hypernode in the matrix $\boldsymbol{W}_1$. Therefore, the extracted matrix $\boldsymbol{W}_1$ can be considered as the first-level summarized graph. In the second layer, the summarized graph $\boldsymbol{W}_1$ is factorized to the second-level hypernode

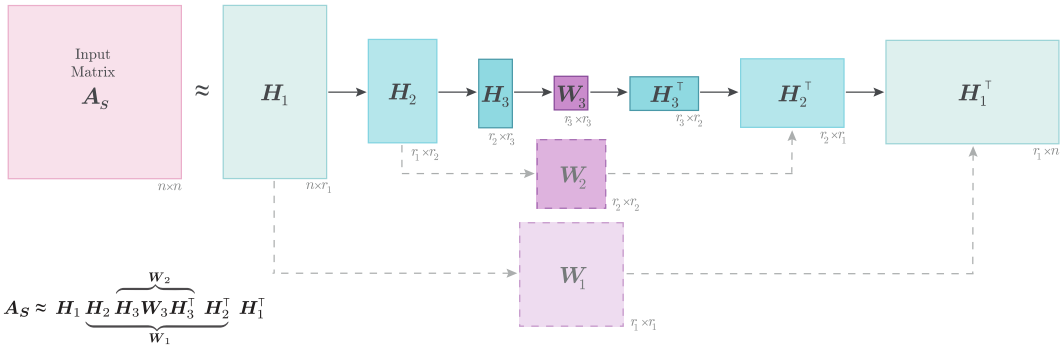


Figure 3: The architecture of Deep AsNMF. For illustration purpose, the depth is fixed at 3.

representation $H_2 \in \mathbb{R}_+^{r_1 \times r_2}$ and mapping matrix $W_2 \in \mathbb{R}_+^{r_2 \times r_2}$. This process is repeated p times until we reach the desired cluster $r_p = k$. The architecture of the Mutli-layer AsNMF model is shown in Figure 2.

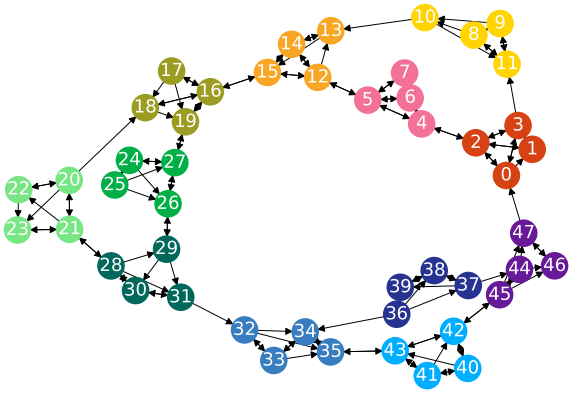
From deep factorization point of view, we propose DAsNMF, aims to introduce additional layers of abstraction of the similarity between nodes from low-level to high-level. Specifically, the input matrix $\boldsymbol{A_s}$ is factorized into $p+1$ nonnegative factors based on the Asymmetric NMF $\boldsymbol{A_s} = \boldsymbol{HWH}^\top$, as follows:

$$\boldsymbol{A_S} \approx \boldsymbol{H}_1 \ldots \boldsymbol{H}_p \boldsymbol{W}_p \boldsymbol{H}_p^\top \ldots \boldsymbol{H}_1^\top \tag{14}$$

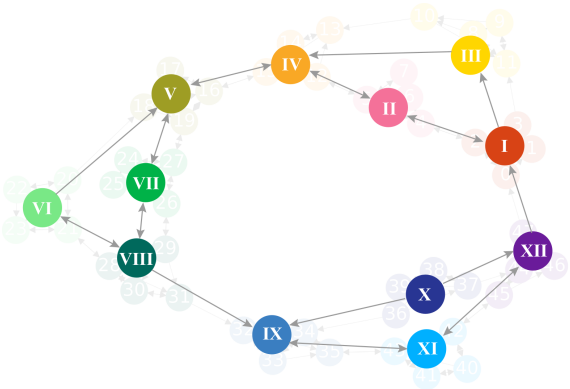
where $\boldsymbol{W}_p \in \mathbb{R}_+^{k \times k}, \boldsymbol{H}_i \in \mathbb{R}_+^{r_{i-1} \times r_i} (1 \le i \le p)$, and we set $n = r_0 \ge r_1 \ge \cdots \ge r_{p-1} \ge r_p = k$. The formulation in Eq. (14) allows for a hierarchy of p layers of abstract understanding of the graph, which can be given by the following factorizations:

$$\begin{aligned} \boldsymbol{W}_{p-1} &\approx \boldsymbol{H}_p \boldsymbol{W}_p \boldsymbol{H}_p^\top \\ &\vdots \\ \boldsymbol{W}_2 &\approx \boldsymbol{H}_3 \ldots \boldsymbol{H}_p \boldsymbol{W}_p \boldsymbol{H}_p^\top \ldots \boldsymbol{H}_3^\top \\ \boldsymbol{W}_1 &\approx \boldsymbol{H}_2 \ldots \boldsymbol{H}_p \boldsymbol{W}_p \boldsymbol{H}_p^\top \ldots \boldsymbol{H}_2^\top \end{aligned} \tag{15}$$

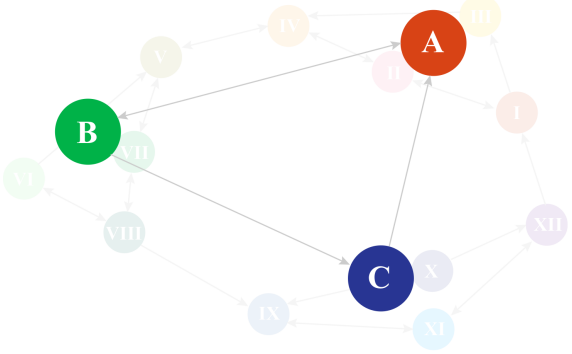
We impose the nonnegativity constraints on $\boldsymbol{W}_i$ and $\boldsymbol{H}_i (1 \le i < p)$. Through this process, every abstraction layer $\boldsymbol{W}_i$ represents the connections among components, ranging from the first-order proximity to the structural identity, and finally the cluster-level interactions. Therefore, the proposed deep architecture is expected to produce more accurate clustering results. The basic Deep Asymmetric



(a) Input Graph $\boldsymbol{A}$



(b) First level clustering result and Inter-cluster Interactions



(c) Second level clustering result and Inter-cluster Interactions

Figure 4: An example of Deep Asymmetric Nonnegative Matrix Factorization

NMF model can be formulated as follows:

$$\min_{\boldsymbol{H}_i, \boldsymbol{W}_p} \mathcal{L} = \|\boldsymbol{A}_{\boldsymbol{S}} - \boldsymbol{H}_1 \ldots \boldsymbol{H}_p \boldsymbol{W}_p \boldsymbol{H}_p^\top \ldots \boldsymbol{H}_1^\top\|_F^2, \text{ s.t. } \quad \boldsymbol{W}_p \geq 0, \boldsymbol{H}_i \geq 0, \forall i = 1, 2, \ldots, p. \tag{16}$$

After minimizing (16), we can achieve the multi-level memberships $\boldsymbol{H}_i (i < p)$ and subsequently the final cluster membership $\boldsymbol{\Psi}$ by hierarchical matrix multiplication $\boldsymbol{\Psi} = \prod_{i=1}^{p} \boldsymbol{H}_i$. The architecture of the Deep AsNMF model is shown in Figure 3.

- *Model interpretability*

This section provides a detailed example of clustering and interpreting Deep Asymmetric NMF (DAsNMF). Figure 4 (a) displays a directed graph with 48 nodes and three clusters. Each cluster consists of 16 nodes that have relatively strong intra-cluster connections. There are also inter-cluster connections, such as a one-way connection from node 4 to node 2. The input graph is clusterd into 12 sub-clusters in the first layer of matrix factorization, and the degree of belonging of each node to each cluster and the degree of interaction of the sub-clusters are obtained in $\boldsymbol{H}_1$ and $\boldsymbol{W}_1$ matrices,

respectively, following the shallow asymmetric NMF. The nodes belonging to a sub-cluster are merged and considered as supernodes I to XII. The degree of interactions between the sub-clusters is captured in the $\boldsymbol{W}_1$ matrix, and it represents the summarized input graph of the second layer of decomposition, as seen in Figure 4 (b). By re-decomposing the summarized graph in Figure 4 (b) into three clusters, the clustering of the second level of the graph occurs. Therefore, the final clustering of the graph and inter-cluster interactions are obtained by analyzing the matrices and interpreting the results, as demonstrated in Figure 4 (c).

3.3. Global Structure Regularization

In real-world networks, the observed links are often sparse, accounting for a small proportion of the overall network. Hence, relying solely on local information is inadequate [36]. To incorporate the side information into the model, most NMF models utilize a graph regularization that imposes graph-based constraints to the objective function of model to learn patterns that are consistent with the underlying graph structure [37]. Similarly, to adapt this prior information of a given graph, we add a global graph penalty term to the objective function (16) as

$$\begin{aligned} \min_{\boldsymbol{H}_i, \boldsymbol{W}_p} \mathcal{L} =& \|\boldsymbol{A}_{\boldsymbol{S}} - \boldsymbol{H}_1 \ldots \boldsymbol{H}_p \boldsymbol{W}_p \boldsymbol{H}_p^\top \ldots \boldsymbol{H}_1^\top\|_F^2 + \lambda \mathcal{R}(\boldsymbol{C}) \\ \text{s.t.} \quad & \boldsymbol{W}_p \geq 0, \boldsymbol{H}_i \geq 0, \forall i = 1, 2, \ldots, p. \end{aligned} \tag{17}$$

where $\mathcal{R}$ is a variable penalty dependent on the information being considered. In this paper, to obtain global topological information, we make use of the PageRank algorithm. Among various methods available to calculate the node score based on graph structure information, the PageRank algorithm is considered one of the most efficient [38]. Therefore, in the case of a graph without self-loops (where $A_{i,i} = 0$), we employ the iterative PageRank algorithm to compute the influence score of each node. The definition of the influence score is as follows:

$$c_i = \frac{1}{n}(1 - \rho) + \rho \sum_{j=1}^{n} \frac{A_{i,j}}{K_j^{out}} c_j \tag{18}$$

where c_i indicates the influence score of the ith node and $c \in [0, 1]$, ρ is the damping coefficient, K_j^{out} means the out-degree of the jth node. We construct the asymmetric influence score matrix of nodes, denoted by $\boldsymbol{C}$, using the following formula:

$$C_{i,j} = \begin{cases} c_i, & \text{if } A_{i,j} \neq 0, \\ 0, & \text{if } A_{i,j} = 0. \end{cases} \tag{19}$$

Therefore, the influence score matrix $\boldsymbol{C}$ contains all the global information of the graph. As our clustering method considers the similarity between two nodes in the original space as prior knowledge, $\mathcal{R}$ can be defined as follows:

$$\begin{aligned} \mathcal{R} &= \sum_{i=1}^{n}\sum_{j=1}^{n}\|\boldsymbol{\psi}_i - \boldsymbol{\psi}_j\|^2 C_{i,j} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}(\boldsymbol{\psi}_i^\top \boldsymbol{\psi}_i C_{i,j} - \boldsymbol{\psi}_i^\top \boldsymbol{\psi}_j C_{i,j} - \boldsymbol{\psi}_j^\top \boldsymbol{\psi}_i C_{i,j} + \boldsymbol{\psi}_j^\top \boldsymbol{\psi}_j C_{i,j}) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}(2\boldsymbol{\psi}_i^\top \boldsymbol{\psi}_i C_{i,j} - \boldsymbol{\psi}_i^\top \boldsymbol{\psi}_j C_{i,j} - \boldsymbol{\psi}_j^\top \boldsymbol{\psi}_i C_{i,j}) \\ &= 2\sum_{i=1}^{n}\boldsymbol{\psi}_i^\top \boldsymbol{\psi}_i D_{i,i} - \sum_{i=1}^{n}\sum_{j=1}^{n}\boldsymbol{\psi}_i^\top \boldsymbol{\psi}_j C_{i,j} - \sum_{i=1}^{n}\sum_{j=1}^{n}\boldsymbol{\psi}_j^\top \boldsymbol{\psi}_i C_{i,j} \\ &= 2\mathrm{Tr}(\boldsymbol{\Psi}\boldsymbol{D}\boldsymbol{\Psi}^\top) - \mathrm{Tr}(\boldsymbol{\Psi}\boldsymbol{C}\boldsymbol{\Psi}^\top) - \mathrm{Tr}(\boldsymbol{\Psi}\boldsymbol{C}^\top\boldsymbol{\Psi}^\top) \end{aligned} \tag{20}$$

where the diagonal matrix $\boldsymbol{D}$ is defined as $D_{ii} = \sum_{j=1}^{n} C_{i,j}$, and $\|\boldsymbol{\psi}_i - \boldsymbol{\psi}_j\|^2$ means the distance between representations of nodes i and j in the cluster space. More specifically, this regularization mandates that nodes possessing high closeness are grouped together within the same cluster. To consider both the similarities of two nodes in the original space and the proximity of their representations, we add a regularization term (denoted by (20)) to the right-hand side of (16). The objective of our regularized DAsNMF algorithm is to solve the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{H}_i, \boldsymbol{W}_p} & \|\boldsymbol{A_S} - \boldsymbol{H}_1 \ldots \boldsymbol{H}_p \boldsymbol{W}_p \boldsymbol{H}_p^\top \ldots \boldsymbol{H}_1^\top\|_F^2 + \lambda[2\mathrm{Tr}(\boldsymbol{\Psi}\boldsymbol{D}\boldsymbol{\Psi}^\top) - \mathrm{Tr}(\boldsymbol{\Psi}\boldsymbol{C}\boldsymbol{\Psi}^\top) - \mathrm{Tr}(\boldsymbol{\Psi}\boldsymbol{C}^\top\boldsymbol{\Psi}^\top)] \\ \text{s.t.} \quad & \boldsymbol{H}_i, \boldsymbol{W}_p \geq 0, \ \forall i \in \{1, \ldots, p\}. \end{aligned} \tag{21}$$

where the λ represents the regularization parameter.

3.4. Optimization

The objective function in Equation (21) is challenging for optimization due to its non-convex nature, which can have multiple local optima, making it hard to find the best solution. A specialized alternating

iterative algorithm has been developed to address this problem [26]. This algorithm cyclically updates factorized matrices in the objective function, breaking the complex problem into simpler subproblems that are iteratively solved. While the algorithm progressively approaches a favorable local optimum, it does not assure the discovery of the global optimum of the objective function. In order to accelerate the optimization of the factor matrices in the DAsNMF model, we utilize a pre-training approach to obtain initial approximations of $\boldsymbol{W}_i$ and $\boldsymbol{H}_i$ for each layer. This pre-training process significantly reduces the training time of the proposed model. The effectiveness of pre-training has been previously demonstrated in the context of deep networks [27]. To perform the pretraining, we first factorize the input matrix $\boldsymbol{A_S} \approx \boldsymbol{H}_1\boldsymbol{W}_1\boldsymbol{H}_1^\top$ by minimizing $\|\boldsymbol{A_S} - \boldsymbol{H}_1\boldsymbol{W}_1\boldsymbol{H}_1^\top\|_F^2$ where $\boldsymbol{H}_1 \in \mathbb{R}_+^{n\times r_1}$ and $\boldsymbol{W}_1 \in \mathbb{R}_+^{r_1\times r_1}$. Then, we decompose the matrix $\boldsymbol{W}_1$ as $\boldsymbol{W}_1 \approx \boldsymbol{H}_2\boldsymbol{W}_2\boldsymbol{H}_2^\top$ by minimizing $\|\boldsymbol{W}_1 - \boldsymbol{H}_2\boldsymbol{W}_2\boldsymbol{H}_2^\top\|_F^2$ where $\boldsymbol{H}_2 \in \mathbb{R}_+^{r_1\times r_2}$ and $\boldsymbol{W}_2 \in \mathbb{R}_+^{r_2\times r_2}$. The pre-training process is continued layer by layer until all layers have been pre-trained. Once pre-training is complete, each layer is fine-tuned using the introduced objective function in Equation (21) through alternating minimization. The updating rules are presented below.

3.4.1. Updating rule for the membership matrices

The objective function in equation (21) can be simplified by holding all variables constant except for $\boldsymbol{H}_i$, resulting in the following expression:

$$\begin{aligned}\min_{\boldsymbol{H}_i} \mathcal{L}(\boldsymbol{H}_i) =& \|\boldsymbol{A_S} - \boldsymbol{\Psi}_{i-1}\boldsymbol{H}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{W}_p\boldsymbol{\Phi}_{i+1}^\top\boldsymbol{H}_i^\top\boldsymbol{\Psi}_{i-1}^\top\|_F^2 \\ &+ 2\lambda\mathrm{Tr}(\boldsymbol{\Phi}_{i+1}^\top\boldsymbol{H}_i^\top\boldsymbol{\Psi}_{i-1}^\top\boldsymbol{D}\boldsymbol{\Psi}_{i-1}\boldsymbol{H}_i\boldsymbol{\Phi}_{i+1}) \\ &- \lambda\mathrm{Tr}(\boldsymbol{\Phi}_{i+1}^\top\boldsymbol{H}_i^\top\boldsymbol{\Psi}_{i-1}^\top\boldsymbol{C}\boldsymbol{\Psi}_{i-1}\boldsymbol{H}_i\boldsymbol{\Phi}_{i+1}) \\ &- \lambda\mathrm{Tr}(\boldsymbol{\Phi}_{i+1}^\top\boldsymbol{H}_i^\top\boldsymbol{\Psi}_{i-1}^\top\boldsymbol{C}^\top\boldsymbol{\Psi}_{i-1}\boldsymbol{H}_i\boldsymbol{\Phi}_{i+1}) \\ \text{s.t.} \quad & \boldsymbol{H}_i \geq 0,\end{aligned} \tag{22}$$

where $\boldsymbol{\Psi}_{i-1} = \boldsymbol{H}_1\ldots\boldsymbol{H}_{i-1}$ and $\boldsymbol{\Phi}_{i+1} = \boldsymbol{H}_{i+1}\ldots\boldsymbol{H}_p$. When $i = 1$, we set $\boldsymbol{\Psi}_0 = \boldsymbol{I}$. Similarly, when $i = p$, we set $\boldsymbol{\Phi}_{p+1} = \boldsymbol{I}$.

We can solve (22) by introducing a Lagrangian multiplier matrix $\boldsymbol{\Theta}_i$ to ensure the non-negativity constraints on $\boldsymbol{H}_i$. This results in an equivalent objective function as follows:

$$\begin{aligned}\min_{\boldsymbol{H}_i,\boldsymbol{\Theta}_i} \mathcal{L}(\boldsymbol{H}_i,\boldsymbol{\Theta}_i) =& \|\boldsymbol{A}_{\boldsymbol{S}} - \boldsymbol{\Psi}_{i-1}\boldsymbol{H}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{W}_p\boldsymbol{\Phi}_{i+1}^\top\boldsymbol{H}_i^\top\boldsymbol{\Psi}_{i-1}^\top\|_F^2 \\ &+ 2\lambda\mathrm{Tr}(\boldsymbol{\Phi}_{i+1}^\top\boldsymbol{H}_i^\top\boldsymbol{\Psi}_{i-1}^\top\boldsymbol{D}\boldsymbol{\Psi}_{i-1}\boldsymbol{H}_i\boldsymbol{\Phi}_{i+1}) \\ &- \lambda\mathrm{Tr}(\boldsymbol{\Phi}_{i+1}^\top\boldsymbol{H}_i^\top\boldsymbol{\Psi}_{i-1}^\top\boldsymbol{C}\boldsymbol{\Psi}_{i-1}\boldsymbol{H}_i\boldsymbol{\Phi}_{i+1}) \\ &- \lambda\mathrm{Tr}(\boldsymbol{\Phi}_{i+1}^\top\boldsymbol{H}_i^\top\boldsymbol{\Psi}_{i-1}^\top\boldsymbol{C}^\top\boldsymbol{\Psi}_{i-1}\boldsymbol{H}_i\boldsymbol{\Phi}_{i+1}) \\ &- \mathrm{Tr}(\boldsymbol{\Theta}_i\boldsymbol{H}_i^\top)\end{aligned} \tag{23}$$

To calculate the gradient of the objective function, we first need to express the function as a trace expression:

$$\begin{aligned}\min_{\boldsymbol{H}_i,\boldsymbol{\Theta}_i} \mathcal{L}(\boldsymbol{H}_i,\boldsymbol{\Theta}_i) =& \mathrm{Tr}(\boldsymbol{A}_{\boldsymbol{S}}^\top\boldsymbol{A}_{\boldsymbol{S}} - 2\boldsymbol{A}_{\boldsymbol{S}}^\top\boldsymbol{\Psi}_{i-1}\boldsymbol{H}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{W}_p\boldsymbol{\Phi}_{i+1}^\top\boldsymbol{H}_i^\top\boldsymbol{\Psi}_{i-1}^\top \\ &+ \boldsymbol{\Psi}_{i-1}\boldsymbol{H}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{W}_p^\top\boldsymbol{\Phi}_{i+1}^\top\boldsymbol{H}_i^\top\boldsymbol{\Psi}_{i-1}^\top\boldsymbol{\Psi}_{i-1}\boldsymbol{H}_i\boldsymbol{\Phi}_{i+1}\boldsymbol{W}_p\boldsymbol{\Phi}_{i+1}^\top\boldsymbol{H}_i^\top\boldsymbol{\Psi}_{i-1}^\top) \\ &+ 2\lambda\mathrm{Tr}(\boldsymbol{\Phi}_{i+1}^\top\boldsymbol{H}_i^\top\boldsymbol{\Psi}_{i-1}^\top\boldsymbol{D}\boldsymbol{\Psi}_{i-1}\boldsymbol{H}_i\boldsymbol{\Phi}_{i+1}) \\ &- \lambda\mathrm{Tr}(\boldsymbol{\Phi}_{i+1}^\top\boldsymbol{H}_i^\top\boldsymbol{\Psi}_{i-1}^\top\boldsymbol{C}\boldsymbol{\Psi}_{i-1}\boldsymbol{H}_i\boldsymbol{\Phi}_{i+1}) \\ &- \lambda\mathrm{Tr}(\boldsymbol{\Phi}_{i+1}^\top\boldsymbol{H}_i^\top\boldsymbol{\Psi}_{i-1}^\top\boldsymbol{C}^\top\boldsymbol{\Psi}_{i-1}\boldsymbol{H}_i\boldsymbol{\Phi}_{i+1}) \\ &- \mathrm{Tr}(\boldsymbol{\Theta}_i\boldsymbol{H}_i^\top),\end{aligned} \tag{24}$$

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

$$\begin{aligned}\boldsymbol{\Theta}_i =& -2\boldsymbol{\Psi}_{i-1}^\top\boldsymbol{A}^\top\boldsymbol{\Psi}\boldsymbol{W}_p\boldsymbol{\Phi}_{i+1}^\top - 2\boldsymbol{\Psi}_{i-1}^\top\boldsymbol{A}\boldsymbol{\Psi}\boldsymbol{W}_p^\top\boldsymbol{\Phi}_{i+1}^\top \\ &+ 2\boldsymbol{\Psi}_{i-1}^\top\boldsymbol{\Psi}\boldsymbol{W}_p^\top\boldsymbol{\Psi}^\top\boldsymbol{\Psi}\boldsymbol{W}_p\boldsymbol{\Phi}_{i+1}^\top + 2\boldsymbol{\Psi}_{i-1}^\top\boldsymbol{\Psi}\boldsymbol{W}_p\boldsymbol{\Psi}^\top\boldsymbol{\Psi}\boldsymbol{W}_p^\top\boldsymbol{\Phi}_{i+1}^\top \\ &- 4\lambda\boldsymbol{\Psi}_{i-1}^\top\boldsymbol{D}\boldsymbol{\Psi}\boldsymbol{\Phi}_{i+1}^\top + 2\lambda\boldsymbol{\Psi}_{i-1}^\top\boldsymbol{C}\boldsymbol{\Psi}\boldsymbol{\Phi}_{i+1}^\top + 2\lambda\boldsymbol{\Psi}_{i-1}^\top\boldsymbol{C}^\top\boldsymbol{\Psi}\boldsymbol{\Phi}_{i+1}^\top,\end{aligned} \tag{25}$$

From the complementary slackness condition of the KarushKuhn-Tucker (KKT) conditions, we obtain:

$$\boldsymbol{\Theta}_i \odot \boldsymbol{H}_i = \mathbf{0}, \tag{26}$$

Equation (26) is the fixed point equation that the solution must satisfy at convergence. By solving this equation, we derive the following updating rule for $\boldsymbol{H}_i$:

Algorithm 1 Deep Asymmetric Nonnegative Matrix Factorization (DAsNMF)

Input: The adjacency matrix of graph $\mathcal{G}$, $\boldsymbol{A}$; layer size of each layer, r_i; scale parameter η; regularization parameter λ; dumping factor $\rho = 0.85$;
Output: $\boldsymbol{W}_i$ $(1 \leq i < p)$, $\boldsymbol{H}_i$ $(1 \leq i < p)$, and the cluster matrix $\boldsymbol{\Psi}$;

Constructing the second-order similarity matrix $\boldsymbol{S}$ by (11);
Constructing the input graph $\boldsymbol{A}_{\boldsymbol{S}}$ by $\boldsymbol{A}_{\boldsymbol{S}} = \boldsymbol{A} + \eta \boldsymbol{S}$;
Constructing the influence score matrix $\boldsymbol{C}$ by (19);
▷ **Pre-training process:**
$\boldsymbol{W}_1, \boldsymbol{H}_1 \leftarrow$ShallowAsNMF$(\boldsymbol{A}_{\boldsymbol{S}}, r_1)$;
for $i = 2$ **to** p **do**
  $\boldsymbol{W}_i, \boldsymbol{H}_i \leftarrow$ ShallowAsNMF$(\boldsymbol{W}_{i-1}, r_i)$;
end for
▷ **Fine-tuning process:**
while convergence not reached **do**
  for $i = 1$ **to** p **do**
    $\boldsymbol{\Psi}_{i-1} \leftarrow \prod_{\tau=1}^{i-1} \boldsymbol{H}_\tau (\boldsymbol{\Psi}_0 \leftarrow \mathbf{I})$;
    $\boldsymbol{\Phi}_{i+1} \leftarrow \prod_{\tau=i+1}^{p} \boldsymbol{H}_\tau (\boldsymbol{\Phi}_{p+1} \leftarrow \mathbf{I})$;
    Update $\boldsymbol{H}_i$ by $\boldsymbol{H}_i \leftarrow \boldsymbol{H}_i \odot \left[\frac{\boldsymbol{\Psi}_{i-1}^\top (\boldsymbol{A}^\top \boldsymbol{\Psi} \boldsymbol{W}_p + \boldsymbol{A} \boldsymbol{\Psi} \boldsymbol{W}_p^\top + \lambda \boldsymbol{C} \boldsymbol{\Psi} + \lambda \boldsymbol{C}^\top \boldsymbol{\Psi}) \boldsymbol{\Phi}_{i+1}^\top}{\boldsymbol{\Psi}_{i-1}^\top (\boldsymbol{\Psi} \boldsymbol{W}_p^\top \boldsymbol{\Psi}^\top \boldsymbol{\Psi} \boldsymbol{W}_p + \boldsymbol{\Psi} \boldsymbol{W}_p \boldsymbol{\Psi}^\top \boldsymbol{\Psi} \boldsymbol{W}_p^\top + 2\lambda \boldsymbol{D} \boldsymbol{\Psi}) \boldsymbol{\Phi}_{i+1}^\top} \right]^{\frac{1}{4}}$;
    $\boldsymbol{\Psi}_i \leftarrow \boldsymbol{\Psi}_{i-1} \boldsymbol{H}_i$;
    Update $\boldsymbol{W}_i$ by $\boldsymbol{W}_i \leftarrow \boldsymbol{W}_i \odot \frac{\boldsymbol{\Psi}_i^\top \boldsymbol{A} \boldsymbol{\Psi}_i}{\boldsymbol{\Psi}_i^\top \boldsymbol{\Psi}_i \boldsymbol{W}_i \boldsymbol{\Psi}_i^\top \boldsymbol{\Psi}_i}$ $(i < p$, optional) or by $\boldsymbol{W}_p \leftarrow \boldsymbol{W}_p \odot \frac{\boldsymbol{\Psi}^\top \boldsymbol{A} \boldsymbol{\Psi}}{\boldsymbol{\Psi}^\top \boldsymbol{\Psi} \boldsymbol{W}_p \boldsymbol{\Psi}^\top \boldsymbol{\Psi}}$ $(i = p)$;
  end for
end while
return $\boldsymbol{W}_i, \boldsymbol{H}_i, \forall i = 1, 2, \ldots, p$;

$$\boldsymbol{H}_i \leftarrow \boldsymbol{H}_i \odot \left[\frac{\boldsymbol{\Psi}_{i-1}^\top (\boldsymbol{A}^\top \boldsymbol{\Psi} \boldsymbol{W}_p + \boldsymbol{A} \boldsymbol{\Psi} \boldsymbol{W}_p^\top + \lambda \boldsymbol{C} \boldsymbol{\Psi} + \lambda \boldsymbol{C}^\top \boldsymbol{\Psi}) \boldsymbol{\Phi}_{i+1}^\top}{\boldsymbol{\Psi}_{i-1}^\top (\boldsymbol{\Psi} \boldsymbol{W}_p^\top \boldsymbol{\Psi}^\top \boldsymbol{\Psi} \boldsymbol{W}_p + \boldsymbol{\Psi} \boldsymbol{W}_p \boldsymbol{\Psi}^\top \boldsymbol{\Psi} \boldsymbol{W}_p^\top + 2\lambda \boldsymbol{D} \boldsymbol{\Psi}) \boldsymbol{\Phi}_{i+1}^\top} \right]^{\frac{1}{4}} \tag{27}$$

3.4.2. Updating rule for the interaction matrix

By fixing all the variables except for $\boldsymbol{W}_p$, the objective function in Eq. (21) is reduced to:

$$\min_{\boldsymbol{W}_p} \mathcal{L}(\boldsymbol{W}_p) = \|\boldsymbol{A}_{\boldsymbol{S}} - \boldsymbol{\Psi} \boldsymbol{W}_p \boldsymbol{\Psi}^\top\|_F^2, \text{ s.t.} \quad \boldsymbol{W}_p \geq 0, \tag{28}$$

Subsequently, it is possible to rewrite the expression as follows:

$$\min_{\boldsymbol{W}_p} \mathcal{L}(\boldsymbol{W}_p) = \mathrm{Tr}(\boldsymbol{A}_{\boldsymbol{S}}^\top \boldsymbol{A}_{\boldsymbol{S}} - 2\boldsymbol{A}_{\boldsymbol{S}}^\top \boldsymbol{\Psi} \boldsymbol{W}_p \boldsymbol{\Psi}^\top + \boldsymbol{\Psi} \boldsymbol{W}_p^\top \boldsymbol{\Psi}^\top \boldsymbol{\Psi} \boldsymbol{W}_p \boldsymbol{\Psi}^\top) - \mathrm{Tr}(\boldsymbol{\Omega} \boldsymbol{W}_p^\top) \tag{29}$$

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

$$\boldsymbol{\Omega} = -2\boldsymbol{\Psi}^\top \boldsymbol{A} \boldsymbol{\Psi} + 2\boldsymbol{\Psi}^\top \boldsymbol{\Psi} \boldsymbol{W}_p \boldsymbol{\Psi}^\top \boldsymbol{\Psi}. \tag{30}$$

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

$$\boldsymbol{W}_p \leftarrow \boldsymbol{W}_p \odot \frac{\boldsymbol{\Psi}^\top \boldsymbol{A} \boldsymbol{\Psi}}{\boldsymbol{\Psi}^\top \boldsymbol{\Psi} \boldsymbol{W}_p \boldsymbol{\Psi}^\top \boldsymbol{\Psi}}. \tag{31}$$

3.4.3. Updating rule for the interaction matrices

Although updating $\boldsymbol{W}_i$ is not essential, as it does not significantly affect the value of the objective function (21), we still aim to extract the latent features in each intermediate layer. Thus, we intend to optimize the following objective function by updating $\boldsymbol{W}_i$:

$$\min_{\boldsymbol{W}_i} \mathcal{L}(\boldsymbol{W}_i) = \|\boldsymbol{A} - \boldsymbol{\Psi}_i \boldsymbol{W}_i \boldsymbol{\Psi}_i^\top\|_F^2, \text{ s.t.} \quad \boldsymbol{W}_i \geq 0, \tag{32}$$

Similar to $\boldsymbol{W}_p$, $\boldsymbol{W}_i$ can be updated by

$$\boldsymbol{W}_i \leftarrow \boldsymbol{W}_i \odot \frac{\boldsymbol{\Psi}_i^\top \boldsymbol{A} \boldsymbol{\Psi}_i}{\boldsymbol{\Psi}_i^\top \boldsymbol{\Psi}_i \boldsymbol{W}_i \boldsymbol{\Psi}_i^\top \boldsymbol{\Psi}_i} \tag{33}$$

We have now completed the derivation of all the updating rules necessary for the optimization process of DAsNMF. Algorithm 1 provides an overview of the overall optimization process, which includes a "ShallowAsNMF" procedure that performs the pretraining step described earlier. The source code for reproducing our results can be found at https://github.com/Hajiveiseh/DAsNMF.

4. Experimental Results

This section presents an empirical evaluation of the effectiveness of our DAsNMF model in comparison with several state-of-the-art methods. We conduct numerical experiments on eight real-world

directed networks to evaluate the performance of the proposed method.

4.1. Experimental setup

To evaluate the proposed approach, we utilize eight real-world directed networks that belong to three types, including three citation networks, two communication networks, and one social network. Table 1 provides the detailed characteristics of these networks. The descriptions of these networks are listed below:

- *WebKB* datasets (*Cornell*, *Texas*, *Washington*, and *Wisconsin*): These datasets contain web-pages collected from four universities by the World Wide Knowledge Base (WebKb) project of the Carnegie Mellon University text learning group.
- *Email*: This communication network is generated using email data from a large European research institution, where the emails represent the communication between the members of the institution.
- *Wiki*: This dataset is a page-page networks on specific fields. Nodes indicate articles and edges are mutual links.
- *Cora* and *CiteSeer*: Two citation networks consisting of scientific publications. The nodes are academic papers from Cora and Citeseer digital libraries.

We conducted experiments with various numbers of hidden layers and found that a deeper model does not necessarily improve performance, but increases computational time. Therefore, we set our model with three hidden layers. As a deep model, the configuration of the layers varies for different networks. The size configuration of each layer is shown in Table 1. Also, the hyperparameters η and λ are analyzed in the range of $\{0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5\}$ and $\{0, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}\}$, respectively. The number of iterations in the pre-training stage and the fine-tuning stage are set to 100 and 500, respectively.

4.2. Compared Methods

In this paper, we assume that DAsNMF can learn hierarchical representations of graphs by considering both local and global information. We hypothesize that this approach can result in a more accurate cluster membership matrix compared to shallow and deep NMF-based models. To verify our assumption, we compare our proposed method with 12 baseline and state-of-the-art models:

- NMF: This model reconstructs the input data using basic matrix factorization with nonnegative constraints on the factor matrices [4].

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

Dataset	#node	#edge	#class	layer configuration
Cornell	195	301	5	195-128-64-5
Texas	187	309	5	187-128-64-5
Washington	230	395	5	230-128-64-5
Wisconsin	197	502	5	197-128-64-5
Email	1005	25571	42	1005-256-128-42
Wiki	2405	16523	19	2405-256-128-19
Cora	2708	5429	7	2708-256-64-7
CiteSeer	3312	4732	6	3312-256-64-6

- PNMF: Projective NMF is a variant of the NMF model that learns part-based subspace representations that are sparse and spatially localized. It is based on positively constrained projections [39].
- ONMF: This model reconstructs the original data based on NMF and constrain the latent matrix to be orthogonal [15].
- SymNMF: This model is a general framework for graph clustering, which factors a similarity matrix to a clustering membership matrix and its transpose [17].
- AsNMF: This model adds extra factor to SymNMF model to handle directed and undirected graph clustering problems [18].
- BigClam: This method proposed a scalable NMF model for community detection in large-scale networks [40].
- M-NMF: Modularized NMF [41] is a model that utilizes the consensus relationship between node representations and community structure by modularizing the problem.
- NNSED: This method utilizes a nonnegative symmetric encoder-decoder approach for community detection, where the modules have only a single-layer mapping [22].
- RANMF: This is a regularized asymmetric nonnegative matrix factorization model for directed graph clustering [19].
- RAsNMF: A directed graph clustering method based on Semi-NMF that relaxes the nonnegativity constraints on the interaction matrix [21].
- DANMF: DANMF is a deep NMF-based model that integrates both deep decoder and encoder modules to learn the community structures [28].

- SDNMF: As a extension of DANMF, SDNMF utilizes second-order similarity matrix as input [29].

4.3. Evaluation Metrics

To evaluate the clustering effectiveness of algorithms when correct labels are available, we utilize three performance measurement methods: NMI (Normalized Mutual Information), Jaccard similarity, ARI (Adjusted Rand Index), Clustering Accuracy (ACC), and F-measure (F1 score), which are briefly introduced below.

The NMI (Normalized Mutual Information) is used as an external measure to evaluate the quality of clustering based on cluster labels. It is capable of comparing different clustering methods with different numbers of clusters. Given two sets of clusters, a and b, NMI is defined as follows:

$$NMI(a,b) = \frac{I(a,b)}{\sqrt{H(a)H(b)}}, \tag{34}$$

where $I(a;b)$ is mutual information between a and b, and $H(a)$ and $H(b)$ are entropies of a and b. Information theoretic based measures, such as normalized mutual information (NMI), are commonly used for evaluating clustering methods. The NMI is equal to 1 if the correct labels and corresponding predicted labels are similar and are close to 0 if they are mostly different.

The Jaccard similarity measures the similarity between two sets by calculating the ratio of the size of their intersection to the size of their union.

$$J(a,b) = \frac{|a \cap b|}{|a \cup b|} = \frac{|a \cap b|}{|a| + |b| - |a \cap b|}. \tag{35}$$

The Adjusted Rand Index (ARI) is a measure used to assess the similarity between two data clusters. It takes a value of 0 when the correspondence between two classes is lower than what would be expected by chance, and a value of 1 when the clusters are identical. If the relationship is weaker than what would be expected by chance, the ARI score may be negative. The equation for ARI is as follows:

$$ARI(c,y) = \frac{\sum_{i,j}\binom{n_{ij}}{2} - \sum_i \binom{n_{i\cdot}}{2}\sum_j \binom{n_{\cdot j}}{2}/\binom{n}{2}}{1/2[\sum_i \binom{n_{i\cdot}}{2} + \sum_j \binom{n_{\cdot j}}{2}] - \sum_i \binom{n_{i\cdot}}{2}\sum_j \binom{n_{\cdot j}}{2}/\binom{n}{2}}, \tag{36}$$

where N is the number of data points in a given data set and N_{ij} is the number of data points of the class label $C_j \in P$ assigned to cluster C_i in partition P. N_i is the number of data points in cluster C_i of partition P, and N_j is the number of data points in class C_j. In general, an ARI value lies between 0 and 1. The index value is equal to 1 only if a partition is completely identical to the intrinsic structure and close to 0 for a random partition.

The Clustering Accuracy (ACC) criterion assesses the proportion of data points for which the generated clusters can accurately correspond to the actual ground-truth classes. Its precise definition is as follows:

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

In this formula, n represents the total number of data samples, y_i denotes a ground truth label, $\bar{y}_i = map(c_i)$ signifies the optimal matching function responsible for permuting all clustering outcomes to achieve the best possible alignment between clustering labels and true labels, and $\delta(\cdot, \cdot)$ serves as the delta function, which evaluates to 1 if $y_i = \bar{y}_i$ and to 0 otherwise.

Clustering and classification algorithms' effectiveness is commonly assessed through the F-Measure, which leverages the precision and recall concepts from information retrieval. Let $C = \{C_1, C_2, \ldots, C_k\}$ denote a clustering of dataset D, and let $C^* = \{C_1, C_2, \ldots, C_l^*\}$ represent the correct class set of D. The recall of cluster j with respect to class i, denoted as $Rec(i, j)$, is defined as $|C_j^* \cap C_i|/|C_j^*|$. Precision, represented as $Prec(i, j)$, measures the precision of cluster j concerning class i and is expressed as $|C_j^* \cap C_i|/|C_i|$. The two values are harmoniously combined in the F-Measure using the following formula:

$$\mathrm{F1}_{i,j} = 2 \times \frac{Prec(i, j) \times Rec(i, j)}{Prec(i, j) + Rec(i, j)}. \tag{38}$$

4.4. Results

This section shows the clustering performance of proposed and compared methods on the eight real-world directed graphs. Tables 2-6 illustrate the results with the best performance for each metric highlighted in bold and second best performance in underline. It is clear from Tables 2–6 that:

- As can be observed from these results, it is clear that the proposed method obtains the highest clustering performance on almost all datasets, demonstrating that DAsNMF can find more appropriate clusters than other methods.

- In comparison to the second-best methods, DAsNMF achieves an average improvement of 0.012, 0.018, 0.036, 0.027, and 0.026 in terms of NMI, ARI, JAC, ACC, and F1 respectively. More specifically, compared to the best shallow and deep NMF models on the Cornell and Wisconsin datasets, our approach significantly increases the NMI value from 0.1679 to 0.2106, and from 0.0889 to 0.1089, respectively.

- Across 40 different cases, our proposed method achieves the highest performance on all evaluation metrics in 30 cases and ranked second-best in 6 out of the remaining 6 cases when compared to

Table 2: NMI results on real-world datasets

Method	Cornell	Texas	Washington	Wisconsin	Email	Wiki	Citeseer	Cora
NMF	0.1339	0.1470	0.1132	0.0788	0.5741	0.2475	0.1157	0.2725
PNMF	0.1195	0.2299	0.1114	0.0611	0.6633	0.2931	0.1575	0.2893
ONMF	0.0936	0.2326	0.1587	0.0691	0.6544	0.2812	0.1424	0.1929
SymNMF	0.1628	0.1549	0.1125	0.0635	0.4932	0.2432	0.1259	0.2598
AsNMF (rnd)	0.1452	0.1457	0.1373	0.0680	0.4848	0.2748	0.1457	0.3458
AsNMF (SVD)	0.1552	0.1616	0.1385	0.0730	0.4890	0.2582	0.1457	0.3458
BigClam	0.0429	0.0684	0.0626	0.0730	0.5649	0.2536	0.0885	0.0922
M-NMF	0.0453	0.0317	0.0475	0.0864	0.5360	0.2175	0.0466	0.0927
NSED	0.0754	0.0746	0.0337	0.0680	0.6700	0.2570	0.1456	0.1833
RANMF (rnd)	0.1085	0.1303	0.1020	0.0691	0.5716	0.2786	0.0835	0.1545
RANMF (SVD)	0.1679	0.1712	0.1703	0.0757	0.5865	0.2845	0.1310	0.3564
RAsNMF	0.1350	**0.3050**	0.2190	0.0730	0.4838	0.2585	0.1480	0.3443
DANMF	0.0980	0.1344	0.0824	0.0889	0.6736	0.2892	0.1102	0.3262
SDNMF	0.1056	0.0932	0.1167	0.0668	0.6902	0.2842	0.1226	0.3210
DAsNMF	**0.2106**	0.2561	**0.2208**	**0.1089**	**0.6988**	**0.2997**	**0.1638**	**0.3677**

Table 3: ARI results on real-world datasets

Method	Cornell	Texas	Washington	Wisconsin	Email	Wiki	Citeseer	Cora
NMF	0.0519	0.1834	0.1761	0.0294	0.3164	0.1247	0.0610	0.1461
PNMF	0.1305	0.3247	0.1305	0.1009	0.0749	0.1346	0.0618	0.2026
ONMF	0.1587	0.2326	0.1587	0.1045	0.0691	0.1181	0.0530	0.1244
SymNMF	0.0278	0.1012	0.0876	0.0512	0.3801	0.1156	0.0544	0.0885
AsNMF (rnd)	0.0944	0.1125	0.0991	0.0401	0.3801	0.1156	0.0623	0.2290
AsNMF (SVD)	0.0944	0.1457	0.0991	0.0401	0.3801	0.1156	0.0674	0.2290
BigClam	0.0654	0.0310	0.0654	0.0425	0.3082	0.0694	0.0691	0.0306
M-NMF	0.0287	0.0049	0.0475	0.0621	0.2808	0.0963	0.0017	0.0033
NSED	0.0588	0.1436	0.0588	0.0391	0.4379	0.1235	0.0568	0.0816
RANMF (rnd)	0.1900	0.2470	0.2056	0.0935	0.5103	0.1355	0.0810	0.1471
RANMF (SVD)	0.0749	0.2875	0.2565	0.1196	0.4442	0.1250	0.0403	0.2500
RAsNMF	0.1130	**0.4090**	**0.3110**	0.0955	0.5199	0.1159	0.0718	0.2282
DANMF	0.0940	0.1993	0.1342	0.0538	0.4788	0.1232	0.0248	**0.2747**
SDNMF	0.0732	0.1464	0.1256	0.1073	0.5014	0.1246	0.0565	0.2109
DAsNMF	**0.2394**	0.3375	0.2276	**0.1275**	**0.562**	**0.1375**	**0.1211**	0.2442

all other methods. This demonstrates that our method outperforms all other methods in most cases.

- In general, experimental results indicate that the effectiveness of different approaches varies when applied to different datasets. For instance, SDNMF is effective in clustering Email and Wiki, but not so much in Cornell, Texas, or Washington. RAsNMF shows good results for Texas and Washington, but not for Email, Wiki, or Cornell. RANMF works much better for Cora compared to Texas and Email. On the other hand, our method consistently produces the best or near-best results across all eight real-world datasets, demonstrating its consistency.

4.5. Parameter Analysis

This section analyzes the impact of hyperparameters on the model performance. The model utilizes two parameters, η and λ, which correspond to the asymmetric cosine similarity contribution and the regularization term, respectively. To evaluate the effectiveness of these parameters, Figure 5 presents the NMI, ARI, and JAC of the proposed method across five real-world datasets with different η and λ values tested. The figure is presented as a heatmap and the two axes correspond to the parameters η and λ. In this figure, the metrics are represented by color, where lighter colors indicate better

Table 4: Jaccard results on real-world datasets

Method	Cornell	Texas	Washington	Wisconsin	Email	Wiki	Citeseer	Cora
NMF	0.1650	0.1650	0.1650	0.1650	0.2176	0.1107	0.1640	0.1595
PNMF	0.1969	0.3126	0.2991	0.1691	0.0253	0.1201	0.1053	0.2048
ONMF	0.2202	0.3518	0.3184	0.1725	0.0223	0.1066	0.1684	0.1556
SymNMF	0.0914	0.2046	0.1853	0.2131	0.2236	0.0901	0.1425	0.0857
AsNMF (rnd)	0.1792	0.2136	0.1960	0.2356	0.0970	0.1005	0.1736	0.2282
AsNMF (SVD)	0.1792	0.2136	0.1960	0.2356	0.0970	0.1005	0.1736	0.2282
BigClam	0.2568	0.0310	0.3132	0.1246	0.0458	0.0981	0.1785	0.1792
M-NMF	0.1343	0.1494	0.1528	0.2045	0.1800	0.0859	0.0952	0.0880
NSED	0.2031	0.3609	0.2991	0.2151	0.2675	0.1042	0.1865	0.1635
RANMF (rnd)	0.2778	0.3434	0.2938	0.2366	0.2961	0.1165	0.1643	0.1124
RANMF (SVD)	0.2026	0.3416	0.3630	0.2704	0.3108	0.1134	0.1702	0.2356
RAsNMF	0.2210	**0.4610**	**0.3751**	0.2380	0.0966	0.1007	0.1740	0.2277
DANMF	0.2415	0.3464	0.2473	0.2406	0.3334	0.1178	0.1605	0.1905
SDNMF	0.2852	0.4146	0.3034	0.2455	0.3533	0.1119	0.1567	0.2190
DAsNMF	**0.3357**	0.4609	0.3595	**0.2936**	**0.4131**	**0.1231**	**0.2910**	**0.2832**

Table 5: Clustering Accuracy results on real-world datasets

Method	Cornell	Texas	Washington	Wisconsin	Email	Wiki	Citeseer	Cora
NMF	0.2341	0.3223	0.3125	0.2933	0.3841	0.2411	0.1587	0.1657
PNMF	0.2580	0.2951	0.2996	0.3112	0.3772	0.3517	0.2158	0.2021
ONMF	0.1782	0.2813	0.2549	0.2357	0.3637	0.3442	0.2531	0.1965
SymNMF	0.2155	0.2399	0.2200	0.2547	0.3632	0.2452	0.1913	0.1583
AsNMF (rnd)	0.3385	0.5508	0.4523	0.4264	0.5791	0.3043	0.3041	0.4095
AsNMF (SVD)	0.3525	0.5412	0.4217	0.4212	0.5791	0.2925	0.2967	0.3969
BigClam	0.2546	0.3266	0.2885	0.3321	0.3921	0.3125	0.2754	0.1354
MNMF	0.1232	0.3531	0.3654	0.3511	0.4120	0.2531	0.2584	0.2369
NSED	0.3945	0.5111	0.5025	0.4805	0.5458	0.3021	0.3542	0.3965
DANMF	0.4244	0.5233	0.4366	0.4394	**0.5944**	0.3244	0.2885	0.3760
RANMF (rnd)	0.3846	0.5729	0.5435	0.4981	0.5413	0.3489	0.3252	0.4660
RANMF (SVD)	0.4126	0.5674	0.5366	0.5008	0.5763	0.3524	0.3542	0.3965
RAsNMF	0.3538	**0.6096**	0.4957	0.4415	0.3592	0.3652	0.3404	0.3812
SDNMF	0.4381	0.5457	0.4471	0.4876	0.5150	0.3264	0.3179	0.4486
DAsNMF	**0.4821**	0.5829	**0.5565**	**0.5132**	0.5861	**0.4586**	**0.3995**	**0.4712**

Table 6: F1-score results on real-world datasets

Method	Cornell	Texas	Washington	Wisconsin	Email	Wiki	Citeseer	Cora
NMF	0.2154	0.2111	0.2158	0.2143	0.3521	0.2411	0.1587	0.1457
PNMF	0.1752	0.3451	0.1753	0.1529	0.4501	0.1893	0.1258	0.1955
ONMF	0.1658	0.1857	0.1886	0.1652	0.4312	0.1582	0.1564	0.2333
SymNMF	0.1587	0.1726	0.1578	0.1255	0.4453	0.2333	0.2071	0.2135
AsNMF (rnd)	0.3164	0.5061	0.3815	0.3602	0.5156	0.2267	0.2590	0.3202
AsNMF (SVD)	0.3164	0.5119	0.3898	0.3687	0.5156	0.2305	0.2628	0.3256
BigClam	0.2565	0.3521	0.3486	0.3181	0.3345	0.2996	0.2358	0.2511
MNMF	0.2438	0.3357	0.2435	0.1685	0.1201	0.1985	0.1965	0.1835
NSED	0.3008	0.5588	0.4216	0.3876	0.4854	0.2259	0.2998	**0.3860**
RANMF (rnd)	0.3438	0.5748	0.5261	0.4237	0.4854	0.2259	0.2997	**0.3860**
RANMF (SVD)	0.4125	0.5389	0.4257	0.4157	**0.5231**	0.2451	0.2996	0.3123
RAsNMF	0.3222	0.5620	0.4695	0.3791	0.1769	0.2712	0.2999	0.2854
DANMF	0.3844	0.4860	0.4008	0.3918	0.5043	0.2095	0.2780	0.2990
SDNMF	0.4114	0.5266	0.4923	0.4114	0.6119	0.1988	0.2698	0.3476
DAsNMF	**0.4570**	**0.5788**	**0.5315**	**0.4478**	0.4946	**0.3615**	**0.3174**	0.3758

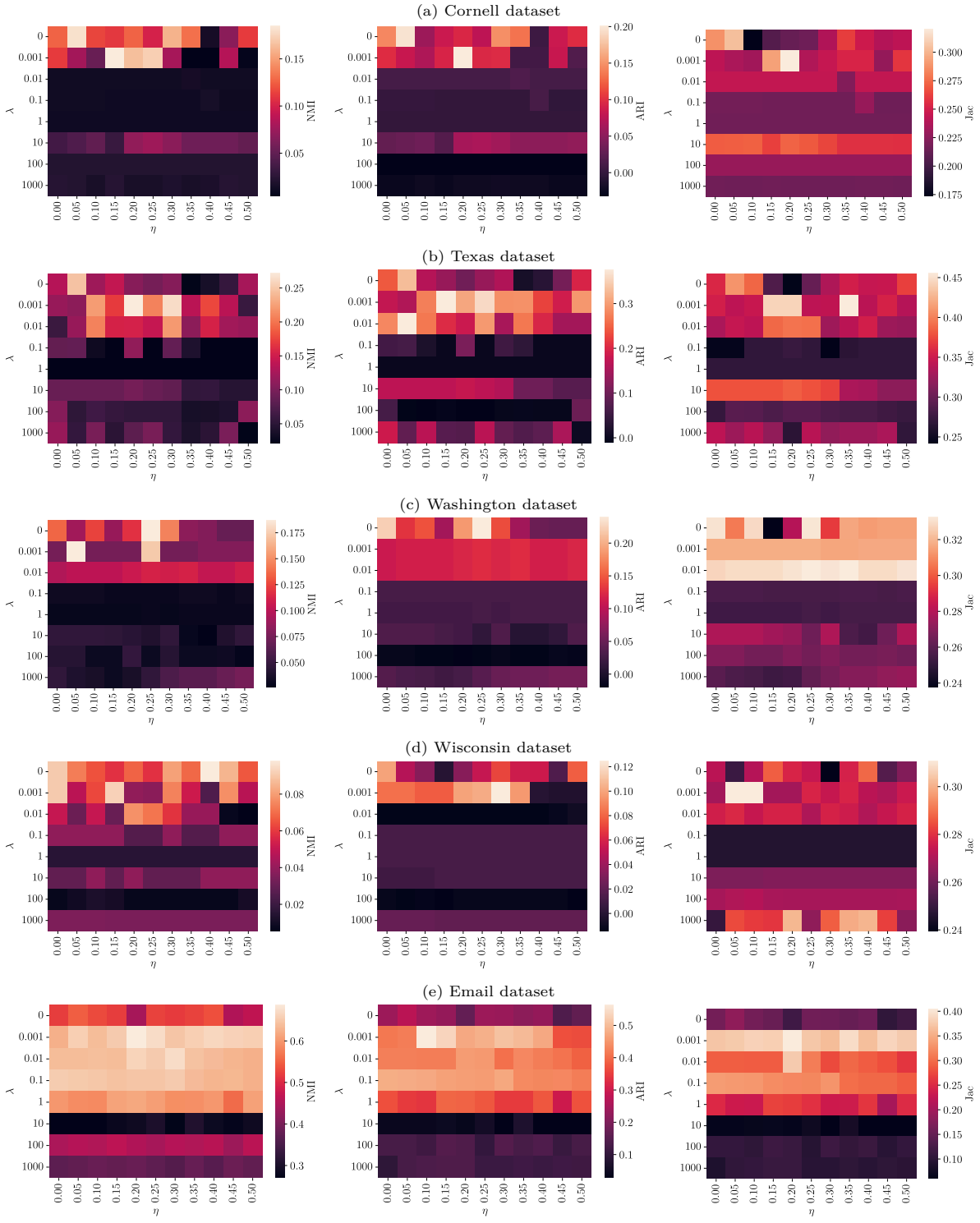


Figure 5: Parameter analysis (in terms of NMI, ARI, and Jaccard measures) on the parameters η and λ, where the lighter color describes the higher values.

Table 7: The optimal η and λ hyperparameter values for each dataset

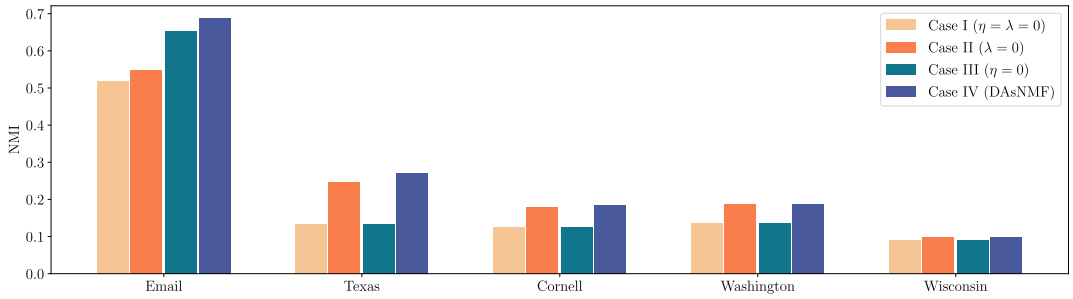
Parameter	Cornell	Texas	Washington	Wisconsin	Email	Wiki	Citeseer	Cora
η	0.2	0.2	0.25	0.30	0.20	0.2	0.15	0.45
λ	0.001	0.001	0.001	0.001	0.001	0.01	0.01	0.01

results. The analysis revealed that both η and λ with relatively large values led to poor performance. In the small-scale datasets such as, Cornell, Texas, Wisconsin, and Washington, smaller η and λ values produced better performance, suggesting similar behavior between the regularization term and asymmetric cosine similarity contribution. On the other hand, for the large-scale datasets such as, Email dataset, a higher value for the parameter λ proved to be effective, and the results were less sensitive to the parameter η. Consequently, the contribution of asymmetric cosine similarity and regularization term in the proposed model impacts the results across different datasets. The optimal η and λ hyperparameter values for each dataset are reported in Table 7. In scenarios where real-world datasets lack ground truth labels, appropriate parameter values can be chosen based on the evaluated datasets that share similar characteristics. For example, when clustering a citation network dataset with sparse links, one could reference other successful clustering efforts on similar evaluated datasets and utilize their parameter values for the task at hand.

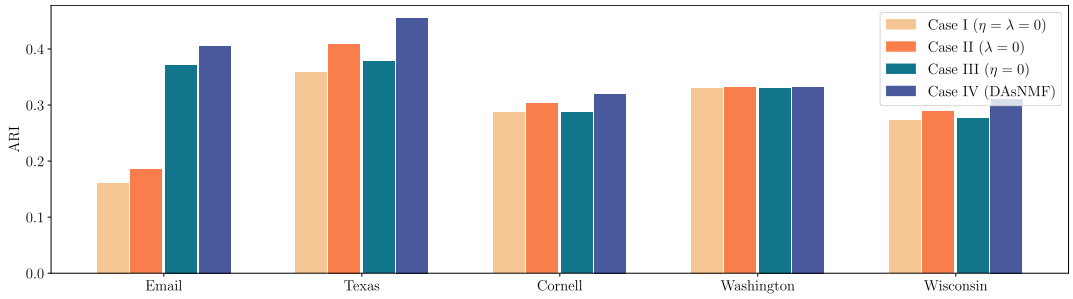
4.6. Ablation study

In this section, we have studied the effectiveness of each part of the model on the performance. As mentioned in the section 3, the DAsNMF model, in addition to its deep structure, has local structure preservation and global structure regularization parts which their impact are controlled by η and λ hyperparameters respectively. In Figure 6, the performance of different cases of the proposed model are shown in term of NMI, ARI and Jaccard metrics. In particular, case I ($\eta = \lambda = 0$) means the proposed method without local and global preservation parts, case II ($\lambda = 0$) indicates that only the local asymmetric cosine similarity is utilized, and case III ($\eta = 0$) indicates that only manifold regularization term based on pageRank is considered. Finally, case IV denotes the DAsNMF model with incorporating both local and global preservation parts. From Figure 6, the following conclusions can be drawn:

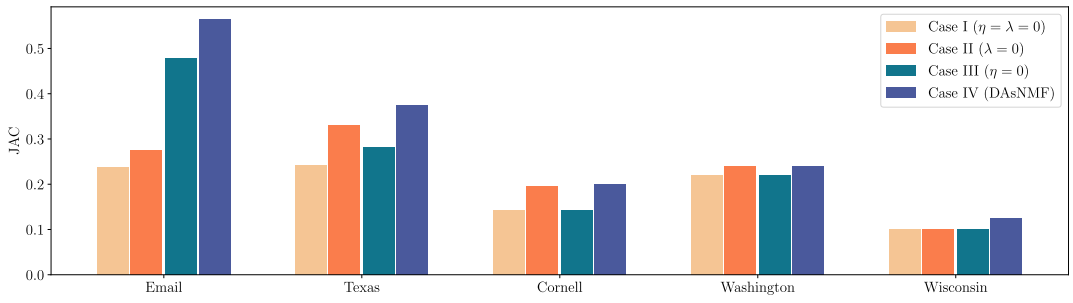
- In the all examined datasets, the results of case II are better than case I in terms of NMI, ARI, and Jaccard, which proves the efficiency of considering the local structure based on the asymmetric cosine similarity.
- The results shows that case III outperforms case I, which proves the efficiency of considering the global graph regularization.



(a) Normalized Mutual Information



(b) Adjusted Rand Index



(c) Jaccard

Figure 6: Ablation study on the DAsNMF model.

- In the sparse datasets such as Texas, Cornell, Washington, and Wisconsin, case II results better performances than case III which indicates that reducing sparsity by adding the asymmetric cosine similarity to the input matrix is more effective than adding the global regularization term to the objective function.

- In the some large-scale networks with complex structures such as Email dataset, case III performs better than case II in terms of NMI, Jaccard, and ARI, which means that considering global structure preservation can be more constructive in the large-scale networks.

- Finally, we can conclude based on case IV that both local and global structure preservation parts significantly contribute to the proposed model and complement each other.

4.7. Convergence Analysis

The iterative updating rules in our optimization algorithm are the basis of its operation. Therefore, we investigate the convergence behavior of the proposed method. Our optimization algorithm comprises of the pre-training stage and the fine-tuning stage. During the pre-training stage, each layer is equivalent to a shallow Asymmetric NMF model, whose convergence has been analyzed in previous work [18]. Therefore, we focus our attention on the fine-tuning stage and analyze its convergence rate, which measures the speed of the objective function value change. We present the results of this analysis on Cornell, Texas, and Washington networks in Figure 7. Comparable results were also obtained for other networks. From Figure 7, we observe that DAsNMF converges quickly, typically within 100 iterations.

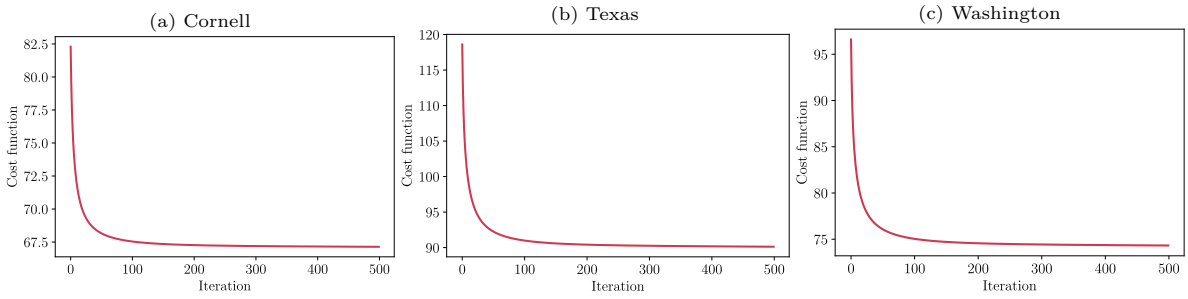


Figure 7: Convergence analysis of the DAsNMF on the real-world datasets.

5. Conclusion

This paper proposed a graph-specific Deep NMF model called Deep Asymmetric NMF (DAsNMF) which has multi-layer factorization structure. Different from prior deep NMF models which are intrinsically graph embedding and usually need further clustering algorithms to result in final clusters, DAsNMF is a graph clustering model which can take edge direction into account. The proposed structure makes DAsNMF be able to cluster an asymmetric graph by learning multi-layer node representation and graph summarization simultaneously. Meanwhile, a combination of first-order and second-order proximity matrices of the graph is selected as the original input matrix, and introduced a tailored asymmetric graph regularization term to retain the graph structure. Furthermore, a comparison is held with other existing Deep NMF and baseline algorithms. The proposed approach is perceived to perform better than the other algorithms under comparison in terms of NMI, ARI, and Jaccard.

Although the proposed method shows promise in extracting network structures, it is essential to acknowledge its inherent simplicity of the linear deep NMF model. To address this limitation, future work could focus on developing a neural matrix factorization model with interpretability properties

that incorporates non-linearities and additional layers, thus allowing for a more faithful representation of intricate network structures. Furthermore, by relaxing its nonnegativity constraint, DAsNMF model can be developed to a Deep Asymmetric Semi-NMF for signed graph analysis. Additionally, due to summarization capability of the DAsNMF, it can be utilized for multi-level graph summarization, especially in dynamic graphs. Finally, the proposed deep graph reconstruction model seems to be suitable for the directed and undirected link prediction problems.

References

[1] S. Fortunato, Community detection in graphs, Physics Reports 486 (3) (2010) 75–174.

[2] E. A. Leicht, M. E. J. Newman, Community structure in directed networks, Phys. Rev. Lett. 100 (2008) 118703.

[3] F. D. Malliaros, M. Vazirgiannis, Clustering and community detection in directed networks: A survey, Physics Reports 533 (4) (2013) 95–142.

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

[5] J. Wang, X.-L. Zhang, Deep nmf topic modeling, Neurocomputing 515 (2023) 157–173.

[6] S. A. Seyedi, P. Moradi, F. A. Tab, A weakly-supervised factorization method with dynamic graph embedding, in: Artificial Intelligence and Signal Processing Conference (AISP), 2017, pp. 213–218.

[7] J. Chavoshinejad, S. A. Seyedi, F. Akhlaghian Tab, N. Salahian, Self-supervised semi-supervised nonnegative matrix factorization for data clustering, Pattern Recognition 137 (2023) 109282.

[8] R. Mahmoodi, S. A. Seyedi, F. Akhlaghian Tab, A. Abdollahpouri, Link prediction by adversarial nonnegative matrix factorization, Knowledge-Based Systems 280 (2023) 110998.

[9] Z. Shajarian, S. A. Seyedi, P. Moradi, A clustering-based matrix factorization method to improve the accuracy of recommendation systems, in: International Conference on Electrical Engineering (ICEE), 2017, pp. 2241–2246.

[10] N. Salahian, F. A. Tab, S. A. Seyedi, J. Chavoshinejad, Deep autoencoder-like nmf with contrastive regularization and feature relationship preservation, Expert Systems with Applications 214 (2023) 119051.

[11] S. A. Seyedi, F. Akhlaghian Tab, A. Lotfi, N. Salahian, J. Chavoshinejad, Elastic adversarial deep nonnegative matrix factorization for matrix completion, Information Sciences 621 (2023) 562–579.

[12] Y. Zhao, F. Deng, J. Pei, X. Yang, Progressive deep non-negative matrix factorization architecture with graph convolution-based basis image reorganization, Pattern Recognition 132 (2022) 108984.

[13] K. Luong, R. Nayak, T. Balasubramaniam, M. A. Bashar, Multi-layer manifold learning for deep non-negative matrix factorization-based multi-view clustering, Pattern Recognition 131 (2022) 108815.

[14] N. Gillis, Nonnegative matrix factorization, SIAM, 2020.

[15] 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.

[16] 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.

[17] D. Kuang, C. Ding, H. Park, Symmetric nonnegative matrix factorization for graph clustering, in: Proceedings of the SIAM International Conference on Data Mining (SDM), 2012, pp. 106–117.

[18] F. Wang, T. Li, X. Wang, S. Zhu, C. Ding, Community discovery using nonnegative matrix factorization, Data Mining and Knowledge Discovery 22 (2011) 493–521.

[19] A. Tosyali, J. Kim, J. Choi, M. K. Jeong, Regularized asymmetric nonnegative matrix factorization for clustering in directed networks, Pattern Recognition Letters 125 (2019) 750–757.

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

[21] R. Abdollahi, S. Amjad Seyedi, M. Reza Noorimehr, Asymmetric semi-nonnegative matrix factorization for directed graph clustering, in: International Conference on Computer and Knowledge Engineering (ICCKE), 2020, pp. 323–328.

[22] B.-J. Sun, H. Shen, J. Gao, W. Ouyang, X. Cheng, A non-negative symmetric encoder-decoder approach for community detection, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017, pp. 597–606.

[23] Z. Zhang, P. Cui, W. Zhu, Deep learning on graphs: A survey, IEEE Transactions on Knowledge and Data Engineering 34 (1) (2022) 249–270.

[24] S. Dong, P. Wang, K. Abbas, A survey on deep learning and its applications, Computer Science Review 40 (2021) 100379.

[25] X. Su, S. Xue, F. Liu, J. Wu, J. Yang, C. Zhou, W. Hu, C. Paris, S. Nepal, D. Jin, Q. Z. Sheng, P. S. Yu, A comprehensive survey on community detection with deep learning, IEEE Transactions on Neural Networks and Learning Systems (2022) 1–21.

[26] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, B. Schuller, A deep semi-nmf model for learning hidden representations, in: Proceedings of the International Conference on Machine Learning (ICML), Vol. 32, 2014, pp. 1692–1700.

[27] P. De Handschutter, N. Gillis, X. Siebert, A survey on deep matrix factorizations, Computer Science Review 42 (2021) 100423.

[28] F. Ye, C. Chen, Z. Zheng, Deep autoencoder-like nonnegative matrix factorization for community detection, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018, pp. 1393–1402.

[29] M. Zhang, Z. Zhou, Structural deep nonnegative matrix factorization for community detection, Applied Soft Computing 97 (2020) 106846.

[30] E. Al-sharoa, B. Rahahleh, Community detection in networks through a deep robust auto-encoder nonnegative matrix factorization, Engineering Applications of Artificial Intelligence 118 (2023) 105657.

[31] C. He, H. Liu, Y. Tang, X. Fei, H. Li, Q. Zhang, Network embedding using deep robust nonnegative matrix factorization, IEEE Access 8 (2020) 85441–85453.

[32] J. Huang, T. Zhang, W. Yu, J. Zhu, E. Cai, Community detection based on modularized deep nonnegative matrix factorization, International Journal of Pattern Recognition and Artificial Intelligence 35 (02) (2021) 2159006.

[33] P. De Handschutter, N. Gillis, W. Blekic, Deep symmetric matrix factorization, in: European Association for Signal Processing (EURASIP), 2023.

[34] P. Pirasteh, D. Hwang, J. J. Jung, Exploiting matrix factorization to asymmetric user similarities in recommendation systems, Knowledge-Based Systems 83 (2015) 51–57.

[35] W. Xu, G. Niu, A. Hyvärinen, M. Sugiyama, Direction Matters: On Influence-Preserving Graph Summarization and Max-Cut Principle for Directed Graphs, Neural Computation 33 (8) (2021) 2128–2162.

[36] G. Chen, C. Xu, J. Wang, J. Feng, J. Feng, Nonnegative matrix factorization for link prediction in directed complex networks using pagerank and asymmetric link clustering information, Expert Systems with Applications 148 (2020) 113290.

[37] D. Cai, X. He, J. Han, T. S. Huang, Graph regularized nonnegative matrix factorization for data representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (8) (2010) 1548–1560.

[38] R. Baeza-Yates, P. Boldi, C. Castillo, Generalizing pagerank: Damping functions for link-based ranking algorithms, in: Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval, 2006, pp. 308–315.

[39] Z. Yuan, E. Oja, Projective nonnegative matrix factorization for image compression and feature extraction, in: Image Analysis: 14th Scandinavian Conference, SCIA 2005, Joensuu, Finland, June 19-22, 2005. Proceedings 14, 2005, pp. 333–342.

[40] J. Yang, J. Leskovec, Overlapping community detection at scale: A nonnegative matrix factorization approach, in: Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, 2013, pp. 587–596.

[41] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, S. Yang, Community preserving network embedding, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 31, 2017, pp. 203–209.