

Article

# Estimating Mixed Memberships in Directed Networks by Spectral Clustering

Huan Qing 

School of Mathematics, China University of Mining and Technology, Xuzhou 221116, China; qinghuan@cumt.edu.cn or qinghuan07131995@163.com

**Abstract:** Community detection is an important and powerful way to understand the latent structure of complex networks in social network analysis. This paper considers the problem of estimating community memberships of nodes in a directed network, where a node may belong to multiple communities. For such a directed network, existing models either assume that each node belongs solely to one community or ignore variation in node degree. Here, a directed degree corrected mixed membership (DiDCMM) model is proposed by considering degree heterogeneity. An efficient spectral clustering algorithm with a theoretical guarantee of consistent estimation is designed to fit DiDCMM. We apply our algorithm to a small scale of computer-generated directed networks and several real-world directed networks.

**Keywords:** community detection; directed networks; overlapping networks; spectral clustering

## 1. Introduction

Many real-world complex networks have community structure such that nodes within the same community (also known as cluster or module) have more links than across communities. For example, in social networks, communities can be groups of students in the same department; in co-authorship networks, a community can be formed by researchers in the same field. However, community structure for a real-world network is usually not directly observable. To process this problem, community detection, also known as graph clustering, is a popular tool for uncovering a latent community structure in a network [1,2]. For decades, many community detection methods have been proposed for non-overlapping undirected networks in which each node belongs to a single community, and the interactions between two nodes are symmetric or undirected. The stochastic block model (SBM) [3] is a popular generative model for non-overlapping undirected networks. In SBM, it is assumed that each node only belongs to one community and that nodes in the same community have the same expectation degrees. Ref. [4] proposes the classical degree corrected stochastic block model (DCSBM) which extends SBM by considering variation in node degree. In recent years, numerous algorithms have been developed to estimate node community for non-overlapping undirected networks generated from SBM and DCSBM, see [5–15]. For recent developments about SBM, see the wonderful review paper [16].

However, in most real-world networks, a node may belong to more than one community at a time. In recent years, the problem of estimating mixed memberships for the undirected network has received a lot of attention [17–29], and references therein. Ref. [17] extends the SBM model from non-overlapping undirected networks to mixed membership undirected networks and designs the mixed membership stochastic block (MMSB) model. Based on the MMSB model, ref. [24] designs a model called the degree corrected mixed membership (DCMM) model by considering degree heterogeneity, where DCMM can also be seen as an extension of the non-overlapping model DCSBM, and ref. [24] also develops an efficient and provably consistent spectral algorithm. Ref. [27] presents a spectral algorithm under MMSB and establishes per-node rates for mixed memberships by sharp



**Citation:** Qing, H. Estimating Mixed Memberships in Directed Networks by Spectral Clustering. *Entropy* **2023**, *25*, 345. <https://doi.org/10.3390/e25020345>

Academic Editors: Valentino Santucci, Marco Bairoletti and Marco Tomassini

Received: 29 December 2022

Revised: 4 February 2023

Accepted: 10 February 2023

Published: 13 February 2023



**Copyright:** © 2023 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

row-wise eigenvector deviation. Ref. [29] proposes an overlapping continuous community assignment model (OCCAM), which is also an extension of MMSB, by considering degree heterogeneity. To fit OCCAM, ref. [29] develops a spectral algorithm requiring a relatively small fraction of mixed nodes when building theoretical frameworks. Ref. [26] finds the cone structure inherent in the normalization of the eigen-decomposition of the population adjacency matrix under DCMM and develops a spectral algorithm to hunt corners in the cone structure.

Though the above works are encouraging and appealing, they focus on undirected networks. In reality, there exist substantial directed networks, such as citation networks, protein–protein interaction networks, and the hyperlink network of websites. In recent years, a lot of works with encouraging results have been developed for directed networks. Ref. [30] proposes a stochastic co-block model (ScBM) and its extension DC-ScBM by considering degree heterogeneity to model non-overlapping directed networks, where ScBM and DC-ScBM can model directed networks whose row nodes may be different from column nodes, and the number of row communities may also be different from the number of column communities. Ref. [31] studies the theoretical guarantees for the algorithm DSCORE [32] and its variants designed under DC-ScBM. Ref. [33] studies the spectral clustering algorithms designed by a data-driven regularization of the adjacency matrix under ScBM. Ref. [34] studies higher-order spectral clustering of directed graphs by designing a nearly linear time algorithm. Based on the fact that the above works only consider non-overlapping directed networks, ref. [35] develops a directed mixed membership stochastic block model (DiMMSB), which is an extension of ScBM, and models directed networks with mixed memberships. DiMMSB can also be seen as a direct extension of MMSB from an undirected network to a directed network.

Recall that DCSBM, DCMM, and DCScBM are extensions of SBM, MMSB, and ScBM by considering node degree variation, respectively, this paper aims at proposing a model as an extension of DiMMSB by considering node degree heterogeneity and building an efficient spectral algorithm to fit the proposed model. In this paper, we focus on the directed network with mixed membership. Our contributions are as follows:

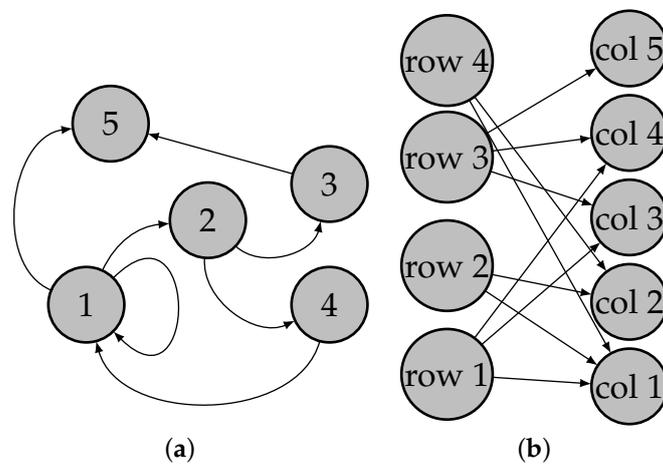
- (i) We propose a novel generative model for directed networks with a mixed membership, the directed degree corrected mixed membership (DiDCMM) model. DiDCMM models a directed network with mixed memberships when row nodes have degree heterogeneities, while column nodes do not. We present the identifiability of DiDCMM under popular conditions which are also required by models modeling mixed membership networks when considering degree heterogeneity. Meanwhile, our results also show that modeling a directed network with mixed membership when considering degree heterogeneity for both row and column nodes needs nontrivial conditions. DiDCMM can be seen as an extension of the DCScBM model from a non-overlapping directed network to an overlapping directed network. DiDCMM also extends the DCMM model from an undirected network to a directed network and extends the DiMMSB model by considering node degree heterogeneity. For a detailed comparison of our DiDCMM with previous models, see Remark 2.
- (ii) To fit DiDCMM, we present a spectral algorithm called DiMSC, which is designed based on the investigation that there exists an ideal cone structure inherent in the normalized version of the left singular vectors and an ideal simplex structure inherent in the right singular vectors of the population adjacency matrix. We prove that our DiMSC exactly recovers the membership matrices for both row and column nodes in the oracle case under DiDCMM, and this also supports the identifiability of DiDCMM. We obtain the upper bounds of error rates for each row (and column) node and show that our method produces asymptotically consistent parameter estimations under mild conditions. Our theoretical results are consistent with classical results when DiDCMM degenerates to SBM and MMSB under mild conditions. Numerical results of simulated directed networks support our theoretical results and show that our approach outperforms its competitors. We also apply our algorithm to several real-

world directed networks to test the existence of highly mixed nodes and asymmetric structures between row and column communities.

**Notations.** We take the following general notations in this paper. For a vector  $x$  and fixed  $q > 0$ ,  $\|x\|_q$  denotes its  $l_q$ -norm. For a matrix  $M$ ,  $M'$  denotes the transpose of the matrix  $M$ ,  $\|M\|$  denotes the spectral norm,  $\|M\|_F$  denotes the Frobenius norm, and  $\|M\|_{2 \rightarrow \infty}$  denotes the maximum  $l_2$ -norm of all the rows of  $M$ . Let  $\text{rank}(M)$  denote the rank of matrix  $M$ . Let  $\sigma_i(M)$  be the  $i$ -th largest singular value of matrix  $M$ , and  $\lambda_i(M)$  denote the  $i$ -th largest eigenvalue of the matrix  $M$  ordered by the magnitude.  $M(i, :)$  and  $M(:, j)$  denote the  $i$ -th row and the  $j$ -th column of matrix  $M$ , respectively.  $M(S_r, :)$  and  $M(:, S_c)$  denote the rows and columns in the index sets  $S_r$  and  $S_c$  of matrix  $M$ , respectively. For any matrix  $M$ , we simply use  $Y = \max(0, M)$  to represent  $Y_{ij} = \max(0, M_{ij})$  for any  $i, j$ . For any matrix  $M \in \mathbb{R}^{m \times m}$ , let  $\text{diag}(M)$  be the  $m \times m$  diagonal matrix whose  $i$ -th diagonal entry is  $M(i, i)$ . Here,  $\mathbf{1}$  and  $\mathbf{0}$  are column vectors with all entries being ones and zeros, respectively;  $e_i$  is a column vector whose  $i$ -th entry is one, while other entries are zero.  $C$  is a positive constant that may vary occasionally.

### 2. The Directed Degree Corrected Mixed Membership Model

Consider a directed network  $\mathcal{N} = (\mathcal{V}_r, \mathcal{V}_c, \mathcal{E})$ , where  $\mathcal{V}_r = \{1, 2, \dots, n_r\}$  is the set of row nodes,  $\mathcal{V}_c = \{1, 2, \dots, n_c\}$  is the set of column nodes ( $n_r$  and  $n_c$  indicate the number of row nodes and the number of column nodes, respectively), and  $\mathcal{E}$  is the set of edges. Note that when  $\mathcal{V}_r = \mathcal{V}_c$  such that row nodes are the same as column nodes,  $\mathcal{N}$  is a traditional directed network [31,36–42]; when  $\mathcal{V}_r \neq \mathcal{V}_c$ ,  $\mathcal{N}$  is a bipartite network (also known as a bipartite graph) [30,33,35,43–45]; see Figure 1 for illustrations of the topological structures for a directed network and a bipartite network. Without confusion, we also call bipartite networks directed networks occasionally in this paper.



**Figure 1.** Illustration for directed network and bipartite network. Panel (a): directed network; Panel (b): bipartite network.

We assume that the row nodes of the directed network  $\mathcal{N}$  belong to  $K$  perceivable communities (called row communities in this paper)

$$\mathcal{C}_r^{(1)}, \mathcal{C}_r^{(2)}, \dots, \mathcal{C}_r^{(K)}, \tag{1}$$

and the column nodes of the directed network  $\mathcal{N}$  belong to  $K$  perceivable communities (called column communities in this paper)

$$\mathcal{C}_c^{(1)}, \mathcal{C}_c^{(2)}, \dots, \mathcal{C}_c^{(K)}. \tag{2}$$

Define an  $n_r \times K$  row nodes membership matrix  $\Pi_r$  and an  $n_c \times K$  column nodes membership matrix  $\Pi_c$  such that  $\Pi_r(i, :)$  is a  $1 \times K$  probability mass function (PMF) for row node  $i$ ,  $\Pi_c(j, :)$  is a  $1 \times K$  PMF for column node  $j$ , and

$$\Pi_r(i, k) \text{ is the weight of row node } i \text{ on } \mathcal{C}_r^{(k)}, 1 \leq k \leq K, \tag{3}$$

$$\Pi_c(j, k) \text{ is the weight of column node } j \text{ on } \mathcal{C}_c^{(k)}, 1 \leq k \leq K. \tag{4}$$

Call row node  $i$  ‘pure’ if  $\Pi_r(i, :)$  is degenerate (i.e., one entry is 1, all other  $K - 1$  entries are 0) and ‘mixed’ otherwise. The same definitions hold for column nodes. Note that mixed nodes considered in this article are not the boundary nodes introduced in [46] since boundary nodes are defined based on non-overlapping networks, while mixed nodes belong to multiple communities.

Let  $A \in \{0, 1\}^{n_r \times n_c}$  be the bi-adjacency matrix of  $\mathcal{N}$  such that for each entry,  $A(i, j) = 1$  if there is a directional edge from row node  $i$  to column node  $j$ , and  $A(i, j) = 0$  otherwise. So, the  $i$ -th row of  $A$  records how row node  $i$  sends edges, and the  $j$ -th column of  $A$  records how column node  $j$  receives edges. Let  $P$  be a  $K \times K$  matrix such that

$$P(k, l) \geq 0 \text{ for } 1 \leq k, l \leq K. \tag{5}$$

Note that since we consider a directed network in this paper,  $P$  may be asymmetric.

Without loss of generality, suppose that row nodes have degree heterogeneities, while column nodes do not i.e., row nodes have variation in degree, while column nodes do not. Note that in a directed network, if column nodes have degree heterogeneities while row nodes do not, to detect memberships of both row nodes and column nodes, we set the transpose of the adjacency matrix as input when applying our algorithm DiMSC. Meanwhile, in a directed network, if both row and column nodes have degree heterogeneity, to model such a directed network with mixed memberships, we need nontrivial constraints on the degree heterogeneities between row nodes and column nodes for model identifiability, for detail, see Remark 1.

Let  $\theta_r$  be an  $n_r \times 1$  vector whose  $i$ -th entry is the positive degree heterogeneity of row node  $i$ . For all pairs of  $(i, j)$  with  $1 \leq i \leq n_r, 1 \leq j \leq n_c$ , DiDCMM models the entries of  $A$  such that  $A(i, j)$  are independent Bernoulli random variables satisfying

$$\mathbb{P}(A(i, j) = 1) = \theta_r(i) \sum_{k=1}^K \sum_{l=1}^K \Pi_r(i, k) \Pi_c(j, l) P(k, l). \tag{6}$$

Equation (6) means that  $\mathbb{P}(A(i, j) = 1) = \theta_r(i) \Pi_r(i, :) P \Pi_c'(j, :)$ , i.e., the probability of generating a directional edge from row node  $i$  to column node  $j$  is  $\theta_r(i) \Pi_r(i, :) P \Pi_c'(j, :)$ , and this probability is controlled by the degree heterogeneity parameter  $\theta_r(i)$  of row node  $i$ , the connecting matrix  $P$ , and the memberships of nodes  $i$  and  $j$ . Equation (6) functions similarly to Equation (1.4) in [24], and both equations define the probability of generating an edge. For comparison, Equation (6) defines the probability of generating a directional edge under DiDCMM for a directed network, while Equation (1.4) in [24] defines the probability of generating an edge under DCMM for an undirected network, i.e., DiDCMM can be seen as an extension of DCMM from an undirected network to a directed network.

Introduce the degree heterogeneity diagonal matrix  $\Theta_r \in \mathbb{R}^{n_r \times n_r}$  for row nodes such that

$$\Theta_r(i, i) = \theta_r(i) \quad \text{for } 1 \leq i \leq n_r. \tag{7}$$

Equation (7) uses a diagonal matrix  $\Theta_r$  to contain all degree heterogeneities, and  $\Theta_r$  is useful for further theoretical analysis through Equation (8).

**Definition 1.** Call model (1)–(6) the directed degree corrected mixed membership (DiDCMM) model, and denote it by  $DiDCMM_{n_r, n_c}(K, P, \Pi_r, \Pi_c, \Theta_r)$ .

The following conditions are sufficient for the identifiability of DiDCMM:

- (I1)  $\text{rank}(P) = K$ , and  $P$  has unit diagonals.
- (I2) There is at least one pure node for each of the  $K$  row and  $K$  column communities.

When building statistical models for a network in which nodes can belong to multiple communities, the full rank requirement of connecting matrix  $P$  and pure nodes condition are always necessary for model identifiability, see models for an undirected network such as MMSB considered in [23,27], DCMM considered in [24,26], and OCCAM considered in [26,29]. Meanwhile, if models modeling networks with mixed memberships consider degree heterogeneity, the unit diagonals requirement on connecting matrix  $P$  is also necessary for model identifiability, see the identifiability requirement of DCMM and OCCAM considered in [24,26,29]. Furthermore, based on the fact that DiDCMM, DCMM, and OCCAM can include the well-known model SBM, letting  $P$  have unit diagonals is not a serious problem since many wonderful works study a special case of SBM when  $P$  has unit diagonals and a network has  $K$  equal size clusters (this special case of SBM is also known as a planted partition model), see [12,47–52].

Let  $\Omega = \mathbb{E}[A]$  be the expectation of the adjacency matrix  $A$ . Under DiDCMM, we have

$$\Omega = \Theta_r \Pi_r P \Pi_c' \tag{8}$$

We refer to  $\Omega$  as the population adjacency matrix. Since  $\text{rank}(\Theta_r) = K, \text{rank}(\Pi_r) = K, \text{rank}(\Pi_c) = K$  and  $\text{rank}(P) = K$  by Equation (7) and Conditions (I1) and (I2), the rank of  $\Omega$  is  $K$ . Recall that  $K$  is the number of communities, and it is much smaller than network size. We see that  $\Omega$  has a low dimensional structure. The form of  $\Omega$  given in Equation (8) is powerful to build the spectral algorithm developed in this paper to fit DiDCMM. Analyzing properties of the population adjacency matrix to build a spectral algorithm fitting statistical model is a common strategy in community detection, for example, references [24,26,27,35] also use this strategy to design their algorithms fitting DCMM, MMSB, and DiDCMM.

For  $1 \leq k \leq K$ , let  $\mathcal{I}_r^{(k)} = \{i \in \{1, 2, \dots, n_r\} : \Pi_r(i, k) = 1\}$  and  $\mathcal{I}_c^{(k)} = \{j \in \{1, 2, \dots, n_c\} : \Pi_c(j, k) = 1\}$ . By Condition (I2),  $\mathcal{I}_r^{(k)}$  and  $\mathcal{I}_c^{(k)}$  are nonempty for all  $1 \leq k \leq K$ . For  $1 \leq k \leq K$ , select one row node from  $\mathcal{I}_r^{(k)}$  to construct the index set  $\mathcal{I}_r$ , i.e.,  $\mathcal{I}_r$  is the indices of row nodes corresponding to  $K$  pure row nodes, one from each community, and  $\mathcal{I}_c$  is defined similarly. W.L.O.G., let  $\Pi_r(\mathcal{I}_r, :) = I_K$  and  $\Pi_c(\mathcal{I}_c, :) = I_K$  (Lemma 2.1 [27] has a similar setting to design their spectral algorithm under MMSB.), where  $I_K$  is the  $K \times K$  identity matrix. The proposition below shows that the DiDCMM model is identifiable.

**Proposition 1.** (Identifiability). *When Conditions (I1) and (I2) hold, DiDCMM is identifiable: for eligible  $(P, \Pi_r, \Pi_c, \Theta_r)$  and  $(\tilde{P}, \tilde{\Pi}_r, \tilde{\Pi}_c, \tilde{\Theta}_r)$ , set  $\Omega = \Theta_r \Pi_r P \Pi_c'$  and  $\tilde{\Omega} = \tilde{\Theta}_r \tilde{\Pi}_r \tilde{P} \tilde{\Pi}_c'$ . If  $\Omega = \tilde{\Omega}$ , then  $\Theta_r = \tilde{\Theta}_r, \Pi_r = \tilde{\Pi}_r, \Pi_c = \tilde{\Pi}_c$  and  $P = \tilde{P}$ .*

**Remark 1.** (The reason that we do not model a directed network with mixed memberships where both row and column nodes have degree heterogeneities). *Suppose both row and column nodes have degree heterogeneities in a mixed membership directed network. To model such a directed network, the probability of generating an edge from row node  $i$  to column node  $j$  is*

$$\mathbb{P}(A(i, j) = 1) = \theta_r(i) \theta_c(j) \sum_{k=1}^K \sum_{l=1}^K \Pi_r(i, k) \Pi_c(j, l) P(k, l),$$

where  $\theta_c$  is an  $n_c \times 1$  vector whose  $j$ -th entry is the degree heterogeneity of column node  $j$ . Set  $\Omega = \mathbb{E}[A]$ , then  $\Omega = \Theta_r \Pi_r P \Pi_c'$ , where  $\Theta_c \in \mathbb{R}^{n_c \times n_c}$  is a diagonal matrix whose  $j$ -th diagonal entry  $\theta_c(j)$ . Set  $\Omega = U \Lambda V'$  as the compact SVD of  $\Omega$ . Follow similar analysis as Lemma 1, we see that  $U = \Theta_r \Pi_r B_r$  and  $V = \Theta_c \Pi_c B_c$  (without causing confusion, we still use  $B_c$  here for convenience.). For model identifiability, follow similar analysis as the proof of Proposition 1, since  $\Omega(\mathcal{I}_r, \mathcal{I}_c) = \Theta_r(\mathcal{I}_r, \mathcal{I}_r) \Pi_r(\mathcal{I}_r, :) P \Pi_c'(\mathcal{I}_c, :) \Theta_c(\mathcal{I}_c, \mathcal{I}_c) = \Theta_r(\mathcal{I}_r, \mathcal{I}_r) P \Theta_c(\mathcal{I}_c, \mathcal{I}_c) = U(\mathcal{I}_r, :$

$)\Lambda V'(\mathcal{I}_c, :)$ , we see that  $\Theta_r(\mathcal{I}_r, \mathcal{I}_r)P\Theta_c(\mathcal{I}_c, \mathcal{I}_c) = U(\mathcal{I}_r, :)\Lambda V'(\mathcal{I}_c, :)$ . To obtain  $\Theta_r(\mathcal{I}_r, \mathcal{I}_r)$  and  $\Theta_c(\mathcal{I}_c, \mathcal{I}_c)$  from  $U(\mathcal{I}_r, :)\Lambda V'(\mathcal{I}_c, :)$ , when  $P$  has unit diagonals, we see that it is impossible to recover  $\Theta_r(\mathcal{I}_r, \mathcal{I}_r)$  and  $\Theta_c(\mathcal{I}_c, \mathcal{I}_c)$  unless we add a condition that  $\Theta_r(\mathcal{I}_r, \mathcal{I}_r) = \Theta_c(\mathcal{I}_c, \mathcal{I}_c)$ . Now, suppose  $\Theta_r(\mathcal{I}_r, \mathcal{I}_r) = \Theta_c(\mathcal{I}_c, \mathcal{I}_c)$  holds and call it Condition (I3); we have  $\Theta_r(\mathcal{I}_r, \mathcal{I}_r)P\Theta_r(\mathcal{I}_r, \mathcal{I}_r) = U(\mathcal{I}_r, :)\Lambda V'(\mathcal{I}_c, :)$ ; hence,  $\Theta_r(\mathcal{I}_r, \mathcal{I}_r) = \Theta_c(\mathcal{I}_c, \mathcal{I}_c) = \sqrt{\text{diag}(U(\mathcal{I}_r, :)\Lambda V'(\mathcal{I}_c, :))}$  when  $P$  has unit diagonals. However, Condition (I3) is nontrivial since it requires  $\Theta_r(\mathcal{I}_r, \mathcal{I}_r) = \Theta_c(\mathcal{I}_c, \mathcal{I}_c)$ , and we always prefer a directed network in which there are no connections between row nodes degree heterogeneities and column nodes degree heterogeneities. For example, when all nodes are pure in a directed network, ref. [30] models such directed network using model DC-ScBM such that  $\Omega = \Theta_r\Pi_r\Pi_r'\Theta_c$  when all nodes are pure, and  $\Theta_r$  and  $\Theta_c$  are independent under DC-ScBM. Because Condition (I3) is nontrivial, we do not model a mixed membership directed network with all nodes having degree heterogeneities.

For DiDCMM's identifiability, the number of row communities should equal that of column communities when both row and column nodes may belong to more than one community. However, when only row nodes have mixed memberships while column nodes do not, the number of row communities can be lesser than that of column communities, and this is also discussed in [53]. All proofs of our theoretical results are provided in the Appendix A.1.

Unless specified, we treat Conditions (I1) and (I2) as default from now on. Proposition 1 is important since it guarantees that our model DiDCMM is well-defined, and we can design efficient spectral algorithms to fit DiDCMM based on its identifiability. The reason that we do not consider degree heterogeneity for column nodes for our DiDCMM is mainly for its identifiability. As analyzed in Remark 1, considering degree heterogeneity for both row and column nodes make the model unidentifiable unless adding some nontrivial conditions on model parameters. Meanwhile, many previous statistical models in the community detection areas are identifiable, and spectral algorithms can be applied to fit them. For examples, SBM [3], DCSBM [4], MMSB [17], DCMM [24], OCCAM [29], ScBM (and DCScBM), [30], and DiMMSB [35] are identifiable. Especially, though different statistical models may have different requirements on model parameters for identifiability, the proof of identifiability enjoys a similar idea as that of Proposition 1, for instance, Proposition 1.1 [24] and Theorem 2.1 [27] build theoretical guarantees on identifiability for DCMM and MMSB, respectively.

**Remark 2.** We compare our DiDCMM with some previous models in this remark.

- When  $\Theta_r = \rho I$  for  $\rho > 0$ , Equation (8) gives  $\Omega = \rho\Pi_r\Pi_r'$  and DiDCMM degenerates to DiMMSB [35], where  $\rho$  is known as a sparsity parameter [9,27,35]. So, DiDCMM includes DiMMSB as a special case, and the relationship between DiDCMM and DiMMSB is similar to that between DCSBM [3,4]. Meanwhile, DiDCMM considers degree heterogeneity parameter  $\Theta_r$  at the cost that DiDCMM requires  $P$  to have unit diagonals for model identifiability, while there is no such requirement for  $P$  on DiMMSB's identifiability. Note that both DiDCMM and DiMMSB are identifiable only when  $P$  is a full-rank square matrix.
- When  $\Theta_r = \rho I$  for  $\rho > 0$  and all nodes are pure, DiDCMM reduces to ScBM [30]. DiDCMM can model a directed network in which nodes enjoy overlapping memberships, while ScBM cannot. Meanwhile, DiDCMM enjoys this advantage at the cost of requiring  $\text{rank}(P) = K$  for model identifiability, while ScBM is identifiable even when  $P$  is not a square matrix, i.e., ScBM can model a directed network in which the number of row communities can be different from the number of column communities. A comparison between DiDCMM and DCScBM [30] is similar.
- When  $\Theta_r = \rho I$  and the network is undirected, DiDCMM reduces to MMSB [17]. However, DiDCMM models directed networks with mixed memberships, while MMSB only models undirected networks with mixed memberships. Again, DiDCMM enjoys its advantage at the cost of  $P$  having unit diagonals for its identifiability (not that DiDCMM allows  $P$  to be asymmetric since DiDCMM models directed networks), while MMSB is identifiable even when

$P$  has non-unit diagonals (note that  $P$  is symmetric under MMSB since it models undirected networks). Meanwhile, the identifiability of both DiDCMM and MMSB requires the square matrix  $P$  to have full rank.

- When  $\Theta_r = \rho I$ , the network is undirected and all nodes are pure, DiDCMM reduces to SBM [3]. For comparison, DiDCMM models directed networks and allows nodes to belong to multiple communities, while SBM only models undirected networks in which a node only belongs to one community. Meanwhile, DiDCMM enjoys these advantages at the cost of requiring  $P$  to be full rank with unit diagonals for its identifiability, while SBM is identifiable even when  $P$  is not full rank and  $P$  has non-unit diagonals. Note that DiDCMM allows  $P$  to be asymmetric, while  $P$  must be symmetric for SBM since DiDCMM models directed networks, while SBM models undirected networks. Comparison between DiDCMM and DCSBM [4] is similar.
- Compared with DCMM introduced in [24] and OCCAM introduced in [29], DCMM, and OCCAM model undirected networks with mixed memberships, while DiDCMM models directed networks with mixed memberships. DiDCMM, DCMM, and OCCAM all consider degree heterogeneity for overlapping networks, and they are identifiable only when the full rank matrix  $P$  has unit diagonals. These three models are identifiable only when the square matrix  $P$  is full rank. Meanwhile, DiDCMM allows  $P$  to be asymmetric, while  $P$  must be symmetric for DCMM and OCCAM since DiDCMM models directed networks, while DCMM and OCCAM model undirected networks.

### 3. Algorithm

The primary goal of the proposed algorithm is to estimate the row membership matrix  $\Pi_r$  and column membership matrix  $\Pi_c$  from the observed adjacency matrix  $A$  with given  $K$ . We start by considering the ideal case when  $\Omega$  is known, and then we extend what we learn in the ideal case to the real case.

#### 3.1. The Ideal Simplex (IS), the Ideal Cone (IC), and the Ideal DiMSC

Recall that  $\text{rank}(\Omega) = K$  under Conditions (I1) and (I2), and  $K$  is much smaller than  $\min\{n_r, n_c\}$ . Let  $\Omega = U\Lambda V'$  be the compact singular value decomposition of  $\Omega$  such that  $U \in \mathbb{R}^{n_r \times K}$ ,  $\Lambda \in \mathbb{R}^{K \times K}$ ,  $V \in \mathbb{R}^{n_c \times K}$ ,  $U'U = I_K$ ,  $V'V = I_K$ . The goal of the ideal case is to use  $U, \Lambda$ , and  $V$  to exactly recover  $\Pi_r$  and  $\Pi_c$ . As stated in [8,24],  $\theta_r$  is one of the major nuisances, and similar to [7], we remove the effect of  $\theta_r$  by normalizing each row of  $U$  to have a unit  $l_2$  norm. Set  $U_* \in \mathbb{R}^{n_r \times K}$  by  $U_*(i, :) = \frac{U(i, :)}{\|U(i, :)\|_F}$ , and let  $N_U$  be the  $n_r \times n_r$  diagonal matrix such that  $N_U(i, i) = \frac{1}{\|U(i, :)\|_F}$  for  $1 \leq i \leq n_r$ . Then,  $U_*$  can be rewritten as  $U_* = N_U U$ . The existences of the ideal cone (IC for short) structure inherent in  $U_*$  and the ideal simplex (IS for short) structure inherent in  $V$  are guaranteed by the following lemma.

**Lemma 1.** (Ideal Simplex and Ideal Cone). Under  $\text{DiDCMM}_{n_r, n_c}(K, P, \Pi_r, \Pi_c, \Theta_r)$ , there exist a unique  $K \times K$  matrix  $B_r$  and a unique  $K \times K$  matrix  $B_c$  such that

- $U = \Theta_r \Pi_r B_r$ , where  $B_r = \Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :)$ , and  $U_* = YU_*(\mathcal{I}_r, :)$  where  $Y = N_M \Pi_r \Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)N_U^{-1}(\mathcal{I}_r, \mathcal{I}_r)$  with  $N_M$  being an  $n_r \times n_r$  diagonal matrix whose diagonal entries are positive. Meanwhile,  $U_*(i, :) = U_*(\bar{i}, :)$  if  $\Pi_r(i, :) = \Pi_r(\bar{i}, :)$  for  $1 \leq i, \bar{i} \leq n_r$ .
- $V = \Pi_c B_c$ , where  $B_c = V(\mathcal{I}_c, :)$ . Meanwhile,  $V(j, :) = V(\bar{j}, :)$  if  $\Pi_c(j, :) = \Pi_c(\bar{j}, :)$  for  $1 \leq j, \bar{j} \leq n_c$ .

Lemma 1 says that the rows of  $V$  form a  $K$ -simplex in  $\mathbb{R}^K$  which we call the ideal simplex (IS), with the  $K$  rows of  $B_c$  being the vertices. Such IS is also found in [24,27,35]. Lemma 1 also shows that the form of  $U_* = YU_*(\mathcal{I}_r, :)$  is actually the ideal cone structure mentioned in [26]. Meanwhile, we remove the influence of  $\theta_r$  by normalizing each row of  $U$  to have a unit norm in this paper. Using the idea of the entry-wise ratio in [8] also works, where ref. [24] develops their spectral algorithms to fit DCMM using the idea of

entry-wise ratio. Designing algorithms based on the nonnegative matrix factorization [25] to fit DiDCMM by adding some constraints on  $\Omega$  may also work. We leave the study of using these ideas to fit DiDCMM or its submodels for our future work.

For column nodes (recall that column nodes have no degree heterogeneities), since  $B_c$  is full rank if  $V$  and  $B_c$  are known in advance, ideally we can exactly recover  $\Pi_c$  by setting  $\Pi_c = VB'_c(B_cB'_c)^{-1} \equiv VB_c^{-1}$ . For convenience, to transfer the ideal case to the real case, set  $Z_c = VB_c^{-1}$ . Since  $Z_c \equiv \Pi_c$ , we have

$$\Pi_c(j, :) = \frac{Z_c(j, :)}{\|Z_c(j, :)\|_1}, 1 \leq j \leq n_c.$$

With given  $V$ , since it enjoys IS structure  $V = \Pi_c B_c \equiv \Pi_c V(\mathcal{I}_c, :)$ , as long as we can obtain  $V(\mathcal{I}_c, :)$  (i.e.,  $B_c$ ), we can recover  $\Pi_c$  exactly. As mentioned in [24,27], for such IS, the successive projection (SP) algorithm [54] (i.e., Algorithm A2 in the Appendix E) can be applied to  $V$  with  $K$  column communities to find the column corner matrix  $B_c$ . The above analysis gives how to recover  $\Pi_c$  with given  $\Omega$  and  $K$  under DiDCMM ideally.

Next, we aim to recover  $\Pi_r$  from  $U$  with the given  $K$ . Since  $\text{rank}(U_*) = K$ ,  $\text{rank}(U_*(\mathcal{I}_r, :)) = K$ . As  $U_*(\mathcal{I}_r, :) \in \mathbb{R}^{K \times K}$ , the inverse of  $U_*(\mathcal{I}_r, :)$  exists. Therefore, Lemma 1 also gives that

$$Y = U_* U_*^{-1}(\mathcal{I}_r, :). \tag{9}$$

Equation (9) holds because  $U_* = YU_*(\mathcal{I}_r, :)$  and  $U_*(\mathcal{I}_r, :)$  is a nonsingular matrix. By Lemma 1, we know that for row nodes, their membership matrix  $\Pi_r$  appears in the expression of  $Y$ . Therefore, we aim to use Equation (9) to find the exact expression of  $\Pi_r$  using  $U, V$ , and  $\Lambda$  by putting  $Y$  at the left-hand side of equality. For our next step, we aim at finding  $\Pi_r$  using Equation (9). Since  $Y = N_M \Pi_r \Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r) N_U^{-1}(\mathcal{I}_r, \mathcal{I}_r)$  by Lemma 1 and  $U_* = N_U U$ , using  $N_M \Pi_r \Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r) N_U^{-1}(\mathcal{I}_r, \mathcal{I}_r)$  and  $N_U U$  to replace  $Y$  and  $U_*$  in Equation (9), respectively, we have  $N_U^{-1} N_M \Pi_r \Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r) N_U^{-1}(\mathcal{I}_r, \mathcal{I}_r) = U U_*^{-1}(\mathcal{I}_r, :)$ , which gives

$$N_U^{-1} N_M \Pi_r = U U_*^{-1}(\mathcal{I}_r, : ) N_U(\mathcal{I}_r, \mathcal{I}_r) \Theta_r(\mathcal{I}_r, \mathcal{I}_r). \tag{10}$$

From Equation (10), we have found the expression of  $\Pi_r$  as a function of  $U, U_*, \Theta_r, N_U$ , and  $\mathcal{I}_r$ , where we do not move  $N_U^{-1} N_M$  to the right-hand side of Equation (10) because it is a diagonal matrix and does not influence the expression of  $\Pi_r$ , see our next step for details. When designing a spectral algorithm in the ideal case with given  $\Omega$  and  $K$ , we aim at recovering  $\Pi_r$  and  $\Pi_c$  by taking advantage of the singular value decomposition of  $\Omega$ . We find that though Equation (10) provides an expression for  $\Pi_r$  by  $\Omega$ 's SVD, there is a term  $\Theta_r(\mathcal{I}_r, \mathcal{I}_r)$  which relates to degree heterogeneity, and we aim at expressing  $\Theta_r(\mathcal{I}_r, \mathcal{I}_r)$  through  $\Omega$ 's SVD. By the proof of Lemma 1, we know that  $\Theta_r(\mathcal{I}_r, \mathcal{I}_r) = \text{diag}(U(\mathcal{I}_r, :)) \Lambda V'(\mathcal{I}_c, :)$  when Condition (I1) holds. Thus, substituting  $\text{diag}(U(\mathcal{I}_r, :)) \Lambda V'(\mathcal{I}_c, :)$  for  $\Theta_r(\mathcal{I}_r, \mathcal{I}_r)$  in Equation (10), we obtain an expression of  $\Pi_r$  such that this expression is directly related to  $\Omega$ 's SVD and two index set  $\mathcal{I}_r$  and  $\mathcal{I}_c$ . For convenience, set  $J_* = N_U(\mathcal{I}_r, \mathcal{I}_r) \Theta_r(\mathcal{I}_r, \mathcal{I}_r) \equiv \text{diag}(U_*(\mathcal{I}_r, :)) \Lambda V'(\mathcal{I}_c, :)$ ,  $Z_r = N_U^{-1} N_M \Pi_r$ ,  $Y_* = U U_*^{-1}(\mathcal{I}_r, :)$ . By Equation (10), we have

$$Z_r = Y_* J_* \equiv U U_*^{-1}(\mathcal{I}_r, : ) \text{diag}(U_*(\mathcal{I}_r, :)) \Lambda V'(\mathcal{I}_c, :). \tag{11}$$

Equation (11) looks similar to Equation (7) of [55]. However, Equation (11) is related to two index sets  $\mathcal{I}_r$  and  $\mathcal{I}_c$ , while Equation (7) of [55] is only related to one index set because Equation (11) aims at designing a spectral algorithm for directed network generated under DiDCMM and Equation (7) of [55] aims at reviewing the generation of the SVM-cone-

DCMMSB algorithm proposed in [26] for undirected network generated under DCMM. Meanwhile, since  $N_U^{-1}N_M$  is an  $n_r \times n_r$  positive diagonal matrix, we have

$$\Pi_r(i, :) = \frac{Z_r(i, :)}{\|Z_r(i, :)\|_1}, 1 \leq i \leq n_r. \tag{12}$$

With given  $\Omega$  and  $K$ , we can obtain  $U, V$ ; thus, the above analysis shows that once the two index sets  $\mathcal{I}_r$  and  $\mathcal{I}_c$  are known, we can exactly recover  $\Pi_r$  by Equations (11) and (12). Meanwhile, from Equation (10), we see that it is important to express  $\Theta_r(\mathcal{I}_r, \mathcal{I}_r)$  as a combination of  $U, V, \Lambda$ , and the two index sets  $\mathcal{I}_r$  and  $\mathcal{I}_c$ , where we successfully obtain an expression of  $\Theta_r(\mathcal{I}_r, \mathcal{I}_r)$  by Condition (I1), the unit diagonal constraint on  $P$ . Otherwise, if  $P$  has no unit diagonals, we cannot obtain an expression of  $\Theta_r(\mathcal{I}_r, \mathcal{I}_r)$  unless adding some nontrivial conditions on model parameters, just as analyzed in Remark 1. Similarly, references [24,26] also design their spectral algorithms to fit DCMM by using the unit diagonal constraint on  $P$  to obtain an expression of a sub-matrix of degree heterogeneity matrix, see Equations (6)–(8) of [55] as an example.

Given  $\Omega$  and  $K$ , to recover  $\Pi_r$  in the ideal case, we need to obtain  $Z_r$  by Equation (11), which means that the only difficulty is in finding the index set  $\mathcal{I}_r$  since  $V(\mathcal{I}_c, :)$  can be obtained by SP algorithm from the IS structure  $V = \Pi_c V(\mathcal{I}_c, :)$ . From Lemma 1, we know that  $U_* = YU_*(\mathcal{I}_r, :)$  forms the IC structure. In [26], their SVM-cone algorithm (i.e., Algorithm A3 in the Appendix F) can exactly obtain the row nodes corner matrix  $U_*(\mathcal{I}_r, :)$  from the ideal cone  $U_* = YU_*(\mathcal{I}_r, :)$  as long as the Condition  $(U_*(\mathcal{I}_r, :))U_*'(\mathcal{I}_r, :))^{-1}\mathbf{1} > 0$  holds (see Lemma 2).

**Lemma 2.** Under  $DiDCMM_{n_r, n_c}(K, P, \Pi_r, \Pi_c, \Theta_r)$ ,  $(U_*(\mathcal{I}_r, :))U_*'(\mathcal{I}_r, :))^{-1}\mathbf{1} > 0$  holds.

Based on the above analysis, we are now ready to give the following four-stage algorithm which we call ideal DiMSC. Input  $\Omega, K$ . Output:  $\Pi_r$  and  $\Pi_c$ .

- Let  $\Omega = U\Lambda V'$  be the compact SVD of  $\Omega$  such that  $U \in \mathbb{R}^{n_r \times K}, V \in \mathbb{R}^{n_c \times K}, \Lambda \in \mathbb{R}^{K \times K}, U'U = I, V'V = I$ . Let  $U_* = N_U U$ , where  $N_U$  is an  $n_r \times n_r$  diagonal matrix whose  $i$ -th diagonal entry is  $\frac{1}{\|U(i, :)\|_F}$  for  $1 \leq i \leq n_r$ .
- Run the SP algorithm on  $V$  assuming that there are  $K$  column communities to obtain the column corner matrix  $V(\mathcal{I}_c, :)$  (i.e.,  $B_c$ ). Run the SVM-cone algorithm on  $U_*$  assuming that there are  $K$  row communities to obtain  $\mathcal{I}_r$ .
- Set  $J_* = \text{diag}(U_*(\mathcal{I}_r, :)\Lambda V'(\mathcal{I}_c, :)), Y_* = UU_*^{-1}(\mathcal{I}_r, :), Z_r = Y_* J_*$  and  $Z_c = VV^{-1}(\mathcal{I}_c, :)$ .
- Recover  $\Pi_r$  and  $\Pi_c$  by setting  $\Pi_r(i, :) = \frac{Z_r(i, :)}{\|Z_r(i, :)\|_1}$  for  $1 \leq i \leq n_r$ , and  $\Pi_c(j, :) = \frac{Z_c(j, :)}{\|Z_c(j, :)\|_1}$  for  $1 \leq j \leq n_c$ .

The following theorem guarantees that ideal DiMSC exactly recovers nodes memberships, and this verifies the identifiability of DiDCMM in turn. Meanwhile, it should be noted that many spectral algorithms designed to fit identifiable statistical models in the community detection area can exactly recover node memberships for the ideal case. For example, the spectral clustering for  $K$  many clusters algorithm addressed in [5] under SBM, the regularized spectral clustering designed in [7] under DCSBM, the SCORE algorithm designed in [8] under DCSBM, the two algorithms designed and studied in [9] under SBM and DCSBM, the RSC- $\tau$  algorithm studied in [11] under SBM, the mixed-SCORE algorithm designed in [24] under DCMM, the DI-SIM algorithm designed in [30] under DCScBM, the D-SCORE algorithm studied in [31,32] under DCScBM, the SVM-cone-DCMMSB algorithm designed in [26] under DCMM, and the SPACL algorithm designed in [27] under MMSB can exactly recover membership matrices under respective models for the ideal case by using the population adjacency matrix to replace the adjacency matrix in the input of these algorithms. The fact that ideal cases for the above spectral algorithms can return community information also supports the identifiability of the above models.

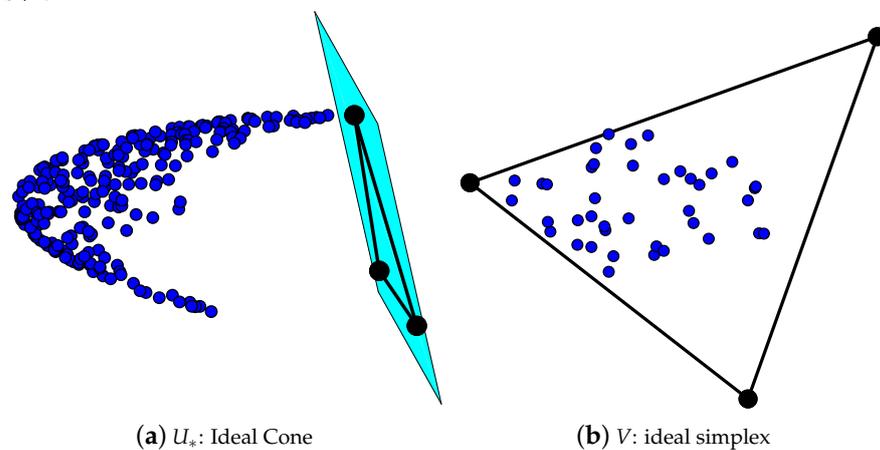
**Theorem 1.** Under  $DiDCMM_{n_r, n_c}(K, P, \Pi_r, \Pi_c, \Theta_r)$ , the ideal DiMSC exactly recovers the row nodes membership matrix  $\Pi_r$  and the column nodes membership matrix  $\Pi_c$ .

To demonstrate that  $U_*$  has the ideal cone structure, we drew Panel (a) of Figure 2. The simulated data used for Panel (a) is generated from  $DiDCMM_{n_r, n_c}(K, P, \Pi_r, \Pi_c, \Theta_r)$  with  $n_r = 600, n_c = 400, K = 3$ ; each row (and column) community has 120 pure nodes. For the 240 mixed row nodes, we set  $\Pi_r(i, 1) = \text{rand}(1)/2, \Pi_r(i, 2) = \text{rand}(1)/2, \Pi_r(i, 3) = 1 - \Pi_r(j, 1) - \Pi_r(j, 2)$ , where  $\text{rand}(1)$  is any random number in  $(0, 1)$ ,

and  $i$  is a mixed row node. For the 40 mixed column nodes, set  $\Pi_c(j, 1) = \text{rand}(1)/2, \Pi_c(j, 2) = \text{rand}(1)/2, \Pi_c(j, 3) = 1 - \Pi_c(j, 1) - \Pi_c(j, 2)$ . For the degree heterogeneity parameter, set  $\theta_r(i) = \text{rand}(1)$  for all row nodes  $i$ . The matrix  $P$  is set as

$$P = \begin{bmatrix} 1 & 0.4 & 0.3 \\ 0.2 & 1 & 0.1 \\ 0.1 & 0.4 & 1 \end{bmatrix}.$$

Under such a setting, after computing  $\Omega$  and obtaining  $U_*, V$  from  $\Omega$ , we can plot Figure 2. Panel (a) shows that all rows respective to mixed row nodes of  $U_*$  are located at one side of the hyperplane formed by the  $K$  rows of  $U_*(\mathcal{I}_r, :)$ , and this phenomenon occurs since each row of  $U_*$  is a scaled convex combination of the  $K$  rows of  $U_*(\mathcal{I}_r, :)$  guaranteed by the IC structure  $U_* = YU_*(\mathcal{I}_r, ;)$ . Thus Panel (a) shows the existence of the ideal cone structure formed by  $U_*$ . Similarly, to demonstrate that  $V$  has the ideal simplex structure, we drew Panel (b) of Figure 2, where Panel (b) is obtained under the same setting as Panel (a). Panel (b) shows that rows respective to mixed column nodes of  $V$  are located inside of the simplex formed by the  $K$  rows of  $V(\mathcal{I}_c, :)$ , and this phenomenon occurs since each row of  $V$  is a convex linear combination of the  $K$  rows of  $V(\mathcal{I}_c, :)$  guaranteed by the IS structure  $V = \Pi_c V(\mathcal{I}_c, ;)$ . Thus Panel (b) shows the existence of the ideal simplex structure formed by  $V$ .



**Figure 2.** Panel (a): plot of  $U_*$  and the hyperplane formed by  $U_*(\mathcal{I}_r, :)$ . Blue points denote rows respective to mixed row nodes of  $U_*$ , and black points denote the  $K$  rows of the corner matrix  $U_*(\mathcal{I}_r, :)$ . The plane in Panel (a) is the hyperplane formed by the triangle of the 3 rows of  $U_*(\mathcal{I}_r, :)$ . Panel (b): plot of  $V$  and the ideal simplex formed by  $V(\mathcal{I}_c, :)$ . Blue points denote rows respective to mixed column nodes of  $V$ , and black points denote the  $K$  rows of the corner matrix  $V(\mathcal{I}_c, :)$ . Since  $K = 3$ , for visualization, we have projected these points from  $\mathbb{R}^3$  to  $\mathbb{R}^2$ .

### 3.2. Dimsc Algorithm

We now extend the ideal case to the real case. Set  $\tilde{A} = \hat{U}\hat{\Lambda}\hat{V}'$  to be the top- $K$ -dimensional SVD of  $A$  such that  $\hat{U} \in \mathbb{R}^{n_r \times K}, \hat{V} \in \mathbb{R}^{n_c \times K}, \hat{\Lambda} \in \mathbb{R}^{K \times K}, \hat{U}'\hat{U} = I_K, \hat{V}'\hat{V} = I_K$ , and  $\hat{\Lambda}$  contains the top  $K$  singular values of  $A$ . Let  $\hat{U}_*$  be the row-wise normalization of  $\hat{U}$  such that  $\hat{U} = N_{\hat{U}}\hat{U}_*$ , where  $N_{\hat{U}} \in \mathbb{R}^{n_r \times n_r}$  is a diagonal matrix whose  $i$ -th diagonal entry is  $\frac{1}{\|\hat{U}(i, :)\|_F}$ . For the real case, we use  $\hat{J}_*, \hat{Y}_*, \hat{Z}_r, \hat{Z}_c, \hat{\Pi}_r, \hat{\Pi}_c$  given in Algorithm 1 to estimate

$J_*, Y_*, Z_r, Z_c, \Pi_r, \Pi_c$ , respectively. Algorithm 1 called directed mixed simplex and cone (DiMSC for short) algorithm is a natural extension of the ideal DiMSC to the real case.

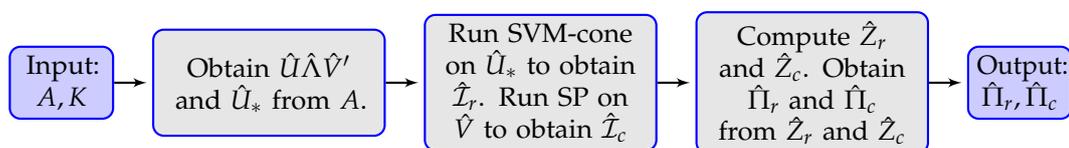
**Algorithm 1** Directed Mixed Simplex and Cone (DiMSC) algorithm

**Require:** The adjacency matrix  $A \in \mathbb{R}^{n_r \times n_c}$  of a directed network, the number of row (column) communities  $K$ .

**Ensure:** The estimated  $n_r \times K$  row membership matrix  $\hat{\Pi}_r$  and the estimated  $n_c \times K$  column membership matrix  $\hat{\Pi}_c$ .

- 1: Obtain  $\tilde{A} = \hat{U}\hat{\Lambda}\hat{V}'$ , the top- $K$ -dimensional SVD of  $A$ . Compute  $\hat{U}_*$  from  $\hat{U}$ .
- 2: Apply SP algorithm (i.e., Algorithm A2) on the rows of  $\hat{V}$  assuming there are  $K$  column communities to obtain  $\hat{\mathcal{I}}_c$ , the index set returned by SP algorithm.
- 3: Similarly, apply SVM-cone algorithm (i.e., Algorithm A3) on the rows of  $\hat{U}_*$  with  $K$  row communities to obtain  $\hat{\mathcal{I}}_r$ , the index set returned by SVM-cone algorithm.
- 4: Set  $\hat{J}_* = \text{diag}(\hat{U}_*(\hat{\mathcal{I}}_r, :)\hat{\Lambda}\hat{V}'(\hat{\mathcal{I}}_c, :)), \hat{Y}_* = \hat{U}\hat{U}_*^{-1}(\hat{\mathcal{I}}_r, :), \hat{Z}_r = \hat{Y}_*\hat{J}_*$  and  $\hat{Z}_c = \hat{V}\hat{V}^{-1}(\hat{\mathcal{I}}_c, :)$ . Then, set  $\hat{Z}_r = \max(0, \hat{Z}_r)$  and  $\hat{Z}_c = \max(0, \hat{Z}_c)$ .
- 5: Estimate  $\Pi_r(i, :)$  by  $\hat{\Pi}_r(i, :) = \hat{Z}_r(i, :)/\|\hat{Z}_r(i, :)\|_1, 1 \leq i \leq n_r$  and estimate  $\Pi_c(j, :)$  by  $\hat{\Pi}_c(j, :) = \hat{Z}_c(j, :)/\|\hat{Z}_c(j, :)\|_1, 1 \leq j \leq n_c$ .

In the third step, we set the negative entries of  $\hat{Z}_r$  as 0 by setting  $\hat{Z}_r = \max(0, \hat{Z}_r)$  for the reason that weights for any row node should be nonnegative, while there may exist some negative entries of  $\hat{Y}_*\hat{J}_*$ . A similar argument holds for  $\hat{Z}_c$ . The flowchart of DiMSC is displayed in Figure 3. Meanwhile, in community detection, researchers often use top- $K$ -dimensional SVD of  $A$  or its variants such as Laplacian matrix or regularized Laplacian matrix to design their spectral clustering algorithms to fit identifiable statistical models such as spectral methods designed or studied in [5,7–9,11,24,26,27,29,31,33,35,56]. Furthermore, as discussed in [57], the SVS<sup>+</sup> and SVS\* algorithms may be used as substitutions of the SP algorithm in our DiMSC for a better estimation of  $\Pi_r$ . When applying the entry-wise normalization idea developed in [8] to deal with  $U$ , as analyzed in [24], we obtain a simplex structure, and we can use the SP algorithm (or the combinatorial vertex search and sketched vertex search approaches developed in [24]) to hunt for the corners. The above ideas suggest that we can design different spectral algorithms to fit our model DiDCMM. We leave them for our future work. In particular, in this paper, we apply the SVM-cone algorithm to hunt for the corners of the cone structure inherent in  $U_*$  mainly for the theoretical convenience of the SVM-cone algorithm because ref. [26] has developed a nice theoretical framework on the performance for the SVM-cone algorithm.



**Figure 3.** Flowchart of Algorithm 1.

3.3. Computational Complexity

The computing cost of DiMSC mainly comes from SVD, SP, and SVM-cone. The computational complexity of SVD is  $O(\max(n_r, n_c)\min(n_r^2, n_c^2))$ . Since the adjacency matrix  $A$  for real-world network data sets is usually sparse, using the power method discussed in [58], the computation complexity for obtaining the top- $K$ -dimensional SVD of  $A$  is only slightly larger than  $O(\max(n_r^2, n_c^2)K)$  [8,24]. The SP algorithm step in DiMSC has a complexity of  $O(\max(n_r, n_c)K^2)$  [24]. The complexity of the one-class SVM step for SVM-cone algorithm is  $O(\max(n_r, n_c)K^2)$  [26,59]. The complexity of the K-means step for SVM-cone algorithm is  $O(\max(n_r, n_c)K^2)$  [60]. Since the number of communities  $K$  considered in this paper is much smaller than the network size, the total complexity of DiMSC is  $O(\max(n_r^2, n_c^2)K)$ . Results in Section 5 show that, for a computer-generated network with 15,000 nodes un-

der SBM, DiMSC takes hundreds of seconds to process a standard personal computer (Thinkpad X1 Carbon Gen 8) using MATLAB R2021b. Meanwhile, many spectral methods developed under models SBM, DCSBM, MMSB, ScBM, DCScBM, OCCAM, DCMM, and DiMMSB for community detection also have complexity  $O(\max(n_r^2, n_c^2)K)$ , see spectral algorithms designed or studied in [5,7–9,11,24,26,27,29–31,33,35,61,62]. Researchers design spectral algorithms for community detection under various identifiable statistical models mainly for their convenience on building a theoretical guarantee of consistent estimation, and we also provide a theoretical guarantee on DiMSC’s estimation consistency in next section.

#### 4. Consistency Results

In this section, we show the consistency of our algorithm for fitting the DiDCMM by proving that the sample-based estimates  $\hat{\Pi}_r$  and  $\hat{\Pi}_c$  concentrate around the true mixed membership matrices  $\Pi_r$  and  $\Pi_c$ . Throughout this paper,  $K$  is a known positive integer. Set  $\theta_{r,\max} = \max_{1 \leq i \leq n_r} \theta_r(i)$  and  $\theta_{r,\min} = \min_{1 \leq i \leq n_r} \theta_r(i)$ . Assume that

**Assumption 1.**  $P_{\max} \max(\|\theta_r\|_1, \theta_{r,\max} n_c) \geq \log(n_r + n_c)$ .

Assumption 1 means that the network cannot be too sparse, and it also means that we allow  $\theta_{r,\max}$  to go to zero with increasing numbers of row nodes and column nodes. When building theoretical guarantees on consistent estimation, controlling network sparsity is popular in the community detection area. For examples, Condition (2.9) of [8], Theorem 3.1 of [9], Condition (2.13) of [24], Assumption 3.1 of [27], and Assumption 2 of [31] all control network sparsity for their theoretical analysis. Especially, when DiDCMM reduces to SBM by letting  $\Theta_r = \rho I, n = n_r = n_c, \Pi_r = \Pi_c$ , and all nodes are pure for  $\rho > 0$ , Assumption 1 requires that  $\rho n \gg \log(n)$ , which is consistent with the sparsity requirement in [8,9,24,31]. As analyzed in [55], we know that our requirement on network sparsity is optimal since it matches the sharp threshold of obtaining a connected Erdős–Rényi (ER) random graph [63] when SBM reduces to an ER random graph by letting  $K = 1$ .

For notation convenience, set  $\omega = \max(\|\hat{U}\hat{U}' - UU'\|_{2 \rightarrow \infty}, \|\hat{V}\hat{V}' - VV'\|_{2 \rightarrow \infty}), \hat{f}_r = \max_{1 \leq i \leq n_r} \|e_i'(\hat{\Pi}_r - \Pi_r \mathcal{P}_r)\|_1, \hat{f}_c = \max_{1 \leq j \leq n_c} \|e_j'(\hat{\Pi}_c - \Pi_c \mathcal{P}_c)\|_1$ , and  $\pi_{r,\min} = \min_{1 \leq k \leq K} \mathbf{1}'\Pi_r e_k$ , where  $\omega$  is the row-wise singular vector deviation which can be bounded by Theorem 4.4 of [64],  $\hat{f}_r$  and  $\hat{f}_c$  measures per node clustering error of DiMSC, and  $\pi_{r,\min}$  measures the minimum summation of row nodes belonging to a certain row community. Increasing  $\pi_{r,\min}$  makes the network tend to be more balanced and vice versa. Meanwhile, row-wise singular vector deviation is important when building a theoretical guarantee of spectral methods fitting models for a network with mixed memberships, for example, refs. [24,26,27,35] also consider  $\omega$  when building consistent estimation for their spectral methods.

The next theorem gives theoretical bounds on estimations of memberships for both row and column nodes, which is the main theoretical result for our DiMSC method.

**Theorem 2.** Under  $DiDCMM_{n_r, n_c}(K, P, \Pi_r, \Pi_c, \Theta_r)$ , let  $\hat{\Pi}_r$  and  $\hat{\Pi}_c$  be obtained from Algorithm 1, when Assumption 1 holds, suppose  $\sigma_K(\Omega) \geq C\sqrt{\theta_{r,\max} P_{\max}(n_r + n_c)\log(n_r + n_c)}$ , with probability at least  $1 - o((n_r + n_c)^{-3})$ , we have

$$\hat{f}_r = O\left(\frac{K^{5.5}\theta_{r,\max}^{15}\omega\kappa^{4.5}(\Pi_r'\Pi_r)\kappa(\Pi_c)\lambda_1^{1.5}(\Pi_r'\Pi_r)}{\theta_{r,\min}^{15}\pi_{r,\min}}\right), \hat{f}_c = O(\omega K\kappa(\Pi_c'\Pi_c)\sqrt{\lambda_1(\Pi_c'\Pi_c)}).$$

In Theorem 2, the Condition  $\sigma_K(\Omega) \geq C\sqrt{\theta_{r,\max} P_{\max}(n_r + n_c)\log(n_r + n_c)}$  is necessary when applying Theorem 4.4 [64] to obtain a theoretical upper bound of  $\omega$ . When building a theoretical guarantee on estimation consistency for spectral methods fitting models modeling network with mixed memberships, it is necessary to have a lower bound requirement on  $\sigma_K(\Omega)$ , see [24,26,27,35]. Actually, this requirement matches with the con-

sistent requirement on  $\frac{\sigma_K(P)}{\sqrt{P_{\max}}}$  obtained from the theoretical upper bound of error rates for a balanced network, see Remark 4 for details. Meanwhile, similar to [7,11,30], we can design a spectral algorithm via an application of regularized Laplacian matrix to fit DiDCMM.

The following corollary is obtained by adding conditions on model parameters similar to Corollary 3.1 in [27], where these conditions give a directed network in which each community has the same order of size, and each node has the same order of degree, i.e., a balanced network.

**Corollary 1.** *Under DiDCMM<sub>n<sub>r</sub>,n<sub>c</sub></sub>(K, P, Π<sub>r</sub>, Π<sub>c</sub>, Θ<sub>r</sub>), when conditions of Theorem 2 hold, suppose  $\lambda_K(\Pi'_r \Pi_r) = O(\frac{n_r}{K})$ ,  $\lambda_K(\Pi'_c \Pi_c) = O(\frac{n_c}{K})$ ,  $\pi_{r,\min} = O(\frac{n_r}{K})$  and  $K = O(1)$ , with probability at least  $1 - o((n_r + n_c)^{-3})$ , we have*

$$\hat{f}_r = O\left(\left(\frac{\theta_{r,\max}}{\theta_{r,\min}}\right)^{15.5} \frac{1}{\sigma_K(P)} \sqrt{\frac{P_{\max} \log(n_r + n_c)}{\theta_{r,\min} n_c}}\right), \hat{f}_c = O\left(\left(\frac{\theta_{r,\max}}{\theta_{r,\min}}\right)^{0.5} \frac{1}{\sigma_K(P)} \sqrt{\frac{P_{\max} \log(n_r + n_c)}{\theta_{r,\min} n_r}}\right).$$

Meanwhile,

- when  $\theta_{r,\max} = O(\rho)$ ,  $\theta_{r,\min} = O(\rho)$  (i.e.,  $\frac{\theta_{r,\min}}{\theta_{r,\max}} = O(1)$ ), we have

$$\hat{f}_r = O\left(\frac{1}{\sigma_K(P)} \sqrt{\frac{P_{\max} \log(n_r + n_c)}{\rho n_c}}\right), \hat{f}_c = O\left(\frac{1}{\sigma_K(P)} \sqrt{\frac{P_{\max} \log(n_r + n_c)}{\rho n_r}}\right).$$

- when  $n_r = O(n)$ ,  $n_c = O(n)$  and  $\theta_{r,\max} = O(\rho)$ ,  $\theta_{r,\min} = O(\rho)$ , we have

$$\hat{f}_r = O\left(\frac{1}{\sigma_K(P)} \sqrt{\frac{P_{\max} \log(n)}{\rho n}}\right), \hat{f}_c = O\left(\frac{1}{\sigma_K(P)} \sqrt{\frac{P_{\max} \log(n)}{\rho n}}\right).$$

Consider a directed mixed membership network under the settings of Corollary 1 when  $\theta_{r,\max} = O(\rho)$ ,  $\theta_{r,\min} = O(\rho)$  for  $\rho > 0$ , to obtain consistent estimations for both row nodes and column nodes, by Corollary 1,  $\frac{\sigma_K(P)}{\sqrt{P_{\max}}}$  should shrink slower than  $\sqrt{\frac{\log(n_r + n_c)}{\rho \min(n_r, n_c)}}$ , where consistent estimation means that the theoretical upper bound of error rate goes to zero when increasing network size. Especially, when  $n_r = O(n)$  and  $n_c = O(n)$ ,  $\frac{\sigma_K(P)}{\sqrt{P_{\max}}}$  should shrink slower than  $\sqrt{\frac{\log(n)}{n}}$ . We further assume that  $P = (2 - \beta)I_K + (\beta - 1)\mathbf{1}\mathbf{1}'$  for  $\beta \in [1, 2) \cup (2, \infty)$  and let  $\tilde{P} = \rho P$  (note that for this  $P$ , we have  $\sigma_K(P) = |\beta - 2|$  and  $P_{\max} = \max(1, \beta - 1)$ ). So the diagonal elements for  $\tilde{P}$  are  $\rho$  and non-diagonal elements are  $\rho(\beta - 1)$ . Set  $p_{\text{in}}$  as the diagonal entries of  $\tilde{P}$ , and  $p_{\text{out}}$  as the non-diagonal entries of  $\tilde{P}$ , we have  $p_{\text{in}} = \rho$ ,  $p_{\text{out}} = \rho(\beta - 1)$ , and  $\frac{|p_{\text{in}} - p_{\text{out}}|}{\sqrt{\max(p_{\text{in}}, p_{\text{out}})}} = \frac{\sqrt{\rho}|\beta - 2|}{\sqrt{\max(1, \beta - 1)}} = \frac{\sqrt{\rho}\sigma_K(P)}{\sqrt{P_{\max}}}$ . Hence, for consistent estimation, we see that  $\frac{|p_{\text{in}} - p_{\text{out}}|}{\sqrt{\max(p_{\text{in}}, p_{\text{out}})}}$  should shrink slower than  $\sqrt{\frac{\log(n_r + n_c)}{\min(n_r, n_c)}}$  by Corollary 1 and should shrink slower than  $\sqrt{\frac{\log(n)}{n}}$  when  $n_r = O(n)$  and  $n_c = O(n)$ , where this result is consistent with classical separation condition for a standard network with two equal-sized clusters by applying the separation condition and sharp threshold criterion developed in [55].

**Remark 3.** *When the network is undirected (i.e.,  $n_r = n_c = n$ ,  $\Pi_r = \Pi_c$ ) with  $K = O(1)$  by setting  $\theta_r(i) = \rho$  for  $1 \leq i \leq n_r$ , DiDCMM degenerates to MMSB considered in [27], the upper bound of error rate for DiMSC is  $O\left(\frac{1}{\sigma_K(P)} \sqrt{\frac{\log(n)}{\rho n}}\right)$  when  $P_{\max} = 1$ . Replacing the  $\Theta$  in [24] by  $\Theta = \sqrt{\rho}I$ , their DCMM model degenerates to MMSB. Then, their conditions in Theorem 2.2 are our Assumption 1 and  $\lambda_K(\Pi'\Pi) = O(\frac{n}{K})$ , where  $\Pi = \Pi_r = \Pi_c$  for MMSB. When  $K = O(1)$ , the error bound in Theorem 2.2 in [24] is  $O\left(\frac{1}{\sigma_K(P)} \sqrt{\frac{\log(n)}{\rho n}}\right)$ , which is consistent with ours.*

**Remark 4.** By Lemma A5 in the Appendix D, we know  $\sigma_K(\Omega) \geq \theta_{r,\min}\sigma_K(P)\sigma_K(\Pi_r)\sigma_K(\Pi_c)$ . To ensure the Condition  $\sigma_K(\Omega) \geq C(\theta_{r,\max}P_{\max}(n_r + n_c)\log(n_r + n_c))^{1/2}$  in Theorem 2 holds, we need

$$\frac{\sigma_K(P)}{\sqrt{P_{\max}}} \geq C\left(\frac{\theta_{r,\max}(n_r + n_c)\log(n_r + n_c)}{\theta_{r,\min}^2\lambda_K(\Pi_r'\Pi_r)\lambda_K(\Pi_c'\Pi_c)}\right)^{1/2}. \tag{13}$$

When  $K = O(1), n_r = O(n), n_c = O(n), \lambda_K(\Pi_r'\Pi_r) = O(\frac{n_r}{K}), \lambda_K(\Pi_c'\Pi_c) = O(\frac{n_c}{K})$  and  $\theta_{r,\max} = O(\rho), \theta_{r,\min} = O(\rho)$ , Equation (13) gives that  $\frac{\sigma_K(P)}{\sqrt{P_{\max}}}$  should shrink slower than  $\sqrt{\frac{\log(n)}{\rho n}}$ , which matches with the consistency requirement on  $\frac{\sigma_K(P)}{\sqrt{P_{\max}}}$  of Corollary 1.

For convenience, we need the following definition.

**Definition 2.** Let  $DiDCMM(n, K, \Pi_r, \Pi_c, \alpha_{in}, \alpha_{out})$  be a special case of  $DiMMDF_{n_r, n_c}(K, P, \Pi_r, \Pi_c, \Theta_r)$  when  $\Theta_r = \rho I, n_r = n_c = n, \lambda_K(\Pi_r'\Pi_r) = O(n/K), \lambda_K(\Pi_c'\Pi_c) = O(n/K), \pi_{r,\min} = O(n/K), K = O(1)$ , and  $\tilde{P} = \rho P$  has diagonal entries  $p_{in} = \alpha_{in}\frac{\log(n)}{n}$  and non-diagonal entries  $p_{out} = \alpha_{out}\frac{\log(n)}{n}$ .

$DiDCMM(n, K, \Pi_r, \Pi_c, \alpha_{in}, \alpha_{out})$  denotes a special directed network such that row communities have nearly equal sizes since  $\lambda_K(\Pi_r'\Pi_r) = O(n/K)$ , and column communities also have nearly equal sizes. By Corollary 1, for consistent estimation, we need

$$\frac{|p_{in} - p_{out}|}{\sqrt{\max(p_{in}, p_{out})}} \gg \sqrt{\frac{\log(n)}{n}} \text{ under } DiDCMM(n, K, \Pi_r, \Pi_c, \alpha_{in}, \alpha_{out}). \text{ Since } \frac{|p_{in} - p_{out}|}{\sqrt{\max(p_{in}, p_{out})}} = \frac{|\alpha_{in} - \alpha_{out}| \sqrt{\frac{\log(n)}{n}}}{\sqrt{\max(\alpha_{in}, \alpha_{out})}}, \text{ for consistent estimation, we need}$$

$$\frac{|\alpha_{in} - \alpha_{out}|}{\sqrt{\max(\alpha_{in}, \alpha_{out})}} \gg 1 \tag{14}$$

Our numerical results in Section 5 support that DiMSC can estimate memberships for both row and column nodes when the threshold  $\frac{|\alpha_{in} - \alpha_{out}|}{\sqrt{\max(\alpha_{in}, \alpha_{out})}} \gg 1$  holds under  $DiDCMM(n, K, \Pi_r, \Pi_c, \alpha_{in}, \alpha_{out})$ .

**Remark 5.** When  $K = 2$ , the network is undirected (i.e.,  $\Pi_r = \Pi_c$ ), all nodes are pure, and each community has an equal size,  $DiDCMM(n, K, \Pi_r, \Pi_c, \alpha_{in}, \alpha_{out})$  reduces to the SBM case such that nodes connect with probability  $p_{in}$  within clusters and  $p_{out}$  across clusters. This case has been well studied in recent years, see [50] and references therein. Especially, for this case, ref. [50] finds that exact recovery is possible if  $|\sqrt{\alpha_{in}} - \sqrt{\alpha_{out}}| > \sqrt{2}$  and impossible if  $|\sqrt{\alpha_{in}} - \sqrt{\alpha_{out}}| < \sqrt{2}$ . For convenience, we use  $SBM(n, p_{in}, p_{out})$  to denote this case. Our numerical results in Section 5 show that DiMSC return consistent estimation under  $SBM(n, p_{in}, p_{out})$  when  $\alpha_{in}$  and  $\alpha_{out}$  are set in the impossible region of exact recovery but satisfy Equation (14).

**Remark 6.** In information theory, Shannon entropy [65] quantifies the amount of information in a variable, and it is a measure of uncertainty information of a probability distribution. We use a node membership entropy (NME) derived from Shannon theory to measure the node's uncertainty about the node and all communities [66,67]. For row node  $i$  with membership  $\Pi_r(i, :)$ , since  $\sum_{k=1}^K \Pi_r(i, k) = 1$  and  $\Pi_r(i, k)$  can be seen as the probability that row node  $i$  belongs to row cluster  $k$  for  $1 \leq k \leq K$ , NME of row node  $i$  is the Shannon entropy related to  $\Pi_r(i, :)$ :

$$NME(i) = - \sum_{k=1}^K \Pi_r(i, k)\log(\Pi_r(i, k)). \tag{15}$$

For column node  $j$  with membership  $\Pi_c(j, :)$ , we can also obtain its NME by Equation (15). In particular, if a node belongs to each cluster with equal probability  $\frac{1}{K}$ , its NME is  $\log(K)$  which is

the maximum among all NME; if a node belongs to two clusters with equal probability  $\frac{1}{2}$ , its NME is  $\log(2)$  which is less than  $\log(K)$  when  $K \geq 3$ . Generally, we see that recovering memberships for mixed nodes is harder than for pure nodes since NME is 0 for pure nodes, while NME is larger than 0 for mixed nodes by the definition of NME.

### 5. Simulations

In this section, several experiments are conducted to investigate the performance of our DiMSC under DiDCMM. We compare our DiMSC with three model-based methods that can be thought of as special cases of our model DiDCMM. Model-based methods we compare include the DISIM algorithm proposed in [30], the DSCORE algorithm studied in [31], and the DiPCA algorithm which is obtained by using the adjacency matrix  $A$  to replace the regularized graph Laplacian matrix in the DISIM algorithm. Similar to [24,27], for simulations, we measure the errors for the inferred community membership matrices instead of simply each node. We measure the performance of DiMSC and its competitors by the mixed Hamming error rate (MHamm for short) defined below

$$\text{MHamm} = \max\left(\frac{\min_{\mathcal{P} \in S_{\mathcal{P}}} \|\hat{\Pi}_r \mathcal{P} - \Pi_r\|_1}{n_r}, \frac{\min_{\mathcal{P} \in S_{\mathcal{P}}} \|\hat{\Pi}_c \mathcal{P} - \Pi_c\|_1}{n_c}\right), \tag{16}$$

where  $S_{\mathcal{P}}$  is the set of  $K \times K$  permutation matrices.

For all simulations in this section, unless specified, we set the parameters  $(n_r, n_c, K, P, \Pi_r, \Pi_c, \Theta_r)$  under DiDCMM as follows: let each row community and each column community have  $n_0$  pure nodes; let all mixed row nodes (and mixed column nodes) have membership  $(1/K, 1/K, \dots, 1/K)$ ; for  $z \geq 1$ , we generate the degree parameters for row nodes as below: let  $\bar{\theta}_r \in \mathbb{R}^{n_r \times 1}$  such that  $1/\bar{\theta}_r(i) \stackrel{iid}{\sim} U(1, z)$  for  $1 \leq i \leq n_r$ , where  $U(1, z)$  denotes the uniform distribution on  $[1, z]$ , and set  $\theta_r = \rho \bar{\theta}_r$ , where we use  $\rho$  to control the sparsity of the network; when  $K = 2$ ,  $P$  is set as

$$P_1 = \begin{bmatrix} 1 & 0.1 \\ 0.2 & 1 \end{bmatrix} \text{ or } P_2 = \begin{bmatrix} 0.8 & 0.1 \\ 0.2 & 0.9 \end{bmatrix};$$

when  $K = 3$ ,

$$P_3 = \begin{bmatrix} 1 & 0.1 & 0.3 \\ 0.2 & 1 & 0.4 \\ 0.5 & 0.2 & 1 \end{bmatrix} \text{ or } P_4 = \begin{bmatrix} 0.8 & 0.1 & 0.3 \\ 0.2 & 0.9 & 0.4 \\ 0.5 & 0.2 & 1 \end{bmatrix};$$

where  $P_2$  and  $P_4$  have non-unit diagonals, and we consider the two cases because we want to investigate DiMSC's sensitivity when  $P$  has non-unit diagonals such that  $P$  disobeys Condition (I1).

After obtaining  $P, \Pi_r, \Pi_c, \theta_r$ , similar to the five simulation steps in [8], each simulation experiment contains the following steps:

(a) Let  $\Theta_r$  be the  $n_r \times n_r$  diagonal matrix such that  $\Theta_r(i, i) = \theta_r(i), 1 \leq i \leq n_r$ . Set  $\Omega = \Theta_r \Pi_r P \Pi_c'$ .

(b) Let  $W$  be an  $n_r \times n_c$  matrix such that  $W(i, j)$  are independent centered-Bernoulli with parameters  $\Omega(i, j)$ . Let  $\tilde{A} = \Omega + W$ .

(c) Set  $\tilde{S}_r = \{i : \sum_{j=1}^{n_c} \tilde{A}(i, j) = 0\}$  and  $\tilde{S}_c = \{j : \sum_{i=1}^{n_r} \tilde{A}(i, j) = 0\}$ , i.e.,  $\tilde{S}_r$  ( $\tilde{S}_c$ ) is the set of row (column) nodes with 0 edges. Let  $A$  be the adjacency matrix obtained by removing rows respective to nodes in  $\tilde{S}_r$  and removing columns respective to nodes in  $\tilde{S}_c$  from  $\tilde{A}$ . Similarly, update  $\Pi_r$  by removing nodes in  $\tilde{S}_r$  and update  $\Pi_c$  by removing nodes in  $\tilde{S}_c$ .

(d) Apply the DiMSC algorithm (and its competitors) to  $A$ . Record MHamm under investigations.

(e) Repeat (b)–(d) 50 times, and report the averaged MHamm over the 50 repetitions.

Let  $n_{r,A}$  be the number of rows of  $A$  and  $n_{c,A}$  be the number of columns of  $A$ . In our experiments,  $n_{r,A}$  and  $n_{c,A}$  are usually very close to  $n_r$  and  $n_c$ ; therefore we do not

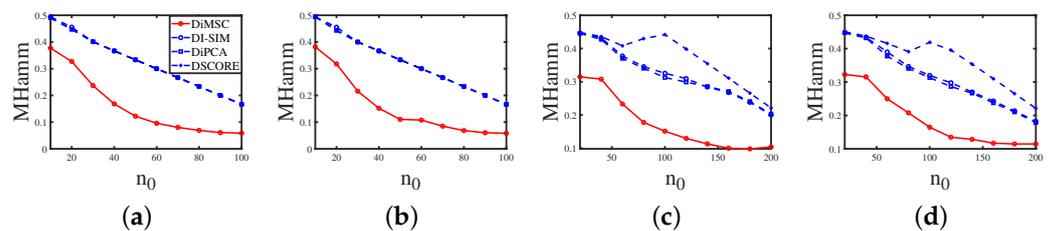
report the exact values of  $n_{r,A}$  and  $n_{c,A}$ . After providing the above steps about how to generate  $A$  numerically under DiDCMM and how to record the error rates, now we describe our experiments in detail. We consider six experiments here. In experiments 1–6, we study the influence of the fraction of pure nodes, degree heterogeneity, connectivity across communities, sparsity, phase transition, and network size on performances of these methods, respectively.

Experiment 1 (a): Fraction of pure nodes. Set  $n_r = 200, n_c = 300, z = 5, \rho = 1$  and  $P$  as  $P_1$ . Let  $n_0$  range in  $\{10, 20, 30, \dots, 100\}$ . The numerical results are shown in Panel (a) of Figure 4. The results show that as the fraction of pure nodes increases for both row and column communities, all approaches perform better. Meanwhile, DiMSC performs best among all methods in Experiment 1 (a).

Experiment 1 (b): Fraction of pure nodes. All parameters are set the same as Experiment 1 (a) except that we set  $P$  as  $P_2$  here. The numerical results are shown in Panel (b) of Figure 4. The results show that all methods perform better as  $n_0$  increases, DiMSC outperforms its competitors, and DiMSC enjoys satisfactory performance even when  $P$  has non-unit diagonals.

Experiment 1 (c): Fraction of pure nodes. Set  $n_r = 600, n_c = 900, z = 5, \rho = 1$ , and  $P$  as  $P_3$ . Let  $n_0$  range in  $\{20, 40, 60, \dots, 200\}$ . The numerical results are shown in Panel (c) of Figure 4, and we see that all methods perform better when there are more pure nodes and our DiMSC performs best.

Experiment 1 (d): Fraction of pure nodes. All parameters are set the same as Experiment 1 (c) except that we set  $P$  as  $P_4$  here. The numerical results are shown in Panel (d) of Figure 4, and the analysis is similar to that of Experiment 1 (b).



**Figure 4.** Errors against increasing  $n_0$ . y-axis: MHamm. Panel (a): Experiment 1 (a); Panel (b): Experiment 1 (b); Panel (c): Experiment 1 (c); Panel (d): Experiment 1 (d).

Experiment 2 (a): Degree heterogeneity. Set  $n_r = 200, n_c = 300, n_0 = 80, \rho = 1$ , and  $P$  as  $P_1$ . Let  $z$  range in  $\{2, 3, 4, \dots, 12\}$ . A larger  $z$  generates lesser edges. The results are displayed in Panel (a) of Figure 5. The results suggest that the error rates of DiMSC for both row and column nodes tend to increase as  $z$  increases. This phenomenon happens because decreasing degree heterogeneities for row nodes lowers the number of edges in the directed network; thus the network becomes harder to be detected for both row and column nodes. Meanwhile, DiMSC outperforms its competitors in this experiment, and it is interesting to see that the error rates of DI-SIM, DiPCA, and DSCORE are almost the same for this experiment.

Experiment 2 (b): Degree heterogeneity. All parameters are set the same as Experiment 2 (a) except that we set  $P$  as  $P_2$  here. The results are displayed in Panel (b) of Figure 5, and we see that DiMSC performs satisfactorily when the directed network is not too sparse (i.e., a small  $z$  case) even when  $P$  has non-unit diagonals. Meanwhile, DiMSC significantly outperforms its competitors in this experiment.

Experiment 2 (c): Degree heterogeneity. Set  $n_r = 600, n_c = 900, n_0 = 150, \rho = 1$ , and  $P$  as  $P_3$ . Let  $z$  range in  $\{2, 3, 4, \dots, 12\}$ . The results are shown in Panel (c) of Figure 5 and can be analyzed similarly to Experiment 2 (a).

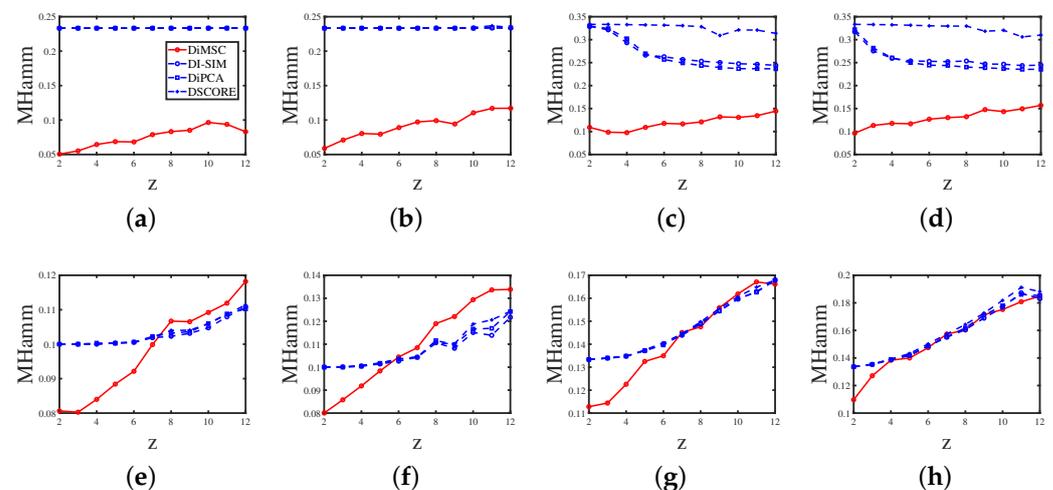
Experiment 2 (d): Degree heterogeneity. All parameters are set the same as Experiment 2 (c) except that we set  $P$  as  $P_4$  here. The results are displayed in Panel (d) of Figure 5 and are similar to that of Experiment 2 (b).

**Experiment 2 (e): Degree heterogeneity.** All parameters are set the same as Experiment 2(a) except that we set  $n_0 = 0$  (so there are no pure nodes in both row and column communities), and all mixed row nodes have two different memberships (0.9, 0.1) and (0.1, 0.9), each with  $\frac{n_r}{K} = 100$  number of row nodes, and all mixed column nodes also have the above two memberships, each with  $\frac{n_c}{K} = 150$  number of column nodes. Panel (e) of Figure 5 shows the results, and we see that DiMSC performs satisfactorily for a small  $z$  even for the case when there are no pure nodes for both row and column communities. Meanwhile, DiMSC performs better than its competitors when  $z < 7$ , and it perform poorer than its competitors when  $z \geq 8$  for this experiment. Furthermore, compared with numerical results of Experiment 2 (a), we see that DI-SIM, DiPCA, and DSCORE have better performances in Experiment 2 (e). The possible reason is the memberships (0.9, 0.1) and (0.1, 0.9) are close to (1, 0) and (0, 1) somewhat.

**Experiment 2 (f): Degree heterogeneity.** All parameters are set the same as Experiment 2 (b) except that we set  $\Pi_r$  and  $\Pi_c$  the same as Experiment 2 (e). The results are shown in Panel (f) of Figure 5 and are similar to that of Experiment 2 (e).

**Experiment 2 (g): Degree heterogeneity.** All parameters are set the same as Experiment 2 (c) except that we set  $n_0 = 0$ , all mixed row nodes have three different memberships (0.8, 0.1, 0.1), (0.1, 0.8, 0.1), and (0.1, 0.1, 0.8), each with  $\frac{n_r}{K} = 200$  number of row nodes, and all mixed column nodes also have the above four memberships, each with  $\frac{n_c}{K} = 300$  number of column nodes. The results are displayed in Panel (g) of Figure 5 and are similar to that of Experiment 2 (e).

**Experiment 2 (h): Degree heterogeneity.** All parameters are set the same as Experiment 2 (d) except that we set  $\Pi_r$  and  $\Pi_c$  the same as Experiment 2 (g). The results are shown in Panel (h) of Figure 5 and are similar to that of Experiment 2 (e).



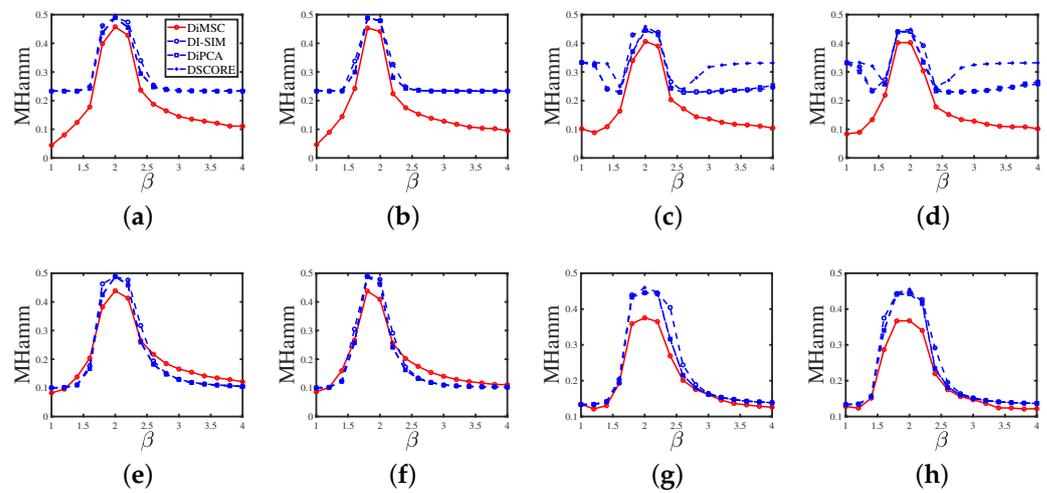
**Figure 5.** Errors against increasing  $z$ .  $y$ -axis: MHammm. Panel (a): Experiment 2 (a); Panel (b): Experiment 2 (b); Panel (c): Experiment 2 (c); Panel (d): Experiment 2 (d); Panel (e): Experiment 2 (e); Panel (f): Experiment 2 (f); Panel (g): Experiment 2 (g); Panel (h): Experiment 2 (h).

**Experiment 3 (a): Connectivity across communities.** Set  $n_r = 200, n_c = 300, n_0 = 80, z = 5, \rho = 1$ . Set

$$P = \begin{bmatrix} 1 & \beta - 1 \\ \beta - 1 & 1 \end{bmatrix}.$$

and let  $\beta$  range in  $\{1, 1.2, 1.4, \dots, 4\}$ . Decreasing  $|\beta - 2|$  increases the hardness of detecting such directed networks. Note that  $\mathbb{P}(A(i, j) = 1) = \Omega(i, j) = \theta_r(i)\Pi_r(i, :)\Pi_c^t(j, :)$  gives  $\max_{i,j}\Omega(i, j) = \theta_{r,\max}P_{\max}$  should be no larger than 1. Since  $P_{\max}$  may be larger than one in this experiment, after obtaining  $\theta_r$ , we need to update  $\theta_r$  as  $\theta_r/P_{\max}$ . The results are displayed in Panel (a) of Figure 6, and they support the arguments given after Corollary 1

such that DiMSC performs better when  $|\beta - 2|$  increases and vice versa. Meanwhile, our DiMSC outperforms its competitors in this experiment.



**Figure 6.** Errors against increasing  $\beta$ .  $y$ -axis: MHammm. Panel (a): Experiment 3 (a); Panel (b): Experiment 3 (b); Panel (c): Experiment 3 (c); Panel (d): Experiment 3 (d); Panel (e): Experiment 3 (e); Panel (f): Experiment 3 (f); Panel (g): Experiment 3 (g); Panel (h): Experiment 3 (h).

Experiment 3 (b): Connectivity across communities. All parameters are set the same as Experiment 3 (a) except that we set

$$P = \begin{bmatrix} 0.8 & \beta - 1 \\ \beta - 1 & 0.9 \end{bmatrix}.$$

The results are displayed in Panel (b) of Figure 6, and we see that DiMSC performs better when  $|\beta - 2|$  increases even for the case that  $P$  has non-unit diagonals. Meanwhile, our DiMSC performs better than its competitors here.

Experiment 3 (c): Connectivity across communities. Set  $n_r = 600, n_c = 900, n_0 = 150, z = 5, \rho = 1$ . Set

$$P = \begin{bmatrix} 1 & \beta - 1 & \beta - 1 \\ \beta - 1 & 1 & \beta - 1 \\ \beta - 1 & \beta - 1 & 1 \end{bmatrix}.$$

and let  $\beta$  range in  $\{1, 1.2, 1.4, \dots, 4\}$ . The results are displayed in Panel (c) of Figure 6 and can be analyzed similarly to Experiment 3 (a).

Experiment 3 (d): Connectivity across communities. All parameters are set the same as Experiment 3(c) except that we set

$$P = \begin{bmatrix} 0.8 & \beta - 1 & \beta - 1 \\ \beta - 1 & 0.9 & \beta - 1 \\ \beta - 1 & \beta - 1 & 1 \end{bmatrix}.$$

The results are displayed in Panel (d) of Figure 6 and can be analyzed similarly to Experiment 3 (b).

Experiment 3 (e): Connectivity across communities. All parameters are set the same as Experiment 3(a) except that we let  $\Pi_r$  and  $\Pi_c$  be the same as that of Experiment 2 (e) (so there are no pure nodes in both row and column communities.). Panel (e) of Figure 6 shows the results, and we see that DiMSC enjoys better performance when  $|\beta - 2|$  increases even in the case that there are no pure nodes for both row and column communities. Meanwhile, all methods have competitive performances for this experiment, and the possible reason that DiMSC’s competitors enjoy better performances here than in Experiment 3 (a) is analyzed in Experiment 2 (e).

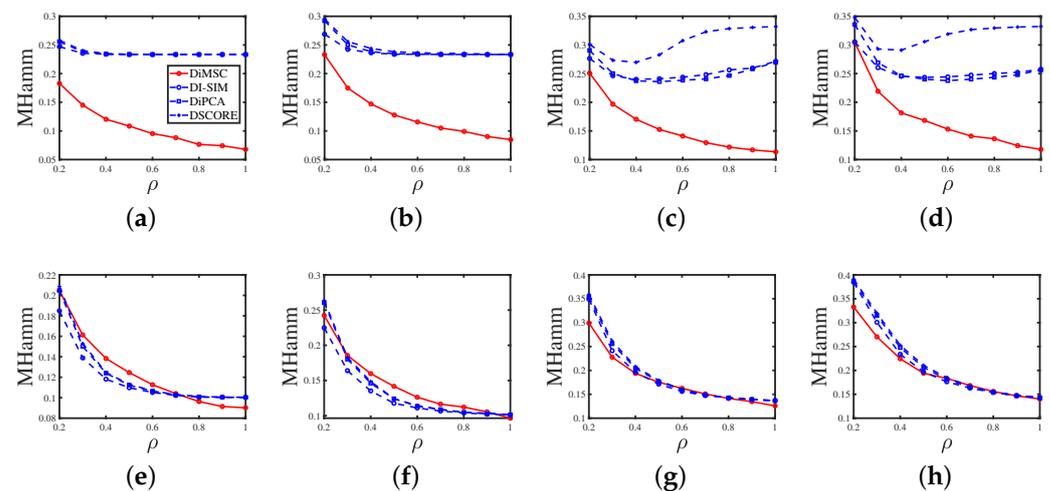
Experiment 3 (f): Connectivity across communities. All parameters are set the same as Experiment 3 (b) except that we set  $\Pi_r$  and  $\Pi_c$  the same as Experiment 2 (e). The results are displayed in Panel (f) of Figure 6 and can be analyzed similarly to Experiment 3 (e).

Experiment 3 (g): Connectivity across communities. All parameters are set the same as Experiment 3 (c) except that we let  $\Pi_r$  and  $\Pi_c$  be the same as that of Experiment 2 (g) (so there are no pure nodes). Panel (g) of Figure 6 shows the results, and the analysis is similar to that of Experiment 3 (b).

Experiment 3 (h): Connectivity across communities. All parameters are set the same as Experiment 3 (d) except that we set  $\Pi_r$  and  $\Pi_c$  the same as Experiment 2 (g). Panel (h) of Figure 6 shows the results, and the analysis is similar to that of Experiment 3 (b).

Experiment 4 (a): Sparsity. Set  $n_r = 200, n_c = 300, n_0 = 80, z = 5$ , and  $P$  as  $P_1$ . Let  $\rho$  range in  $\{0.2, 0.3, \dots, 1\}$ . A larger  $\rho$  indicates a denser network. Panel (a) in Figure 7 displays the simulation results of this experiment. We see that DiMSC performs better as the simulated directed network becomes denser, and DiMSC significantly outperforms its competitors in this experiment.

Experiment 4 (b): Sparsity. All parameters are set the same as Experiment 4 (a) except that  $P$  is set as  $P_2$ . Panel (b) of Figure 7 shows the results, and the analysis is similar to that of Experiment 2 (b).



**Figure 7.** Errors against increasing  $\rho$ . y-axis: MHammm: Panel (a): Experiment 4 (a); Panel (b): Experiment 4 (b); Panel (c): Experiment 4 (c); Panel (d): Experiment 4 (d); Panel (e): Experiment 4 (e); Panel (f): Experiment 4 (f); Panel(g): Experiment 4 (g); Panel (h): Experiment 4 (h).

Experiment 4 (c): Sparsity. Set  $n_r = 600, n_c = 900, n_0 = 150, z = 5$ , and  $P$  as  $P_3$ . Let  $\rho$  range in  $\{0.2, 0.3, \dots, 1\}$ . Panel (c) of Figure 7 shows the results, and the analysis is similar to that of Experiment 4 (a).

Experiment 4 (d): Sparsity. All parameters are set the same as Experiment 4 (c) except that  $P$  is set as  $P_4$ . Panel (d) of Figure 7 displays the results, and the analysis is similar to that of Experiment 4 (b).

Experiment 4 (e): Sparsity. All parameters are set the same as Experiment 4 (a) except that we let  $\Pi_r$  and  $\Pi_c$  be the same as that of Experiment 2 (e). Panel (e) of Figure 7 shows the results, and we see that DiMSC’s error rates decrease for a denser directed network even when all nodes are mixed. Meanwhile, all methods enjoy similar performances in this experiment.

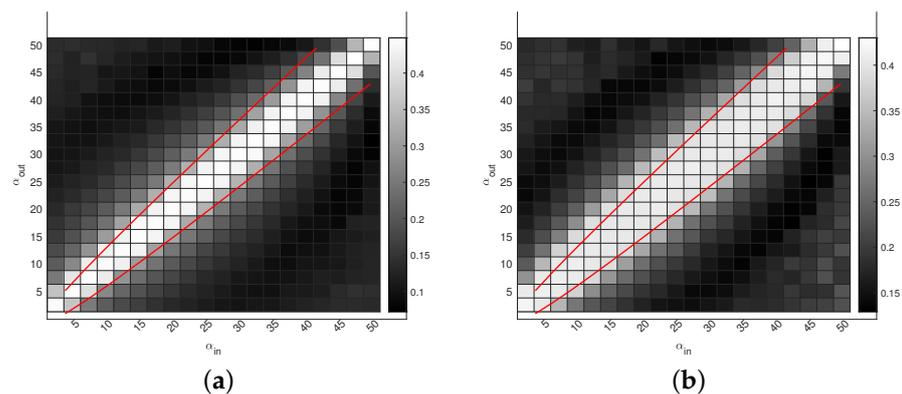
Experiment 4 (f): Sparsity. All parameters are set the same as Experiment 4 (b) except that we set  $\Pi_r$  and  $\Pi_c$  the same as Experiment 2(e). Panel (f) of Figure 7 shows the results, and the analysis is similar to that of Experiment 4 (e).

Experiment 4 (g): Sparsity. All parameters are set the same as Experiment 4 (c) except that we let  $\Pi_r$  and  $\Pi_c$  be the same as that of Experiment 2 (g). Panel (g) of Figure 7 shows the results, and the analysis is similar to that of Experiment 4 (e).

Experiment 4 (h): Sparsity. All parameters are set the same as Experiment 4 (d) except that we set  $\Pi_r$  and  $\Pi_c$  the same as Experiment 2(g). Panel (h) of Figure 7 shows the results, and the analysis is similar to that of Experiment 4 (e).

Experiment 5 (a): Phase transition. Under  $DiDCMM(n, K, \Pi_r, \Pi_c, \alpha_{in}, \alpha_{out})$ , set  $K = 2, n = n_r = n_c = 300$ . Let each row community have 100 pure nodes, each column community have 120 pure nodes, and all mixed nodes have membership  $(1/2, 1/2)$ . Since  $\max(p_{in}, p_{out}) = \max(\alpha_{in}, \alpha_{out}) \frac{\log(n)}{n} \leq 1$ ,  $\alpha_{in}$  and  $\alpha_{out}$  should be set in  $(0, \frac{n}{\log(n)}]$ . We let  $\alpha_{in}$  and  $\alpha_{out}$  be in the range of  $\{2.5, 5, 7.5, \dots, 50\}$ . Panel (a) of Figure 8 displays the results. We see that DiMSC performs satisfactorily when  $\alpha_{in}$  and  $\alpha_{out}$  satisfy Equation (14), and this means that DiMSC achieves the threshold provided in Equation (14) under  $DiDCMM(n, K, \Pi_r, \Pi_c, \alpha_{in}, \alpha_{out})$ .

Experiment 5 (b): Phase transition. Under  $DiDCMM(n, K, \Pi_r, \Pi_c, \alpha_{in}, \alpha_{out})$ , set  $K = 3, n = n_r = n_c = 300$ . Let each row community have 60 pure nodes, each column community have 80 pure nodes, and all mixed nodes have membership  $(1/3, 1/3, 1/3)$ . We also let  $\alpha_{in}$  and  $\alpha_{out}$  be in the range of  $\{2.5, 5, 7.5, \dots, 50\}$ . Panel (b) of Figure 8 displays the results, and the analysis is similar to that of Experiment 5 (a).



**Figure 8.** Phase transition for DiMSC: darker pixels represent lower error rates. The red lines represent  $\frac{|\alpha_{in} - \alpha_{out}|}{\sqrt{\max(\alpha_{in}, \alpha_{out})}} = 1$ . Panel (a): Experiment 5 (a); Panel (b): Experiment 5 (b).

For Experiments 1–5, we can conclude that DiMSC outperforms its competitors, and this supports our analysis in Remark 6 because DiMSC is designed to estimate mixed memberships, while its competitors are designed for community partition of pure nodes.

Experiment 6: Network size. Under  $SBM(n, p_{in}, p_{out})$ , let  $\alpha_{in} = 2$  and  $\alpha_{out} = 0.0001$ . On the one hand, we have  $\sqrt{\alpha_{in}} - \sqrt{\alpha_{out}} = \sqrt{2} - 0.01 < \sqrt{2}$ , i.e.,  $\alpha_{in}$  and  $\alpha_{out}$  locates in the impossible region of exact recovery introduced in [50]. On the other hand, we have  $\frac{\alpha_{in} - \alpha_{out}}{\sqrt{\alpha_{in}}} > 1$ , i.e.,  $\alpha_{in}$  and  $\alpha_{out}$  satisfy Equation (14) for DiMSC’s consistent estimation. Let  $n$  range in  $\{1000, 2000, 3000, \dots, 15000\}$ . For each  $n$  in this experiment, we report the averaged error rate and running time of DiMSC over 10 independent repetitions. The results are shown in Figure 9. From Panel (a) of Figure 9, we see that DiMSC enjoys satisfactory performance with a small error rate for this experiment. Panels (b) of Figure 9 says that DiMSC processes computer-generated networks of up to 15,000 nodes within hundreds of seconds.

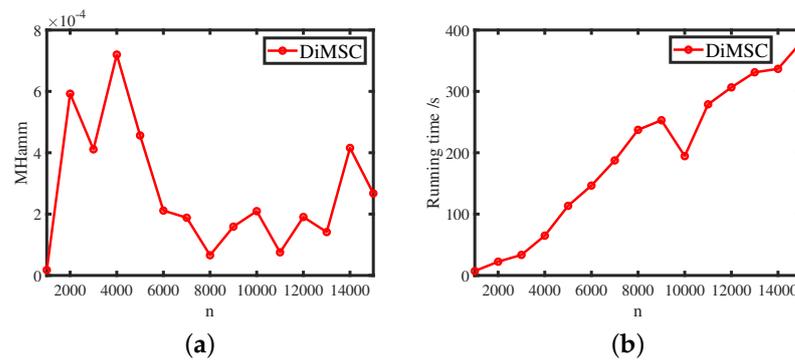


Figure 9. Numerical results for Experiment 6. Panel (a): MHammm; Panel (b): running time.

**Remark 7.** For visibility, we provide some examples of different types of directed networks generated under DiDCMM in this remark. Let  $\theta_r(i) = 0.9 + \frac{i^2}{9n_r^2}$  for  $1 \leq i \leq n_r$ . Let each row community has  $n_{r,0}$  pure nodes, and each column community has  $n_{c,0}$  pure nodes. Let all mixed nodes have membership  $(1/K, \dots, 1/K)$ . For the setting of  $P$ , we set it as

$$P_a = \begin{bmatrix} 0.9 & 0.05 \\ 0.1 & 0.95 \end{bmatrix} \text{ or } P_b = \begin{bmatrix} 0.1 & 0.95 \\ 0.9 & 0.05 \end{bmatrix} \text{ or } P_c = \begin{bmatrix} 12 & 1 \\ 0 & 12 \end{bmatrix} \frac{\log(n_r)}{n_r} \text{ or } P_d = \begin{bmatrix} 0 & 12 \\ 12 & 1 \end{bmatrix} \frac{\log(n_r)}{n_r} \text{ or } P_e = \begin{bmatrix} 12 & 1 & 0 \\ 0 & 12 & 0 \\ 1 & 0 & 12 \end{bmatrix} \frac{\log(n_r)}{n_r} \text{ or } P_f = \begin{bmatrix} 1 & 0 & 12 \\ 12 & 0 & 0 \\ 0 & 12 & 1 \end{bmatrix} \frac{\log(n_r)}{n_r},$$

where  $K = 2$  when  $P$  is  $P_a, P_b, P_c$  or  $P_d$ , and  $K = 3$  when  $P$  is  $P_e$  or  $P_f$ . Meanwhile, we can generate different types of directed networks under DiDCMM by considering the above six different settings of  $P$ , where these different types are also considered in Experiments 1–6, and we mainly provide the visibility for these directed networks with different structures provided in different  $P$  for this remark. Note that we allow  $P$  to have non-unit diagonals here because Condition (I1) is mainly for our theoretical buildings, and results for previous experiments show that DiMSC performs stable even when  $P$  has non-unit diagonals. We consider below eight settings.

**Model Setup 1:** Set  $n_r = 16, n_{r,0} = 6, n_c = 16, n_{c,0} = 7$ , and  $P$  as  $P_a$ . For this setup, a directed network with 16 row nodes and 16 column nodes is generated from DiDCMM. Figure 10 shows a directed network  $\mathcal{N}$  generated under Model Setup 1, where we also report DiMSC’s error rate. Figure 10 says that there are more directed edges sent from row nodes 1–6 to column nodes 1–7 than from row nodes 7–12 to column nodes 1–7 for  $P_a$ . With given adjacency matrix  $A$  and known memberships  $\Pi_r$  and  $\Pi_c$  for this setup, readers can apply our DiMSC directly to  $A$  given in Panel (a) of Figure 10 to check the effectiveness of DiMSC.

**Model Setup 2:** All settings are the same as Model Setup 1 except that we let  $P$  be  $P_b$ . The directed network  $\mathcal{N}$  and its adjacency matrix are shown in Figure 11. We see that there are more directed edges sent from row nodes 1–6 to column nodes 10–16 than from row nodes 7–12 to column nodes 10–16 for  $P_b$ , which means that directed network generated using  $P_b$  and directed network from  $P_a$  has different structures.

**Model Setup 3:** Set  $n_r = 32, n_{r,0} = 14, n_c = 28, n_{c,0} = 12$ , and  $P$  as  $P_a$ . For this setup, a bipartite network with 32 row nodes and 28 column nodes are generated from DiDCMM. Figure 12 shows this bipartite network and its adjacency matrix.

**Model Setup 4:** All settings are the same as Model Setup 3 except that we let  $P$  be  $P_b$ . Figure 13 displays the results, and we see that the bipartite network from  $P_b$  also has a different structure compared with the one generated from using  $P_a$  under DiDCMM.

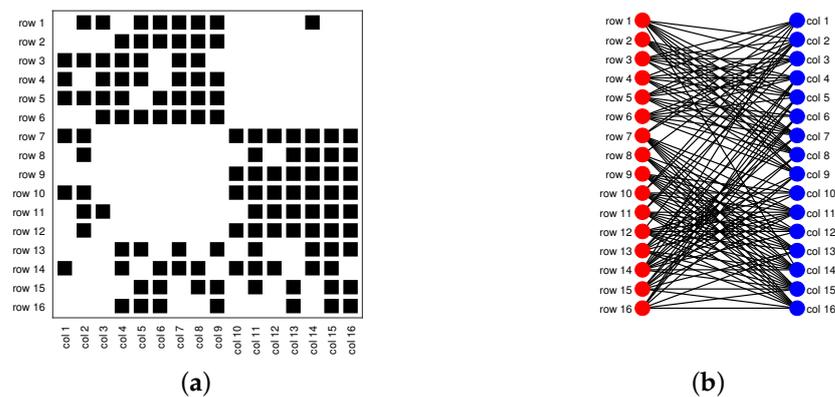
**Model Setup 5:** Set  $n_r = 100, n_{r,0} = 48, n_c = 100, n_{c,0} = 45$ , and  $P$  as  $P_c$ . Figure 14 shows the row and column communities for a directed network generated from Setup 5 under DiDCMM, where we plot the directed network directly.

**Model Setup 6:** All settings are the same as Model Setup 5 except that we let  $P$  be  $P_d$ . Figure 15 shows a directed network obtained from this setup, and we see that the structure of the

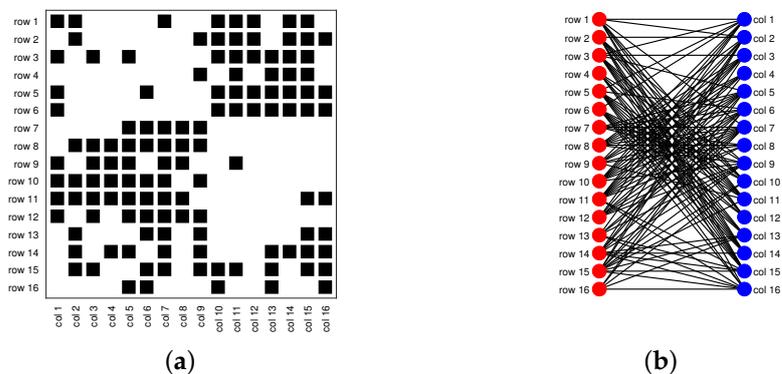
directed network from  $P_d$  in Figure 15 differs a lot from that of the directed network from  $P_c$  shown in Figure 14.

Model Setup 7: Set  $n_r = 100, n_{r,0} = 30, n_c = 100, n_{c,0} = 32$ , and  $P$  as  $P_e$ . Figure 16 shows a directed network generated from this setup.

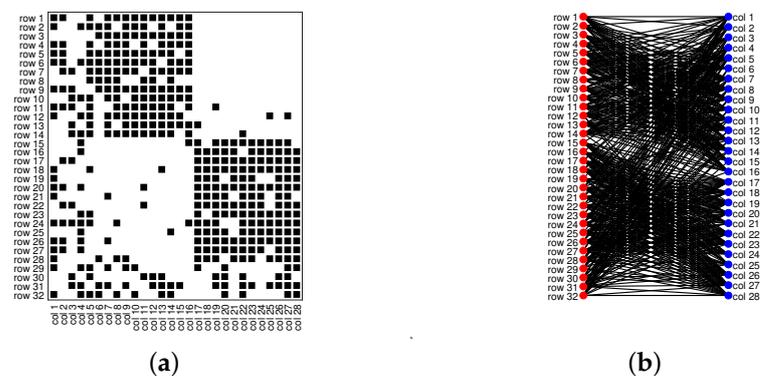
Model Setup 8: All settings are the same as Model Setup 7 except that we let  $P$  be  $P_f$ . Figure 17 displays a directed network generated from this setup, and we see that directed networks from  $P_f$  and  $P_e$  have different structures by comparing Figures 16 and 17.



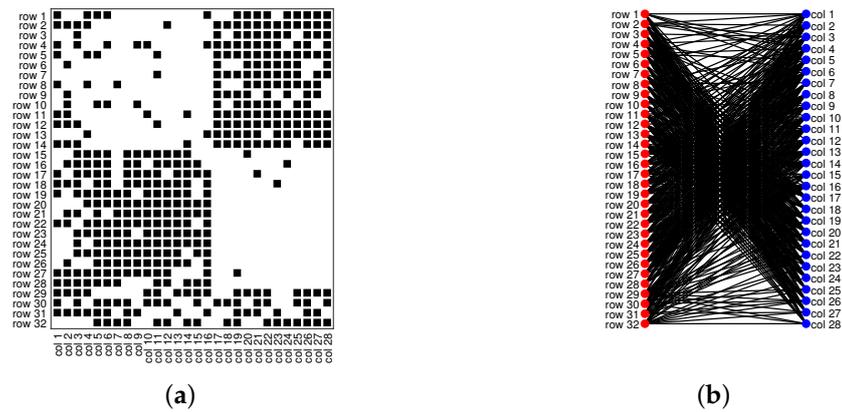
**Figure 10.** Illustration for a directed network under Model Setup 1. Panel (a): adjacency matrix of  $\mathcal{N}$ , where black square denotes 1; Panel (b): directed network  $\mathcal{N}$ , where red (blue) points indicate row (column) nodes. The error rate MHam defined in Equation (16) of our DiSMC algorithm for this directed network  $\mathcal{N}$  is 0.0377.



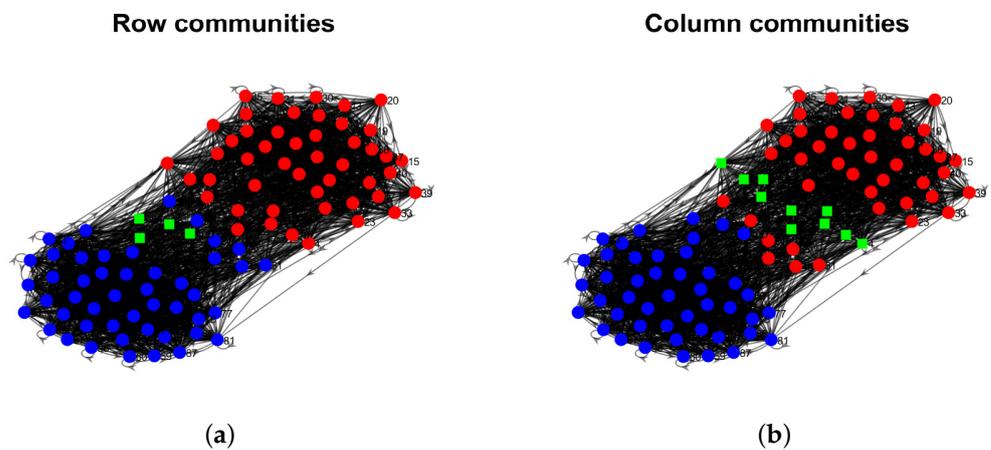
**Figure 11.** Illustration for a directed network under Model Setup 2. Panel (a): adjacency matrix  $A$ ; Panel (b): directed network  $\mathcal{N}$ . MHam of DiMSC for this directed network  $\mathcal{N}$  is 0.0424.



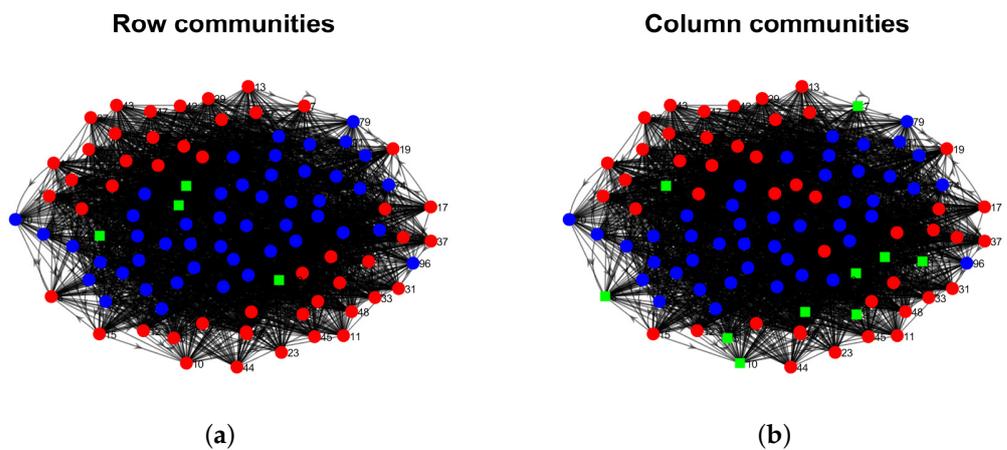
**Figure 12.** Illustration for a bipartite network under Model Setup 3. Panel (a): adjacency matrix  $A$ ; Panel (b): bipartite network  $\mathcal{N}$ . MHam of DiMSC for this bipartite network  $\mathcal{N}$  is 0.0313.



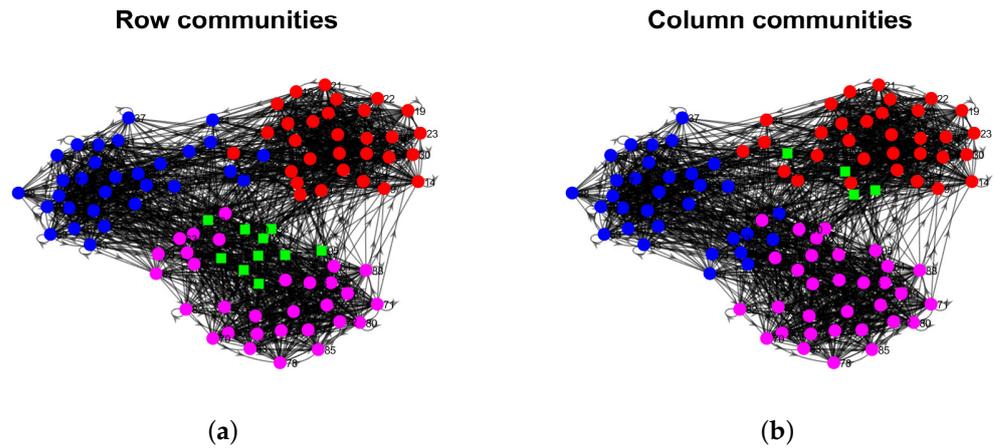
**Figure 13.** Illustration for a bipartite network under Model Setup 4. Panel (a): adjacency matrix  $A$ ; Panel (b): bipartite network  $\mathcal{N}$ . MHamm of DiMSC for this bipartite network  $\mathcal{N}$  is 0.0320.



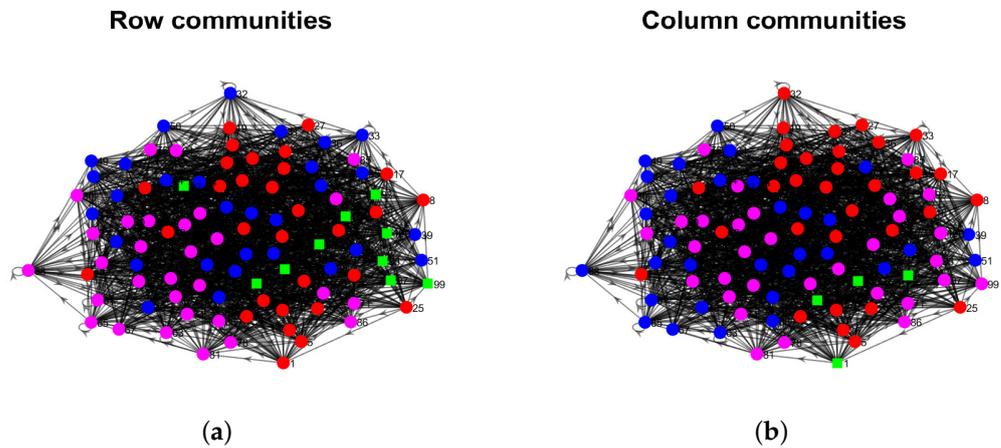
**Figure 14.** Illustration for a directed network under Model Setup 5. Panels (a,b) show the row and column communities, respectively. In these two panels, dots in the same color are pure nodes in the same communities, and a square indicates mixed nodes. MHamm of DiMSC for this directed network  $\mathcal{N}$  is 0.0181.



**Figure 15.** Illustration for a directed network under Model Setup 6. Panels (a,b) show the row and column communities, respectively. MHamm of DiMSC for this directed network  $\mathcal{N}$  is 0.0185.



**Figure 16.** Illustration for a directed network under Model Setup 7. Panels (a,b) show the row and column communities, respectively. MHamm of DiMSC for this directed network  $\mathcal{N}$  is 0.0266.



**Figure 17.** Illustration for a directed network under Model Setup 8. Panels (a,b) show the row and column communities, respectively. MHamm of DiMSC for this directed network  $\mathcal{N}$  is 0.0279.

**6. Application to Real-World Directed Networks**

For the empirical directed networks considered here, row nodes are always the same as column nodes, so we let  $n_r = n_c = n$ . For  $\hat{\Pi}_r$ , we call node  $i$  highly mixed node if  $0.8 \geq \max_{1 \leq k \leq K} \hat{\Pi}_r(i, k)$ , similar for  $\hat{\Pi}_c$ . A highly mixed node tells us whether a node has mixed memberships and belongs to multiple communities. Let  $\tau_r = \frac{|i:0.8 \geq \max_{1 \leq k \leq K} \hat{\Pi}_r(i, k)|}{n}$  be the proportion of highly mixed nodes among all nodes to measure the mixability of all row communities. Define  $\tau_c$  similar to  $\tau_r$ . Let  $\hat{\ell}_r$  be a vector such that  $\hat{\ell}_r(i) = \operatorname{argmax}_{1 \leq k \leq K} \hat{\Pi}_r(i, k)$  for  $1 \leq i \leq n$ , where we use  $\hat{\ell}_r(i)$  to denote the home base row community of node  $i$ . Define  $\hat{\ell}_c$  similar to  $\hat{\ell}_r$ . To measure the asymmetric structure of a directed network, we use

$$\text{Hamm}_{rc} = \frac{\min_{\mathcal{P} \in \mathcal{S}_{\mathcal{P}}} \|\hat{\Pi}_c \mathcal{P} - \hat{\Pi}_r\|_1}{n},$$

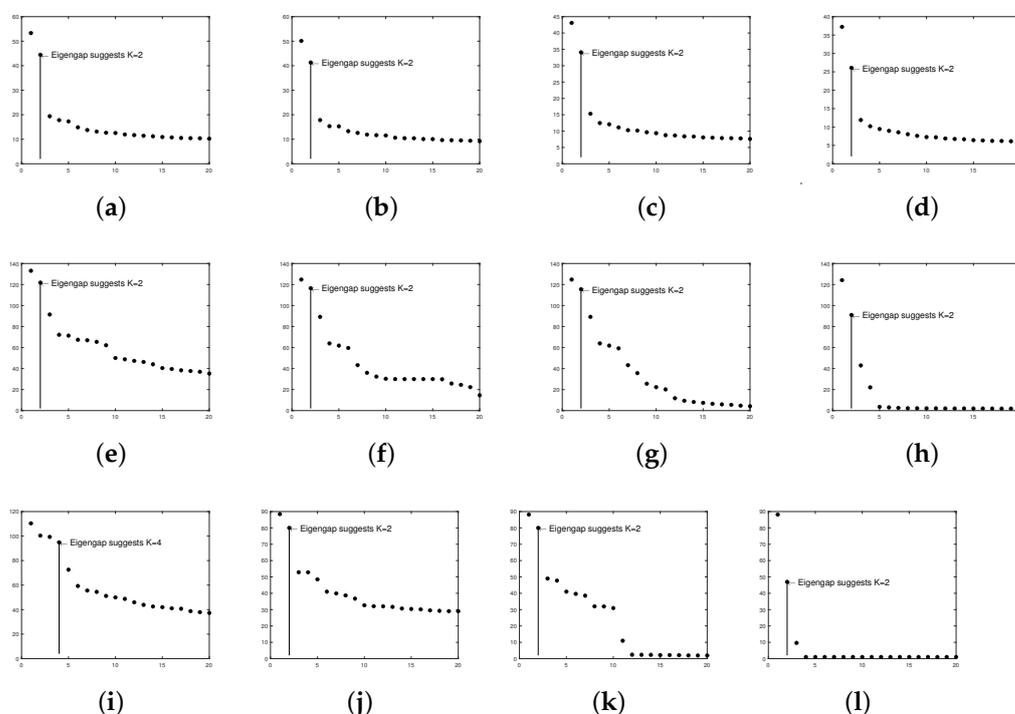
where a large  $\text{Hamm}_{rc}$  means that the structure of row clusters differs a lot from that of column clusters. For  $1 \leq i \leq n$ , let  $d_r(i) = \sum_{j=1}^n A(i, j)$  be the number of edges sent by node  $i$ ,  $d_c(i) = \sum_{j=1}^n A(j, i)$  be the number of edges received by node  $i$ , where  $d_r(i)$  (and  $d_c(i)$ ) is the out degree (in degree) of node  $i$ . Since there are many nodes with zero in degree or out degree for real-world directed network, we need the below pre-processing: for any directed network  $\mathcal{N}$ , we let  $A_m$  be its adjacency matrix for any positive integer  $m$  such that  $A_m$  is connected, and every node has at least  $m$  in degree and  $m$  out degree in  $A_m$ .

We apply DiMSC to the following real-world directed networks to discover their mixability, asymmetries, and directional communities.

**Political blogs:** This is a directed network of hyperlinks between weblogs on US politics [68]. In this data, node means a blog, and edge means a hyperlink. This data can be downloaded from <http://www-personal.umich.edu/~mejn/netdata/> (accessed on 28 August 2022). It is well-known that there are two parties, “liberal” and “conservative”, so  $K = 2$  for this data. There are 1490 nodes in the original data. After pre-processing,  $A_1 \in \{0,1\}^{813 \times 813}$ ,  $A_3 \in \{0,1\}^{495 \times 495}$ ,  $A_6 \in \{0,1\}^{285 \times 285}$ ,  $A_9 \in \{0,1\}^{158 \times 158}$ , where we focus on the cases when  $m = 1, 3, 6, 9$  for this data here. Meanwhile, we use political blogs  $A_m$  to denote this network when its adjacency matrix is  $A_m$ , where every node has a degree at least  $m$ . Similar notations hold for other real-world directed networks used in this paper.

**Wikipedia links (gan):** This directed network consists of the Wikilinks of Wikipedia in the Gan Chinese language (gan). In this data, node means an article, and the directed edge is a Wikilink [69]. This data can be downloaded from [http://konect.cc/networks/wikipedia\\_link\\_gan](http://konect.cc/networks/wikipedia_link_gan) (accessed on 28 August 2022). There are 9189 nodes in the original data. After pre-processing,  $A_1 \in \{0,1\}^{6012 \times 6012}$ ,  $A_{30} \in \{0,1\}^{820 \times 820}$ ,  $A_{60} \in \{0,1\}^{559 \times 559}$ ,  $A_{90} \in \{0,1\}^{240 \times 240}$ , where we study the cases  $m = 1, 30, 60, 90$  for this data. The leading 20 singular values of  $A_1, A_{30}, A_{60}, A_{90}$  shown in Panels (e)–(h) of Figure 18 suggest  $K = 2$  for these four adjacency matrices, where [30] also uses eigengap to estimate  $K$ .

**Wikipedia links (nah):** This network consists of the Wikilinks of the Nāhuatl language (nah) [69] and can be downloaded from [http://konect.cc/networks/wikipedia\\_link\\_nah/](http://konect.cc/networks/wikipedia_link_nah/) (accessed on 28 August 2022). The original data has 10285 nodes. After pre-processing,  $A_1 \in \{0,1\}^{6924 \times 6924}$ ,  $A_{20} \in \{0,1\}^{1057 \times 1057}$ ,  $A_{30} \in \{0,1\}^{486 \times 486}$ ,  $A_{40} \in \{0,1\}^{136 \times 136}$ . Panel (i) of Figure 18 suggests  $K = 4$  for  $A_1$ , and Panels (j)–(l) of Figure 18 suggest  $K = 2$  for  $A_{20}, A_{30}$ , and  $A_{40}$ . Note that it only takes around 4 seconds for DiMSC to estimate memberships of Wikipedia links (nah)  $A_1$ .

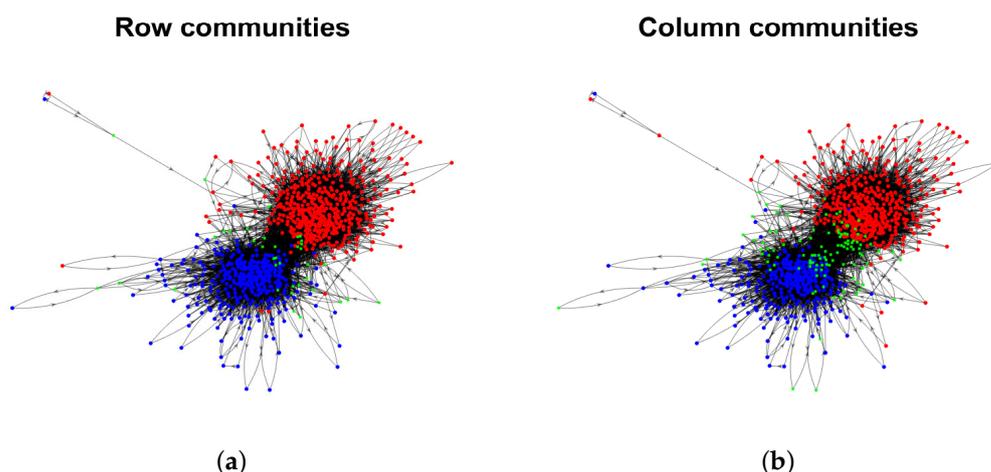


**Figure 18.** Leading 20 singular values of real-world directed networks used in this paper. Panel (a): political blogs  $A_1$ ; Panel (b): political blogs  $A_3$ ; Panel (c): political blogs  $A_6$ ; Panel (d): political blogs  $A_9$ ; Panel (e): Wikipedia links (gan)  $A_1$ ; Panel (f): Wikipedia links (gan)  $A_{30}$ ; Panel (g): Wikipedia links (gan)  $A_{60}$ ; Panel (h): Wikipedia links (gan)  $A_{90}$ ; Panel (i): Wikipedia links (nah)  $A_1$ ; Panel (j): Wikipedia links (nah)  $A_{20}$ ; Panel (k): Wikipedia links (nah)  $A_{30}$ ; Panel (l): Wikipedia links (nah)  $A_{40}$ .

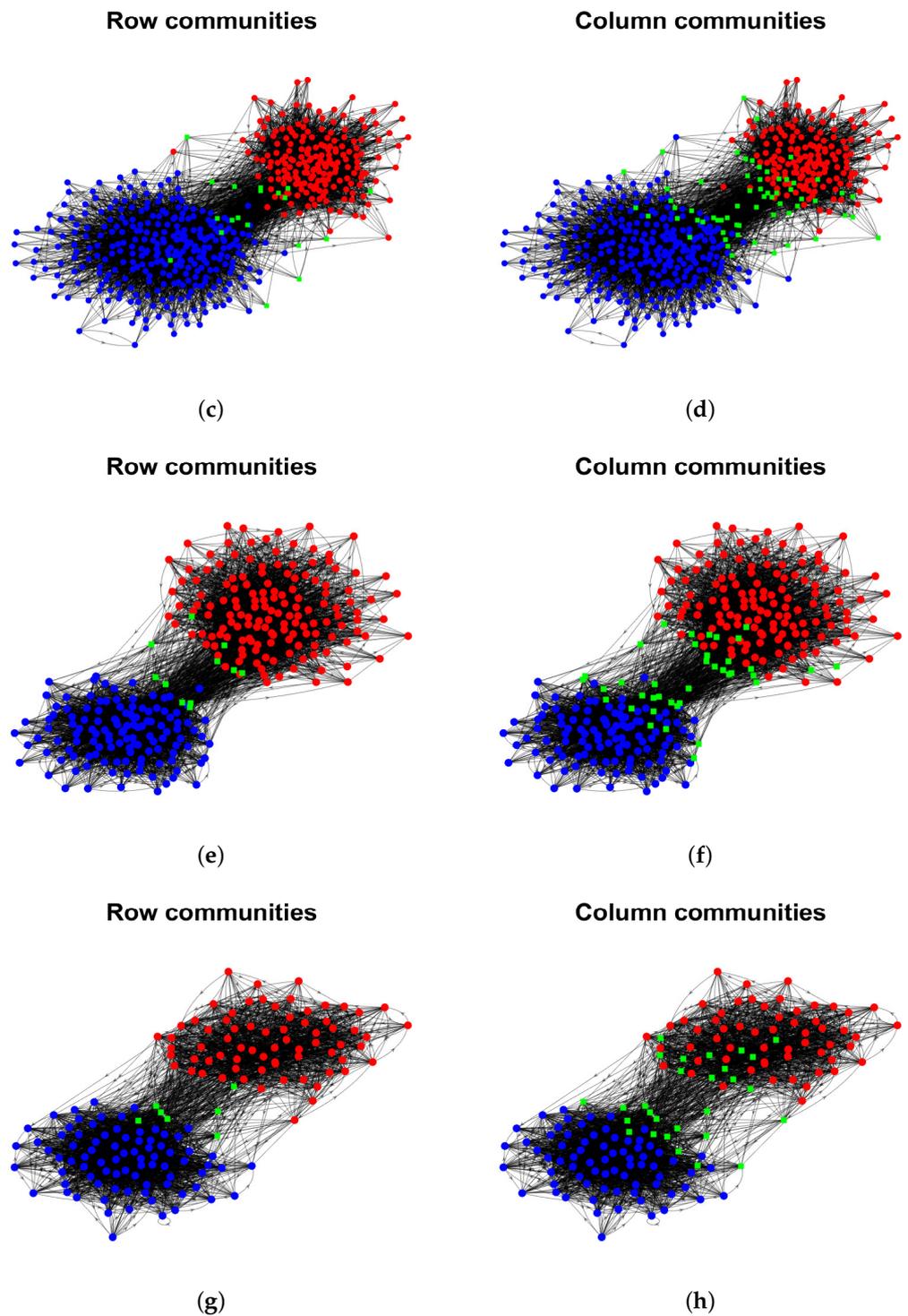
The proportions of highly mixed nodes and  $Hamm_{rc}$  when applying DiMSC on the above real-world directed networks are reported in Table 1. For the political blogs network, small  $\tau_r, \tau_c$ , and  $Hamm_{rc}$  indicate that there are only a few highly mixed nodes, and the structure of row communities is similar to that of column communities, i.e., there is a slight asymmetry for this data. For Wikipedia links (gan)  $A_1$  and Wikipedia links (nah)  $A_1$ , they have a large proposition of highly mixed nodes in both row and column communities, and the row communities differ a lot from column communities, suggesting heavy asymmetric structure between row and column communities for these two data. For Wikipedia links (gan)  $A_{30}, A_{60}$ , and Wikipedia links (nah)  $A_{20}$ , we see that the proportion of highly mixed nodes for row (column) communities is small (large), and there is a slight asymmetric for these data. For Wikipedia links (gan)  $A_{90}$  and Wikipedia links (nah)  $A_{30}, A_{40}$ , there is no highly mixed node, and the structure of row clusters is similar to that of column clusters. For visualization, we plot the row and column communities as well as highly mixed nodes by applying DiMSC to some of these directed networks in Figures 19 and 20.

**Table 1.**  $\tau_r, \tau_c$ , and  $Hamm_{rc}$  obtained from DiMSC for real-world directed networks used in this paper.

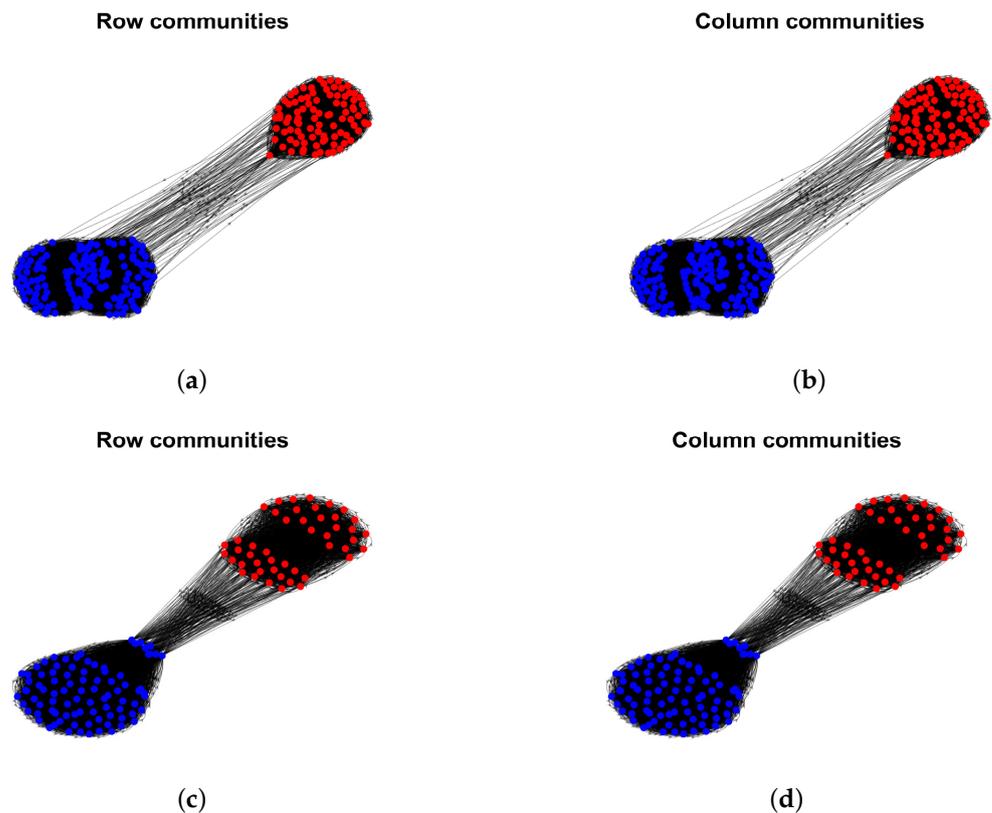
Data	$\tau_r$	$\tau_c$	$Hamm_{rc}$
Political blogs $A_1$	0.0455	0.1353	0.0893
Political blogs $A_3$	0.0481	0.1570	0.0705
Political blogs $A_6$	0.0386	0.1368	0.0662
Political blogs $A_9$	0.0443	0.1772	0.0771
Wikipedia links (gan) $A_1$	0.1505	0.6051	0.3528
Wikipedia links (gan) $A_{30}$	0.0817	0.1902	0.0547
Wikipedia links (gan) $A_{60}$	0.0054	0.1145	0.0664
Wikipedia links (gan) $A_{90}$	0	0	0.0203
Wikipedia links (nah) $A_1$	0.2718	0.3521	0.2065
Wikipedia links (nah) $A_{20}$	0.0937	0.1722	0.0488
Wikipedia links (nah) $A_{30}$	0	0	0.0046
Wikipedia links (nah) $A_{40}$	0	0	0



**Figure 19.** Cont.



**Figure 19.** Row and column communities detected by DiMSC for political blogs. Colors indicate clusters, and a green square indicates highly mixed nodes, where the row and column communities are obtained from  $\hat{\ell}_r$  and  $\hat{\ell}_c$ , respectively. Panel (a): political blogs  $A_1$ ; Panel (b): political blogs  $A_1$ ; Panel (c): political blogs  $A_3$ ; Panel (d): political blogs  $A_3$ ; Panel (e): political blogs  $A_6$ ; Panel (f): political blogs  $A_6$ ; Panel (g): political blogs  $A_9$ ; Panel (h): political blogs  $A_9$ .



**Figure 20.** Row and column communities detected by DiMSC for Wikipedia links (gan)  $A_{90}$  and Wikipedia links (nah)  $A_{40}$ . Colors indicate clusters, where the row and column communities are obtained from  $\hat{\ell}_r$  and  $\hat{\ell}_c$ , respectively. Panel (a): Wikipedia links (gan)  $A_{90}$ ; Panel (b): Wikipedia links (gan)  $A_{90}$ ; Panel (c): Wikipedia links (nah)  $A_{40}$ ; Panel (d): Wikipedia links (nah)  $A_{40}$ .

## 7. Discussion and Conclusions

In this paper, we propose a novel directed degree corrected mixed membership (DiDCMM) model. DiDCMM models a directed network with mixed memberships for row nodes with degree heterogeneities and column nodes without degree heterogeneities. DiDCMM is identifiable when the two well-used Conditions (I1) and (I2) hold. It should be mentioned that a model modeling a directed network with mixed memberships for both row and column nodes with degree heterogeneities is unidentifiable unless considering some nontrivial conditions. To fit the model, we propose a provably consistent spectral algorithm called DiMSC to infer community memberships for both row and column nodes in a directed network generated by DiDCMM. DiMSC is designed based on the SVD of the adjacency matrix, where we apply the SP algorithm to hunt for the corners in the simplex structure and the SVM-cone algorithm to hunt for the corners in the cone structure. The theoretical results of DiMSC show that it consistently recovers memberships of both row nodes and column nodes under mild conditions. Meanwhile, when DiDCMM degenerates to MMSB, our theoretical results match that of Theorem 2.2 [24] when their DCMM degenerates to MMSB under mild conditions. Experiments conducted on synthetic directed networks generated from DiDCMM verify the effectiveness and the stability of Conditions (I1) and (I2) of DiMSC. Results for real-world directed networks show that DiMSC reveals highly mixed nodes and asymmetries in the structure of row and column communities. The model DiDCMM and the algorithm DiMSC developed in this paper are useful to discover asymmetry for a directed network with mixed memberships. DiDCMM can also generate an artificially directed network with mixed memberships as a benchmark directed network for research purposes. We wish that DiDCMM and DiMSC can be widely applied in social network analysis.

The proposed model DiDCMM and the algorithm DiMSC can be extended in many ways. Similar to [24,57], we may obtain an ideal simplex from  $U$  using the idea of the entry-wise ratio proposed in [8]. Meanwhile, DiMSC is designed based on the SVD of the adjacency matrix, and similar to [5,7,11,30], we may design spectral algorithms based on the regularized Laplacian matrix under DiDCMM. Extending DiDCMM from an unweighted directed network to a weighted directed network with an application of the distribution-free idea introduced in [62] is one of our future research directions. The SVD step of DiMSC can be accelerated by the random projection and random sampling ideas introduced in [70] to process large-scale directed networks. Instead of simply using eigengap to find  $K$ , in our future work, it is worth focusing on estimating the number of communities in a directed network generated under ScBM (and DCScBM) [30] and DiDCMM. Ref. [46] proposes an algorithm to uncover boundary nodes that spread information between communities in undirected social networks. It is an interesting topic to extend works in [46] to directed networks generated from ScBM, DCScBM, and DiDCMM. We leave them for our future work.

**Funding:** This research was funded by the Scientific research start-up fund of CUMT NO. 102520253, the High-level personal project of Jiangsu Province NO. JSSCBS20211218.

**Institutional Review Board Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The author declares no conflict of interest.

### Abbreviations

The following abbreviations are used in this manuscript:

SBM	Stochastic Blockmodel
DCSBM	Degree Corrected Stochastic Blockmodel
MMSB	Mixed Membership Stochastic Blockmodel
DCMM	Degree Corrected Mixed Membership model
OCCAM	Overlapping Continuous Community Assignment model
ScBM	Stochastic co-Blockmodel
DC-ScBM	Degree Corrected Stochastic co-Blockmodel
DiMMSB	Directed Mixed Membership Stochastic Blockmodel
DiDCMM	Directed Degree Corrected Mixed Membership model
SP	Successive projection algorithm
SVD	Singular value decomposition
DiMSC	Directed Mixed Simplex & Cone algorithm

### Appendix A. Proof for Identifiability

#### Appendix A.1. Proof of Proposition 1

**Proof.** Let  $\Omega = U\Lambda V'$  be the compact singular value decomposition of  $\Omega$ . Lemma 1 gives  $V = \Pi_c B_c \equiv \Pi_c V(\mathcal{I}_c, :)$ . Since  $\Omega = \tilde{\Omega}$ ,  $V$  also equals to  $\tilde{\Pi}_c V(\mathcal{I}_c, :)$ , which gives that  $\Pi_c = \tilde{\Pi}_c$ .

Since  $\Omega(\mathcal{I}_r, \mathcal{I}_c) = \Theta_r(\mathcal{I}_r, \mathcal{I}_r)\Pi_r(\mathcal{I}_r, :)\Pi\Pi'_c(\mathcal{I}_c, :) = \Theta_r(\mathcal{I}_r, \mathcal{I}_r)P = U(\mathcal{I}_r, :)\Lambda V'(\mathcal{I}_c, :)$  by Condition (I2), we have  $\Theta_r(\mathcal{I}_r, \mathcal{I}_r)P = U(\mathcal{I}_r, :)\Lambda V'(\mathcal{I}_c, :)$ , which gives that  $\Theta_r(\mathcal{I}_r, \mathcal{I}_r) = \text{diag}(U(\mathcal{I}_r, :)\Lambda V'(\mathcal{I}_c, :))$ . From this step, we see that if  $P$ 's diagonal entries are not ones, we cannot obtain  $\Theta_r(\mathcal{I}_r, \mathcal{I}_r) = \text{diag}(U(\mathcal{I}_r, :)\Lambda V'(\mathcal{I}_c, :))$  which leads to a consequence that  $\Theta_r(\mathcal{I}_r, \mathcal{I}_r)$  does not equal to  $\tilde{\Theta}_r(\mathcal{I}_r, \mathcal{I}_r)$ ; hence Condition (I1) is necessary by Condition (I1). Since  $\Omega = \tilde{\Omega}$ , we also have  $\tilde{\Theta}_r(\mathcal{I}_r, \mathcal{I}_r) = \text{diag}(U(\mathcal{I}_r, :)\Lambda V'(\mathcal{I}_c, :))$ , which gives that  $\Theta_r(\mathcal{I}_r, \mathcal{I}_r) = \tilde{\Theta}_r(\mathcal{I}_r, \mathcal{I}_r)$ . Since  $\tilde{\Theta}_r(\mathcal{I}_r, \mathcal{I}_r)\tilde{P}$  also equals to  $U(\mathcal{I}_r, :)\Lambda V'(\mathcal{I}_c, :)$ , we have  $P = \tilde{P}$ .

Lemma 1 gives that  $U = \Theta_r\Pi_r B_r$ , where  $B_r = \Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :)$ . Since  $\Omega = \tilde{\Omega}$ , we also have  $U = \tilde{\Theta}_r\tilde{\Pi}_r\tilde{B}_r$ . Since  $\tilde{B}_r = \tilde{\Theta}_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :) = \Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :)$ , we have  $\tilde{B}_r = B_r$ . Since  $U = \Theta_r\Pi_r B_r = \tilde{\Theta}_r\tilde{\Pi}_r\tilde{B}_r = \tilde{\Theta}_r\tilde{\Pi}_r B_r$ , we have  $\Theta_r\Pi_r = \tilde{\Theta}_r\tilde{\Pi}_r$ . Since each row of  $\Pi_r$  or  $\tilde{\Pi}_r$  is a PMF,  $\Theta_r = \tilde{\Theta}_r$ ,  $\Pi_r = \tilde{\Pi}_r$ , and the claim follows.  $\square$

### Appendix B. Ideal Simplex, Ideal Cone

#### Appendix B.1. Proof of Lemma 1

**Proof.** First, we consider  $U$  and  $V$ . Since  $\Omega = U\Lambda V'$ , we have  $U = \Omega V\Lambda^{-1}$  since  $V'V = I_K$ . Recall that  $\Omega = \Theta_r \Pi_r P \Pi_c'$ , we have  $U = \Theta_r \Pi_r P \Pi_c' V \Lambda^{-1} = \Theta_r \Pi_r B_r$ , where we set  $B_r = P \Pi_c' V \Lambda^{-1}$  and sure it is unique. Since  $U(\mathcal{I}_r, :) = \Theta_r(\mathcal{I}_r, \mathcal{I}_r) \Pi_r(\mathcal{I}_r, :) B_r = \Theta_r(\mathcal{I}_r, \mathcal{I}_r) B_r$ , we have  $B_r = \Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r) U(\mathcal{I}_r, :)$ .

Similarly, since  $\Omega = U\Lambda V'$ , we have  $V' = \Lambda^{-1} U' \Omega$  since  $U'U = I_K$ , hence  $V = \Omega' U \Lambda^{-1}$ . Recall that  $\Omega = \Theta_r \Pi_r P \Pi_c'$ , we have  $V = (\Theta_r \Pi_r P \Pi_c')' U \Lambda^{-1} = \Pi_c P' \Pi_r' \Theta_r U \Lambda^{-1} = \Pi_c B_c$ , where we set  $B_c = P' \Pi_r' \Theta_r U \Lambda^{-1}$  and sure it is unique. Since  $V(\mathcal{I}_c, :) = \Pi_c(\mathcal{I}_c, :) B_c = B_c$ , we have  $B_c = V(\mathcal{I}_c, :)$ . Meanwhile, for  $1 \leq j \leq n_c$ , we have  $V(j, :) = e_j' \Pi_c B_c = \Pi_c(j, :) B_c$ . Hence, we have  $V(j, :) = V(\bar{j}, :)$  as long as  $\Pi_c(j, :) = \Pi_c(\bar{j}, :)$ .

Now, we show the ideal cone structure that appears in  $U_*$ . For convenience, set  $M = \Pi_r B_r$ , hence  $U = \Theta_r \Pi_r B_r$  gives  $U = \Theta_r M$ . Hence, we have  $U(i, :) = e_i' U = \Theta_r(i, i) M(i, :)$ . Therefore,  $U_*(i, :) = \frac{U(i, :)}{\|U(i, :)\|_F} = \frac{M(i, :)}{\|M(i, :)\|_F}$ , combine it with the fact that  $B_r = \Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r) U(\mathcal{I}_r, :)$ , we have

$$\begin{aligned}
 U_* &= \begin{bmatrix} \frac{1}{\|M(1, :)\|_F} & & & \\ & \frac{1}{\|M(2, :)\|_F} & & \\ & & \ddots & \\ & & & \frac{1}{\|M(n_r, :)\|_F} \end{bmatrix} \Pi_r B_r \\
 &= \begin{bmatrix} \Pi_r(1, :)/\|M(1, :)\|_F \\ \Pi_r(2, :)/\|M(2, :)\|_F \\ \vdots \\ \Pi_r(n_r, :)/\|M(n_r, :)\|_F \end{bmatrix} \Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r) N_U^{-1}(\mathcal{I}_r, \mathcal{I}_r) U_*(\mathcal{I}_r, :).
 \end{aligned}$$

Therefore, we have

$$Y = \begin{bmatrix} \Pi_r(1, :)/\|M(1, :)\|_F \\ \Pi_r(2, :)/\|M(2, :)\|_F \\ \vdots \\ \Pi_r(n_r, :)/\|M(n_r, :)\|_F \end{bmatrix} \Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r) N_U^{-1}(\mathcal{I}_r, \mathcal{I}_r) = N_M \Pi_r \Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r) N_U^{-1}(\mathcal{I}_r, \mathcal{I}_r),$$

where  $N_M$  is a diagonal matrix with  $N_M(i, i) = \frac{1}{\|M(i, :)\|_F}$  for  $1 \leq i \leq n_r$ . All entries of  $Y$  are nonnegative, and since we assume that each community has at least one pure node, no row of  $Y$  is 0.

Then, we prove that  $U_*(i, :) = U_*(\bar{i}, :)$  when  $\Pi_r(i, :) = \Pi_r(\bar{i}, :)$ . For  $1 \leq i \leq n_r$ , we have

$$U_*(i, :) = e_i' U_* = \frac{1}{\|M(i, :)\|_F} e_i' M = \frac{1}{\|\Pi_r(i, :)\|_F} \Pi_r(i, :) B_r,$$

and the claim follows immediately.  $\square$

#### Appendix B.2. Proof of Lemma 2

**Proof.** Since  $I = U'U = B_r' \Pi_r' \Theta_r^2 \Pi_r B_r = U'(\mathcal{I}_r, :)\Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)\Pi_r'\Theta_r^2\Pi_r\Theta^{-1}(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :)$  and  $\text{rank}(U(\mathcal{I}_r, :)) = K$  (i.e., the inverse of  $U(\mathcal{I}_r, :)$  exists), we have  $(U(\mathcal{I}_r, :)U'(\mathcal{I}_r, :))^{-1} = \Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)\Pi_r'\Theta_r^2\Pi_r\Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)$ . Since  $U_*(\mathcal{I}_r, :) = N_U(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :)$ , we have

$$(U_*(\mathcal{I}_r, :)U_*(\mathcal{I}_r, :))^{-1} = N_U^{-1}(\mathcal{I}_r, \mathcal{I}_r)\Theta^{-1}(\mathcal{I}_r, \mathcal{I}_r)\Pi_r'\Theta_r^2\Pi_r\Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)N_U^{-1}(\mathcal{I}_r, \mathcal{I}_r).$$

Since all entries of  $N_U^{-1}(\mathcal{I}_r, \mathcal{I}_r), \Pi_r, \Theta_r$  and nonnegative and  $N, \Theta_r$  are diagonal matrices, we see that all entries of  $(U_*(\mathcal{I}_r, :)U'_*(\mathcal{I}_r, :))^{-1}$  are nonnegative, and its diagonal entries are strictly positive, hence we have  $(U_*(\mathcal{I}_r, :)U'_*(\mathcal{I}_r, :))^{-1}\mathbf{1} > 0$ .  $\square$

Appendix B.3. Proof of Theorem 1

**Proof.** For column nodes, Remark A1 guarantees that SP algorithm returns  $\mathcal{I}_c$  when the input is  $V$  with  $K$  column communities, hence ideal DiMSC recovers  $\Pi_c$  exactly. For row nodes, Remark A2 guarantees that SVM-cone algorithm returns  $\mathcal{I}_r$  when the input is  $U_*$  with  $K$  row communities, hence ideal DiMSC recovers  $\Pi_r$  exactly, and this theorem follows.  $\square$

Appendix C. Equivalence Algorithm

In this subsection, we design one algorithm DiMSC-equivalence which returns the same estimations as DiMSC. Set  $U_2 = UU' \in \mathbb{R}^{n_r \times n_r}, \hat{U}_2 = \hat{U}\hat{U}' \in \mathbb{R}^{n_r \times n_r}, V_2 = VV' \in \mathbb{R}^{n_c \times n_c}, \hat{V}_2 = \hat{V}\hat{V}' \in \mathbb{R}^{n_c \times n_c}$ . Set  $U_{*,2} \in \mathbb{R}^{n_r \times n_r}$  as  $U_{*,2}(i, :) = \frac{U_2(i, :)}{\|U_2(i, :)\|_F}$  for  $1 \leq i \leq n_r$ .  $\hat{U}_{*,2}$  is defined similarly. The next lemma guarantees that  $V_2$  enjoys IS structure, and  $U_{*,2}$  enjoys IC structure.

**Lemma A1.** Under  $DiDCMM_{n_r, n_c}(K, P, \Pi_r, \Pi_c, \Theta_r)$ , we have  $V_2 = \Pi_c V_2(\mathcal{I}_c, :)$ , and  $U_{*,2} = YU_{*,2}(\mathcal{I}_r, :)$ .

**Proof.** By Lemma 1, we know that  $V = \Pi_c V(\mathcal{I}_c, :)$ , which gives that  $V_2 = VV' = \Pi_c V(\mathcal{I}_c, :)' = \Pi_c(VV')(\mathcal{I}_c, :) = \Pi_c V_2(\mathcal{I}_c, :)$ . For  $U$ , since  $U = \Theta_r \Pi_r \Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :)$  by Lemma 1, we have  $U_2 = UU' = \Theta_r \Pi_r \Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :)'U' = \Theta_r \Pi_r \Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)(UU')(\mathcal{I}_r, :) = \Theta_r \Pi_r \Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U_2(\mathcal{I}_r, :)$ . Set  $M_2 = \Pi_r \Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U_2(\mathcal{I}_r, :)$ , we have  $U_2 = \Theta_r M_2$ . Then, follow similar proof as Lemma 1, we have  $U_{*,2} = Y_2 U_{*,2}(\mathcal{I}_r, :)$ , where  $Y_2 = N_{M_2} \Pi_r \Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)N_{U_2}^{-1}(\mathcal{I}_r, \mathcal{I}_r)$ , and  $N_{M_2}, N_{U_2}$  are  $n_r \times n_r$  diagonal matrices whose  $i$ -th diagonal entries are  $\frac{1}{\|M_2(i, :)\|_F}, \frac{1}{\|U_2(i, :)\|_F}$ , respectively. Since  $\|U_2(i, :)\|_F = \|U(i, :)'U'\|_F = \|U(i, :)\|_F$ , we have  $N_{U_2} = N_U$ . Since  $\|M_2(i, :)\|_F = \|\Pi_r \Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U_2(\mathcal{I}_r, :)\|_F = \|\Pi_r \Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :)'U'\|_F = \|M(i, :)\|_F$ , we have  $N_{M_2} = N_M$ . Hence,  $Y_2 \equiv Y$  and the claim follows.  $\square$

Since  $U_{*,2}(\mathcal{I}_r, :) \in \mathbb{R}^{K \times n_r}$  and  $V_2(\mathcal{I}_c, :) \in \mathbb{R}^{K \times n_c}$ ,  $U_{*,2}(\mathcal{I}_r, :)$  and  $V_2(\mathcal{I}_c, :)$  are singular matrix with rank  $K$  by Condition (I1), while the inverses of  $U_{*,2}(\mathcal{I}_r, :)'U'_{*,2}(\mathcal{I}_r, :)$  and  $V_2(\mathcal{I}_c, :)'V'_2(\mathcal{I}_c, :)$  exist. Therefore, Lemma A1 gives that

$$Y = U_{*,2}U'_{*,2}(\mathcal{I}_r, :)(U_{*,2}(\mathcal{I}_r, :)'U'_{*,2}(\mathcal{I}_r, :))^{-1}, \Pi_c = V_2V'_2(\mathcal{I}_c, :)(V_2(\mathcal{I}_c, :)'V'_2(\mathcal{I}_c, :))^{-1}.$$

Since  $U_{*,2} = N_U U_2$  and  $Y = N_M \Pi_r \Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)N_U^{-1}(\mathcal{I}_r, \mathcal{I}_r)$ , we see that  $Y_*$  also equals to  $U_2U'_{*,2}(\mathcal{I}_r, :)(U_{*,2}(\mathcal{I}_r, :)'U'_{*,2}(\mathcal{I}_r, :))^{-1}$  by basic algebra.

Based on the above analysis, we are now ready to give the ideal DiMSC-equivalence. Input  $\Omega$ . Output:  $\Pi_r$  and  $\Pi_c$ .

- Obtain  $U, \Lambda, V, U_{*,2}, V_2$  from  $\Omega$ .
- Run SP algorithm on  $V_2$  with  $K$  column communities to obtain  $V_2(\mathcal{I}_c, :)$ . Run SVM-cone algorithm on  $U_{*,2}$  with  $K$  row communities to obtain  $\mathcal{I}_r$ .
- Set  $J_* = \text{diag}(U_*(\mathcal{I}_r, :)\Lambda V'(\mathcal{I}_c, :)), Y_* = U_2U'_{*,2}(\mathcal{I}_r, :)(U_{*,2}(\mathcal{I}_r, :)'U'_{*,2}(\mathcal{I}_r, :))^{-1}, Z_r = Y_*J_*$  and  $Z_c = V_2V'_2(\mathcal{I}_c, :)(V_2(\mathcal{I}_c, :)'V'_2(\mathcal{I}_c, :))^{-1}$ .
- Recover  $\Pi_r$  and  $\Pi_c$  by setting  $\Pi_r(i, :) = \frac{Z_r(i, :)}{\|Z_r(i, :)\|_1}$  for  $1 \leq i \leq n_r$ , and  $\Pi_c(j, :) = \frac{Z_c(j, :)}{\|Z_c(j, :)\|_1}$  for  $1 \leq j \leq n_c$ .

For the real case, set  $\hat{U}_2 = \hat{U}\hat{U}', \hat{V}_2 = \hat{V}\hat{V}', \hat{U}_{*,2} = N_{\hat{U}}\hat{U}_2$ . We now extend the ideal case to the real one given by Algorithm A1.

**Algorithm A1** DiMSC-equivalence

**Require:** The adjacency matrix  $A \in \mathbb{R}^{n_r \times n_c}$  of a directed network, the number of row communities (column communities)  $K$ .

**Ensure:** The estimated  $n_r \times K$  row membership matrix  $\hat{\Pi}_{r,2}$  and the estimated  $n_c \times K$  column membership matrix  $\hat{\Pi}_{c,2}$ .

- 1: Obtain  $\hat{A} = \hat{U}\hat{\Lambda}\hat{V}'$ , the top- $K$ -dimensional SVD of  $A$ . Compute  $\hat{U}_*, \hat{U}_2, \hat{V}_2, \hat{U}_{*,2}$ .
- 2: Apply SP algorithm on the rows of  $\hat{V}_2$  assuming there are  $K$  column communities to obtain  $\hat{\mathcal{I}}_{c,2}$ , the index set returned by SP algorithm.
- 3: Apply SVM-cone algorithm on the rows of  $\hat{U}_{*,2}$  with  $K$  row communities to obtain  $\hat{\mathcal{I}}_{r,2}$ , the index set returned by SVM-cone algorithm.
- 4: Set  $\hat{Y}_{*,2} = \text{diag}(\hat{U}_*(\hat{\mathcal{I}}_{r,2},:) \hat{\Lambda} \hat{V}'(\hat{\mathcal{I}}_{c,2},:)), \hat{Y}_{*,2} = \hat{U}_2 \hat{U}'_{*,2}(\hat{\mathcal{I}}_{r,2},:)(\hat{U}_{*,2}(\hat{\mathcal{I}}_{r,2},:))^{-1}$ ,  $\hat{Z}_{r,2} = \hat{Y}_{*,2} \hat{J}_{*,2}$  and  $\hat{Z}_{c,2} = \hat{V}_2 \hat{V}'_2(\hat{\mathcal{I}}_{c,2},:)(\hat{V}_2(\hat{\mathcal{I}}_{c,2},:))^{-1}$ . Then, set  $\hat{Z}_{r,2} = \max(0, \hat{Z}_{r,2})$  and  $\hat{Z}_{c,2} = \max(0, \hat{Z}_{c,2})$ .
- 5: Estimate  $\Pi_r(i, :)$  by  $\hat{\Pi}_{r,2}(i, :) = \hat{Z}_{r,2}(i, :)/\|\hat{Z}_{r,2}(i, :)\|_1, 1 \leq i \leq n_r$  and estimate  $\Pi_c(j, :)$  by  $\hat{\Pi}_{c,2}(j, :) = \hat{Z}_{c,2}(j, :)/\|\hat{Z}_{c,2}(j, :)\|_1, 1 \leq j \leq n_c$ .

**Lemma A2.** (Equivalence). For the empirical case, we have  $\hat{\mathcal{I}}_{r,2} \equiv \hat{\mathcal{I}}_r, \hat{\mathcal{I}}_{c,2} \equiv \hat{\mathcal{I}}_c, \hat{U}_{*,2}(\hat{\mathcal{I}}_{r,2},:) \hat{U}'_{*,2}(\hat{\mathcal{I}}_{r,2},:) \equiv \hat{U}_*(\hat{\mathcal{I}}_r,:) \hat{U}'_*(\hat{\mathcal{I}}_r,:), \hat{Y}_{*,2} \equiv \hat{Y}_*, \hat{J}_{*,2} \equiv \hat{J}_*, \hat{Z}_{r,2} \equiv \hat{Z}_r, \hat{Z}_{c,2} \equiv \hat{Z}_c, \hat{\Pi}_{r,2} \equiv \hat{\Pi}_r$  and  $\hat{\Pi}_{c,2} \equiv \hat{\Pi}_c$ .

**Proof.** For column nodes, Lemma 3.2 [27] gives  $\hat{\mathcal{I}}_c = \hat{\mathcal{I}}_{c,2}$  (i.e., SP algorithm will return the same indices on both  $\hat{V}$  and  $\hat{V}_2$ ), which gives that  $\hat{V}_2 \hat{V}'_2(\hat{\mathcal{I}}_{c,2},:) = \hat{V}_2 \hat{V}'_2(\hat{\mathcal{I}}_c,:) = \hat{V} \hat{V}'((\hat{V} \hat{V}')'(\hat{\mathcal{I}}_c, :))' = \hat{V} \hat{V}'(\hat{V}(\hat{\mathcal{I}}_c, :)) \hat{V}' = \hat{V} \hat{V}' \hat{V} \hat{V}'(\hat{\mathcal{I}}_c, :) = \hat{V} \hat{V}'(\hat{\mathcal{I}}_c, :)$ , and  $\hat{V}_2(\hat{\mathcal{I}}_{c,2},:) \hat{V}'_2(\hat{\mathcal{I}}_{c,2},:) = \hat{V}_2(\hat{\mathcal{I}}_c, :) \hat{V}'_2(\hat{\mathcal{I}}_c, :) = \hat{V}(\hat{\mathcal{I}}_c, :) \hat{V}'(\hat{V}(\hat{\mathcal{I}}_c, :)) \hat{V}' = \hat{V}(\hat{\mathcal{I}}_c, :) \hat{V}'(\hat{\mathcal{I}}_c, :)$ . Therefore, we have  $\hat{Z}_{c,2} = \hat{Z}_c, \hat{\Pi}_{c,2} = \hat{\Pi}_c$ .

For row nodes, Lemma G.1 [26] guarantees that  $\hat{\mathcal{I}}_r = \hat{\mathcal{I}}_{r,2}$  (i.e., SVM-cone algorithm will return the same indices on both  $\hat{U}_*$  and  $\hat{U}_{*,2}$ ), so immediately we have  $\hat{J}_{*,2} = \hat{J}_*$ . Since  $\hat{U}_{*,2}(\hat{\mathcal{I}}_{r,2},:) = \hat{U}_{*,2}(\hat{\mathcal{I}}_r, :) = N_{\hat{U}}(\hat{\mathcal{I}}_r, \hat{\mathcal{I}}_r) \hat{U}_2(\hat{\mathcal{I}}_r, :) = N_{\hat{U}}(\hat{\mathcal{I}}_r, \hat{\mathcal{I}}_r) \hat{U}(\hat{\mathcal{I}}_r, :) \hat{U}' = \hat{U}_*(\hat{\mathcal{I}}_r, :) \hat{U}'$ , we have  $\hat{U}_2 \hat{U}'_{*,2}(\hat{\mathcal{I}}_{r,2},:) = \hat{U}_2 \hat{U}'_{*,2}(\hat{\mathcal{I}}_r, :) = \hat{U} \hat{U}' \hat{U} \hat{U}'_*(\hat{\mathcal{I}}_r, :) = \hat{U} \hat{U}'_*(\hat{\mathcal{I}}_r, :)$  and  $(\hat{U}_{*,2}(\hat{\mathcal{I}}_{r,2},:))^{-1} \hat{U}'_{*,2}(\hat{\mathcal{I}}_{r,2},:) = (\hat{U}_{*,2}(\hat{\mathcal{I}}_r, :))^{-1} \hat{U}'_{*,2}(\hat{\mathcal{I}}_r, :)$ , which give that  $\hat{Y}_{*,2} = \hat{Y}_*$ , and the claim follows immediately.  $\square$

Lemma A2 guarantees that the DiMSC and DiMSC-equivalence return same estimations for both row and column nodes’s memberships. In this article, we introduce the DiMSC-equivalence algorithm since it is helpful to build a theoretical framework for DiMSC, see Remark A3 and A4 for detail.

**Appendix D. Basic Properties of  $\Omega$**

**Lemma A3.** Under  $DiDCMM_{n_r, n_c}(K, P, \Pi_r, \Pi_c, \Theta_r)$ , we have

$$\frac{\theta_{r, \min}}{\theta_{r, \max} \sqrt{K \lambda_1(\Pi_r' \Pi_r)}} \leq \|U(i, :)\|_F \leq \frac{\theta_{r, \max}}{\theta_{r, \min} \sqrt{\lambda_K(\Pi_r' \Pi_r)}}, \quad 1 \leq i \leq n_r,$$

$$\sqrt{\frac{1}{K \lambda_1(\Pi_c' \Pi_c)}} \leq \|V(j, :)\|_F \leq \sqrt{\frac{1}{\lambda_K(\Pi_c' \Pi_c)}}, \quad 1 \leq j \leq n_c.$$

**Proof.** Since  $I = U'U = U'(\mathcal{I}_r, :)\Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)\Pi_r'\Theta_r^2\Pi_r\Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :)$ , we have

$$((\Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :))((\Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :)))^{-1} = \Pi_r'\Theta_r^2\Pi_r,$$

which gives that

$$\begin{aligned} \max_k \|e'_k(\Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :))\|_F^2 &= \max_k e'_k(\Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :))(\Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :))' e_k \\ &\leq \max_{\|x\|_F=1} x'(\Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :))(\Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :))' x \end{aligned}$$

$$\begin{aligned}
 &= \lambda_1((\Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :))(\Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :)))' \\
 &= \frac{1}{\lambda_K(\Pi_r' \Theta_r^2 \Pi_r)} \leq \frac{1}{\theta_{r,\min}^2 \lambda_K(\Pi_r' \Pi_r)}.
 \end{aligned}$$

Similarly, we have

$$\min_k \|e'_k(\Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :))\|_F^2 \geq \frac{1}{\lambda_1(\Pi_r' \Theta_r^2 \Pi_r)} \geq \frac{1}{\theta_{r,\max}^2 \lambda_1(\Pi_r' \Pi_r)}.$$

Since  $U(i, :) = e'_i U = e'_i \Theta_r \Pi_r \Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :) = \theta_r(i) \Pi_r(i, :) \Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :)$  for  $1 \leq i \leq n_r$ , we have

$$\begin{aligned}
 \|U(i, :)\|_F &= \|\theta_r(i) \Pi_r(i, :) \Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :)\|_F = \theta_r(i) \|\Pi_r(i, :) \Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :)\|_F \\
 &\leq \theta_r(i) \max_i \|\Pi_r(i, :)\|_F \max_i \|e'_i(\Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :))\|_F \\
 &\leq \theta_r(i) \max_i \|e'_i(\Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :))\|_F \leq \frac{\theta_{r,\max}}{\theta_{r,\min} \sqrt{\lambda_K(\Pi_r' \Pi_r)}}.
 \end{aligned}$$

Similarly, we have

$$\begin{aligned}
 \|U(i, :)\|_F &\geq \theta_r(i) \min_i \|\Pi_r(i, :)\|_F \min_i \|e'_i(\Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :))\|_F \\
 &\geq \theta_r(i) \min_i \|e'_i(\Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :))\|_F / \sqrt{K} \geq \frac{\theta_{r,\min}}{\theta_{r,\max} \sqrt{K \lambda_1(\Pi_r' \Pi_r)}}.
 \end{aligned}$$

For  $\|V(j, :)\|_F$ , since  $V = \Pi_c B_c$ , we have

$$\begin{aligned}
 \min_j \|e'_j V\|_F^2 &= \min_j e'_j V V' e_j = \min_j \Pi_c(j, :) B_c B_c' \Pi_c'(j, :) \\
 &= \min_j \|\Pi_c(j, :)\|_F^2 \frac{\Pi_c(j, :)}{\|\Pi_c(j, :)\|_F} B_c B_c' \frac{\Pi_c'(j, :)}{\|\Pi_c'(j, :)\|_F} \\
 &\geq \min_j \|\Pi_c(j, :)\|_F^2 \min_{\|x\|_F=1} x' B_c B_c' x = \min_j \|\Pi_c(j, :)\|_F^2 \lambda_K(B_c B_c') \\
 &\stackrel{\text{By Lemma A4}}{=} \frac{\min_j \|\Pi_c(j, :)\|_F^2}{\lambda_1(\Pi_c' \Pi_c)} \geq \frac{1}{K \lambda_1(\Pi_c' \Pi_c)}.
 \end{aligned}$$

Meanwhile,

$$\begin{aligned}
 \max_j \|e'_j V\|_F^2 &= \max_j \|\Pi_c(j, :)\|_F^2 \frac{\Pi_c(j, :)}{\|\Pi_c(j, :)\|_F} B_c B_c' \frac{\Pi_c'(j, :)}{\|\Pi_c'(j, :)\|_F} \\
 &\leq \max_j \|\Pi_c(j, :)\|_F^2 \max_{\|x\|_F=1} x' B_c B_c' x = \max_j \|\Pi_c(j, :)\|_F^2 \lambda_K(B_c B_c') \\
 &\stackrel{\text{By Lemma A4}}{=} \frac{\max_j \|\Pi_c(j, :)\|_F^2}{\lambda_K(\Pi_c' \Pi_c)} \leq \frac{1}{\lambda_K(\Pi_c' \Pi_c)}.
 \end{aligned}$$

□

**Lemma A4.** Under DiDCMM<sub>n<sub>r</sub>,n<sub>c</sub></sub>(K, P, Π<sub>r</sub>, Π<sub>c</sub>, Θ<sub>r</sub>), we have

$$\frac{\theta_{r,\min}^2 \lambda_K(\Pi_r' \Pi_r)}{\theta_{r,\max}^2 \lambda_1(\Pi_r' \Pi_r)} \leq \lambda_K(U_*(\mathcal{I}_r, :))U'_*(\mathcal{I}_r, :), \lambda_1(U_*(\mathcal{I}_r, :))U'_*(\mathcal{I}_r, :) \leq \frac{\theta_{r,\max}^2 K \lambda_1(\Pi_r' \Pi_r)}{\theta_{r,\min}^2 \lambda_K(\Pi_r' \Pi_r)},$$

$$\text{and } \lambda_1(B_c B_c') = \frac{1}{\lambda_K(\Pi_c' \Pi_c)}, \lambda_K(B_c B_c') = \frac{1}{\lambda_1(\Pi_c' \Pi_c)}.$$

**Proof.** Recall that  $V = \Pi_c B_c$  and  $V'V = I$ , we have  $I = B_c' \Pi_c' \Pi_c B_c$ . As  $B_c$  is full rank, we have  $\Pi_c' \Pi_c = (B_c B_c')^{-1}$ , which gives

$$\lambda_1(B_c B_c') = \frac{1}{\lambda_K(\Pi_c' \Pi_c)}, \lambda_K(B_c B_c') = \frac{1}{\lambda_1(\Pi_c' \Pi_c)}.$$

By the proof of Lemma 2, we know that

$$(U_*(\mathcal{I}_r, :)U'_*(\mathcal{I}_r, :))^{-1} = N_U^{-1}(\mathcal{I}_r, \mathcal{I}_r)\Theta^{-1}(\mathcal{I}_r, \mathcal{I}_r)\Pi'_r\Theta_r^2\Pi_r\Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)N_U^{-1}(\mathcal{I}_r, \mathcal{I}_r),$$

which gives that

$$U_*(\mathcal{I}_r, :)U'_*(\mathcal{I}_r, :) = N_U(\mathcal{I}_r, \mathcal{I}_r)\Theta(\mathcal{I}_r, \mathcal{I}_r)(\Pi'_r\Theta_r^2\Pi_r)^{-1}\Theta_r(\mathcal{I}_r, \mathcal{I}_r)N_U(\mathcal{I}_r, \mathcal{I}_r).$$

Then, we have

$$\begin{aligned} \lambda_1(U_*(\mathcal{I}_r, :)U'_*(\mathcal{I}_r, :)) &= \lambda_1(N_U(\mathcal{I}_r, \mathcal{I}_r)\Theta(\mathcal{I}_r, \mathcal{I}_r)(\Pi'_r\Theta_r^2\Pi_r)^{-1}\Theta_r(\mathcal{I}_r, \mathcal{I}_r)N_U(\mathcal{I}_r, \mathcal{I}_r)) \\ &= \lambda_1(N_U^2(\mathcal{I}_r, \mathcal{I}_r)\Theta_r^2(\mathcal{I}_r, \mathcal{I}_r)(\Pi'_r\Theta_r^2\Pi_r)^{-1}) \leq \lambda_1^2(N_U(\mathcal{I}_r, \mathcal{I}_r)\Theta_r(\mathcal{I}_r, \mathcal{I}_r))\lambda_1((\Pi'_r\Theta_r^2\Pi_r)^{-1}) \\ &= \lambda_1^2(N_U(\mathcal{I}_r, \mathcal{I}_r)\Theta_r(\mathcal{I}_r, \mathcal{I}_r)) / \lambda_K(\Pi'_r\Theta_r^2\Pi_r) \leq (\max_{i \in \mathcal{I}_r} \theta_r(i) / \|U(i, :)\|_F)^2 / \lambda_K(\Pi'_r\Theta_r^2\Pi_r) \\ &\leq \frac{\theta_{r, \max}^2 K \lambda_1(\Pi'_r \Pi_r)}{\lambda_K(\Pi'_r \Theta_r^2 \Pi_r)} \leq \frac{\theta_{r, \max}^2 K \lambda_1(\Pi'_r \Pi_r)}{\theta_{r, \min}^2 \lambda_K(\Pi'_r \Pi_r)}. \end{aligned}$$

Similarly, we have

$$\begin{aligned} \lambda_K(U_*(\mathcal{I}_r, :)U'_*(\mathcal{I}_r, :)) &= \lambda_K(N_U(\mathcal{I}_r, \mathcal{I}_r)\Theta(\mathcal{I}_r, \mathcal{I}_r)(\Pi'_r\Theta_r^2\Pi_r)^{-1}\Theta_r(\mathcal{I}_r, \mathcal{I}_r)N_U(\mathcal{I}_r, \mathcal{I}_r)) \\ &= \lambda_K(N_U^2(\mathcal{I}_r, \mathcal{I}_r)\Theta_r^2(\mathcal{I}_r, \mathcal{I}_r)(\Pi'_r\Theta_r^2\Pi_r)^{-1}) \geq \lambda_K^2(N_U(\mathcal{I}_r, \mathcal{I}_r)\Theta_r(\mathcal{I}_r, \mathcal{I}_r))\lambda_K((\Pi'_r\Theta_r^2\Pi_r)^{-1}) \\ &= \lambda_K^2(N_U(\mathcal{I}_r, \mathcal{I}_r)\Theta_r(\mathcal{I}_r, \mathcal{I}_r)) / \lambda_1(\Pi'_r\Theta_r^2\Pi_r) \geq (\min_{i \in \mathcal{I}_r} \theta_r(i) / \|U(i, :)\|_F)^2 / \lambda_1(\Pi'_r\Theta_r^2\Pi_r) \\ &\geq \frac{\theta_{r, \min}^2 \lambda_K(\Pi'_r \Pi_r)}{\lambda_1(\Pi'_r \Theta_r^2 \Pi_r)} \geq \frac{\theta_{r, \min}^2 \lambda_K(\Pi'_r \Pi_r)}{\theta_{r, \max}^2 \lambda_1(\Pi'_r \Pi_r)}. \end{aligned}$$

□

**Lemma A5.** Under DiDCMM<sub>n<sub>r</sub>,n<sub>c</sub></sub>(K, P, Π<sub>r</sub>, Π<sub>c</sub>, Θ<sub>r</sub>), we have

$$\sigma_K(\Omega) \geq \theta_{r, \min} \sigma_K(P) \sigma_K(\Pi_r) \sigma_K(\Pi_c), \sigma_1(\Omega) \leq \theta_{r, \max} \sigma_1(P) \sigma_1(\Pi_r) \sigma_1(\Pi_c).$$

**Proof.** For σ<sub>K</sub>(Ω), we have

$$\begin{aligned} \sigma_K^2(\Omega) &= \lambda_K(\Omega\Omega') = \lambda_K(\Theta_r \Pi_r P \Pi'_c \Pi_c P' \Pi'_r \Theta_r) = \lambda_K(\Theta_r^2 \Pi_r P \Pi'_c \Pi_c P' \Pi'_r) \\ &\geq \theta_{r, \min}^2 \lambda_K(\Pi'_r \Pi_r P \Pi'_c \Pi_c P') \geq \theta_{r, \min}^2 \lambda_K(\Pi'_r \Pi_r) \lambda_K(P \Pi'_c \Pi_c P') = \theta_{r, \min}^2 \lambda_K(\Pi'_r \Pi_r) \lambda_K(\Pi'_c \Pi_c P' P) \\ &\geq \theta_{r, \min}^2 \lambda_K(\Pi'_r \Pi_r) \lambda_K(\Pi'_c \Pi_c) \lambda_K(P P') = \theta_{r, \min}^2 \sigma_K^2(\Pi_r) \sigma_K^2(\Pi_c) \sigma_K^2(P), \end{aligned}$$

where we have used the fact for any matrices X, Y, the nonzero eigenvalues of XY are the same as the nonzero eigenvalues of YX. Following a similar analysis, the lemma follows. □

**Lemma A6.** Under DiDCMM<sub>n<sub>r</sub>,n<sub>c</sub></sub>(K, P, Π<sub>r</sub>, Π<sub>c</sub>, Θ<sub>r</sub>), when Assumption 1 holds, with probability at least 1 - o((n<sub>r</sub> + n<sub>c</sub>)<sup>-3</sup>), we have

$$\|A - \Omega\| = O(\sqrt{P_{\max} \max(\|\theta_r\|_1, \theta_{r, \max} n_c) \log(n_r + n_c)}).$$

**Proof.** Since the proof is similar to that of Lemma 7 [35], we omit most of the details. Let e<sub>i</sub> be an n<sub>r</sub> × 1 vector, where e<sub>i</sub>(i) = 1 and 0 elsewhere, for row nodes 1 ≤ i ≤ n<sub>r</sub>, and ẽ<sub>j</sub> be an n<sub>c</sub> × 1 vector, where ẽ<sub>j</sub>(j) = 1 and 0 elsewhere, for column nodes 1 ≤ j ≤ n<sub>c</sub>. Set W = ∑<sub>i=1</sub><sup>n<sub>r</sub></sup> ∑<sub>j=1</sub><sup>n<sub>c</sub></sup> W(i, j) e<sub>i</sub> ẽ<sub>j</sub>', where W = A - Ω. Set W<sup>(i,j)</sup> = W(i, j) e<sub>i</sub> ẽ<sub>j</sub>', for 1 ≤ i ≤ n<sub>r</sub>, 1 ≤ j ≤ n<sub>c</sub>. Then, we have E(W<sup>(i,j)</sup>) = 0. For 1 ≤ i ≤ n<sub>r</sub>, 1 ≤ j ≤ n<sub>c</sub>, we have

$$\|W^{(i,j)}\| = \|W(i, j) e_i \tilde{e}_j'\| = |A(i, j) - \Omega(i, j)| \leq 1.$$

Next, we consider the variance parameter

$$\sigma^2 := \max(\|\sum_{i=1}^{n_r} \sum_{j=1}^{n_c} \mathbb{E}(W^{(i,j)}(W^{(i,j)})')\|, \|\sum_{i=1}^{n_r} \sum_{j=1}^{n_c} \mathbb{E}((W^{(i,j)})'W^{(i,j)})\|).$$

Since

$$\mathbb{E}(W^2(i, j)) = \mathbb{E}((A(i, j) - \Omega(i, j))^2) = \text{Var}(A(i, j)),$$

where  $\text{Var}(A(i, j))$  denotes the variance of the Bernoulli random variable  $A(i, j)$ , we have

$$\begin{aligned} \mathbb{E}(W^2(i, j)) &= \text{Var}(A(i, j)) = \mathbb{P}(A(i, j) = 1)(1 - \mathbb{P}(A(i, j) = 1)) \\ &\leq \mathbb{P}(A(i, j) = 1) = \Omega(i, j) = e'_i \Theta_r \Pi_r P \Pi'_c \tilde{e}_j = \theta_r(i) e'_i \Pi_r P \Pi'_c \tilde{e}_j \leq \theta_r(i) P_{\max}. \end{aligned}$$

Since  $e_i e'_i$  is an  $n_r \times n_r$  diagonal matrix with  $(i, i)$ -th entry being one and other entries being zero, we have

$$\|\sum_{i=1}^{n_r} \sum_{j=1}^{n_c} \mathbb{E}(W^{(i,j)}(W^{(i,j)})')\| = \|\sum_{i=1}^{n_r} \sum_{j=1}^{n_c} \mathbb{E}(W^2(i, j)) e_i e'_i\| = \max_{1 \leq i \leq n_r} |\sum_{j=1}^{n_c} \mathbb{E}(W^2(i, j))| \leq \theta_{r, \max} P_{\max} n_c.$$

Similarly, we have  $\|\sum_{i=1}^{n_r} \sum_{j=1}^{n_c} \mathbb{E}((W^{(i,j)})'W^{(i,j)})\| \leq P_{\max} \|\theta_r\|_1$ , which gives that

$$\sigma^2 = \max(\|\sum_{i=1}^{n_r} \sum_{j=1}^{n_c} \mathbb{E}(W^{(i,j)}(W^{(i,j)})')\|, \|\sum_{i=1}^{n_r} \sum_{j=1}^{n_c} \mathbb{E}((W^{(i,j)})'W^{(i,j)})\|) \leq P_{\max} \max(\|\theta_r\|_1, \theta_{r, \max} n_c).$$

By the rectangular version of the Bernstein inequality [71], combining with

$$\sigma^2 \leq P_{\max} \max(\|\theta_r\|_1, \theta_{r, \max} n_c), R = 1, d_1 + d_2 = n_r + n_c, \text{ set}$$

$t = \frac{\alpha + 1 + \sqrt{\alpha^2 + 20\alpha + 19}}{3} \sqrt{P_{\max} \max(\|\theta_r\|_1, \theta_{r, \max} n_c) \log(n_r + n_c)}$  for any  $\alpha > 0$ , we have

$$\begin{aligned} \mathbb{P}(\|W\| \geq t) &= \mathbb{P}(\|\sum_{i=1}^{n_r} \sum_{j=1}^{n_c} W^{(i,j)}\| \geq t) \leq (n_r + n_c) \exp(-\frac{t^2/2}{\sigma^2 + \frac{Rt}{3}}) \\ &\leq (n_r + n_c) \exp(-\frac{t^2/2}{P_{\max} \max(\|\theta_r\|_1, \theta_{r, \max} n_c) + t/3}) \\ &= (n_r + n_c) \exp(-(\alpha + 1) \log(n_r + n_c) \cdot \frac{1}{\frac{18}{(\sqrt{\alpha+19} + \sqrt{\alpha+1})^2} + \frac{2\sqrt{\alpha+1}}{\sqrt{\alpha+19} + \sqrt{\alpha+1}} \sqrt{P_{\max} \max(\|\theta_r\|_1, \theta_{r, \max} n_c)}})} \\ &\leq (n_r + n_c) \exp(-(\alpha + 1) \log(n_r + n_c)) = \frac{1}{(n_r + n_c)^\alpha}, \end{aligned}$$

where we have used Assumption 1 in the last inequality. Set  $\alpha = 3$ , and the claim follows.  $\square$

### Appendix E. Proof of Consistency for DiMSC

Similar to [24,26,27], for our DiMSC, the main theoretical results (i.e., Theorem 2) rely on the row-wise singular vector deviation bounds for the singular eigenvectors of the adjacency matrix.

**Lemma A7.** (Row-wise singular vector deviation) Under  $DiDCMM_{n_r, n_c}(K, P, \Pi_r, \Pi_c, \Theta_r)$ , when Assumption 1 holds, suppose  $\sigma_K(\Omega) \geq C \sqrt{\theta_{r, \max}(n_r + n_c) \log(n_r + n_c)}$ , with probability at least  $1 - o((n_r + n_c)^{-3})$ , we have

$$\max(\|\hat{U}\hat{U}' - UU'\|_{2 \rightarrow \infty}, \|\hat{V}\hat{V}' - VV'\|_{2 \rightarrow \infty}) = O(\frac{\sqrt{P_{\max} \theta_{r, \max} K \log(n_r + n_c)}}{\theta_{r, \min} \sigma_K(P) \sigma_K(\Pi_r) \sigma_K(\Pi_c)}).$$

**Proof.** Let  $H_{\hat{U}} = \hat{U}'U$ , and  $H_{\hat{U}} = U_{H_{\hat{U}}} \Sigma_{H_{\hat{U}}} V'_{H_{\hat{U}}}$  be the SVD decomposition of  $H_{\hat{U}}$  with  $U_{H_{\hat{U}}}, V_{H_{\hat{U}}} \in \mathbb{R}^{n_r \times K}$ , where  $U_{H_{\hat{U}}}$  and  $V_{H_{\hat{U}}}$  represent, respectively, the left and right sin-

gular matrices of  $H_{\hat{U}}$ . Define  $\text{sgn}(H_{\hat{U}}) = U_{H_{\hat{U}}} V'_{H_{\hat{U}}}$ ;  $\text{sgn}(H_{\hat{V}})$  is defined similarly. Since  $\mathbb{E}(A(i, j) - \Omega(i, j)) = 0$ ,  $\mathbb{E}[(A(i, j) - \Omega(i, j))^2] \leq \theta_r(i) P_{\max} \leq \theta_{r, \max} P_{\max}$  by the proof of Lemma A6,  $\frac{1}{\sqrt{\theta_{r, \max} P_{\max} \min(n_r, n_c) / (\mu \log(n_r + n_c))}} \leq O(1)$  holds by Assumption 1, where  $\mu$  is the incoherence parameter defined as  $\mu = \max(\frac{n_r \|U\|_{2 \rightarrow \infty}^2}{K}, \frac{n_c \|V\|_{2 \rightarrow \infty}^2}{K})$ . By Theorem 4.4 [64], with high probability, we have below row-wise singular vector deviation

$$\begin{aligned} & \max(\|\hat{U} \text{sgn}(H_{\hat{U}}) - U\|_{2 \rightarrow \infty}, \|\hat{V} \text{sgn}(H_{\hat{V}}) - V\|_{2 \rightarrow \infty}) \\ & \leq C \frac{\sqrt{P_{\max} \theta_{r, \max} K} (\kappa(\Omega) \sqrt{\frac{\max(n_r, n_c) \mu}{\min(n_r, n_c)}} + \sqrt{\log(n_r + n_c)})}{\sigma_K(\Omega)} \\ & \leq C \frac{\sqrt{P_{\max} \theta_{r, \max} K \log(n_r + n_c)}}{\sigma_K(\Omega)}, \end{aligned} \tag{A1}$$

provided that  $c_1 \sigma_K(\Omega) \geq \sqrt{\theta_{r, \max} P_{\max} (n_r + n_c) \log(n_r + n_c)}$  for some sufficiently small constant  $c_1$ , and here we set  $\sqrt{\frac{\max(n_r, n_c) \mu}{\min(n_r, n_c)}} = O(1)$  for convenience since this term has little effect on the error bounds of DiMSC, especially for the case when  $\frac{n_r}{n_c} = O(1)$ .

Since  $U'U = I, \hat{U}'\hat{U} = I$ , we have  $\|\hat{U}\hat{U}' - UU'\|_{2 \rightarrow \infty} \leq 2\|U - \hat{U} \text{sgn}(H_{\hat{U}})\|_{2 \rightarrow \infty}$  by basic algebra. Now, we are ready to bound  $\|\hat{U}\hat{U}' - UU'\|_{2 \rightarrow \infty}$ :

$$\begin{aligned} \|\hat{U}\hat{U}' - UU'\|_{2 \rightarrow \infty} &= \max_{1 \leq i \leq n_r} \|e'_i(UU' - \hat{U}\hat{U}')\|_F \leq 2\|U - \hat{U} \text{sgn}(H_{\hat{U}})\|_{2 \rightarrow \infty} \\ &\leq C \frac{\sqrt{P_{\max} \theta_{r, \max} K \log(n_r + n_c)}}{\sigma_K(\Omega)} \stackrel{\text{By Lemma A5}}{\leq} C \frac{\sqrt{P_{\max} \theta_{r, \max} K \log(n_r + n_c)}}{\theta_{r, \min} \sigma_K(P) \sigma_K(\Pi_r) \sigma_K(\Pi_c)}. \end{aligned}$$

The lemma holds by following similar proof for  $\|\hat{V}\hat{V}' - VV'\|_{2 \rightarrow \infty}$ .  $\square$

When  $\Theta_r = \rho I, n_r = n_c, \Pi_r = \Pi_c = \Pi, P_{\max} = O(1)$ , and DiCCMM degenerates to MMSB, the bound in Lemma A7 is  $O(\frac{\sqrt{K \log(n)}}{\sigma_K(P) \sqrt{\rho} \lambda_K(\Pi' \Pi)})$ . if we further assume that  $\lambda_K(\Pi' \Pi) = O(\frac{n}{K})$  and  $K = O(1)$ , the bound is of order  $O(\frac{1}{\sigma_K(P)} \frac{1}{\sqrt{n}} \sqrt{\frac{\log(n)}{\rho n}})$ . Set the  $\Theta$  in [24] as  $\sqrt{\rho} I$ , their DCMM degenerates to MMSB, their assumptions are translated to our  $\lambda_K(\Pi' \Pi) = O(\frac{n}{K})$ , when  $K = O(1)$ , the row-wise singular vector deviation bound in the fourth bullet of Lemma 2.1 [24] is  $O(\frac{1}{\sigma_K(P)} \frac{1}{\sqrt{n}} \sqrt{\frac{\log(n)}{\rho n}})$ , which is consistent with ours. Meanwhile, if we further assume that  $\sigma_K(P) = O(1)$ , the bound is of order  $\frac{1}{\sqrt{n}} \sqrt{\frac{\log(n)}{\rho n}}$ .

The next lemma is the cornerstone to characterizing the behaviors of DiMSC.

**Lemma A8.** Under DiDCMM $_{n_r, n_c}(K, P, \Pi_r, \Pi_c, \Theta_r)$ , when conditions of Lemma A7 hold, there exist two permutation matrices  $\mathcal{P}_r, \mathcal{P}_c \in \mathbb{R}^{K \times K}$  such that with probability at least  $1 - o((n_r + n_c)^{-3})$ , we have

$$\begin{aligned} \max_{1 \leq k \leq K} \|e'_k(\hat{U}_{*,2}(\hat{\mathcal{I}}_r, :) - \mathcal{P}'_r U_{*,2}(\mathcal{I}_r, :))\|_F &= O\left(\frac{K^3 \theta_{r, \max}^{11} \omega \kappa^3(\Pi'_r \Pi_r) \lambda_1^{1.5}(\Pi'_r \Pi_r)}{\theta_{r, \min}^{11} \pi_{r, \min}}\right), \\ \max_{1 \leq k \leq K} \|e'_k(\hat{V}_2(\hat{\mathcal{I}}_c, :) - \mathcal{P}'_c V_2(\mathcal{I}_c, :))\|_F &= O(\omega \kappa(\Pi'_c \Pi_c)). \end{aligned}$$

**Proof.** First, we consider column nodes. The detail of the SP algorithm is in Algorithm A2.

---

**Algorithm A2** Successive Projection (SP) [54]

---

**Require:** Near-separable matrix  $Y_{sp} = S_{sp}M_{sp} + Z_{sp} \in \mathbb{R}_+^{m \times n}$ , where  $S_{sp}, M_{sp}$  should satisfy Assumption 1 [54], the number  $r$  of columns to be extracted.

**Ensure:** Set of indices  $\mathcal{K}$  such that  $Y(\mathcal{K}, :) \approx S$  (up to permutation)

- 1: Compute  $\hat{U}_r \in \mathbb{R}^{n_r \times K_r}$  and  $\hat{U}_c \in \mathbb{R}^{n_c \times K_r}$  from the top- $K_r$ -dimensional SVD of  $A$ .
  - 2: Let  $R = Y_{sp}, \mathcal{K} = \{ \}, k = 1$ .
  - 3: **While**  $R \neq 0$  and  $k \leq r$  **do**
  - 4:      $k_* = \operatorname{argmax}_k \|R(k, :)\|_F$ .
  - 5:      $u_k = R(k_*, :)$ .
  - 6:      $R \leftarrow (I - \frac{u_k u_k'}{\|u_k\|_F^2})R$ .
  - 7:      $\mathcal{K} = \mathcal{K} \cup \{k_*\}$ .
  - 8:      $k = k + 1$ .
  - 9: **end while**
- 

Based on Algorithm A2, the following theorem is Theorem 1.1 in [54].

**Theorem A1.** Fix  $m \geq r$  and  $n \geq r$ . Consider a matrix  $Y_{sp} = S_{sp}M_{sp} + Z_{sp}$ , where  $S_{sp} \in \mathbb{R}^{m \times r}$  has a full column rank,  $M_{sp} \in \mathbb{R}^{r \times n}$  is a nonnegative matrix such that the sum of each column is at most 1, and  $Z_{sp} = [Z_{sp,1}, \dots, Z_{sp,n}] \in \mathbb{R}^{m \times n}$ . Suppose  $M_{sp}$  has a submatrix equal to  $I_r$ . Write  $\epsilon \leq \max_{1 \leq i \leq n} \|Z_{sp,i}\|_F$ . Suppose  $\epsilon = O(\frac{\sigma_{\min}(S_{sp})}{\sqrt{r\kappa^2(S_{sp})}})$ , where  $\sigma_{\min}(S_{sp})$  and  $\kappa(S_{sp})$  are the minimum singular value and condition number of  $S_{sp}$ , respectively. If we apply the SP algorithm to columns of  $Y_{sp}$ , then it outputs an index set  $\mathcal{K} \subset \{1, 2, \dots, n\}$  such that  $|\mathcal{K}| = r$  and  $\max_{1 \leq k \leq r} \min_{j \in \mathcal{K}} \|S_{sp}(:, k) - Y_{sp}(:, j)\|_F = O(\epsilon \kappa^2(S_{sp}))$ , where  $S_{sp}(:, k)$  is the  $k$ -th column of  $S_{sp}$ .

Let  $m = K, r = K, n = n_c, Y_{sp} = \hat{V}'_2, Z_{sp} = \hat{V}'_2 - V'_2, S_{sp} = V'_2(\mathcal{I}_c, :)$ , and  $M_{sp} = \Pi'_c$ . By Condition (I2),  $M_{sp}$  has an identity submatrix  $I_K$ . By Lemma A7, we have

$$\epsilon_c = \max_{1 \leq j \leq n_c} \|\hat{V}_2(j, :) - V_2(j, :)\|_F = \|\hat{V}_2(j, :) - V_2(j, :)\|_{2 \rightarrow \infty} \leq \omega.$$

By Theorem A1, there exists a permutation matrix  $\mathcal{P}_c$  such that

$$\max_{1 \leq k \leq K} \|e'_k(\hat{V}_2(\hat{\mathcal{I}}_c, :) - \mathcal{P}'_c V_2(\mathcal{I}_c, :))\|_F = O(\epsilon_c \kappa^2(V_2(\mathcal{I}_c, :)) \sqrt{K}) = O(\omega \kappa^2(V_2(\mathcal{I}_c, :))).$$

Since  $\kappa^2(V_2(\mathcal{I}_c, :)) = \kappa(V_2(\mathcal{I}_c, :)V'_2(\mathcal{I}_c, :)) = \kappa(V(\mathcal{I}_c, :)V'(\mathcal{I}_c, :)) = \kappa(\Pi'_c \Pi_c)$ , where the last equality holds by Lemma A4, we have

$$\max_{1 \leq k \leq K} \|e'_k(\hat{V}_2(\hat{\mathcal{I}}_c, :) - \mathcal{P}'_c V_2(\mathcal{I}_c, :))\|_F = O(\omega \kappa(\Pi'_c \Pi_c)).$$

**Remark A1.** For the ideal case, let  $m = K, r = K, n = n_c, Y_{sp} = V', Z_{sp} = V' - V' \equiv 0, S_{sp} = V'(\mathcal{I}_c, :)$ , and  $M_{sp} = \Pi'_c$ . Then, we have  $\max_{1 \leq j \leq n_c} \|V(j, :) - V(j, :)\|_F = 0$ . By Theorem A1, SP algorithm returns  $\mathcal{I}_c$  when the input is  $V$  assuming there are  $K$  column communities.

Now, we consider row nodes. From Lemma 2, we see that  $U_*(\mathcal{I}_r, :)$  satisfies Condition 1 in [26]. Meanwhile, since  $(U_*(\mathcal{I}_r, :)U'_*(\mathcal{I}_r, :))^{-1} \mathbf{1} > 0$ , we have  $(U_*(\mathcal{I}_r, :)U'_*(\mathcal{I}_r, :))^{-1} \mathbf{1} \geq \eta \mathbf{1}$ , hence  $U_*(\mathcal{I}_r, :)$  satisfies Condition 2 in [26]. Now, we give a lower bound for  $\eta$  to show that  $\eta$  is strictly positive. By the proof of Lemma A4, we have

$$\begin{aligned} (U_*(\mathcal{I}_r, :)U'_*(\mathcal{I}_r, :))^{-1} &= N_U^{-1}(\mathcal{I}_r, \mathcal{I}_r)\Theta^{-1}(\mathcal{I}_r, \mathcal{I}_r)\Pi'_r \Theta^2 \Pi_r \Theta^{-1}(\mathcal{I}_r, \mathcal{I}_r)N_U^{-1}(\mathcal{I}_r, \mathcal{I}_r) \\ &\geq \frac{\theta_{r,\min}^2}{\theta_{r,\max}^2 N_{U,\max}^2} \Pi'_r \Pi_r \geq \frac{\theta_{r,\min}^4}{\theta_{r,\max}^4 K \lambda_1(\Pi'_r \Pi_r)} \Pi'_r \Pi_r, \end{aligned}$$

where we set  $N_{U,\max} = \max_{1 \leq i \leq n_r} N_U(i, i)$ , and we have used the facts that  $N_U, \Theta_r$  are diagonal matrices, and  $N_{U,\max} \leq \frac{\theta_{r,\max} \sqrt{K\lambda_1(\Pi'_r \Pi_r)}}{\theta_{r,\min}}$  by Lemma A3. Then, we have

$$\begin{aligned} \eta &= \min_{1 \leq k \leq K} ((U_*(\mathcal{I}_r, :) U_*'(\mathcal{I}_r, :))^{-1} \mathbf{1})(k) \geq \frac{\theta_{r,\min}^4}{\theta_{r,\max}^4 K \lambda_1(\Pi'_r \Pi_r)} \min_{1 \leq k \leq K} e'_k \Pi'_r \Pi_r \mathbf{1} \\ &= \frac{\theta_{r,\min}^4}{\theta_{r,\max}^4 K \lambda_1(\Pi'_r \Pi_r)} \min_{1 \leq k \leq K} e'_k \Pi'_r \mathbf{1} = \frac{\theta_{r,\min}^4 \pi_{r,\min}}{\theta_{r,\max}^4 K \lambda_1(\Pi'_r \Pi_r)}, \end{aligned}$$

i.e.,  $\eta$  is strictly positive. Since  $U_{*,2}(\mathcal{I}_r, :) U_{*,2}'(\mathcal{I}_r, :) \equiv U_*(\mathcal{I}_r, :) U_*'(\mathcal{I}_r, :)$ , we have  $U_{*,2}(\mathcal{I}_r, :)$  also satisfies Conditions 1 and 2 in [26]. The above analysis shows that we can directly apply Lemma F.1 of [26] since the ideal DiMSC algorithm satisfies Conditions 1 and 2 in [26], therefore there exists a permutation matrix  $\mathcal{P}_r \in \mathbb{R}^{K \times K}$  such that

$$\max_{1 \leq k \leq K} \|e'_k (\hat{U}_{*,2}(\hat{\mathcal{I}}_r, :) - \mathcal{P}'_r U_{*,2}(\mathcal{I}_r, :))\|_F = O\left(\frac{\sqrt{K} \zeta \epsilon_r}{\lambda_K^{1.5}(U_{*,2}(\mathcal{I}_r, :) U_{*,2}'(\mathcal{I}_r, :))}\right),$$

where  $\zeta \leq \frac{4K}{\eta \lambda_K^{1.5}(U_{*,2}(\mathcal{I}_r, :) U_{*,2}'(\mathcal{I}_r, :))} = O\left(\frac{K}{\eta \lambda_K^{1.5}(U_{*,2}(\mathcal{I}_r, :) U_{*,2}'(\mathcal{I}_r, :))}\right)$ , and  $\epsilon_r = \max_{1 \leq i \leq n_r} \|\hat{U}_{*,2}(i, :) - U_{*,2}(i, :)\|$ . Next, we bound  $\epsilon_r$  as below

$$\begin{aligned} \|\hat{U}_{*,2}(i, :) - U_{*,2}(i, :)\|_F &= \left\| \frac{\hat{U}_2(i, :)\|U_2(i, :)\|_F - U_2(i, :)\|\hat{U}_2(i, :)\|_F}{\|\hat{U}_2(i, :)\|_F \|U_2(i, :)\|_F} \right\|_F \leq \frac{2\|\hat{U}_2(i, :) - U_2(i, :)\|_F}{\|U_2(i, :)\|_F} \\ &\leq \frac{2\|\hat{U}_2 - U_2\|_{2 \rightarrow \infty}}{\|U_2(i, :)\|_F} \leq \frac{2\omega}{\|U_2(i, :)\|_F} = \frac{2\omega}{\|(UU')(i, :)\|_F} = \frac{2\omega}{\|U(i, :)\|_F} = \frac{2\omega}{\|U(i, :)\|_F} \\ &\leq 2\omega \frac{\theta_{r,\max} \sqrt{K\lambda_1(\Pi'_r \Pi_r)}}{\theta_{r,\min}}, \end{aligned}$$

where the last inequality holds by Lemma A3. Then, we have  $\epsilon_r = O\left(\omega \frac{\theta_{r,\max} \sqrt{K\lambda_1(\Pi'_r \Pi_r)}}{\theta_{r,\min}}\right)$ . Finally, by Lemma A4, we have

$$\max_{1 \leq k \leq K} \|e'_k (\hat{U}_{*,2}(\hat{\mathcal{I}}_r, :) - \mathcal{P}'_r U_{*,2}(\mathcal{I}_r, :))\|_F = O\left(\frac{K^3 \theta_{r,\max}^{11} \omega \kappa^3 (\Pi'_r \Pi_r) \lambda_1^{1.5} (\Pi'_r \Pi_r)}{\theta_{r,\min}^{11} \pi_{r,\min}}\right).$$

**Remark A2.** For the ideal case, when setting  $U_*$  as the input of the SVM-cone algorithm assuming there are  $K$  row communities, since  $\|U_* - U_*\|_{2 \rightarrow \infty} = 0$ , Lemma F.1; [26] guarantees that SVM-cone algorithm returns  $\mathcal{I}_r$  exactly. Meanwhile, another view to see that the SVM-cone algorithm exactly obtains  $\mathcal{I}_r$  when the input is  $U_*$  (also  $U_{2,*}$ ) is given in Appendix F, which focuses on following the three steps of SVM-cone algorithm to show that it returns  $\mathcal{I}_r$  with input  $U_*$  (also  $U_{*,2}$ ), instead of simply applying Lemma F.1. [26].

□

**Lemma A9.** Under  $DiDCMM_{n_r, n_c}(K, P, \Pi_r, \Pi_c, \Theta_r)$ , when conditions of Lemma A7 hold, with probability at least  $1 - o((n_r + n_c)^{-3})$ , we have

$$\begin{aligned} \max_{1 \leq i \leq n_r} \|e'_i (\hat{Z}_r - Z_r \mathcal{P}_r)\|_F &= O\left(\frac{K^5 \theta_{r,\max}^{15} \omega \kappa^{4.5} (\Pi'_r \Pi_r) \kappa (\Pi_c) \lambda_1^{1.5} (\Pi'_r \Pi_r)}{\theta_{r,\min}^{14} \pi_{r,\min}}\right), \\ \max_{1 \leq j \leq n_c} \|e'_j (\hat{Z}_c - Z_c \mathcal{P}_c)\|_F &= O(\omega \kappa (\Pi'_c \Pi_c) \sqrt{K \lambda_1 (\Pi'_c \Pi_c)}). \end{aligned}$$

**Proof.** First, we consider column nodes. Recall that  $V(\mathcal{I}_c, :) = B_c$ . For convenience, set  $\hat{V}(\hat{\mathcal{I}}_c, :) = \hat{B}_c, V_2(\mathcal{I}_c, :) = B_{2c}, \hat{V}_2(\hat{\mathcal{I}}_c, :) = \hat{B}_{2c}$ . We bound  $\|e'_j (\hat{Z}_c - Z_c \mathcal{P}_c)\|_F$  when the input

is  $\hat{V}$  in the SP algorithm. Recall that  $Z_c = \max(VV'(\mathcal{I}_c, :)(V(\mathcal{I}_c, :))^{-1}, 0) \equiv \Pi_c$ , for  $1 \leq j \leq n_c$ , we have

$$\begin{aligned} & \|e'_j(\hat{Z}_c - Z_c \mathcal{P}_c)\|_F = \|e'_j(\max(0, \hat{V} \hat{B}'_c (\hat{B}_c \hat{B}'_c)^{-1}) - VB'_c(B_c B'_c)^{-1} \mathcal{P}_c)\|_F \\ & \leq \|e'_j(\hat{V} \hat{B}'_c (\hat{B}_c \hat{B}'_c)^{-1} - VB'_c(B_c B'_c)^{-1} \mathcal{P}_c)\|_F \\ & = \|e'_j(\hat{V} - V(V' \hat{V})) \hat{B}'_c (\hat{B}_c \hat{B}'_c)^{-1} + e'_j(V(V' \hat{V}) \hat{B}'_c (\hat{B}_c \hat{B}'_c)^{-1} - V(V' \hat{V})(\mathcal{P}'_c(B_c B'_c)(B'_c)^{-1}(V' \hat{V}))^{-1})\|_F \\ & \leq \|e'_j(\hat{V} - V(V' \hat{V})) \hat{B}'_c (\hat{B}_c \hat{B}'_c)^{-1}\|_F + \|e'_j V(V' \hat{V})(\hat{B}'_c (\hat{B}_c \hat{B}'_c)^{-1} - (\mathcal{P}'_c(B_c B'_c)(B'_c)^{-1}(V' \hat{V}))^{-1})\|_F \\ & \leq \|e'_j(\hat{V} - V(V' \hat{V}))\|_F \|\hat{B}_c^{-1}\|_F + \|e'_j V(V' \hat{V})(\hat{B}'_c (\hat{B}_c \hat{B}'_c)^{-1} - (\mathcal{P}'_c(B_c B'_c)(B'_c)^{-1}(V' \hat{V}))^{-1})\|_F \\ & \leq \sqrt{K} \|e'_j(\hat{V} - V(V' \hat{V}))\|_F / \sqrt{\lambda_K(\hat{B}_c \hat{B}'_c)} + \|e'_j V(V' \hat{V})(\hat{B}_c^{-1} - (\mathcal{P}'_c B_c (V' \hat{V}))^{-1})\|_F \\ & = \sqrt{K} \|e'_j(\hat{V} \hat{V}' - VV') \hat{V}\|_F O(\sqrt{\lambda_1(\Pi'_c \Pi_c)}) + \|e'_j V(V' \hat{V})(\hat{B}_c^{-1} - (\mathcal{P}'_c B_c (V' \hat{V}))^{-1})\|_F \\ & \leq \sqrt{K} \|e'_j(\hat{V} \hat{V}' - VV')\|_F O(\sqrt{\lambda_1(\Pi'_c \Pi_c)}) + \|e'_j V(V' \hat{V})(\hat{B}_c^{-1} - (\mathcal{P}'_c B_c (V' \hat{V}))^{-1})\|_F \\ & \leq \sqrt{K} \omega O(\sqrt{\lambda_1(\Pi'_c \Pi_c)}) + \|e'_j V(V' \hat{V})(\hat{B}_c^{-1} - (\mathcal{P}'_c B_c (V' \hat{V}))^{-1})\|_F \\ & = O(\omega \sqrt{K \lambda_1(\Pi'_c \Pi_c)}) + \|e'_j V(V' \hat{V})(\hat{B}_c^{-1} - (\mathcal{P}'_c B_c (V' \hat{V}))^{-1})\|_F, \end{aligned}$$

where we have used similar idea in the proof of Lemma VII.3 in [27] such that apply  $O(\frac{1}{\lambda_K(\hat{B}_c \hat{B}'_c)})$  to estimate  $\frac{1}{\lambda_K(\hat{B}_c \hat{B}'_c)}$ , then by Lemma A4, we have  $\frac{1}{\lambda_K(\hat{B}_c \hat{B}'_c)} = O(\lambda_1(\Pi'_c \Pi_c))$ .

Now, we aim to bound  $\|e'_j V(V' \hat{V})(\hat{B}_c^{-1} - (\mathcal{P}'_c B_c (V' \hat{V}))^{-1})\|_F$ . For convenience, set  $T_c = V' \hat{V}, S_c = \mathcal{P}'_c B_c T_c$ . We have

$$\begin{aligned} & \|e'_j V(V' \hat{V})(\hat{B}_c^{-1} - (\mathcal{P}'_c B_c (V' \hat{V}))^{-1})\|_F = \|e'_j V T_c S_c^{-1} (S_c - \hat{B}_c) \hat{B}_c^{-1}\|_F \\ & \leq \|e'_j V T_c S_c^{-1} (S_c - \hat{B}_c)\|_F \|\hat{B}_c^{-1}\|_F \leq \|e'_j V T_c S_c^{-1} (S_c - \hat{B}_c)\|_F \frac{\sqrt{K}}{|\lambda_K(\hat{B}_c)|} \\ & = \|e'_j V T_c S_c^{-1} (S_c - \hat{B}_c)\|_F \frac{\sqrt{K}}{\sqrt{\lambda_K(\hat{B}_c \hat{B}'_c)}} \leq \|e'_j V T_c S_c^{-1} (S_c - \hat{B}_c)\|_F O(\sqrt{K \lambda_1(\Pi'_c \Pi_c)}) \\ & = \|e'_j V T_c T_c^{-1} B'_c (B_c B'_c)^{-1} \mathcal{P}_c (S_c - \hat{B}_c)\|_F O(\sqrt{K \lambda_1(\Pi'_c \Pi_c)}) \\ & = \|e'_j V B'_c (B_c B'_c)^{-1} \mathcal{P}_c (S_c - \hat{B}_c)\|_F O(\sqrt{K \lambda_1(\Pi'_c \Pi_c)}) \\ & = \|e'_j Z_c \mathcal{P}_c (S_c - \hat{B}_c)\|_F O(\sqrt{K \lambda_1(\Pi'_c \Pi_c)}) \stackrel{\text{By } Z_c = \Pi_c}{\leq} \max_{1 \leq k \leq K} \|e'_k (S_c - \hat{B}_c)\|_F O(\sqrt{K \lambda_1(\Pi'_c \Pi_c)}) \\ & = \max_{1 \leq k \leq K} \|e'_k (\hat{B}_c - \mathcal{P}'_c B_c V' \hat{V})\|_F O(\sqrt{K \lambda_1(\Pi'_c \Pi_c)}) \\ & = \max_{1 \leq k \leq K} \|e'_k (\hat{B}_c \hat{V}' - \mathcal{P}'_c B_c V') \hat{V}\|_F O(\sqrt{K \lambda_1(\Pi'_c \Pi_c)}) \\ & \leq \max_{1 \leq k \leq K} \|e'_k (\hat{B}_c \hat{V}' - \mathcal{P}'_c B_c V')\|_F O(\sqrt{K \lambda_1(\Pi'_c \Pi_c)}) \\ & = \max_{1 \leq k \leq K} \|e'_k (\hat{B}_{2c} - \mathcal{P}'_c B_{2c})\|_F O(\sqrt{K \lambda_1(\Pi'_c \Pi_c)}) \tag{A2} \\ & = O(\omega \kappa (\Pi'_c \Pi_c) \sqrt{K \lambda_1(\Pi'_c \Pi_c)}). \end{aligned}$$

**Remark A3.** Equation (A2) supports our statement that building the theoretical framework of DiMSC benefits a lot by introducing the DiMSC-equivalence algorithm since  $\|\hat{B}_{2c} - \mathcal{P}'_c B_{2c}\|_{2 \rightarrow \infty}$  is obtained from DiMSC-equivalence (i.e., inputting  $\hat{V}_2$  in the SP algorithm obtains  $\|\hat{B}_{2c} - \mathcal{P}'_c B_{2c}\|_{2 \rightarrow \infty}$ ).

Then, we have

$$\begin{aligned} & \|e'_j(\hat{Z}_c - Z_c \mathcal{P}_c)\|_F \leq O(\omega \sqrt{K \lambda_1(\Pi'_c \Pi_c)}) + \|e'_j V(V' \hat{V})(\hat{B}_c^{-1} - (\mathcal{P}'_c B_c (V' \hat{V}))^{-1})\|_F \\ & \leq O(\omega \sqrt{K \lambda_1(\Pi'_c \Pi_c)}) + O(\omega \kappa (\Pi'_c \Pi_c) \sqrt{K \lambda_1(\Pi'_c \Pi_c)}) = O(\omega \kappa (\Pi'_c \Pi_c) \sqrt{K \lambda_1(\Pi'_c \Pi_c)}). \end{aligned}$$

Next, we consider row nodes. For  $1 \leq i \leq n_r$ , since  $Z_r = Y_* J_*$ ,  $\hat{Z}_r = \hat{Y}_* \hat{J}_*$ , we have

$$\begin{aligned} \|e'_i(\hat{Z}_r - Z_r \mathcal{P}_r)\|_F &= \|e'_i(\max(0, \hat{Y}_* \hat{J}_*) - Y_* J_* \mathcal{P}_r)\|_F \leq \|e'_i(\hat{Y}_* \hat{J}_* - Y_* J_* \mathcal{P}_r)\|_F \\ &= \|e'_i(\hat{Y}_* - Y_* \mathcal{P}_r) \hat{J}_* + e'_i Y_* \mathcal{P}_r (\hat{J}_* - \mathcal{P}'_r J_* \mathcal{P}_r)\|_F \\ &\leq \|e'_i(\hat{Y}_* - Y_* \mathcal{P}_r)\|_F \|\hat{J}_*\|_F + \|e'_i Y_* \mathcal{P}_r\|_F \|\hat{J}_* - \mathcal{P}'_r J_* \mathcal{P}_r\|_F \\ &= \|e'_i(\hat{Y}_* - Y_* \mathcal{P}_r)\|_F \|\hat{J}_*\|_F + \|e'_i Y_*\|_F \|\hat{J}_* - \mathcal{P}'_r J_* \mathcal{P}_r\|_F. \end{aligned}$$

Therefore, the bound of  $\|e'_i(\hat{Z}_r - Z_r \mathcal{P}_r)\|_F$  can be obtained as long as we bound  $\|e'_i(\hat{Y}_* - Y_* \mathcal{P}_r)\|_F$ ,  $\|\hat{J}_*\|_F$ ,  $\|e'_i Y_*\|_F$  and  $\|\hat{J}_* - \mathcal{P}'_r J_* \mathcal{P}_r\|_F$ . We bound the four terms as below:

- We bound  $\|e'_i(\hat{Y}_* - Y_* \mathcal{P}_r)\|_F$  first. Similar as bounding  $\|e'_i(\hat{Z}_c - Z_c \mathcal{P}_c)\|_F$ , we set  $U_*(\mathcal{I}_r, :)$   $= B_R, \hat{U}_*(\hat{\mathcal{I}}_r, :) = \hat{B}_R, U_{*,2}(\mathcal{I}_r, :) = B_{2R}, \hat{U}_{*,2}(\hat{\mathcal{I}}_r, :) = \hat{B}_{2R}$  for convenience. We bound  $\|e'_i(\hat{Y}_* - Y_* \mathcal{P}_r)\|_F$  when the input is  $\hat{U}_*$  in the SVM-cone algorithm. For  $1 \leq i \leq n_r$ , we have

$$\begin{aligned} \|e'_i(\hat{Y}_* - Y_* \mathcal{P}_r)\|_F &= \|e'_i(\hat{U} \hat{B}'_R (\hat{B}_R \hat{B}'_R)^{-1} - U B'_R (B_R B'_R)^{-1} \mathcal{P}_r)\|_F \\ &= \|e'_i(\hat{U} - U(U' \hat{U})) \hat{B}'_R (\hat{B}_R \hat{B}'_R)^{-1} + e'_i(U(U' \hat{U}) \hat{B}'_R (\hat{B}_R \hat{B}'_R)^{-1} \\ &\quad - U(U' \hat{U}) (\mathcal{P}'_r (B_R B'_R) (B'_R)^{-1} (U' \hat{U}))^{-1})\|_F \\ &\leq \|e'_i(\hat{U} - U(U' \hat{U})) \hat{B}'_R (\hat{B}_R \hat{B}'_R)^{-1}\|_F + \|e'_i U(U' \hat{U}) (\hat{B}'_R (\hat{B}_R \hat{B}'_R)^{-1} \\ &\quad - (\mathcal{P}'_r (B_R B'_R) (B'_R)^{-1} (U' \hat{U}))^{-1})\|_F \\ &\leq \|e'_i(\hat{U} - U(U' \hat{U}))\|_F \|\hat{B}_R^{-1}\|_F + \|e'_i U(U' \hat{U}) (\hat{B}'_R (\hat{B}_R \hat{B}'_R)^{-1} - (\mathcal{P}'_r (B_R B'_R) (B'_R)^{-1} (U' \hat{U}))^{-1})\|_F \\ &\leq \sqrt{K} \|e'_i(\hat{U} - U(U' \hat{U}))\|_F / \sqrt{\lambda_K(\hat{B}_R \hat{B}'_R)} + \|e'_i U(U' \hat{U}) (\hat{B}_R^{-1} - (\mathcal{P}'_r B_R (U' \hat{U}))^{-1})\|_F \\ &\stackrel{(i)}{=} \sqrt{K} \|e'_i(\hat{U} \hat{U}' - U U') \hat{U}\|_F O\left(\frac{\theta_{r, \max} \sqrt{\kappa(\Pi'_r \Pi_r)}}{\theta_{r, \min}}\right) + \|e'_i U(U' \hat{U}) (\hat{B}_R^{-1} - (\mathcal{P}'_r B_R (U' \hat{U}))^{-1})\|_F \\ &\leq \sqrt{K} \|e'_i(\hat{U} \hat{U}' - U U')\|_F O\left(\frac{\theta_{r, \max} \sqrt{\kappa(\Pi'_r \Pi_r)}}{\theta_{r, \min}}\right) + \|e'_i U(U' \hat{U}) (\hat{B}_R^{-1} - (\mathcal{P}'_r B_R (U' \hat{U}))^{-1})\|_F \\ &\leq \sqrt{K} \omega O\left(\frac{\theta_{r, \max} \sqrt{\kappa(\Pi'_r \Pi_r)}}{\theta_{r, \min}}\right) + \|e'_i U(U' \hat{U}) (\hat{B}_R^{-1} - (\mathcal{P}'_r B_R (U' \hat{U}))^{-1})\|_F \\ &= O\left(\frac{\theta_{r, \max} \sqrt{K \kappa(\Pi'_r \Pi_r)}}{\theta_{r, \min}}\right) + \|e'_i U(U' \hat{U}) (\hat{B}_R^{-1} - (\mathcal{P}'_r B_R (U' \hat{U}))^{-1})\|_F, \end{aligned}$$

where we have used similar idea in the proof of Lemma VII.3 in [27] such that we apply  $O\left(\frac{1}{\lambda_K(\hat{B}_R \hat{B}'_R)}\right)$  to estimate  $\frac{1}{\lambda_K(\hat{B}_R \hat{B}'_R)}$ , hence (i) holds by Lemma A4.

Now, we aim to bound  $\|e'_i U(U' \hat{U}) (\hat{B}_R^{-1} - (\mathcal{P}'_r B_R (U' \hat{U}))^{-1})\|_F$ . For convenience, set  $T_r = U' \hat{U}, S_r = \mathcal{P}'_r B_R T_r$ . We have

$$\begin{aligned} \|e'_i U(U' \hat{U}) (\hat{B}_R^{-1} - (\mathcal{P}'_r B_R (U' \hat{U}))^{-1})\|_F &= \|e'_i U T_r S_r^{-1} (S_r - \hat{B}_R) \hat{B}_R^{-1}\|_F \\ &\leq \|e'_i U T_r S_r^{-1} (S_r - \hat{B}_R)\|_F \|\hat{B}_R^{-1}\|_F \leq \|e'_i U T_r S_r^{-1} (S_r - \hat{B}_R)\|_F \frac{\sqrt{K}}{|\lambda_K(\hat{B}_R)|} \\ &= \|e'_i U T_r S_r^{-1} (S_r - \hat{B}_R)\|_F \frac{\sqrt{K}}{\sqrt{\lambda_K(\hat{B}_R \hat{B}'_R)}} \leq \|e'_i U T_r S_r^{-1} (S_r - \hat{B}_R)\|_F O\left(\frac{\theta_{r, \max} \sqrt{K \kappa(\Pi'_r \Pi_r)}}{\theta_{r, \min}}\right) \\ &= \|e'_i U T_r T_r^{-1} B'_R (B_R B'_R)^{-1} \mathcal{P}_r (S_r - \hat{B}_R)\|_F O\left(\frac{\theta_{r, \max} \sqrt{K \kappa(\Pi'_r \Pi_r)}}{\theta_{r, \min}}\right) \\ &= \|e'_i U B'_R (B_R B'_R)^{-1} \mathcal{P}_r (S_r - \hat{B}_R)\|_F O\left(\frac{\theta_{r, \max} \sqrt{K \kappa(\Pi'_r \Pi_r)}}{\theta_{r, \min}}\right) \\ &= \|e'_i Y_* \mathcal{P}_r (S_r - \hat{B}_R)\|_F O\left(\frac{\theta_{r, \max} \sqrt{K \lambda_1(\Pi'_r \Pi_r)}}{\theta_{r, \min}}\right) \leq \|e'_i Y_*\|_F \|S_r - \hat{B}_R\|_F O\left(\frac{\theta_{r, \max} \sqrt{K \lambda_1(\Pi'_r \Pi_r)}}{\theta_{r, \min}}\right) \end{aligned}$$

$$\begin{aligned}
 &\text{By Equation (A4)} \leq \frac{\theta_{r,\max}^2 \sqrt{K\lambda_1(\Pi_r' \Pi_r)}}{\theta_{r,\min}^2 \lambda_K(\Pi_r' \Pi_r)} \max_{1 \leq k \leq K} \|e'_k(S_r - \hat{B}_R)\|_F O\left(\frac{\theta_{r,\max} K \sqrt{\kappa(\Pi_r' \Pi_r)}}{\theta_{r,\min}}\right) \\
 &= \max_{1 \leq k \leq K} \|e'_k(\hat{B}_R - \mathcal{P}'_r B_R U' \hat{U})\|_F O\left(\frac{\theta_{r,\max}^3 K^{1.5} \kappa(\Pi_r' \Pi_r)}{\theta_{r,\min}^3 \sqrt{\lambda_K(\Pi_r' \Pi_r)}}\right) \\
 &= \max_{1 \leq k \leq K} \|e'_k(\hat{B}_R \hat{U}' - \mathcal{P}'_r B_R U') \hat{U}\|_F O\left(\frac{\theta_{r,\max}^3 K^{1.5} \kappa(\Pi_r' \Pi_r)}{\theta_{r,\min}^3 \sqrt{\lambda_K(\Pi_r' \Pi_r)}}\right) \\
 &\leq \max_{1 \leq k \leq K} \|e'_k(\hat{B}_R \hat{U}' - \mathcal{P}'_r B_R U')\|_F O\left(\frac{\theta_{r,\max}^3 K^{1.5} \kappa(\Pi_r' \Pi_r)}{\theta_{r,\min}^3 \sqrt{\lambda_K(\Pi_r' \Pi_r)}}\right) \\
 &= \max_{1 \leq k \leq K} \|e'_k(\hat{B}_{2R} - \mathcal{P}'_r B_{2R})\|_F O\left(\frac{\theta_{r,\max}^3 K^{1.5} \kappa(\Pi_r' \Pi_r)}{\theta_{r,\min}^3 \sqrt{\lambda_K(\Pi_r' \Pi_r)}}\right) \tag{A3} \\
 &\stackrel{\text{By Lemma A8}}{=} O\left(\frac{K^{4.5} \theta_{r,\max}^{14} \omega \kappa^{4.5}(\Pi_r' \Pi_r) \lambda_1(\Pi_r' \Pi_r)}{\theta_{r,\min}^{14} \pi_{r,\min}}\right).
 \end{aligned}$$

**Remark A4.** Similar as Equation (A2), Equation (A3) supports our statement that building the theoretical framework of DiMSC benefits a lot by introducing the DiMSC-equivalence algorithm since  $\|\hat{B}_{2R} - \mathcal{P}'_r B_{2R}\|_{2 \rightarrow \infty}$  is obtained from DiMSC-equivalence (i.e., inputting  $\hat{U}_{*,2}$  in the SVM-cone algorithm obtains  $\|\hat{B}_{2R} - \mathcal{P}'_r B_{2R}\|_{2 \rightarrow \infty}$ ).

Then, we have

$$\begin{aligned}
 \|e'_i(\hat{Y}_* - Y_* \mathcal{P}_r)\|_F &\leq O\left(\omega \frac{\theta_{r,\max} \sqrt{K\kappa(\Pi_r' \Pi_r)}}{\theta_{r,\min}}\right) + \|e'_i U(U' \hat{U})(\hat{B}_R^{-1} - (\mathcal{P}'_r B_R U' \hat{U})^{-1})\|_F \\
 &\leq O\left(\omega \frac{\theta_{r,\max} \sqrt{K\kappa(\Pi_r' \Pi_r)}}{\theta_{r,\min}}\right) + O\left(\frac{K^{4.5} \theta_{r,\max}^{14} \omega \kappa^{4.5}(\Pi_r' \Pi_r) \lambda_1(\Pi_r' \Pi_r)}{\theta_{r,\min}^{14} \pi_{r,\min}}\right) \\
 &= O\left(\frac{K^{4.5} \theta_{r,\max}^{14} \omega \kappa^{4.5}(\Pi_r' \Pi_r) \lambda_1(\Pi_r' \Pi_r)}{\theta_{r,\min}^{14} \pi_{r,\min}}\right).
 \end{aligned}$$

- for  $\|e'_i Y_*\|_F$ , since  $Y_* = U U_*^{-1}(\mathcal{I}_r, :)$ , by Lemmas (A3) and (A4), we have

$$\|e'_i Y_*\|_F \leq \|U(i, :)\|_F \|U_*^{-1}(\mathcal{I}_r, :)\|_F \leq \frac{\sqrt{K} \|U(i, :)\|_F}{\sqrt{\lambda_K(U_*(\mathcal{I}_r, :)) U_*^{-1}(\mathcal{I}_r, :)}} \leq \frac{\theta_{r,\max}^2 \sqrt{K\lambda_1(\Pi_r' \Pi_r)}}{\theta_{r,\min}^2 \lambda_K(\Pi_r' \Pi_r)}. \tag{A4}$$

- for  $\|\hat{f}_*\|_F$ , recall that  $\hat{f}_* = \text{diag}(\hat{U}_*(\hat{I}_r, :)) \hat{\Lambda} \hat{V}'(\hat{\mathcal{I}}_c, :)$ , we have

$$\begin{aligned}
 \|\hat{f}_*\| &= \max_{1 \leq k \leq K} \hat{f}_*(k, k) = \max_{1 \leq k \leq K} e'_k \hat{U}_*(\hat{I}_r, :)\hat{\Lambda} \hat{V}'(\hat{\mathcal{I}}_c, :)\mathbf{e}_k \\
 &= \max_{1 \leq k \leq K} \|e'_k \hat{U}_*(\hat{I}_r, :)\hat{\Lambda} \hat{V}'(\hat{\mathcal{I}}_c, :)\mathbf{e}_k\| \\
 &\leq \max_{1 \leq k \leq K} \|e'_k \hat{U}_*(\hat{I}_r, :)\| \|\hat{\Lambda}\| \|\hat{V}'(\hat{\mathcal{I}}_c, :)\mathbf{e}_k\| \leq \max_{1 \leq k \leq K} \|e'_k \hat{U}_*(\hat{I}_r, :)\|_F \|\hat{\Lambda}\| \|\hat{V}'(\hat{\mathcal{I}}_c, :)\mathbf{e}_k\| \\
 &= \max_{1 \leq k \leq K} \|A\| \|\hat{V}'(\hat{\mathcal{I}}_c, :)\mathbf{e}_k\| = \max_{1 \leq k \leq K} \|A\| \|(e'_k \hat{V}(\hat{\mathcal{I}}_c, :))'\| = \max_{1 \leq k \leq K} \|A\| \|e'_k \hat{V}(\hat{\mathcal{I}}_c, :)\| \\
 &\leq \max_{1 \leq k \leq K} \|A\| \|e'_k \hat{V}(\hat{\mathcal{I}}_c, :)\|_F \leq \|A\| \|\hat{V}\|_{2 \rightarrow \infty} = \|A\| \|\hat{V} \text{sgn}(H_{\hat{V}}) - V + V\|_{2 \rightarrow \infty} \\
 &\leq \|A\| (\|\hat{V} \text{sgn}(H_{\hat{V}}) - V\|_{2 \rightarrow \infty} + \|V\|_{2 \rightarrow \infty}).
 \end{aligned}$$

By Lemmas (A6) and (A5),  $\|A\| = \|A - \Omega + \Omega\| \leq \|A - \Omega\| + \sigma_1(\Omega) \leq \|A - \Omega\| + \theta_{r,\max} \sigma_1(P) \sigma_1(\Pi_r) \sigma_1(\Pi_c) = O(\theta_{r,\max} \sigma_1(\Pi_r) \sigma_1(\Pi_c))$ . By Lemma (A5) and Equation (A1),  $\|\hat{V} \text{sgn}(H_{\hat{V}}) - V\|_{2 \rightarrow \infty} \leq C \frac{\sqrt{\theta_{r,\max} K \log(n_r + n_c)}}{\theta_{r,\min} \sigma_K(P) \sigma_K(\Pi_r) \sigma_K(\Pi_c)}$ . By Lemma A3,  $\|V\|_{2 \rightarrow \infty} \leq \sqrt{\frac{1}{\lambda_K(\Pi_c' \Pi_c)}}$ , which gives  $\|\hat{V} \text{sgn}(H_{\hat{V}}) - V\|_{2 \rightarrow \infty} + \|V\|_{2 \rightarrow \infty} = O\left(\sqrt{\frac{1}{\lambda_K(\Pi_c' \Pi_c)}}\right)$  (this can be seen as simply using  $\|V\|_{2 \rightarrow \infty}$  to estimate  $\|\hat{V}\|_{2 \rightarrow \infty}$  since  $\sqrt{\frac{1}{\lambda_K(\Pi_c' \Pi_c)}}$  is the same order as

- $\frac{\sqrt{\theta_{r,\max}K\log(n_r+n_c)}}{\theta_{r,\min}\sigma_K(P)\sigma_K(\Pi_r)\sigma_K(\Pi_c)}$ ). Then, we have  $\|\hat{J}_*\| = O(\theta_{r,\max}\sigma_1(\Pi_r)\kappa(\Pi_c))$ , which gives that  $\|\hat{J}_*\|_F = O(\theta_{r,\max}\sqrt{K}\sigma_1(\Pi_r)\kappa(\Pi_c))$ .
- for  $\|\hat{J}_* - \mathcal{P}'_r J_* \mathcal{P}_r\|_F$ , since  $J_* = N_U(\mathcal{I}_r, \mathcal{I}_r)\Theta_r(\mathcal{I}_r, \mathcal{I}_r)$ , we have  $\|J_*\| \leq N_{U,\max}\theta_{r,\max} \leq \frac{\theta_{r,\max}^2\sqrt{K\lambda_1(\Pi'_r\Pi_r)}}{\theta_{r,\min}}$ , which gives that  $\|J_*\|_F \leq \frac{\theta_{r,\max}^2K\sigma_1(\Pi_r)}{\theta_{r,\min}}$ . Thus, we have  $\|\hat{J}_* - \mathcal{P}'_r J_* \mathcal{P}_r\|_F = O(\frac{\theta_{r,\max}^2K\sigma_1(\Pi_r)}{\theta_{r,\min}})$ .

Combining the above results, we have

$$\begin{aligned} \|e'_i(\hat{Z}_r - Z_r \mathcal{P}_r)\|_F &\leq \|e'_i(\hat{Y}_* - Y_* \mathcal{P}_r)\|_F \|\hat{J}_*\|_F + \|e'_i Y_*\|_F \|\hat{J}_* - \mathcal{P}'_r J_* \mathcal{P}_r\|_F \\ &= O\left(\frac{K^{4.5}\theta_{r,\max}^{14}\omega\kappa^{4.5}(\Pi'_r\Pi_r)\lambda_1(\Pi'_r\Pi_r)}{\theta_{r,\min}^{14}\pi_{r,\min}}\right)O(\theta_{r,\max}\sqrt{K}\sigma_1(\Pi_r)\kappa(\Pi_c)) \\ &\quad + \frac{\theta_{r,\max}^2\sqrt{K\lambda_1(\Pi'_r\Pi_r)}}{\theta_{r,\min}^2\lambda_K(\Pi_r\Pi_r)}O\left(\frac{\theta_{r,\max}^2K\sigma_1(\Pi_r)}{\theta_{r,\min}}\right) = O\left(\frac{K^5\theta_{r,\max}^{15}\omega\kappa^{4.5}(\Pi'_r\Pi_r)\kappa(\Pi_c)\lambda_1^{1.5}(\Pi'_r\Pi_r)}{\theta_{r,\min}^{14}\pi_{r,\min}}\right). \end{aligned}$$

□

Appendix E.1. Proof of Theorem 2

**Proof.** We bound  $\|e'_j(\hat{\Pi}_c - \Pi_c \mathcal{P}_c)\|_1$  first. Recall that  $Z_c = \Pi_c, \Pi_c(j, \cdot) = \frac{Z_c(j, \cdot)}{\|Z_c(j, \cdot)\|_1}, \hat{\Pi}_c(i, \cdot) = \frac{\hat{Z}_c(i, \cdot)}{\|\hat{Z}_c(i, \cdot)\|_1}$ , for  $1 \leq j \leq n_c$ , since

$$\begin{aligned} \|e'_j(\hat{\Pi}_c - \Pi_c \mathcal{P}_c)\|_1 &= \left\| \frac{e'_j \hat{Z}_c}{\|e'_j \hat{Z}_c\|_1} - \frac{e'_j Z_c \mathcal{P}_c}{\|e'_j Z_c \mathcal{P}_c\|_1} \right\|_1 = \left\| \frac{e'_j \hat{Z}_c \|e'_j Z_c\|_1 - e'_j Z_c \mathcal{P}_c \|e'_j \hat{Z}_c\|_1}{\|e'_j \hat{Z}_c\|_1 \|e'_j Z_c\|_1} \right\|_1 \\ &= \left\| \frac{e'_j \hat{Z}_c \|e'_j Z_c\|_1 - e'_j \hat{Z}_c \|e'_j \hat{Z}_c\|_1 + e'_j \hat{Z}_c \|e'_j \hat{Z}_c\|_1 - e'_j Z_c \mathcal{P}_c \|e'_j \hat{Z}_c\|_1}{\|e'_j \hat{Z}_c\|_1 \|e'_j Z_c\|_1} \right\|_1 \\ &\leq \frac{\|e'_j \hat{Z}_c \|e'_j Z_c\|_1 - e'_j \hat{Z}_c \|e'_j \hat{Z}_c\|_1\|_1 + \|e'_j \hat{Z}_c \|e'_j \hat{Z}_c\|_1 - e'_j Z_c \mathcal{P}_c \|e'_j \hat{Z}_c\|_1\|_1}{\|e'_j \hat{Z}_c\|_1 \|e'_j Z_c\|_1} \\ &= \frac{\| \|e'_j Z_c\|_1 - \|e'_j \hat{Z}_c\|_1 \| + \|e'_j \hat{Z}_c - e'_j Z_c \mathcal{P}_c\|_1}{\|e'_j Z_c\|_1} \leq \frac{2\|e'_j(\hat{Z}_c - Z_c \mathcal{P}_c)\|_1}{\|e'_j Z_c\|_1} \\ &= \frac{2\|e'_j(\hat{Z}_c - Z_c \mathcal{P}_c)\|_1}{\|e'_j \Pi_c\|_1} = 2\|e'_j(\hat{Z}_c - Z_c \mathcal{P}_c)\|_1 \leq 2\sqrt{K}\|e'_j(\hat{Z}_c - Z_c \mathcal{P}_c)\|_F, \end{aligned}$$

by Lemma A9, we have

$$\|e'_j(\hat{\Pi}_c - \Pi_c \mathcal{P}_c)\|_1 = O(\sqrt{K}\|e'_j(\hat{Z}_c - Z_c \mathcal{P}_c)\|_F) = O(\omega K \kappa(\Pi'_c \Pi_c) \sqrt{\lambda_1(\Pi'_c \Pi_c)}).$$

For row nodes  $1 \leq i \leq n_r$ , recall that  $Z_r = Y_* J_* \equiv N_U^{-1} N_M \Pi_r, \hat{Z}_r = \hat{Y}_* \hat{J}_*, \Pi_r(i, \cdot) = \frac{Z_r(i, \cdot)}{\|Z_r(i, \cdot)\|_1}$  and  $\hat{\Pi}_r(i, \cdot) = \frac{\hat{Z}_r(i, \cdot)}{\|\hat{Z}_r(i, \cdot)\|_1}$ , where  $N_M$  and  $M$  are defined in the proof of Lemma 1 such that  $U = \Theta_r M \equiv \Theta_r \Pi_r B_r$  and  $N_M(i, i) = \frac{1}{\|M(i, \cdot)\|_F}$ , similar as the proof for column nodes, we have

$$\|e'_i(\hat{\Pi}_r - \Pi_r \mathcal{P}_r)\|_1 \leq \frac{2\|e'_i(\hat{Z}_r - Z_r \mathcal{P}_r)\|_1}{\|e'_i Z_r\|_1} \leq \frac{2\sqrt{K}\|e'_i(\hat{Z}_r - Z_r \mathcal{P}_r)\|_F}{\|e'_i Z_r\|_1}.$$

Now, we provide a lower bound of  $\|e'_i Z_r\|_1$  as below

$$\|e'_i Z_r\|_1 = \|e'_i N_U^{-1} N_M \Pi_r\|_1 = \|N_U^{-1}(i, i) e'_i N_M \Pi_r\|_1 = N_U^{-1}(i, i) \|N_M(i, i) e'_i \Pi_r\|_1 = \frac{N_M(i, i)}{N_U(i, i)}$$

$$\begin{aligned}
 &= \|U(i, :)\|_F N_M(i, i) = \|U(i, :)\|_F \frac{1}{\|M(i, :)\|_F} = \|U(i, :)\|_F \frac{1}{\|e'_i M\|_F} \\
 &= \|U(i, :)\|_F \frac{1}{\|e'_i \Theta_r^{-1} U\|_F} = \|U(i, :)\|_F \frac{1}{\|\Theta_r^{-1}(i, i) e'_i U\|_F} = \theta_r(i) \geq \theta_{r, \min}.
 \end{aligned}$$

Therefore, by Lemma A9, we have

$$\begin{aligned}
 \|e'_i(\hat{\Pi}_r - \Pi_r \mathcal{P}_r)\|_1 &\leq \frac{2\sqrt{K} \|e'_i(\hat{Z}_r - Z_r \mathcal{P}_r)\|_F}{\|e'_i Z_r\|_1} \leq \frac{2\sqrt{K} \|e'_i(\hat{Z}_r - Z_r \mathcal{P}_r)\|_F}{\theta_{r, \min}} \\
 &= O\left(\frac{K^{5.5} \theta_{r, \max}^{15} \omega \kappa^{4.5} (\Pi'_r \Pi_r) \kappa(\Pi_c) \lambda_1^{1.5} (\Pi'_r \Pi_r)}{\theta_{r, \min}^{15} \pi_{r, \min}}\right).
 \end{aligned}$$

□

Appendix E.2. Proof of Corollary 1

**Proof.** Under conditions of Corollary 1, we have

$$\begin{aligned}
 \|e'_i(\hat{\Pi}_r - \Pi_r \mathcal{P}_r)\|_1 &= O\left(\frac{K^{5.5} \theta_{r, \max}^{15} \omega \kappa^{4.5} (\Pi'_r \Pi_r) \kappa(\Pi_c) \lambda_1^{1.5} (\Pi'_r \Pi_r)}{\theta_{r, \min}^{15} \pi_{r, \min}}\right) = O\left(\frac{\theta_{r, \max}^{15} \omega \sqrt{n_r}}{\theta_{r, \min}^{15}}\right), \\
 \|e'_j(\hat{\Pi}_c - \Pi_c \mathcal{P}_c)\|_1 &= O(\omega K \kappa(\Pi'_c \Pi_c) \sqrt{\lambda_1(\Pi'_c \Pi_c)}) = O(\omega \sqrt{n_c}).
 \end{aligned}$$

Under conditions of Corollary 1, Lemma A7 gives  $\omega = O\left(\frac{\sqrt{P_{\max} \theta_{r, \max} \log(n_r + n_c)}}{\theta_{r, \min} \sigma_K(P) \sqrt{n_r n_c}}\right)$ , which gives that

$$\begin{aligned}
 \|e'_i(\hat{\Pi}_r - \Pi_r \mathcal{P}_r)\|_1 &= O\left(\frac{\theta_{r, \max}^{15} \omega \sqrt{n_r}}{\theta_{r, \min}^{15}}\right) = O\left(\frac{\theta_{r, \max}^{15.5} \sqrt{P_{\max} \log(n_r + n_c)}}{\theta_{r, \min}^{16} \sigma_K(P) \sqrt{n_c}}\right), \\
 \|e'_j(\hat{\Pi}_c - \Pi_c \mathcal{P}_c)\|_1 &= O(\omega \sqrt{n_c}) = O\left(\frac{\sqrt{P_{\max} \theta_{r, \max} \log(n_r + n_c)}}{\theta_{r, \min} \sigma_K(P) \sqrt{n_r}}\right).
 \end{aligned}$$

By basic algebra, this corollary follows. □

Appendix F. SVM-Cone Algorithm

For readers' convenience, we briefly introduce the SVM-cone algorithm given in [26] and provide another view that the SVM-cone algorithm exactly recovers  $\Pi_r$  when the input is  $U_*$  (or  $U_{*,2}$ ). Let  $S$  be a matrix whose rows have unit  $l_2$  norm, and  $S$  can be written as  $S = H S_C$ , where  $H \in \mathbb{R}^{n \times K}$  with nonnegative entries, no row of  $H$  is 0, and  $S_C \in \mathbb{R}^{K \times m}$  corresponding to  $K$  rows of  $S$  (i.e., there exists an index set  $\mathcal{I}$  with  $K$  entries such that  $S_C = S(\mathcal{I}, :)$ ). Inferring  $H$  from  $S$  is called the ideal cone problem, i.e., Problem 1 in [26]. The ideal cone problem can be solved by applying one-class SVM to the rows of  $S$ , and the  $K$  rows of  $S_C$  are the support vectors found by one-class SVM:

$$\text{maximize } b \text{ s.t. } \mathbf{w}' S(i, :) \geq b \text{ (for } i = 1, 2, \dots, n) \text{ and } \|\mathbf{w}\|_F \leq 1. \tag{A5}$$

The solution  $(\mathbf{w}, b)$  for the ideal cone problem when  $(S_C S'_C)^{-1} \mathbf{1} > 0$  is given by

$$\mathbf{w} = b^{-1} \cdot S'_C \frac{(S_C S'_C)^{-1} \mathbf{1}}{\mathbf{1}' (S_C S'_C)^{-1} \mathbf{1}}, \quad b = \frac{1}{\sqrt{\mathbf{1}' (S_C S'_C)^{-1} \mathbf{1}}}. \tag{A6}$$

For the empirical case, let  $\hat{S} \in \mathbb{R}^{n \times m}$  be a matrix where all rows have unit  $l_2$  norm, infer  $H$  from  $\hat{S}$  with given  $K$  is called the empirical cone problem, i.e., Problem 2 in [26]. For the empirical cone problem, one-class SVM is applied to all rows of  $\hat{S}$  to obtain  $\mathbf{w}$  and  $b$ 's estimations  $\hat{\mathbf{w}}$  and  $\hat{b}$ . Then, apply the K-means algorithm to rows of  $\hat{S}$  that are close to the

hyperplane into  $K$  clusters, and an estimation of the index set  $\mathcal{I}$  can be obtained from the  $K$  clusters provided. Algorithm A3 below is the SVM-cone algorithm provided in [26].

**Algorithm A3 SVM-cone [26]**

- Require:**  $\hat{S} \in \mathbb{R}^{n \times m}$  with rows having unit  $l_2$  norm, number of corners  $K$ , estimated distance corners from hyperplane  $\gamma$ .  
**Ensure:** The near-corner index set  $\hat{\mathcal{I}}$ .
- 1: Run one-class SVM on  $\hat{S}(i, :)$  to obtain  $\hat{\mathbf{w}}$  and  $\hat{b}$ .
  - 2: Run K-means algorithm to the set  $\{\hat{S}(i, :)|\hat{S}(i, :)\hat{\mathbf{w}} \leq \hat{b} + \gamma\}$  that are close to the hyperplane into  $K$  clusters.
  - 3: Pick one point from each cluster to obtain the near-corner set  $\hat{\mathcal{I}}$ .

As suggested in [26], we can start  $\gamma = 0$  and incrementally increase it until  $K$  distinct clusters are found.

Now, turn to our DiMSC algorithm, and focus on estimating  $\mathcal{I}_r$  with given  $U_*, U_{*,2}$ , and  $K$ . By Lemmas 1 and A1, we know that  $U_*$  and  $U_{*,2}$  enjoy the ideal cone structure, and Lemma 2 guarantees that one-class SVM can be applied to rows of  $U_*$  and  $U_{*,2}$ . Set  $\mathbf{w}_1 = b_1^{-1}U'_*(\mathcal{I}_r, :)\frac{(U_*(\mathcal{I}_r, :)\mathbf{1})^{-1}}{\mathbf{1}'(U_*(\mathcal{I}_r, :)\mathbf{1})^{-1}}$ ,  $b_1 = \frac{1}{\sqrt{\mathbf{1}'(U_*(\mathcal{I}_r, :)\mathbf{1})^{-1}}}$ , and  $\mathbf{w}_2 = b_2^{-1}U'_{*,2}(\mathcal{I}_r, :)\frac{(U_{*,2}(\mathcal{I}_r, :)\mathbf{1})^{-1}}{\mathbf{1}'(U_{*,2}(\mathcal{I}_r, :)\mathbf{1})^{-1}}$ ,  $b_2 = \frac{1}{\sqrt{\mathbf{1}'(U_{*,2}(\mathcal{I}_r, :)\mathbf{1})^{-1}}}$ . Now that  $\mathbf{w}_1$  and  $b_1$  are solutions of the one-class SVM in Equation (A5) by setting  $S = U_*$ , and  $\mathbf{w}_2$  and  $b_2$  are solutions of the one-class SVM in Equation (A5) by setting  $S = U_{*,2}$ . Lemma A11 says that if row node  $i$  is a pure node, we have  $U_*(i, :)\mathbf{w}_1 = b_1$ , and this suggests that in the SVM-cone algorithm, if the input matrix is  $U_*$ , by setting  $\gamma = 0$ , we can find all pure row nodes, i.e., the set  $\{U_*(i, :)|U_*(i, :)\mathbf{w}_1 = b_1\}$  contains all rows of  $U_*$  respective to pure row nodes while including mixed row nodes. By Lemma 1, these pure row nodes belong to the  $K$  distinct row communities such that if row nodes  $i, \bar{i}$  are in the same row community, then we have  $U_*(i, :) = U_*(\bar{i}, :)$ , and this is the reason that we need to apply the K-means algorithm on the set obtained in Step 2 in the SVM-cone algorithm to obtain the  $K$  distinct row communities, and this is also the reason that we said the SVM-cone algorithm returns the index set  $\mathcal{I}$  exactly when the input is  $U_*$ . These conclusions also hold when we set the input in the SVM-cone algorithm as  $U_{*,2}$ .

**Lemma A10.** Under  $DiDCMM_{n_r, n_c}(K, P, \Pi_r, \Pi_c, \Theta_r)$ , for  $1 \leq i \leq n_r$ ,  $U_*(i, :)$  can be written as  $U_*(i, :) = r_1(i)\Phi_1(i, :)\mathbf{1}U_*(\mathcal{I}_r, :)$ , where  $r_1(i) \geq 1$ . Meanwhile,  $r_1(i) = 1$  and  $\Phi_1(i, :) = e'_k$  if  $i$  is a pure node such that  $\Pi_r(i, k) = 1$ ;  $r_1(i) > 1$  and  $\Phi_1(i, :) \neq e'_k$  if  $\Pi_r(i, k) < 1$  for  $1 \leq k \leq K$ . Similarly,  $U_{*,2}(i, :)$  can be written as  $U_{*,2}(i, :) = r_2(i)\Phi_2(i, :)\mathbf{1}U_{*,2}(\mathcal{I}_r, :)$ , where  $r_2(i) \geq 1$ . Meanwhile,  $r_2(i) = 1$  and  $\Phi_2(i, :) = e'_k$  if  $\Pi_r(i, k) = 1$ ;  $r_2(i) > 1$  and  $\Phi_2(i, :) \neq e'_k$  if  $\Pi_r(i, k) < 1$  for  $1 \leq k \leq K$ .

**Proof.** Since  $U_* = YU_*(\mathcal{I}_r, :)$  by Lemma 1, for  $1 \leq i \leq n_r$ , we have

$$U_*(i, :) = Y(i, :)\mathbf{1}U_*(\mathcal{I}_r, :) = Y(i, :)\mathbf{1}\frac{Y(i, :)}{Y(i, :)\mathbf{1}}U_*(\mathcal{I}_r, :) = r_1(i)\Phi_1(i, :)\mathbf{1}U_*(\mathcal{I}_r, :),$$

where we set  $r_1(i) = Y(i, :)\mathbf{1}$ ,  $\Phi_1(i, :) = \frac{Y(i, :)}{Y(i, :)\mathbf{1}}$ , and  $\mathbf{1}$  is a  $K \times 1$  vector with all entries being ones.

By the proof of Lemma 1,  $Y(i, :) = \frac{\Pi_r(i, :)}{\|M(i, :)\|_F}\Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)N_U^{-1}(\mathcal{I}_r, \mathcal{I}_r)$ , where  $M = \Pi_r\Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :)$ . For convenience, set  $T = \Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)$ ,  $Q = N_U^{-1}(\mathcal{I}_r, \mathcal{I}_r)$ , and  $R = U(\mathcal{I}_r, :)$  (such setting of  $T, Q, R$  is only for used for notation convenience for the proof of Lemma A10).

On the one hand, if row node  $i$  is pure such that  $\Pi_r(i, k) = 1$  for certain  $k$  among  $\{1, 2, \dots, K\}$  (i.e.,  $\Pi_r(i, :) = e_k$  if  $\Pi_r(i, k) = 1$ ), we have  $M(i, :) = \Pi_r(i, :)\Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :)$

) =  $T(k, k)R(k, :)$ , and  $\Pi_r(i, :)TQ = T(k, k)Q(k, :)$ , which give that  $Y(i, :) = \frac{T(k, k)Q(k, :)}{\|T(k, k)R(k, :)\|_F} = \frac{Q(k, :)}{\|R(k, :)\|_F}$ . Recall that the  $k$ -th diagonal entry of  $N_U^{-1}(\mathcal{I}_r, \mathcal{I}_r)$  is  $\|U(\mathcal{I}_r, :)\|_F(k, :)$ , i.e.,  $Q(k, :)\mathbf{1} = \|R(k, :)\|_F$ , which gives that  $r_1(i) = Y(i, :)\mathbf{1} = 1$  and  $\Phi_1(i, :) = e'_k$  when  $\Pi_r(i, k) = 1$ .

On the other hand, if  $i$  is a mixed node, since  $\|M(i, :)\|_F = \|\Pi_r(i, :)\Theta_r^{-1}(\mathcal{I}_r, \mathcal{I}_r)U(\mathcal{I}_r, :)\|_F = \|\sum_{k=1}^K \Pi_r(i, k)T(k, k)R(k, :)\|_F < \sum_{k=1}^K \Pi_r(i, k)T(k, k)\|R(k, :)\|_F = \sum_{k=1}^K \Pi_r(i, k)T(k, k)Q(k, k)$ , combine it with  $\Pi_r(i, :)TQ\mathbf{1} = \sum_{k=1}^K \Pi_r(i, k)T(k, k)Q(k, k)$ , so  $r_1(i) = Y(i, :)\mathbf{1} = \frac{\Pi_r(i, :)\mathbf{1}TQ\mathbf{1}}{\|M(i, :)\|_F} > 1$ . The lemma follows by a similar analysis for  $U_{*,2}$ .  $\square$

**Lemma A11.** Under  $DiDCMM_{n_r, n_c}(K, P, \Pi_r, \Pi_c, \Theta_r)$ , for  $1 \leq i \leq n_r$ , if row node  $i$  is a pure node such that  $\Pi_r(i, k) = 1$  for certain  $k$ , we have

$$U_*(i, :)\mathbf{w}_1 = b_1 \quad \text{and} \quad U_{*,2}(i, :)\mathbf{w}_2 = b_2,$$

Meanwhile, if row node  $i$  is a mixed node, the above equalities do not hold.

**Proof.** For the claim that  $U_*(i, :)\mathbf{w}_1 = b_1$  holds when  $i$  is pure, by Lemma A10, when  $i$  is a pure node such that  $\Pi_r(i, k) = 1$ ,  $U_*(i, :)$  can be written as  $U_*(i, :) = e'_k U_*(\mathcal{I}_r, :)$ , so  $U_*(i, :)\mathbf{w}_1 = b_1$  holds surely. When  $i$  is a mixed node, by Lemma A10,  $r_1(i) > 1$  and  $\Phi_1(i, :) \neq e'_k$  for any  $k = 1, 2, \dots, K$ ; hence  $U_*(i, :) \neq e'_k U_*(\mathcal{I}_r, :)$  if  $i$  is mixed, which gives the result. Follow a similar analysis, we obtain the results associated with  $U_{*,2}$ , and the lemma follows.  $\square$

## References

- Fortunato, S. Community detection in graphs. *Phys. Rep.* **2010**, *486*, 75–174. [\[CrossRef\]](#)
- Fortunato, S.; Hric, D. Community detection in networks: A user guide. *Phys. Rep.* **2016**, *659*, 1–44. [\[CrossRef\]](#)
- Holland, P.W.; Laskey, K.B.; Leinhardt, S. Stochastic blockmodels: First steps. *Soc. Netw.* **1983**, *5*, 109–137. [\[CrossRef\]](#)
- Karrer, B.; Newman, M.E.J. Stochastic blockmodels and community structure in networks. *Phys. Rev. E* **2011**, *83*, 16107. [\[CrossRef\]](#)
- Rohe, K.; Chatterjee, S.; Yu, B. Spectral clustering and the high-dimensional stochastic blockmodel. *Ann. Stat.* **2011**, *39*, 1878–1915. [\[CrossRef\]](#)
- Zhao, Y.; Levina, E.; Zhu, J. Consistency of community detection in networks under degree-corrected stochastic block models. *Ann. Stat.* **2012**, *40*, 2266–2292. [\[CrossRef\]](#)
- Qin, T.; Rohe, K. Regularized spectral clustering under the degree-corrected stochastic blockmodel. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 3120–3128.
- Jin, J. Fast community detection by SCORE. *Ann. Stat.* **2015**, *43*, 57–89. [\[CrossRef\]](#)
- Lei, J.; Rinaldo, A. Consistency of spectral clustering in stochastic block models. *Ann. Stat.* **2015**, *43*, 215–237. [\[CrossRef\]](#)
- Cai, T.T.; Li, X. Robust and computationally feasible community detection in the presence of arbitrary outlier nodes. *Ann. Stat.* **2015**, *43*, 1027–1059. [\[CrossRef\]](#)
- Joseph, A.; Yu, B. Impact of regularization on spectral clustering. *Ann. Stat.* **2016**, *44*, 1765–1791. [\[CrossRef\]](#)
- Chen, Y.; Li, X.; Xu, J. Convexified modularity maximization for degree-corrected stochastic block models. *Ann. Stat.* **2018**, *46*, 1573–1602. [\[CrossRef\]](#)
- Passino, F.S.; Heard, N.A. Bayesian estimation of the latent dimension and communities in stochastic blockmodels. *Stat. Comput.* **2020**, *30*, 1291–1307. [\[CrossRef\]](#)
- Li, X.; Chen, Y.; Xu, J. Convex Relaxation Methods for Community Detection. *Stat. Sci.* **2021**, *36*, 2–15. [\[CrossRef\]](#)
- Jing, B.; Li, T.; Ying, N.; Yu, X. Community detection in sparse networks using the symmetrized Laplacian inverse matrix (SLIM). *Stat. Sin.* **2022**, *32*, 1–22. [\[CrossRef\]](#)
- Abbe, E. Community detection and stochastic block models: Recent developments. *J. Mach. Learn. Res.* **2017**, *18*, 6446–6531.
- Airoldi, E.M.; Blei, D.M.; Fienberg, S.E.; Xing, E.P. Mixed Membership Stochastic Blockmodels. *J. Mach. Learn. Res.* **2008**, *9*, 1981–2014.
- Ball, B.; Karrer, B.; Newman, M.E.J. Efficient and principled method for detecting communities in networks. *Phys. Rev. E* **2011**, *84*, 36103. [\[CrossRef\]](#)
- Wang, F.; Li, T.; Wang, X.; Zhu, S.; Ding, C. Community discovery using nonnegative matrix factorization. *Data Min. Knowl. Discov.* **2011**, *22*, 493–521. [\[CrossRef\]](#)
- Gopalan, P.K.; Blei, D.M. Efficient discovery of overlapping communities in massive networks. *Proc. Natl. Acad. Sci. USA* **2013**, *110*, 14534–14539. [\[CrossRef\]](#)
- Anandkumar, A.; Ge, R.; Hsu, D.; Kakade, S.M. A tensor approach to learning mixed membership community models. *J. Mach. Learn. Res.* **2014**, *15*, 2239–2312.

22. Kaufmann, E.; Bonald, T.; Lelarge, M. A spectral algorithm with additive clustering for the recovery of overlapping communities in networks. *Theor. Comput. Sci.* **2017**, *742*, 3–26. [[CrossRef](#)]
23. Panov, M.; Slavnov, K.; Ushakov, R. Consistent estimation of mixed memberships with successive projections. In Proceedings of the International Conference on Complex Networks and their Applications, Lyon, France, 29 November–1 December 2017; Springer: Cham, Switzerland, 2017; pp. 53–64.
24. Jin, J.; Ke, Z.T.; Luo, S. Mixed membership estimation for social networks. *arXiv* **2017**, arXiv:1708.07852.
25. Mao, X.; Sarkar, P.; Chakrabarti, D. On Mixed Memberships and Symmetric Nonnegative Matrix Factorizations. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 2324–2333.
26. Mao, X.; Sarkar, P.; Chakrabarti, D. Overlapping Clustering Models, and One (class) SVM to Bind Them All. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; Volume 31, pp. 2126–2136.
27. Mao, X.; Sarkar, P.; Chakrabarti, D. Estimating Mixed Memberships With Sharp Eigenvector Deviations. *J. Am. Stat. Assoc.* **2020**, *116*, 1928–1940. [[CrossRef](#)]
28. Wang, Y.; Bu, Z.; Yang, H.; Li, H.J.; Cao, J. An effective and scalable overlapping community detection approach: Integrating social identity model and game theory. *Appl. Math. Comput.* **2021**, *390*, 125601. [[CrossRef](#)]
29. Zhang, Y.; Levina, E.; Zhu, J. Detecting Overlapping Communities in Networks Using Spectral Methods. *SIAM J. Math. Data Sci.* **2020**, *2*, 265–283. [[CrossRef](#)]
30. Rohe, K.; Qin, T.; Yu, B. Co-clustering directed graphs to discover asymmetries and directional communities. *Proc. Natl. Acad. Sci. USA* **2016**, *113*, 12679–12684. [[CrossRef](#)]
31. Wang, Z.; Liang, Y.; Ji, P. Spectral Algorithms for Community Detection in Directed Networks. *J. Mach. Learn. Res.* **2020**, *21*, 1–45.
32. Ji, P.; Jin, J. Coauthorship and citation networks for statisticians. *Ann. Appl. Stat.* **2016**, *10*, 1779–1812. [[CrossRef](#)]
33. Zhou, Z.; Amini, A.A. Analysis of spectral clustering algorithms for community detection: The general bipartite setting. *J. Mach. Learn. Res.* **2019**, *20*, 1–47.
34. Laenen, S.; Sun, H. Higher-order spectral clustering of directed graphs. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 941–951.
35. Qing, H.; Wang, J. Directed mixed membership stochastic blockmodel. *arXiv* **2021**, arXiv:2101.02307.
36. Wang, Y.J.; Wong, G.Y. Stochastic Blockmodels for Directed Graphs. *J. Am. Stat. Assoc.* **1987**, *82*, 8–19. [[CrossRef](#)]
37. Fagiolo, G. Clustering in complex directed networks. *Phys. Rev. E* **2007**, *76*, 026107. [[CrossRef](#)] [[PubMed](#)]
38. Leicht, E.A.; Newman, M.E. Community structure in directed networks. *Phys. Rev. Lett.* **2008**, *100*, 118703. [[CrossRef](#)] [[PubMed](#)]
39. Kim, Y.; Son, S.W.; Jeong, H. Finding communities in directed networks. *Phys. Rev. E* **2010**, *81*, 016103. [[CrossRef](#)]
40. Malliaros, F.D.; Vazirgiannis, M. Clustering and Community Detection in Directed Networks: A Survey. *Phys. Rep.* **2013**, *533*, 95–142. [[CrossRef](#)]
41. Zhang, X.; Lian, B.; Lewis, F.L.; Wan, Y.; Cheng, D. Directed Graph Clustering Algorithms, Topology, and Weak Links. *IEEE Trans. Syst. Man, Cybern. Syst.* **2021**, *52*, 3995–4009. [[CrossRef](#)]
42. Zhang, J.; Wang, J. Identifiability and parameter estimation of the overlapped stochastic co-block model. *Stat. Comput.* **2022**, *32*, 1–14. [[CrossRef](#)]
43. Florescu, L.; Perkins, W. Spectral thresholds in the bipartite stochastic block model. In Proceedings of the Conference on Learning Theory. PMLR, New York, NY, USA, 23–26 June 2016; pp. 943–959.
44. Neumann, S. Bipartite stochastic block models with tiny clusters. In 32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, QC, Canada, 3–8 December 2018.
45. Ndaoud, M.; Sigalla, S.; Tsybakov, A.B. Improved clustering algorithms for the bipartite stochastic block model. *IEEE Trans. Inf. Theory* **2021**, *68*, 1960–1975. [[CrossRef](#)]
46. Mantzaris, A.V. Uncovering nodes that spread information between communities in social networks. *EPJ Data Sci.* **2014**, *3*, 1–17. [[CrossRef](#)]
47. McSherry, F. Spectral partitioning of random graphs. In Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science, Newport Beach, CA, USA, 8–11 October 2001; pp. 529–537.
48. Massoulié, L. Community detection thresholds and the weak Ramanujan property. In Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing, New York, NY, USA, 31 May–3 June 2014; pp. 694–703.
49. Mossel, E.; Neeman, J.; Sly, A. Reconstruction and estimation in the planted partition model. *Probab. Theory Relat. Fields* **2015**, *162*, 431–461. [[CrossRef](#)]
50. Abbe, E.; Bandeira, A.S.; Hall, G. Exact recovery in the stochastic block model. *IEEE Trans. Inf. Theory* **2015**, *62*, 471–487. [[CrossRef](#)]
51. Hajek, B.; Wu, Y.; Xu, J. Achieving exact cluster recovery threshold via semidefinite programming. *IEEE Trans. Inf. Theory* **2016**, *62*, 2788–2797. [[CrossRef](#)]
52. Mossel, E.; Neeman, J.; Sly, A. A proof of the block model threshold conjecture. *Combinatorica* **2018**, *38*, 665–708. [[CrossRef](#)]
53. Qing, H. Studying Asymmetric Structure in Directed Networks by Overlapping and Non-Overlapping Models. *Entropy* **2022**, *24*, 1216. [[CrossRef](#)]
54. Gillis, N.; Vavasis, S.A. Semidefinite Programming Based Preconditioning for More Robust Near-Separable Nonnegative Matrix Factorization. *SIAM J. Optim.* **2015**, *25*, 677–698. [[CrossRef](#)]
55. Qing, H. A useful criterion on studying consistent estimation in community detection. *Entropy* **2022**, *24*, 1098. [[CrossRef](#)] [[PubMed](#)]

56. Von Luxburg, U. A tutorial on spectral clustering. *Stat. Comput.* **2007**, *17*, 395–416. [[CrossRef](#)]
57. Ke, Z.T.; Jin, J. The SCORE normalization, especially for highly heterogeneous network and text data. *arXiv* **2022**, arXiv:2204.11097.
58. Newman, M.E. Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA* **2006**, *103*, 8577–8582. [[CrossRef](#)] [[PubMed](#)]
59. Chang, C.C.; Lin, C.J. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol. (Tist)* **2011**, *2*, 1–27. [[CrossRef](#)]
60. Xu, R.; Wunsch, D. Survey of clustering algorithms. *IEEE Trans. Neural Netw.* **2005**, *16*, 645–678. [[CrossRef](#)] [[PubMed](#)]
61. Palmer, W.R.; Zheng, T. Spectral clustering for directed networks. In Proceedings of the International Conference on Complex Networks and Their Applications, Madrid, Spain, 1–3 December 2020; Springer: Cham, Switzerland, 2020; pp. 87–99.
62. Qing, H. Degree-corrected distribution-free model for community detection in weighted networks. *Sci. Rep.* **2022**, *12*, 15153. [[CrossRef](#)] [[PubMed](#)]
63. Erdős, P.; Rényi, A. On the evolution of random graphs. In *The Structure and Dynamics of Networks*; Princeton University Press: Princeton, NJ, USA, 2011; pp. 38–82. [[CrossRef](#)]
64. Chen, Y.; Chi, Y.; Fan, J.; Ma, C. Spectral Methods for Data Science: A Statistical Perspective. *Found. Trends Mach. Learn.* **2021**, *14*, 566–806. [[CrossRef](#)]
65. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [[CrossRef](#)]
66. Žalik, K.R.; Žalik, B. Memetic algorithm using node entropy and partition entropy for community detection in networks. *Inf. Sci.* **2018**, *445*, 38–49. [[CrossRef](#)]
67. Feutrill, A.; Roughan, M. A review of Shannon and differential entropy rate estimation. *Entropy* **2021**, *23*, 1046. [[CrossRef](#)]
68. Adamic, L.A.; Glance, N. The political blogosphere and the 2004 US election: Divided they blog. In Proceedings of the 3rd International Workshop on Link Discovery, Chicago, IL, USA, 21–25 August 2005; pp. 36–43.
69. Kunegis, J. Konect: The koblenz network collection. In Proceedings of the 22nd International Conference on World Wide Web, Rio de Janeiro, Brazil, 13–17 May 2013; pp. 1343–1350.
70. Zhang, H.; Guo, X.; Chang, X. Randomized spectral clustering in large-scale stochastic block models. *J. Comput. Graph. Stat.* **2022**, *31*, 887–906. [[CrossRef](#)]
71. Tropp, J.A. User-Friendly Tail Bounds for Sums of Random Matrices. *Found. Comput. Math.* **2012**, *12*, 389–434. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.