Flexible and Robust Multi-Network Clustering

Jingchao Ni [1], Hanghang Tong [2], Wei Fan [3], and Xiang Zhang [1]
[1]Department of Electrical Engineering and Computer Science, Case Western Reserve University
[2]School of Computing, Informatics, Decision Systems Engineering, Arizona State University
[3]Baidu Research Big Data Lab
[1]{jingchao.ni, xiang.zhang}@case.edu, [2]hanghang.tong@asu.edu,
[3]wei.fan@gmail.com

ABSTRACT

Integrating multiple graphs (or networks) has been shown to be a promising approach to improve the graph clustering accuracy. Various multi-view and multi-domain graph clustering methods have recently been developed to integrate multiple networks. In these methods, a network is treated as a view or domain. The key assumption is that there is a common clustering structure shared across all views (domains), and different views (domains) provide compatible and complementary information on this underlying clustering structure. However, in many emerging real-life applications, different networks have different data distributions, where the assumption that all networks share a single common clustering structure does not hold. In this paper, we propose a flexible and robust framework that allows multiple underlying clustering structures across different networks. Our method models the domain similarity as a network, which can be utilized to regularize the clustering structures in different networks. We refer to such a data model as a network of networks (NoN). We develop NoNClus, a novel method based on non-negative matrix factorization (NMF), to cluster an NoN. We provide rigorous theoretical analysis of NoNClus in terms of its correctness, convergence and complexity. Extensive experimental results on synthetic and real-life datasets show the effectiveness of our method.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data mining*

Keywords

Network of Networks; Graph clustering

1. INTRODUCTION

Graph (or network[1]) clustering is a fundamental problem with numerous applications. Traditional clustering methods are usually designed for a single network [29, 18, 34]. In many emerging real-life applications, networks collected from different conditions or domains are becoming available. For example, gene co-expression networks are being collected from different tissues of model organisms [24, 5, 26]; co-author networks can be constructed for different research areas [28]. Since a single network can be noisy and incomplete, a promising approach is to exploit the shared clustering structure in multiple networks to improve the accuracy of the results.

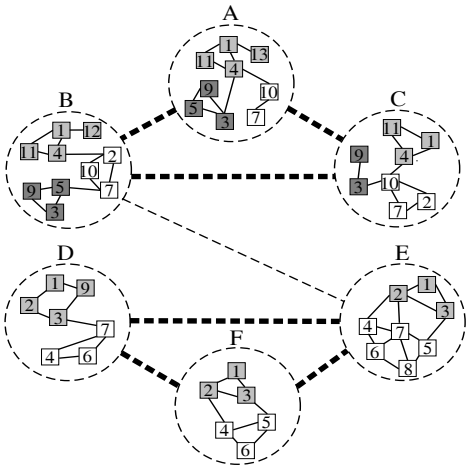


Figure 1: An example of NoN. The main network is represented by dashed nodes and edges. The domain-specific networks are represented by solid nodes and edges.

Several approaches have been recently developed to cluster multiple networks. One popular approach is multi-view clustering [37, 20, 19]. In this approach, a set of data objects may have multiple representations (views). Different views provide compatible and complementary information on the underlying data distribution. The existing multi-view graph clustering methods assume that all views consist of the same set of data objects and are generated from the same underlying distribution. Multi-domain graph clustering [8] generalizes multi-view clustering to allow many-to-many relationships between the nodes in different networks. Thus different networks may consist of different sets of nodes and have different sizes. Similar to multi-view graph clustering, this approach also assumes that there is a single underlying

[1]In this paper, we use graph and network interchangeably.

KDD'15, August 10-13, 2015, Sydney, NSW, Australia.

DOI: http://dx.doi.org/10.1145/2783258.2783262.

clustering structure shared across different domains. The ensemble clustering methods [30, 12, 13] aim at integrating multiple intermediate clustering results into a consensus one. The intermediate results can be obtained by applying the same clustering method on different views or different methods on the same view.

The key assumption of the existing multi-network clustering methods is that different networks share the same underlying clustering structure. However, this assumption may not hold in real-life applications. Figure 1 shows an example with six domains $\{A, B, C, D, E, F\}$, each of which corresponds to a network. Domains $\{A, B, C\}$ are similar to each other and so are domains $\{D, E, F\}$. But these two sets of domains are not similar to each other. Clearly, we cannot assume domain sets $\{A, B, C\}$ and $\{D, E, F\}$ share a common clustering structure.

Note that the similarity among different domains can also be modeled as a network (represented by dashed lines in Figure 1). We refer to the structure shown in Figure 1 as a *network of networks* (NoN). The dashed network represents the *main network* among six domains $\{A, B, C, D, E, F\}$. Each node in the main network corresponds to a *domain-specific network* represented by solid lines.

Consider an important bioinformatics problem: clustering gene co-expression networks [15, 24]. In a gene co-expression network, each node is a gene and an edge represents the functional association between two connected genes. To improve the clustering accuracy, we can utilize multiple gene co-expression networks collected in different tissues. Some tissues are similar to each other while others are not. The similarities among tissues can be modeled as a network. For example, in Figure 1, the main network of domains $\{A, B, C, D, E, F\}$ may represent the similarity among six different tissues. For each tissue, its domain-specific network represents the tissue-specific gene co-expression network. As another example, consider the co-author networks of different research areas. The main network may represent the similarity among different areas and a domain-specific network may represent the co-author network in a particular area.

The NoN setting illustrated in Figure 1 is different from the existing multi-view or multi-domain clustering scenario. In the NoN setting, there can be more than one underlying clustering structures among the domain-specific networks. In Figure 1, the clustering structure shared by domain-specific networks $\{A, B, C\}$ may be different from the one shared by $\{D, E, F\}$. For example, in domains $\{D, E, F\}$, nodes $\{1, 2, 3\}$ are very likely to be in the same cluster, but they are in three different clusters in domains $\{A, B, C\}$. Thus assuming one common clustering structure for all six domain-specific networks is not reasonable. Consider the previous tissue-specific gene co-expression network and area-specific co-author network examples. Given a set of tissue-specific gene co-expression networks, the same set of genes may form a cluster (e.g., a pathway or functional module) in several related tissues but not in others. Given a set of area-specific co-author networks, an author can be in different clusters (e.g., research sub-communities) in different areas.

In this paper, we propose NONCLUS, a robust and flexible multi-network clustering method that allows multiple underlying clustering structures. Our contributions are summarized as follows.

- We investigate a new clustering problem in the novel network setting, NoN, where multiple underlying clustering structures can co-exist among domain-specific networks. It generalizes the single clustering structure assumption of the existing multi-view and multi-domain clustering methods and has wider applicability in many emerging real-life problems.
- We develop a novel two phase clustering framework, NONCLUS, which can simultaneously cluster domain-specific networks with the guidance from the main network. NONCLUS allows partial mapping across different sized domain-specific networks, which is more general and realistic than the multi-view setting. We also provide rigorous theoretical analysis of NONCLUS in terms of its correctness, convergence and complexity.
- We perform extensive experiments on both synthetic and real-life datasets to evaluate the effectiveness of the proposed method.

The rest of the paper is organized as follows. Sec. 2 is the related work. Sec. 3 formulates the problem. Sec. 4 presents NONCLUS and its theoretical analysis. Sec. 5 presents the experimental evaluations. Sec. 6 gives concluding remarks.

2. RELATED WORK

The existing multi-network clustering methods are mostly developed for multi-view networks [37, 20, 19]. In multi-view clustering, views can be networks [37, 20, 19] or data-feature matrices [1, 23, 35]. Ensemble clustering [30, 12, 13] is related to multi-view clustering, where a consensus clustering is obtained by applying the same clustering algorithm on different views or applying different clustering algorithms on the same view. All these methods assume different views are compatible and share the same underlying clustering structure. Moreover, they are usually designed for views with the same set of data objects and the same number of clusters.

Several methods [33, 11, 22] extend the traditional multi-view clustering to allow incomplete views. The method in [33] requires at least one view to be complete. The method in [11] focuses on constrained clustering where a set of must-link and cannot-link constraints are needed. The two-view NMF based model proposed in [22] may suffer efficiency problem when applied to multi-view scenario due to its pairwise regularization between views. A recent work on multi-domain graph clustering [8] allows flexible network sizes and number of clusters. This method also uses pairwise regularization between domains thus may have efficiency problem. In contrast to [22] and [8], our method allows efficient regularization by centroid matrices. More importantly, the methods in [33, 11, 22, 8] have the same single underlying clustering structure assumption as other multi-view clustering methods do.

Tensor factorization methods [17, 16] can be applied to co-cluster multiple matrices [32, 14]. However, the existing tensor factorization methods, such as CP and Tucker decompositions [16], are not designed for graph data where two modes of the tensor are symmetric. Furthermore, when applied to cluster multi-view graphs, tensor factorization methods also have the limitation that all views must have the same size and share a single underlying clustering structure.

3. THE PROBLEM

We first introduce the definition of a network of networks. The main symbols used in this paper are listed in Table 1.

Table 1: Main symbols

Symbol	Meaning
$\mathbf{G}$	the $g \times g$ main network
$\mathbf{A}^{(i)}$	the i^{th} domain-specific network
$\mathbf{U}^{(i)}$	the factor matrix of $\mathbf{A}^{(i)}$
$\mathbf{V}^{(j)}$	the j^{th} hidden factor matrix
$\mathbf{O}^{(ij)}$	the mapping matrix between $\mathbf{U}^{(i)}$ and $\mathbf{V}^{(j)}$
$\mathbf{D}^{(ij)}$	the mapping matrix $\mathbf{D}^{(ij)} = \mathbf{O}^{(ij)}(\mathbf{O}^{(ij)})'$
g	the number of nodes in the main network
n_i	the number of nodes in $\mathbf{A}^{(i)}$
k	the number of main clusters
t_i	the number of domain clusters in $\mathbf{A}^{(i)}$
$\mathcal{A}$	domain-specific networks $\mathcal{A} = \{\mathbf{A}^{(1)}, ..., \mathbf{A}^{(g)}\}$
$\mathcal{V}^{(i)}$	the set of nodes in $\mathbf{A}^{(i)}$
$\mathcal{R}$	a network of networks $\mathcal{R} =< \mathbf{G}, \mathcal{A} >$

DEFINITION 1. *A* **Network of Networks (NoN)** *is defined as* $\mathcal{R} =< \mathbf{G}, \mathcal{A} >$*, where* $\mathbf{G}$ *is the* $g \times g$ **main** *network,* $\mathcal{A} = \{\mathbf{A}^{(1)}, ..., \mathbf{A}^{(g)}\}$ *is a set of* g **domain-specific** *networks. Node* i *(*$i = 1, ..., g$*) in the main network* $\mathbf{G}$ *corresponds to domain-specific network* $\mathbf{A}^{(i)}$*.*

We refer to the nodes in the main network and domain-specific networks as the *main nodes* and *domain nodes* respectively. We use $\mathcal{V}^{(i)}$ to represent the set of domain nodes in domain-specific network $\mathbf{A}^{(i)}$, and $\mathcal{I}^{(ij)}$ to represent the common nodes between $\mathbf{A}^{(i)}$ and $\mathbf{A}^{(j)}$, i.e., $\mathcal{I}^{(ij)} = \mathcal{V}^{(i)} \cap \mathcal{V}^{(j)}$.

For example, in Figure 1, the dashed network is the main network $\mathbf{G}$, which has six main nodes {*A, B, C, D, E, F*}. Each of these main nodes corresponds to a domain-specific network (the solid network). The main node *A* corresponds to the domain-specific network with nine domain nodes {*1, 3, 4, 5, 7, 9, 10, 11, 13*}. The common nodes between domain-specific networks of *A* and *B* are {*1, 3, 4, 5, 7, 9, 10, 11*}.

We refer to the clusters in the main network and domain-specific networks as the *main clusters* and *domain clusters*, respectively. For example, in Figure 1, there are two main clusters {*A, B, C*} and {*D, E, F*}. In the domain-specific network *B*, there are three domain clusters {*1, 4, 11, 12*}, {*2, 7, 10*} and {*3, 5, 9*}.

Our goal is to partition the domain-specific networks while respecting the clustering structure in the main network. More formally, let $\mathcal{H} = \{\mathcal{H}_1, ..., \mathcal{H}_k\}$ be a partition of the main network, we want to find $\mathcal{C} = \{\mathcal{C}^{(1)}, ..., \mathcal{C}^{(g)}\}$, the collection of partitions of all domain-specific networks, where $\mathcal{C}^{(i)}$ $(i = 1, ..., g)$ is a partition of domain-specific network $\mathbf{A}^{(i)}$, with respect to the main clusters in $\mathcal{H}$. Note that in this paper, we focus on finding non-overlapping clusters. This is also the goal of the existing multi-view and multi-domain graph clustering methods [37, 20, 19, 8, 33].

4. THE NONCLUS ALGORITHM

Our NONCLUS method clusters a NoN using a two-phase approach. In Phase I, we partition the main network. To partition the domain-specific networks, in Phase II, we develop a regularized non-negative matrix factorization (NMF) clustering method, which respects the clustering information obtained in Phase I.

4.1 Main Network Clustering

In Phase I, we treat the main network clustering problem as a single network clustering problem. We adopt the widely used non-negative matrix factorization (NMF) approach [21]. In particular, we use the symmetric version of NMF (SNMF) [10, 18] to partition the main network, which minimizes the following objective function

$$J_M = \|\mathbf{G} - \mathbf{H}\mathbf{H}'\|_F^2 \tag{1}$$

where $\|\cdot\|_F$ is the Frobenius norm and $\mathbf{H} \in \mathbb{R}_+^{g \times k}$ is the factor matrix of the main network. An element h_{ij} of $\mathbf{H}$ indicates to which degree main node i belongs to the j^{th} main cluster. We solve Eq. (1) using the method in [10]:

$$\mathbf{H} \leftarrow \mathbf{H} \circ \left(1 - \beta + \beta \frac{\mathbf{G}\mathbf{H}}{\mathbf{H}(\mathbf{H})'\mathbf{H}}\right) \tag{2}$$

$\circ$ and $\frac{[\cdot]}{[\cdot]}$ are element-wise operators and $0 \leq \beta \leq 1$ is a parameter which is suggested to be set to 0.5 in practice.

4.2 Domain-specific Network Clustering

In Phase II, we incorporate the main cluster information to cluster domain-specific networks. We first consider a simple case, where every domain-specific network has a set of n nodes and t clusters. Note that in general, different domain-specific networks may have different number of nodes and clusters.

4.2.1 The Simplified Case

For any domain node x, let $\mathbf{u}_{x*}^{(i)} \in \mathbb{R}_+^{1 \times t}$ $(i = 1, ..., g)$ represent its domain cluster assignment vector in $\mathbf{A}^{(i)}$. We assume that domain-specific networks in the same main cluster share a common underlying clustering structure. Since there are k main clusters $\{\mathcal{H}_1, ..., \mathcal{H}_k\}$, for domain node x, we introduce k *hidden* domain cluster assignment vectors $\mathbf{v}_{x*}^{(j)} \in \mathbb{R}_+^{1 \times t}$ $(j = 1, ..., k)$ to regularize $\mathbf{u}_{x*}^{(i)}$. If $\mathbf{A}^{(i)}$ belongs to main cluster $\mathcal{H}_j$, we want to minimize the cluster assignments inconsistency between $\mathbf{u}_{x*}^{(i)}$ and $\mathbf{v}_{x*}^{(j)}$, i.e., $\|\mathbf{u}_{x*}^{(i)} - \mathbf{v}_{x*}^{(j)}\|_F^2$.

Furthermore, recall that h_{ij} denotes the strength of the main cluster membership. For domain node x, we can collectively penalize the inconsistencies between its domain cluster assignment vectors and hidden domain cluster assignment vectors by minimizing

$$J_x = \sum_{i=1}^{g} \sum_{j=1}^{k} h_{ij} \|\mathbf{u}_{x*}^{(i)} - \mathbf{v}_{x*}^{(j)}\|_F^2 \tag{3}$$

Note that if two domain-specific networks $\mathbf{A}^{(p)}$ and $\mathbf{A}^{(q)}$ have high h_{pj} and h_{qj} values, i.e., they are likely to belong to the same main cluster $\mathcal{H}_j$, the inconsistency between cluster assignments $\mathbf{u}_{x*}^{(p)}$ and $\mathbf{u}_{x*}^{(q)}$ of node x will be penalized through $\mathbf{v}_{x*}^{(j)}$. This is intuitive since if $\mathbf{A}^{(p)}$ and $\mathbf{A}^{(q)}$ are in the same main cluster, the clustering structures of $\mathbf{A}^{(p)}$ and $\mathbf{A}^{(q)}$ should be similar.

Generalizing Eq. (3) to all domain nodes, we have the following objective function:

$$\min_{\substack{\mathbf{U}^{(i)}\geq 0\ (i=1,\ldots,g)\\ \mathbf{V}^{(j)}\geq 0\ (j=1,\ldots k)}} J_D = \underbrace{\sum_{i=1}^{g}\|\mathbf{A}^{(i)}-\mathbf{U}^{(i)}(\mathbf{U}^{(i)})'\|_F^2}_{domain-specific\ network\ clustering} + a\underbrace{\sum_{i=1}^{g}\sum_{j=1}^{k}h_{ij}\|\mathbf{U}^{(i)}-\mathbf{V}^{(j)}\|_F^2}_{main\ cluster\ guided\ regularization} \quad (4)$$

In Eq. (4), $\mathbf{U}^{(i)} \in \mathbb{R}_+^{n\times t}$ is the factor matrix of $\mathbf{A}^{(i)}$, and $\mathbf{V}^{(j)} \in \mathbb{R}_+^{n\times t}$ represents the underlying clustering structure of domain-specific networks in main cluster $\mathcal{H}_j$. In the next, we refer to $\mathbf{V}^{(j)}$ as the j^{th} *hidden factor matrix*.

4.2.2 The General Case

In general, the domain-specific networks may have different node sets and sizes. To generalize the basic model discussed in the previous section, we allow factor matrix $\mathbf{U}^{(i)}$ to have different number of rows (nodes) for different domain-specific networks. We further allow hidden factor matrix $\mathbf{V}^{(j)}$ to contain all nodes in the domain-specific networks that belong to main cluster $\mathcal{H}_j$. That is, the set of nodes in $\mathbf{V}^{(j)}$ is $\mathcal{V}_V^{(j)} = \bigcup \mathcal{V}^{(i)}$ $(i \in \mathcal{H}_j)$. Thus $\mathbf{V}^{(1)}, ..., \mathbf{V}^{(k)}$ also have different number of rows (nodes).

Furthermore, different domain-specific networks may share some common nodes. For example, a gene may be expressed in multiple tissues; a user may have accounts in multiple social networks. Let $n_i = |\mathcal{V}^{(i)}|$ and $\tilde{n}_j = |\mathcal{V}_V^{(j)}|$. We introduce mapping matrices $\mathbf{O}^{(ij)} \in \mathbb{R}_+^{n_i\times\tilde{n}_j}$, such that $\mathbf{O}^{(ij)}(x,y) = 1$ if the x^{th} row of $\mathbf{U}^{(i)}$ and the y^{th} row of $\mathbf{V}^{(j)}$ represent the same data object; $\mathbf{O}^{(ij)}(x,y) = 0$ otherwise. Note that each row of $\mathbf{O}^{(ij)}$ has at most one 1 because of the one-to-one relationship between the common nodes in different domain-specific networks.

Since not all nodes in $\mathbf{A}^{(i)}$ have corresponding rows in $\mathbf{V}^{(j)}$, we also introduce the diagonal mapping matrices $\mathbf{D}^{(ij)} \in \mathbb{R}_+^{n_i\times n_i}$, such that $\mathbf{D}^{(ij)}(x,x) = 1$ if the x^{th} row of $\mathbf{U}^{(i)}$ has a corresponding row in $\mathbf{V}^{(j)}$; $\mathbf{D}^{(ij)}(x,x) = 0$ otherwise. Note that $\mathbf{D}^{(ij)} = \mathbf{O}^{(ij)}(\mathbf{O}^{(ij)})'$.

Next, we further generalize our method to allow domain-specific networks to have different number of clusters. If h_{ij} is large, i.e., strong main cluster membership, we want the same rows in $\mathbf{D}^{(ij)}\mathbf{U}^{(i)}$ and $\mathbf{O}^{(ij)}\mathbf{V}^{(j)}$ to be similar, since they denote the cluster assignments for the common nodes. However, different number of clusters will result in different number of columns in $\mathbf{U}^{(i)}$ and $\mathbf{V}^{(j)}$. This makes the direct inconsistency penalty of domain clusters in the simple case Eq. (4) no longer applicable. We address this issue by taking an indirect regularization.

Let $\hat{\mathbf{U}}^{(ij)} = \mathbf{D}^{(ij)}\mathbf{U}^{(i)}$ and $\hat{\mathbf{V}}^{(ij)} = \mathbf{O}^{(ij)}\mathbf{V}^{(j)}$. Consider two nodes x and y in $\mathbf{A}^{(i)}$ that have similar cluster assignments $\hat{\mathbf{u}}_{x*}^{(ij)}$ and $\hat{\mathbf{u}}_{y*}^{(ij)}$. If h_{ij} is large, their corresponding cluster assignments $\hat{\mathbf{v}}_{x*}^{(ij)}$ and $\hat{\mathbf{v}}_{y*}^{(ij)}$ should be similar. For example, in Figure 1, if nodes 1 and 3 have similar cluster assignments in domain-specific network D, their cluster assignments in the underlying clustering structure shared by $\{D, E, F\}$ should be similar as well. We measure cluster assignment similarity by their inner product, and minimize the inconsistency $(\hat{\mathbf{u}}_{x*}^{(ij)}(\hat{\mathbf{u}}_{y*}^{(ij)})' - \hat{\mathbf{v}}_{x*}^{(ij)}(\hat{\mathbf{v}}_{y*}^{(ij)})')^2$.

Summing up the inconsistencies over all domain nodes, we have the following objective function that allows partially aligned domain-specific networks to have different sizes and number of clusters:

$$\min_{\substack{\mathbf{U}^{(i)}\geq 0\ (i=1,\ldots,g)\\ \mathbf{V}^{(j)}\geq 0\ (j=1,\ldots k)}} J_D = \sum_{i=1}^{g} J_A + a\sum_{i=1}^{g}\sum_{j=1}^{k} h_{ij}J_R \quad (5)$$

where

$$J_A = \|\mathbf{A}^{(i)} - \mathbf{U}^{(i)}(\mathbf{U}^{(i)})'\|_F^2$$

$$J_R = \sum_{x=1}^{n_i}\sum_{y=1}^{n_i}(\hat{\mathbf{u}}_{x*}^{(ij)}(\hat{\mathbf{u}}_{y*}^{(ij)})' - \hat{\mathbf{v}}_{x*}^{(ij)}(\hat{\mathbf{v}}_{y*}^{(ij)})')^2$$
$$= \|(\mathbf{D}^{(ij)}\mathbf{U}^{(i)})(\mathbf{D}^{(ij)}\mathbf{U}^{(i)})' - (\mathbf{O}^{(ij)}\mathbf{V}^{(j)})(\mathbf{O}^{(ij)}\mathbf{V}^{(j)})'\|_F^2$$

In Eq. (5), a is a regularization parameter for the relative importance between the domain-specific network clustering and the main cluster guided regularization. Intuitively, the more reliable the main network, the larger the value of a.

Discussions: The existing NMF based multi-view clustering methods either assume a single shared factor matrix among all views [1] or regularize all factor matrices towards a single centroid factor matrix [23]. In contrast, NoNClus introduces multiple hidden factor matrices to differentially regularize domain-specific clusters guided by the main clusters. If there is only one main cluster, NoNClus degenerates to a multi-view graph clustering method. Moreover, NoNClus allows different network sizes and number of clusters among domain-specific networks. Therefore, NoNClus can be viewed as a generalization of the existing multi-view graph clustering methods to different sized networks with multiple underlying clustering structures.

4.3 Learning Algorithm

Since the objective function Eq. (5) is not jointly convex, we optimize it by an alternating minimization approach, i.e., the objective function is alternately minimized with respect to one variable while fixing others. This procedure repeats until convergence.

Theorem 1 in the following gives the solution of $\mathbf{U}^{(\tau)}$ $(1 \leq \tau \leq g)$ when fixing other variables. Theorem 2 gives the solution of $\mathbf{V}^{(\eta)}$ $(1 \leq \eta \leq k)$ when fixing other variables.

THEOREM 1. **Updating $\mathbf{U}^{(\tau)}$.** *When other variables are fixed, updating $\mathbf{U}^{(\tau)}$ according to Eq. (6) monotonically decreases Eq. (5) until convergence. At convergence, the solution is a KKT fixed point.*

$$\mathbf{U}^{(\tau)} \leftarrow \mathbf{U}^{(\tau)} \circ \left(\frac{\mathbf{A}^{(\tau)}\mathbf{U}^{(\tau)} + a\sum_{j=1}^{k} h_{\tau j}\mathbf{W}^{(\tau j)}\mathbf{U}^{(\tau)}}{\mathbf{U}^{(\tau)}(\mathbf{U}^{(\tau)})'\mathbf{U}^{(\tau)} + a\sum_{j=1}^{k} h_{\tau j}\mathbf{Y}^{(\tau j)}}\right)^{\frac{1}{4}} \quad (6)$$

where

$$\mathbf{W}^{(\tau j)} = (\mathbf{D}^{(\tau j)})'(\mathbf{O}^{(\tau j)}\mathbf{V}^{(j)})(\mathbf{O}^{(\tau j)}\mathbf{V}^{(j)})'\mathbf{D}^{(\tau j)}$$
$$\mathbf{Y}^{(\tau j)} = (\mathbf{D}^{(\tau j)})'\mathbf{D}^{(\tau j)}\mathbf{U}^{(\tau)}(\mathbf{U}^{(\tau)})'(\mathbf{D}^{(\tau j)})'\mathbf{D}^{(\tau j)}\mathbf{U}^{(\tau)}$$

THEOREM 2. **Updating $\mathbf{V}^{(\eta)}$.** *When other variables are fixed, updating $\mathbf{V}^{(\eta)}$ according to Eq. (7) monotonically decreases Eq. (5) until convergence. At convergence, the solution is a KKT fixed point.*

$$\mathbf{V}^{(\eta)} \leftarrow \mathbf{V}^{(\eta)} \circ \left(\frac{\sum_{i=1}^{g} h_{i\eta}\mathbf{Q}^{(i\eta)}\mathbf{V}^{(\eta)}}{\sum_{i=1}^{g} h_{i\eta}\mathbf{R}^{(i\eta)}}\right)^{\frac{1}{4}} \quad (7)$$

Algorithm 1: NoNClus

Input: (1) a network of networks $\mathcal{R} =< \mathbf{G}, \mathcal{A} >$; (2) the mapping matrices $\{\mathbf{O}^{(ij)}\}$ and $\{\mathbf{D}^{(ij)}\}$; (3) the number of main clusters k; (4) the number of domain clusters in $\{\mathbf{A}^{(i)}\}$ and $\{\mathbf{V}^{(j)}\}$; (5) the parameter a

Output: a collection of partitions $\mathcal{C} = \{\mathcal{C}^{(1)}, ..., \mathcal{C}^{(g)}\}$ of all domain-specific networks

Normalize $\mathbf{G}$, $\mathbf{A}^{(i)}$ $(i = 1, ..., g)$ by Frobenius norm;
Phase I:
Initialize $\mathbf{H}$ with random values within $(0, 1]$;
repeat
  Update $\mathbf{H}$ by Eq. (2);
until *Convergence*
Normalize $\mathbf{H}$ by $\mathbf{H} \leftarrow \mathbf{D}_{\mathbf{H}}^{-1}\mathbf{H}$;
Phase II:
for $\tau \leftarrow 1$ *to* g **do**
  Initialize $\mathbf{U}^{(\tau)}$ with random values within $(0, 1]$;
end
for $\eta \leftarrow 1$ *to* k **do**
  Determine the size of $\mathbf{V}^{(\eta)}$ based on main cluster membership of domain-specific networks;
  Initialize $\mathbf{V}^{(\eta)}$ with random values within $(0, 1]$;
end
repeat
  for $\tau \leftarrow 1$ *to* g **do**
    Update $\mathbf{U}^{(\tau)}$ by Eq. (6);
  end
  for $\eta \leftarrow 1$ *to* k **do**
    Update $\mathbf{V}^{(\eta)}$ by Eq. (7);
  end
until *Convergence*
return $\mathcal{C} = \{\mathcal{C}^{(1)}, ..., \mathcal{C}^{(g)}\}$ based on $\mathbf{U}^{(1)}, ..., \mathbf{U}^{(g)}$.

where

$$\mathbf{Q}^{(i\eta)} = (\mathbf{O}^{(i\eta)})'(\mathbf{D}^{(i\eta)}\mathbf{U}^{(i)})(\mathbf{D}^{(i\eta)}\mathbf{U}^{(i)})'\mathbf{O}^{(i\eta)}$$
$$\mathbf{R}^{(i\eta)} = (\mathbf{O}^{(i\eta)})'\mathbf{O}^{(i\eta)}\mathbf{V}^{(\eta)}(\mathbf{V}^{(\eta)})'(\mathbf{O}^{(i\eta)})'\mathbf{O}^{(i\eta)}\mathbf{V}^{(\eta)}$$

In Eq. (6) and Eq. (7), $\circ$, $\frac{[\cdot]}{[\cdot]}$ and $(\cdot)^{\frac{1}{4}}$ are element-wise operators. Algorithm 1 summarizes our alternating minimization algorithm according to Theorems 1 and 2.

4.4 Correctness Analysis

In the next, we provide theoretical analysis of the updating rule in Theorem 1. We first prove the correctness of Eq. (6) according to the Karush-Kuhn-Tucker (KKT) condition [7]. Then we analyze its convergence using the auxiliary function approach [21]. The proofs for Theorem 2 are similar and omitted here.

THEOREM 3. **Correctness of Eq. (6).** *At convergence, the solution found by updating $\mathbf{U}^{(\tau)}$ according to Eq. (6) is a KKT fixed point.*

PROOF. Omitted for brevity. The formal proof can be found in an online Supplementary Material[2]. □

4.5 Convergence Analysis

Next we prove the convergence of Eq. (6) using the auxiliary function approach [21].

DEFINITION 2. *[21] A function $Z(h, \tilde{h})$ is an* **auxiliary function** *for a given function $J(h)$ if the conditions $Z(h, \tilde{h}) \geq J(h)$ and $Z(h, h) = J(h)$ are satisfied.*

[2]http://filer.case.edu/jxn154/NoNClus

LEMMA 1. *[21] If Z is an auxiliary function for J, then J is non-increasing under the update $h^{(t+1)} = \arg\min_h Z(h, h^{(t)})$.*

Theorem 4 gives the auxiliary function for the objective function Eq. (5) w.r.t. $\mathbf{U}^{(\tau)}$.

THEOREM 4. **Auxiliary function of $J(\mathbf{U}^{(\tau)})$.** *Let $J(\mathbf{U}^{(\tau)})$ denote the sum of all terms in Eq. (5) that contains $\mathbf{U}^{(\tau)}$, then the following function*

$$Z(\mathbf{U}^{(\tau)}, \tilde{\mathbf{U}}^{(\tau)}) = -2\sum_{pqr}\mathbf{A}^{(\tau)}_{rp}\tilde{\mathbf{U}}^{(\tau)}_{pq}\tilde{\mathbf{U}}^{(\tau)}_{rq}\left(1 + \log\frac{\mathbf{U}^{(\tau)}_{pq}\mathbf{U}^{(\tau)}_{rq}}{\tilde{\mathbf{U}}^{(\tau)}_{pq}\tilde{\mathbf{U}}^{(\tau)}_{rq}}\right)$$
$$+\sum_{pq}(\tilde{\mathbf{U}}^{(\tau)}(\tilde{\mathbf{U}}^{(\tau)})'\tilde{\mathbf{U}}^{(\tau)})_{pq}\frac{(\mathbf{U}^{(\tau)}_{pq})^4}{(\tilde{\mathbf{U}}^{(\tau)}_{pq})^3} + a\sum_j h_{\tau j}\sum_{pq}\tilde{\mathbf{Y}}^{(\tau j)}_{pq}\frac{(\mathbf{U}^{(\tau)}_{pq})^4}{(\tilde{\mathbf{U}}^{(\tau)}_{pq})^3}$$
$$-2a\sum_j h_{\tau j}\sum_{pqr}\mathbf{W}^{(\tau j)}_{rp}\tilde{\mathbf{U}}^{(\tau)}_{pq}\tilde{\mathbf{U}}^{(\tau)}_{rq}\left(1 + \log\frac{\mathbf{U}^{(\tau)}_{pq}\mathbf{U}^{(\tau)}_{rq}}{\tilde{\mathbf{U}}^{(\tau)}_{pq}\tilde{\mathbf{U}}^{(\tau)}_{rq}}\right) \tag{8}$$

where

$$\mathbf{W}^{(\tau j)} = (\mathbf{D}^{(\tau j)})'(\mathbf{O}^{(\tau j)}\mathbf{V}^{(j)})(\mathbf{O}^{(\tau j)}\mathbf{V}^{(j)})'\mathbf{D}^{(\tau j)}$$
$$\tilde{\mathbf{Y}}^{(\tau j)} = (\mathbf{D}^{(\tau j)})'\mathbf{D}^{(\tau j)}\tilde{\mathbf{U}}^{(\tau)}(\tilde{\mathbf{U}}^{(\tau)})'(\mathbf{D}^{(\tau j)})'\mathbf{D}^{(\tau j)}\tilde{\mathbf{U}}^{(\tau)}$$

is an auxiliary function for $J(\mathbf{U}^{(\tau)})$. It is also a convex function in $\mathbf{U}^{(\tau)}$ and its global minimum is

$$\mathbf{U}^{(\tau)} = \tilde{\mathbf{U}}^{(\tau)} \circ \left(\frac{\mathbf{A}^{(\tau)}\tilde{\mathbf{U}}^{(\tau)} + a\sum_{j=1}^k h_{\tau j}\mathbf{W}^{(\tau j)}\tilde{\mathbf{U}}^{(\tau)}}{\tilde{\mathbf{U}}^{(\tau)}(\tilde{\mathbf{U}}^{(\tau)})'\tilde{\mathbf{U}}^{(\tau)} + a\sum_{j=1}^k h_{\tau j}\tilde{\mathbf{Y}}^{(\tau j)}}\right)^{\frac{1}{4}} \tag{9}$$

PROOF. Omitted for brevity. The formal proof can be found in the Supplementary Material. □

Next we show the convergence of updating $\mathbf{U}^{(\tau)}$ by Eq. (6) in Theorem 5.

THEOREM 5. **Convergence of Eq. (6).** *When other variables are fixed, updating $\mathbf{U}^{(\tau)}$ according to Eq. (6) monotonically decreases Eq. (5) until convergence.*

PROOF. According to Definition 2, Lemma 1 and Theorem 4 (note Eq. (9) is consistent with Eq. (6)), at any iteration $\kappa \geq 0$ during updating $\mathbf{U}^{(\tau)}$, we have

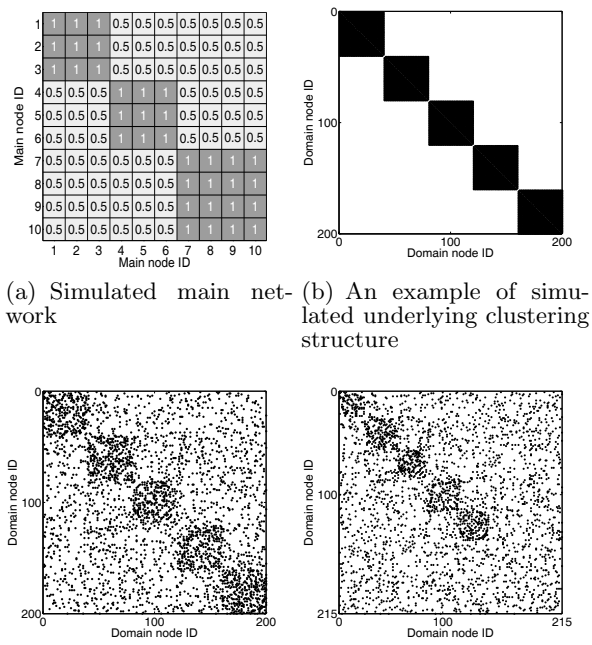
$$J((\mathbf{U}^{(\tau)})^{(\kappa)}) = Z((\mathbf{U}^{(\tau)})^{(\kappa)}, (\mathbf{U}^{(\tau)})^{(\kappa)})$$
$$\geq Z((\mathbf{U}^{(\tau)})^{(\kappa+1)}, (\mathbf{U}^{(\tau)})^{(\kappa)}) \geq J((\mathbf{U}^{(\tau)})^{(\kappa+1)})$$

where $(\mathbf{U}^{(\tau)})^{(\kappa)}$ denotes the updated $\mathbf{U}^{(\tau)}$ at κ^{th} iteration. Thus $J(\mathbf{U}^{(\tau)})$ monotonically decreases. Since the objective function Eq. (5) is bounded below by 0, the updating of $\mathbf{U}^{(\tau)}$ will converge. □

Similarly, Theorem 2 can be proved. Therefore, alternately updating $\mathbf{U}^{(\tau)}$ and $\mathbf{V}^{(\eta)}$ by Eq. (6) and Eq. (7) monotonically decreases Eq. (5) until convergence and the stationary point is a KKT fixed point, which guarantees the correctness and convergence of Algorithm 1.

4.6 Complexity Analysis

Let N be the maximal number of nodes in any domain-specific network. There can be at most N non-zero entries in $\mathbf{O}^{(ij)}$ and $\mathbf{D}^{(ij)}$ $(i = 1, ..., g, j = 1, ..., k)$ because of the one-to-one mapping between common nodes in different domain-specific networks. In practice, $\mathbf{A}^{(1)}, ..., \mathbf{A}^{(g)}$ and $\mathbf{G}$ can be



(a) Simulated main network

(b) An example of simulated underlying clustering structure

(c) Embedding noise

(d) An example of simulated domain-specific network

Figure 2: Synthetic dataset generation

sparse. Let M and m be the maximal number of non-zero entries in any $\mathbf{A}^{(i)}$ and $\mathbf{G}$, respectively. Let T be the maximal number of clusters in any $\mathbf{A}^{(i)}$.

Based on Eq. (6) and Eq. (7), updating each $\mathbf{U}^{(\tau)}$ and $\mathbf{V}^{(\eta)}$ require $O(MT + kNT^2)$ and $O(gNT^2)$ time, respectively. Thus the overall time complexity of Algorithm 1 is $O(I_m(mk + gk^2) + I_d(gMT + gkNT^2))$ considering both Phase I and Phase II, where I_m and I_d are the total number of iterations before the convergence of Phase I and Phase II, respectively. Moreover, I_m, g, m and k are usually much smaller than I_d, N, M and T, respectively. T is much smaller than N, and k can be regarded as a small constant. Therefore, the actual time complexity can be denoted as $O(I_d g(MT + NT^2))$. The experimental results show that the our algorithm is almost linear with respect to M.

5. EXPERIMENTAL RESULTS

In this section, we evaluate NoNClus on synthetic and real-world datasets and compare it with the state-of-the-art multi-view and multi-domain graph clustering methods.

5.1 Effectiveness Evaluation

5.1.1 Simulation Study

We first evaluate our method using synthetic datasets. We generate a main network containing three main clusters with sizes 3, 3, 4 as shown in Figure 2(a). The domain-specific networks are generated as follows. We first generate an underlying domain-specific clustering structure for each main cluster. Figure 2(b) shows an example containing five clusters of the same size, where non-zero entries are set to 1. Then domain-specific networks are generated from each underlying clustering structure. To embed noise, we randomly flip α ($0 \leq \alpha \leq 1$) fraction of 1 entries in the matrix to 0 and β ($0 \leq \beta \leq 1$) fraction of 0 entries to 1. An example is shown in Figure 2(c) with $\alpha = 80\%$ and $\beta = 5\%$. To generate domain-specific networks with different sizes, we randomly remove or add ε fraction of nodes in the previous matrix. ε follows normal distribution with mean μ and standard deviation σ and its value is set between 0 and 1. An example with 215 domain nodes generated by $\mu = 0.3$ and $\sigma = 0.05$ is shown in Figure 2(d).

Using the above approach, we generate two different types of synthetic datasets. In the first dataset, all three underlying clustering structures contain 5 clusters, and all domain-specific networks have the same set of 200 nodes. α and β are set to 80% and 5% respectively to simulate noise. We refer to this dataset as the SynNoN-view dataset. This dataset is used to evaluate the multi-view graph clustering methods, since they assume all views have the same size.

In the second dataset, the three underlying clustering structures contain 5, 6, 7 clusters, and 200, 300, 350 nodes, respectively. They share 100 common nodes. The domain-specific networks are generated with $\alpha = 80\%$, $\beta = 5\%$, $\mu = 0.3$ and $\sigma = 0.05$. We refer to this dataset as the SynNoN-dom dataset. This dataset is used to evaluate the multi-domain graph clustering methods, which allow different domain sizes.

We compare NoNClus with several state-of-the-art clustering methods, including (1) Symmetric NMF (SNMF) [18]; (2) spectral clustering (Spectral) [29]; (3) multi-view co-training spectral clustering (CTSC) [19]; (4) multi-view pairwise co-regularized spectral clustering (PairCRSC) [20]; (5) multi-view centroid-based co-regularized spectral clustering (CentCRSC) [20]; (6) Tensor Factorization (TF) [16]; and (7) multi-domain co-regularized graph clustering (CGC) [8].

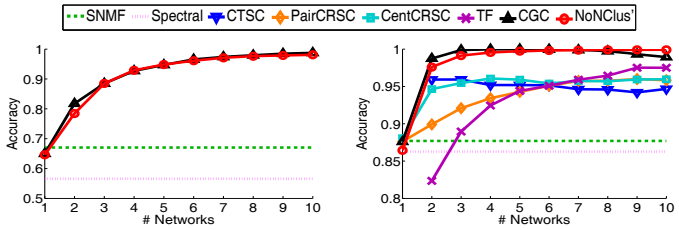
Note that the SNMF and spectral clustering methods can only be applied to a single network. CTSC, PairCRSC and CentCRSC are multi-view graph clustering methods and can only be applied on the SynNoN-view dataset. For TF, we test both CP and Tucker decompositions [16] and use three different strategies to assign a data object to a cluster: (1) the highest value in a row of the factor matrix; (2) the highest absolute value in a row of the factor matrix [32]; and (3) applying k-means [25] on the factor matrix. The best results are reported. Note that TF is similar to multi-view methods thus can only be applied on the SynNoN-view dataset. Moreover, TF does not distinguish individual networks and only gives an overall clustering result of all nodes. CGC is a recent multi-domain graph clustering method that can be applied on the SynNoN-dom dataset. The common node relationships between different domain specific networks are used as the cross-domain relationships in CGC.

Table 2 shows the averaged accuracy of different methods over 500 runs. The parameters are tuned for optimal performance of all methods. It can be seen that NoNClus achieves better individual and overall performance compared to other methods on both datasets. The multi-view/domain methods, CTSC, PairCRSC, CentCRSC, TF and CGC assume a single underlying clustering structure. In contrast, NoNClus allows more flexible underlying clustering structures. This demonstrates that utilizing domain similarity network can dramatically improve the accuracy.

Next, we study a degraded version of NoNClus, which assumes that all domain-specific networks share the same

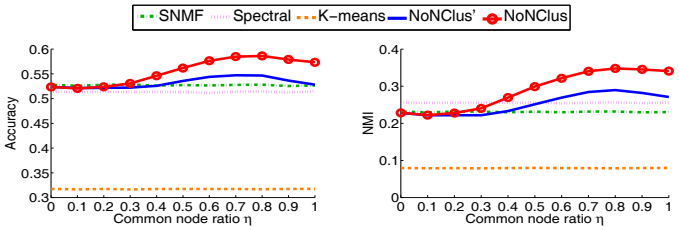
Table 2: Accuracy of different methods on synthetic datasets

Dataset	Method	Main cluster 1			Main cluster 2			Main Cluster 3				
		Net 1	Net 2	Net 3	Net 4	Net 5	Net 6	Net 7	Net 8	Net 9	Net 10	Overall
view	SNMF	0.8751	0.8716	0.8735	0.8796	0.8732	0.8754	0.8722	0.8690	0.8682	0.8746	0.8732
	Spectral	0.8587	0.8586	0.8675	0.8619	0.8571	0.8624	0.8626	0.8582	0.8583	0.8622	0.8607
	CTSC	0.6249	0.6258	0.6279	0.6221	0.6236	0.6196	0.9157	0.9118	0.9106	0.9181	0.7400
	PairCRSC	0.9166	0.9174	0.9227	0.9186	0.9176	0.9173	0.9355	0.9335	0.9378	0.9353	0.9252
	CentCRSC	0.9050	0.9031	0.9090	0.9021	0.9090	0.9077	0.9391	0.9408	0.9342	0.9378	0.9188
	TF	–	–	–	–	–	–	–	–	–	–	0.6505
	CGC	0.6364	0.6337	0.6407	0.6385	0.6273	0.6316	0.7332	0.7365	0.7251	0.7210	0.6724
	NoNClus	**0.9444**	**0.9403**	**0.9463**	**0.9447**	**0.9435**	**0.9418**	**0.9617**	**0.9621**	**0.9643**	**0.9629**	**0.9512**
dom	SNMF	0.6584	0.6687	0.6583	0.7123	0.7063	0.7129	0.6558	0.6596	0.6620	0.6630	0.6787
	Spectral	0.5554	0.5618	0.5556	0.5799	0.5768	0.5811	0.5167	0.5188	0.5241	0.5242	0.5490
	CGC	0.7303	0.7297	0.7229	0.7992	0.7962	0.7965	0.7859	0.7840	0.7837	0.7876	0.7797
	NoNClus	**0.7882**	**0.7960**	**0.7914**	**0.8649**	**0.8650**	**0.8654**	**0.8409**	**0.8363**	**0.8367**	**0.8389**	**0.8388**



(a) Networks of different sizes (b) Networks of same size

Figure 3: Clustering accuracy with different number of networks when all networks share a single common underlying clustering structure



(a) Overall accuracy (b) Overall NMI

Figure 4: Clustering performance on 20-Newsgroup dataset with various rate of common nodes

underlying clustering structure, but allows different sized domain-specific networks. We refer to this degraded version as NoNClus'. As discussed in Sec. 4.2.2, NoNClus' has the same assumption as the multi-view clustering methods and can be treated as a generalization of these methods on different sized networks.

We generate multiple domain-specific networks with different sizes sharing the same underlying clustering structure by setting $\alpha = 80\%$, $\beta = 5\%$, $\mu = 0.3$, $\sigma = 0.05$. Note that the multi-view graph clustering methods cannot be applied to this dataset. Figure 3(a) shows the accuracy when varying the number of domain-specific networks. All results are averaged over 500 runs. It can be seen that NoNClus' is effective in incorporating information from multiple domain-specific networks. It performs better than single network clustering methods and is similar to CGC on this dataset.

We also generate domain-specific networks with the same size by setting $\alpha = 80\%$, $\beta = 5\%$, $\mu = 0$, $\sigma = 0$, so that the multi-view graph clustering methods can be applied. Figure 3(b) shows the results on this dataset (CTSC and TF requires at least two views to run). From the results, we can observe that NoNClus' is comparable to the multi-view methods when applied to networks of the same size but is more general than them. Also, NoNClus' is similar to the multi-domain method CGC. Note that CGC uses a pairwise regularization. NoNClus' utilizes a centroid regularization, thus is more efficient than CGC (see Sec. 5.2). Because of its competitive performance and efficiency, in the following, we use NoNClus' as an alternative to multi-view/domain graph clustering methods for datasets with different-sized domain-specific networks.

5.1.2 20-Newsgroup Dataset

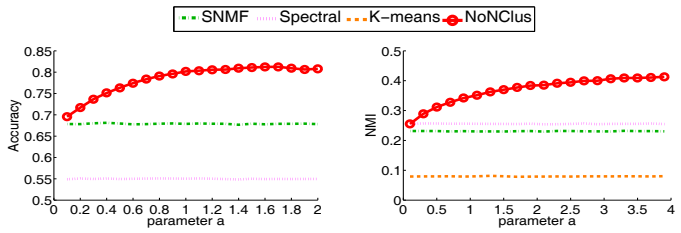
We further evaluate the effectiveness of NoNClus using 20-Newsgroup dataset[3]. Following a similar approach as in [9], we preprocess the data by removing stop words, ignoring headers and subject lines. For each newsgroup, we select the top 2000 words using the mutual information based feature selection method.

We use 12 news groups of three categories, Comp, Rec and Talk[4], corresponding to three underlying clustering structures, each with four clusters (news groups). In this study, we generate 10 domain-specific networks from each category. Each domain-specific network contains randomly sampled 200 documents from the 4 news groups (50 documents from each group) in a category. The affinity matrix of documents is computed based on cosine similarity. The main network is generated by the cosine similarity between the overall word frequencies of domain-specific networks. As a result, the main network contains 30 main nodes forming three main clusters corresponding to the three categories. Each main node corresponds to a domain-specific network.

The common nodes in different domain-specific networks are generated as follows. For any two domain-specific networks generated from the same underlying clustering structure, a document in one domain-specific network is randomly mapped to a document with the same cluster label (e.g., comp.graphics) in another domain-specific network. For any two domain-specific networks generated from different un-

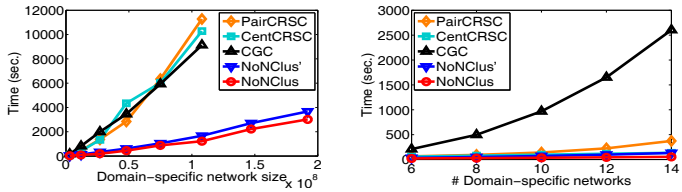
[3]http://qwone.com/%7Ejason/20Newsgroups/

[4]**Comp**: comp.graphics, comp.os.ms-windows.misc, comp.sys.ibm.pc.hardware, comp.sys.mac.pc.hardware; **Rec**: rec.autos, rec.motorcycles, rec.sport.baseball, rec.sport.hockey; **Talk**: talk.politics.guns, talk.politics.mideast, talk.politics.misc, talk.religion.misc.



(a) SynNoN-dom dataset (b) 20-Newsgroup dataset

Figure 5: Parameter sensitivity evaluation



(a) Varying network size (b) Varying # of Networks

Figure 6: Running time evaluation

derlying clustering structures, the documents are randomly mapped with one-to-one relationship. We vary the ratio of common nodes, η, from 0 to 1 to evaluate its effect.

The clustering performance is evaluated using both purity accuracy and normalized mutual information (NMI) according to the labels provided in the dataset. Since multi-view clustering methods CTSC, PairCRSC, CentCRSC and TF cannot be applied on networks with partial common nodes, NoNClus' is used as a generalization of the multi-view clustering methods. Also, single network clustering methods SNMF and Spectral clustering (Spectral) are performed on individual domain-specific networks. The widely used k-means method [25] is also selected as a baseline method in this comparison. It is applied on the original document-word matrix instead of the network data.

Figures 4 shows the average accuracy and NMI on all domain-specific networks when varying η. All results are averaged over 100 runs. As can be seen from the figures, NoNClus becomes better than SNMF when there are around 40% common nodes. NoNClus' performs worse than NoNClus and does not obviously increase the accuracy over SNMF. This is because NoNClus' cannot handle multiple underlying clustering structures. The results demonstrate that NoNClus can effectively improve the accuracy with a small number of common nodes among different networks.

5.2 Performance Evaluation

In this section, we evaluate NoNClus in terms of its sensitivity to the regularization parameter a and its scalability. The evaluation of its convergence property can be found in the Supplementary Material. The datasets used include SynNoN-view, SynNoN-dom and 20-Newsgroup.

Figure 5 shows the clustering accuracy and NMI when varying a. The results of single network clustering methods are used as references. The accuracy and NMI are averaged over 100 runs. We observe that NoNClus is not sensitive to the regularization parameter a. The accuracy and NMI increase as a increases and become stable after $a \geq 1$.

Table 3: Tissue-specific gene co-expression networks

Tissue-specific network	# nodes	# edges
Blood	633	2,573
Lymph node	648	2,256
Tonsil	682	2,480
Thymus	786	2,939
Brain	746	3,135
Caudate nucleus	640	2,578
Hypothalamus	641	2,500
Cerebellum	679	2,636
Total	5,455	21,097

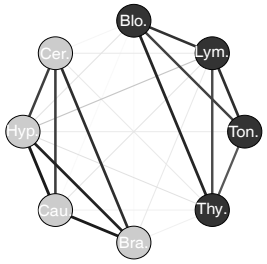


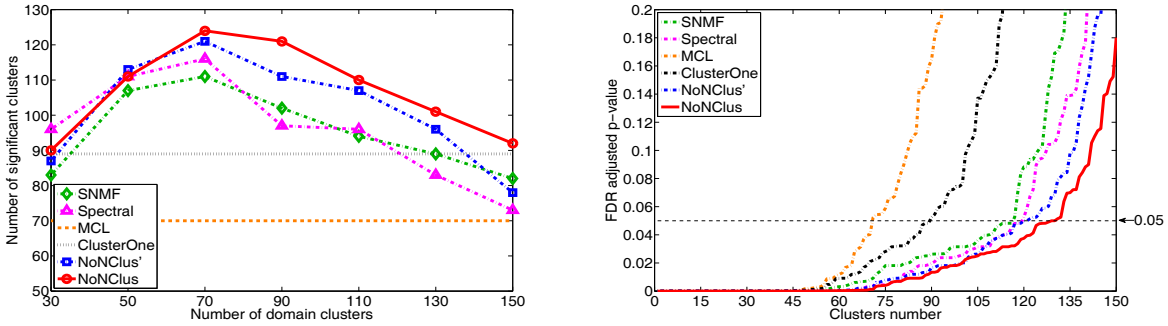
Figure 7: Tissue-tissue similarity network (the main network in NoN)

Next, we evaluate the efficiency of NoNClus using the SynNoN-view dataset. Other multi-view/domain graph clustering methods are also evaluated as references (CTSC is omitted since it does not guarantee convergence). The experiments are performed on a 2.10GHz machine with 48GB memory. The reported results are averaged over 10 runs.

Figure 6(a) shows the running time when varying the size of the domain-specific networks. There are 6 domain-specific networks. The network size is measured by the total number of edges in all domain-specific networks. Figure 6(b) shows the running time when varying the number of domain-specific networks. There are 2,500 nodes in each network. We omit some results of PairCRSC, CentCRSC and CGC because of their high memory or running time costs. As can be seen, the running time of NoNClus is almost linear w.r.t. the size and number of domain-specific networks. This is consistent with the time complexity analysis in Sec. 4.6. In addition, NoNClus is faster than other methods since PairCRSC and CGC require pairwise regularizations, and the eigendecomposition process of PairCRSC and CentCRSC for non-sparse matrices are time and space consuming. NoNClus runs faster than NoNClus' because of its faster convergence rate.

5.3 A Case Study of Tissue-Specific Gene Co-Expression Networks

In this section, we apply NoNClus on tissue-specific gene co-expression networks. We use the recently published global map of human gene expression dataset [24] to generate tissue-specific gene co-expression networks. The dataset contains 5372 samples for 128 different tissues in four different cell types, i.e., normal, disease, neoplasm and cell line. Following a similar approach as in [5], we consider tissues of normal



(a) Number of significant clusters

(b) FDR adjusted p-values

Figure 8: Performance comparison on tissue-specific gene co-expression NoN

status and experiments with at least five replicates. We select 8 tissues, i.e., blood, lymph node, tonsil, thymus, brain, caudate nucleus, hypothalamus and cerebellum to form the main network. The tissue similarity matrix is constructed using the pairwise correlation (normalized between $[0, 1]$) of the expression data of tissue-specific genes. The main network is shown in Figure 7, which contains two main clusters.

For each tissue, we construct the tissue-specific gene co-expression network using the gene expression data for that tissue. We extract genes that are expressed in each tissue (with expression values greater than 10). The edges in a gene co-expression network are weighted by the Pearson's correlation coefficient (normalized between $[0, 1]$) between two connected genes. The statistics of the gene co-expression networks are summarized in Table 3.

We compare NoNClus with (1) SNMF; (2) Spectral clustering (Spectral); (3) Markov clustering (MCL) [34]; (4) ClusterOne [27]; and (5) NoNClus'. MCL has been widely applied to detect functional modules in biological networks [3]. ClusterOne can detect overlapping clusters. Its overlapping rate is set such that any two clusters of size 5 can have at most 3 common genes (i.e., match coefficient [27] 0.36). This rate also applies to clusters of other sizes. Note that multi-view clustering methods cannot be applied because of the different network sizes. We use NoNClus' as an alternative.

The clustering performance are evaluated using the standard Gene Set Enrichment Analysis (GSEA) [31]. The most significant Gene Ontology (GO) term in the biological process category [2] is assigned to each identified gene cluster (we evaluate clusters with sizes at least 5). The significance is assessed by Hypergeometric distribution [3]. Raw p-values are adjusted for multiple testing [36] by False Discovery Rate (FDR) [4].

We first assume that all gene co-expression networks have the same number of clusters. If a method needs initialization, we run it with 10 random initializations and report the optimal performance. Figure 8(a) shows the total number of significant clusters detected in all gene co-expression networks w.r.t. the input number of domain clusters. As we can see, for the methods that need input cluster number, the best performance occurs when the number of clusters is set to 70. Before that, all methods perform similarly because of the limited numbers of clusters. After that, NoNClus is more stable than NoNClus', since NoNClus allows multiple underlying clustering structures. NoNClus also detects more significant clusters than other methods do.

Table 4: Number of significant clusters

Method	# significant clusters	p-value
SNMF	116	4.64×10^{-5}
Spectral	119	6.66×10^{-3}
MCL	70	6.45×10^{-17}
ClusterOne	89	1.43×10^{-10}
NoNClus'	121	4.87×10^{-2}
NoNClus	**130**	1

Next we present the results of the selected methods when their parameters are tuned for their optimal performance. In particular, the numbers of domain clusters for NoNClus' and NoNClus can be tuned by using the optimal values individually given by SNMF or spectral clustering. In this experiment, they are around 70.

The p-values of detected clusters are shown in Figure 8(b). The clusters are sorted in ascending order of their p-values. We observe that the clusters detected by NoNClus are more significant than those identified by other methods.

Table 4 shows the number of significant clusters identified by different methods using a significance threshold 0.05. It can be seen that NoNClus detects more significant clusters than other methods do. For each alternative method, we further perform the two-sample t-test on the p-values of the 155 most significant clusters detected by that method and those by NoNClus. The significance of the test results are reported in the third column of Table 4. Clearly, NoNClus performs significantly better than other methods.

The reason for the better performance of NoNClus is that the gene co-expression networks are very noisy. Single network clustering methods can be sensitive to these noises. Utilizing common clustering structure shared by similar tissues can help improve the robustness of the method. On the other hand, the same set of genes forming a cluster in similar tissues may not form a cluster in dissimilar tissues. In particular, there are some housekeeping genes that are universally expressed in different tissues. These genes achieve their functions in different tissues by interacting with genes that are tissue specific. These tissue specific genes are expressed only in some tissues but not in others [6]. Thus it is more reasonable to distinguish different tissue (main)

clusters when integrating multiple tissue specific gene co-expression networks.

6. CONCLUSION

Clustering multiple networks has been widely recognized as promising to improve graph clustering performance. Existing multiple network clustering methods, such as multi-view/domain graph clustering, assume a single underlying clustering structure is shared among all networks. In this paper, we propose a new clustering framework that clusters multiple domain-specific networks sharing multiple underlying clustering structures. We model domain similarity as a main network where main nodes represent domain-specific networks and formulate the clustering problem on this novel network of networks (NoN) setting as a two phase regularized optimization problem. We develop NoNClus to solve this problem and provide rigorous theoretical analysis concerning its correctness, convergence and complexity. Experimental results on both synthetic and real-world datasets demonstrate the effectiveness of NoNClus.

7. ACKNOWLEDGEMENT

This work was partially supported by the National Science Foundation grants IIS-1162374, IIS-1218036 and IIS-1017415, by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053, by National Institutes of Health under the grant number R01LM011986, and by Region II University Transportation Center under the project number 49997-33 25.

8. REFERENCES

[1] Z. Akata, C. Thurau, C. Bauckhage, et al. Non-negative matrix factorization in multimodality data for segmentation and label prediction. In *16th Computer Vision Winter Workshop*, 2011.

[2] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, et al. Gene ontology: tool for the unification of biology. *Nat. Genet.*, 25(1):25–29, 2000.

[3] S. Asur, D. Ucar, and S. Parthasarathy. An ensemble framework for clustering protein–protein interaction networks. *Bioinformatics*, 23(13):i29–i40, 2007.

[4] Y. Benjamini and D. Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Ann. Stat.*, pages 1165–1188, 2001.

[5] D. Börnigen, T. H. Pers, L. Thorrez, C. Huttenhower, Y. Moreau, and S. Brunak. Concordance of gene expression in human protein complexes reveals tissue specificity and pathology. *Nucleic Acids Res.*, 41(18):e171–e171, 2013.

[6] A. Bossi and B. Lehner. Tissue specificity and the human protein interaction network. *Mol. Syst. Biol.*, 5(1), 2009.

[7] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2009.

[8] W. Cheng, X. Zhang, Z. Guo, Y. Wu, P. F. Sullivan, and W. Wang. Flexible and robust co-regularized multi-domain graph clustering. In *KDD*, 2013.

[9] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *KDD*, 2003.

[10] C. H. Ding, X. He, and H. D. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *SDM*, 2005.

[11] E. Eaton, M. Desjardins, and S. Jacob. Multi-view clustering with constraint propagation for learning with an incomplete mapping between views. In *CIKM*, 2010.

[12] X. Z. Fern and C. E. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *ICML*, 2004.

[13] D. Greene and P. Cunningham. A matrix factorization approach for integrating multiple data views. In *ECML PKDD*, 2009.

[14] J. C. Ho, J. Ghosh, and J. Sun. Marble: high-throughput phenotyping from electronic health records via sparse nonnegative tensor factorization. In *KDD*, 2014.

[15] H. Hu, X. Yan, Y. Huang, J. Han, and X. J. Zhou. Mining coherent dense subgraphs across massive biological networks for functional discovery. *Bioinformatics*, 21(suppl 1):i213–i221, 2005.

[16] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[17] T. G. Kolda and J. Sun. Scalable tensor decompositions for multi-aspect data mining. In *ICDM*, 2008.

[18] D. Kuang, H. Park, and C. H. Ding. Symmetric nonnegative matrix factorization for graph clustering. In *SDM*, 2012.

[19] A. Kumar and H. Daumé. A co-training approach for multi-view spectral clustering. In *ICML*, 2011.

[20] A. Kumar, P. Rai, and H. Daume. Co-regularized multi-view spectral clustering. In *NIPS*, 2011.

[21] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, 2000.

[22] S.-Y. Li, Y. Jiang, and Z.-H. Zhou. Partial multi-view clustering. In *AAAI*, 2014.

[23] J. Liu, C. Wang, J. Gao, and J. Han. Multi-view clustering via joint nonnegative matrix factorization. In *SDM*, 2013.

[24] M. Lukk, M. Kapushesky, J. Nikkilä, H. Parkinson, A. Goncalves, W. Huber, E. Ukkonen, and A. Brazma. A global map of human gene expression. *Nat. Biotechnol.*, 28(4):322–324, 2010.

[25] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. California, USA, 1967.

[26] O. Magger, Y. Y. Waldman, E. Ruppin, and R. Sharan. Enhancing the prioritization of disease-causing genes through tissue specific protein interaction networks. *PLoS Comput. Biol.*, 8(9):e1002690, 2012.

[27] T. Nepusz, H. Yu, and A. Paccanaro. Detecting overlapping protein complexes in protein-protein interaction networks. *Nat. Methods*, 9(5):471–472, 2012.

[28] J. Ni, H. Tong, W. Fan, and X. Zhang. Inside the atoms: ranking on a network of networks. In *KDD*, 2014.

[29] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000.

[30] A. Strehl and J. Ghosh. Cluster ensembles-a knowledge reuse framework for combining partitionings. In *AAAI/IAAI*, 2002.

[31] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. USA*, 102(43):15545–15550, 2005.

[32] J. Sun, D. Tao, and C. Faloutsos. Beyond streams and graphs: dynamic tensor analysis. In *KDD*, 2006.

[33] A. Trivedi, P. Rai, H. Daumé III, and S. L. DuVall. Multiview clustering with incomplete views. In *NIPS Workshop*, 2010.

[34] S. Van Dongen. A cluster algorithm for graphs. In *Centrum voor Wiskunde en Informatica (CWI)*, 2000.

[35] H. Wang, F. Nie, and H. Huang. Multi-view clustering and feature learning via structured sparsity. In *ICML*, 2013.

[36] P. H. Westfall. *Resampling-based multiple testing: Examples and methods for p-value adjustment*, volume 279. John Wiley & Sons, 1993.

[37] D. Zhou and C. J. Burges. Spectral clustering and transductive learning with multiple views. In *ICML*, 2007.