Fusion and Community Detection in Multi-layer Graphs

Vladimir Gligorijević, Yannis Panagakis, Stefanos Zafeiriou
Department of Computing,
Imperial College London
SW7 2AZ London, UK
Email: v.gligorijevic@imperial.ac.uk

***Abstract*—Relational data arising in many domains can be represented by networks (or graphs) with nodes capturing entities and edges representing relationships between these entities. Community detection in networks has become one of the most important problems having a broad range of applications. Until recently, the vast majority of papers have focused on discovering community structures in a single network. However, with the emergence of multi-view network data in many real-world applications and consequently with the advent of multi-layer graph representation, community detection in multi-layer graphs has become a new challenge. Multi-layer graphs provide complementary views of connectivity patterns of the same set of vertices. Fusion of the network layers is expected to achieve better clustering performance. In this paper, we propose two novel methods, coined as *WSSNMTF* (Weighted Simultaneous Symmetric Non-Negative Matrix Tri-Factorization) and *NG-WSSNMTF* (Natural Gradient WSSNMTF), for fusion and clustering of multi-layer graphs. Both methods are robust with respect to missing edges and noise. We compare the performance of the proposed methods with two baseline methods, as well as with three *state-of-the-art* methods on synthetic and three real-world datasets. The experimental results indicate superior performance of the proposed methods.**

I. Introduction

Networks (or graphs[1]) are powerful mathematical tools to represent, model, and analyse many data types in which entities correspond to nodes and relationships between entities correspond to edges. *Community detection* in networks (also known as graph clustering) is one of the foremost problem in network science [1]. It aims to group densely connected nodes into clusters (i.e., *communities*). Due to the abundance of network data in many domains, community detection has become an important task in many research areas, such as biology, sociology, economy, physics, computer science, chemistry etc.

However, in many domains, different experiments or measurements can provide different types of relations between entities. For example, in sociology, users in a social networks can be related by different types of interactions (e.g., personal, professional, social, etc.). In biology an example include a genetic interaction network in which genes representing nodes can interact via different types of molecular interactions. Reducing these networks to a single type interactions is often a very crude approximation that fails to capture a rich complexity of the system and consequently lead to poor clustering results. In order to encompass a multimodal nature of these relations, a *multi-layer graph* representation has been proposed [2]. Multi-layer graphs (also known as *multi-view* or *multiplex* graphs) have recently attracted a lot of attention in network science community. They can be represented as a set of graph layers that share a common set of vertices, V, but a different set of edges, E_i in each layer; that is, $G_i = \{V, E_i, \omega_i\}$, where ω_i represents the corresponding edge weights. With the emergence of this graph representation, community detection in multi-layer graphs has become a new challenge [3].

Traditional single-network graph algorithms fail to properly address the community detection problem on multi-layer networks. Namely, they can only be used in mining connectivity patterns of each graph layer independently, without taking into account the correlated information coming from other graph layers; consequently, in terms of clustering accuracy, this often results in under performance. Therefore, various types of methods for mining multi-layer networks have been proposed [3]. Their goal is to properly combine the connectivity information from all layers in order to find the most consensus community structure across different graph layers.

Recently the community detection problem in multi-layer graph has gained an increasing attention. Thus far, a few *state-of-the-art* methods have been proposed and applied in different domains. In [4], the authors propose a framework called Co-regularized Graph Clustering based on Non-Negative Matrix Factorization (*CGC-NMF*) for multi-domain graph clustering. Within the same objective function, they factorize each adjacency matrix (representing connectivity of each graph layer) independently while minimizing the distance between the low-dimensional cluster indicator matrices. In [5], the authors introduce *GraphFuse*, a method for clustering multi-layered graphs. They cluster multi-layer graphs by using tensor decomposition. That is, they use a variant of PARAFAC decomposition with Sparse Latent Factors (SLF) to co-cluster nodes in different layers. In addition to matrix and tensor factorization methods, several spectral clustering methods for multi-layer graphs have been proposed [6]–[8]. They construct unified graph Laplacian taking into account all the network layers, and then apply spectral clustering algorithm.

We propose two novel methods for network fusion and com-

[1]we use terms *graphs* and *network* interchangeably throughout this paper

munity extraction from multiple network layers. Both methods are based on Symmetric Non-negative Matrix Tri-Factorization (*SNMTF*) [9], [10], in which an adjacency matrix, representing a graph, is decomposed into a product of three low-dimensional non-negative latent feature matrices. We extend this approach on multiple graph layers, and we introduce our methods Weighted Simultaneous SNMTF (*WSSNMTF*) and WSSNMTF with Natural Gradient (*NG-WSSNMTF*). Both methods simultaneously factorize adjacency matrices representing different network layers by using SNMTF, while keeping the low-dimensional latent feature matrices shared across the decomposition; this enables the potential correlations between graph layers to be taken into account during decomposition. Furthermore, both methods take into account the missing values in each matrix (i.e., the missing links in each network layer). To this end, for each layer, we introduce a *symmetric weight matrix*, in which the entries are equal to one for the observed edges and zero for the missing ones. Thus, we formulate the objective function to factorize only observed values. Furthermore, we impose sparsity penalties to our learned low-dimensional latent feature matrices by adding ℓ_1 norm regularization terms to our objective function. This will make the learned matrices more robust to noise and outliers, which is the case for many incomplete graph layers. Consequently, this leads to much stable clustering results. To the best of our knowledge, these are the first methods for multi-layer community detection that consider missing and incomplete network layers. While *WSSNMTF* uses regular gradient, *NG-WSSNMTF* uses natural gradient to optimize the objective function by imposing the low-dimensional latent feature matrices to be orthonormal, and thus, leading to better clustering results.

The rest of the paper is organized as follows. In Section II, we mathematically formulate the proposed two methods and provide the corresponding algorithms. In Section III, we introduce synthetic and real-world multi-layer networks and give a brief description of methods used for comparison with our proposed methods. Concluding remarks are given in Section IV.

II. Weighted Simultaneous SNMTF (WSSNMTF)

A. Formulation of the problem

A multi-layer graph of N layers can be represented by a set of binary, symmetric *adjacency matrices*, $\mathbf{A}^{(i)} \in \{0,1\}^{n\times n}$, where $i = \{1,\dots,N\}$. The method can also be applied to *weighted* graphs where instead of binary entries in the adjacency matrix we have some real values. Each $\mathbf{A}^{(i)}$ is constructed over the same number of nodes n. For some network layers, this may result in $\mathbf{A}^{(i)}$ having zero rows and columns. Therefore, to treat this as missing values properly, for each network layer we define a *weight matrix*, $\mathbf{W}^{(i)}$, introduces in the following way:

$$\forall i, \quad \mathbf{W}^{(i)}_{ab} = \begin{cases} 1 & \text{if } a \text{ is connected with } b \text{ in layer i,} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

In analogy with the SNMTF, we tri-factorize each symmetric matrix $\mathbf{A}^{(i)}$ into a product of three non-negative matrices in the following way:

$$\mathbf{A}^{(i)} \approx \mathbf{H}\mathbf{S}^{(i)}\mathbf{H}^T, \quad \text{where, } \mathbf{S}^{(i)} \geq 0, \quad \mathbf{H} \geq 0$$

Note that matrix $\mathbf{H}$, also called *soft cluster indicator matrix*, is shared across the decompositions. To factorize only the observed entries in matrices (i.e., the observed links) we formulate the following objective function:

$$\mathcal{L} = \sum_{i=1}^{N} \parallel \mathbf{W}^{(i)} \circ (\mathbf{A}^{(i)} - \mathbf{H}\mathbf{S}^{(i)}\mathbf{H}^T) \parallel_F^2 + \sum_{i=1}^{N} \eta_i \parallel \mathbf{S}^{(i)} \parallel_1 \tag{2}$$

where $\circ$ denotes Hadamard product of matrices; ℓ_1 norm of the second terms imposes sparsity constraint to $\mathbf{S}^{(i)}$ matrices. Parameters η_i for $i \in \{1,\dots,N\}$ are trade-off parameters for balancing the influence between the two terms.

B. Optimization with ordinary gradient

We minimize the objective function shown in (2) under the following constraints: $\mathbf{H} \geq 0$ and $\mathbf{S}^{(i)} \geq 0$ for $\forall i \in \{1,\dots,N\}$. We derive the update rules for minimizing the objective function following the procedure from the constrained optimization theory [11]. Since the objective function is not jointly convex in both $\mathbf{H}$ and $\mathbf{S}^{(i)}$, we present an alternating procedure to find a local optimal solution. That is, we optimize the objective function with respect to one variable while fixing others. The procedure is repeated until convergence.

We follow the strategy employed in the derivation of NMF [12] to obtain the multiplicative update rules for both $\mathbf{S}^{(i)}$ and $\mathbf{H}$ matrices that can be used for finding a local minimum of the optimization problem (2).

The derivative of the objective function with respect to $\mathbf{S}^{(i)}$ is as follows[2]:

$$\begin{aligned} \nabla_{\mathbf{S}^{(i)}}\mathcal{L} = &- \overbrace{2\mathbf{H}^T(\mathbf{W}^{(i)} \circ \mathbf{A}^{(i)})\mathbf{H}}^{[\nabla_{\mathbf{S}^{(i)}}\mathcal{L}]^-} \\ &+ \overbrace{2\mathbf{H}^T(\mathbf{W}^{(i)} \circ \mathbf{H}\mathbf{S}^{(i)}\mathbf{H}^T)\mathbf{H} + 2\frac{1}{2}\eta_i}^{[\nabla_{\mathbf{S}^{(i)}}\mathcal{L}]^+} \\ &= [\nabla_{\mathbf{S}^{(i)}}\mathcal{L}]^+ - [\nabla_{\mathbf{S}^{(i)}}\mathcal{L}]^- \end{aligned}$$

where, $[\nabla_{\mathbf{S}^{(i)}}\mathcal{L}]^+ \geq 0$, $[\nabla_{\mathbf{S}^{(i)}}\mathcal{L}]^- \geq 0$ are non-negative terms.

Using the Karush-Kuhn-Tucker (KKT) complementarity condition [11] for the non-negative constraint on $\mathbf{S}^{(i)}$, $\mathbf{S}^{(i)} \leftarrow$

[2]In the derivation procedure we use the following mathematical properties: $Tr(\mathbf{ABC}) = Tr(\mathbf{BCA}) = Tr(\mathbf{CAB})$; $Tr(\mathbf{A}^T) = \text{Tr}(\mathbf{A})$ and $\parallel \mathbf{A} \parallel_F^2 = Tr(\mathbf{A}\mathbf{A}^T)$; if $\mathbf{A} \geq 0$ then $\parallel \mathbf{A} \parallel_1 = Tr(\mathbb{1}^T\mathbf{A})$;$Tr((\mathbf{W}^T \circ \mathbf{A}^T)(\mathbf{W} \circ \mathbf{B})) = Tr(\mathbf{A}^T(\mathbf{W} \circ \mathbf{W} \circ \mathbf{B}))$

$\mathbf{S}^{(i)} \circ \left(\frac{[\nabla_{\mathbf{S}^{(i)}}\mathcal{L}]^-}{[\nabla_{\mathbf{S}^{(i)}}\mathcal{L}]^+}\right)^{\frac{1}{2}}$, we obtain the following multiplicative update rule for each $\mathbf{S}^{(i)}$ matrix:

$$\mathbf{S}^{(i)} \leftarrow \mathbf{S}^{(i)} \circ \sqrt{\frac{\mathbf{H}^T(\mathbf{W}^{(i)} \circ \mathbf{A}^{(i)})\mathbf{H}}{\mathbf{H}^T(\mathbf{W}^{(i)} \circ \mathbf{H}\mathbf{S}^{(i)}\mathbf{H}^T)\mathbf{H} + \frac{1}{2}\eta_i}} \tag{3}$$

The derivative of the objective function with respect to $\mathbf{H}$ is as follows:

$$\nabla_{\mathbf{H}}\mathcal{L} = \overbrace{-\sum_{i=1}^{N} 4(\mathbf{W}^{(i)} \circ \mathbf{A}^{(i)})\mathbf{H}\mathbf{S}^{(i)}}^{[\nabla_{\mathbf{H}}\mathcal{L}]^-} + \underbrace{\sum_{i=1}^{N} 4(\mathbf{W}^{(i)} \circ \mathbf{H}\mathbf{S}^{(i)}\mathbf{H}^T)\mathbf{H}\mathbf{S}^{(i)}}_{[\nabla_{\mathbf{H}}\mathcal{L}]^+}$$

Again, from the Karush-Kuhn-Tucker (KKT) complementarity condition [11] for the non-negative constraint on $\mathbf{H}$, $\mathbf{H} \leftarrow \mathbf{H} \circ \left(\frac{[\nabla_{\mathbf{H}}\mathcal{L}]^-}{[\nabla_{\mathbf{H}}\mathcal{L}]^+}\right)^{\frac{1}{4}}$, we obtain the following update rule for $\mathbf{H}$ matrix:

$$\mathbf{H} \leftarrow \mathbf{H} \circ \left(\frac{\sum_{i=1}^{N}(\mathbf{W}^{(i)} \circ \mathbf{A}^{(i)})\mathbf{H}\mathbf{S}^{(i)}}{\sum_{i=1}^{N}(\mathbf{W}^{(i)} \circ \mathbf{H}\mathbf{S}^{(i)}\mathbf{H}^T)\mathbf{H}\mathbf{S}^{(i)}}\right)^{\frac{1}{4}} \tag{4}$$

Given randomly initialized non-negative matrices $\mathbf{H}$ and $\mathbf{S}^{(i)}$, we use multiplicative update rules (3) and (4) to find a local minimum of the optimization problem (2).

C. Optimization with natural gradient

Ding *et al.* [10] have proved that the orthonormality constraint imposed on non-negative low-dimensional latent feature matrix, $\mathbf{H}$, in SNMTF leads to a more rigorous clustering interpretation. Moreover, under this constraint, they showed the equivalence between NMF and k-means clustering [10]. We adopt the same idea in our optimization problem (2) by imposing the orthonormality constraint on the low-dimensional non-negative latent feature matrix $\mathbf{H}$: $\mathbf{H}^T\mathbf{H} = \mathbf{I}$. On the other hand, it can be shown that columns of $\mathbf{H}$ matrix span a vector subspace known as Grassmann manifold $\mathbb{G}(k, n)$, i.e., $span(\mathbf{H}) \in \mathbb{G}(k, n)$ [13]. Moreover, Amari in [14] has showed that when an optimization problem is defined over a Grassmann manifold, the ordinary gradient of the optimization function does not represent its steepest direction, but *natural gradient* does.

Therefore, we define a natural gradient to optimize our objective function (2) under the orthornormality constraint. Following Panagakis *et al.* [13], we introduce a tangent vector, $\mathbf{\Delta} \in \mathbb{R}^{n \times k}$ at $\mathbf{H}$ that satisfies: $\mathbf{H}^T\mathbf{\Delta} = 0$. The projection onto the tangent space can be given as: $\mathbf{\Pi}_{\mathbf{H}} = \mathbf{I} - \mathbf{H}\mathbf{H}^T$. By fixing $\mathbf{S}_i$ and using the projection, we can define a natural gradient of $\mathcal{L}(\mathbf{H})$ on Grassmann manifold at $\mathbf{H}$ in the following way:

$$\widetilde{\nabla}_{\mathbf{H}}\mathcal{L} = \mathbf{\Pi}_{\mathbf{H}}\nabla_{\mathbf{H}}\mathcal{L} = \nabla_{\mathbf{H}}\mathcal{L} - \mathbf{H}\mathbf{H}^T\nabla_{\mathbf{H}}\mathcal{L}$$

where, $\nabla_{\mathbf{H}}\mathcal{L}$ is the ordinary gradient, computed as in II-B.

Following the Karush-Kuhn-Tucker (KKT) complementarity condition [11] and preserving the non-negativity of $\mathbf{H}$, we re-write the update rule for $\mathbf{H}$ matrix using the natural gradient: $\mathbf{H} \leftarrow \mathbf{H} \circ \left(\frac{[\widetilde{\nabla}_{\mathbf{H}}\mathcal{L}]^-}{[\widetilde{\nabla}_{\mathbf{H}}\mathcal{L}]^+}\right)^{\frac{1}{4}}$, where $\widetilde{\nabla}_{\mathbf{H}}\mathcal{L}$ can be written in the following way:

$$\begin{aligned}\widetilde{\nabla}_{\mathbf{H}}\mathcal{L} &= [\nabla_{\mathbf{H}}\mathcal{L}]^+ - [\nabla_{\mathbf{H}}\mathcal{L}]^- \\ &\quad - \mathbf{H}\mathbf{H}^T[\nabla_{\mathbf{H}}\mathcal{L}]^+ + \mathbf{H}\mathbf{H}^T[\nabla_{\mathbf{H}}\mathcal{L}]^-\end{aligned}$$

The multiplicative update rule for $\mathbf{H}$ is:

$$\mathbf{H} \leftarrow \mathbf{H} \circ \left(\frac{[\nabla_{\mathbf{H}}\mathcal{L}]^- + \mathbf{H}\mathbf{H}^T[\nabla_{\mathbf{H}}\mathcal{L}]^+}{[\nabla_{\mathbf{H}}\mathcal{L}]^+ + \mathbf{H}\mathbf{H}^T[\nabla_{\mathbf{H}}\mathcal{L}]^-}\right)^{\frac{1}{4}} \tag{5}$$

The update rule for $\mathbf{S}$ stays the same as in (3).

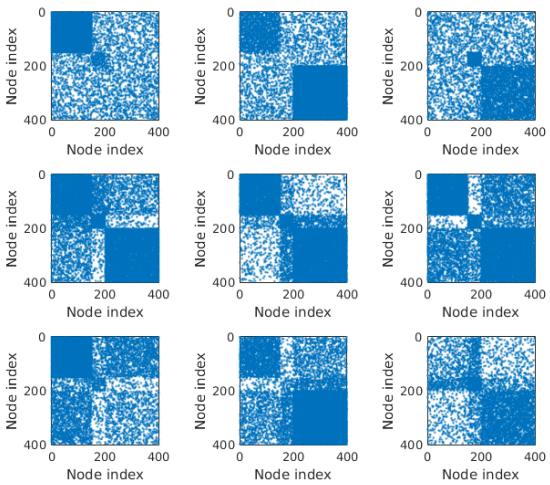

Fig. 1. Spy-plots of three random synthetic multi-layer networks with three layers. (top) SYNT-1 is composed of three complementary layers with communities of different connectivity probabilities and fixed probability of noise between communities. (middle) SYNT-2 is composed of three layers with different levels of noise between communities and fixed connectivity probability of communities. (bottom) SYNT-MIX is composed of three layers generated by combining approaches from SYNT-1 and SYNT-2.

III. Experiments

In this section, we evaluate the performance of our proposed algorithms *WSSNMTF* and *NG-WSSNMTF* on several synthetic and real world multi-layer networks. First, we describe the multi-layer networks that we use for clustering evaluation, and then provide a brief description of algorithms that we use

for clustering comparison. Finally, we present the results in terms of the three different clustering evaluation measures.

A. Datasets

a) Synthetic data generation: We generate three synthetic multi-layer graphs each with $N_l = 3$ different layers (see Fig. 1) by using the planted partition model approach described in [5]. For each multi-layer network, we choose $N = 400$ nodes partitioned into three communities with $N_1 = 150$, $N_2 = 50$ and $N_3 = 200$ nodes respectively in each community. For each layer, we split the corresponding adjacency matrix into blocks defined by the partition. Entries in each diagonal block, are filled with ones randomly, with probability p_{ii}, representing the edge connectivity probability; it is also referred as community's edge density. We also add random noise between each pair of blocks, ij, with probability p_{ij}, representing links between communities. The larger the values of p_{ij} are the harder the clustering is. Similarly, the smaller the values of p_{ii} are the harder the clustering is.

To demonstrate the ability of *WSSNMTF* and *NG-WSSNMTF* to efficiently integrate *complementary* network layers and to demonstrate their *robustness* against different levels of noise we generate three different synthetic random multi-layer networks. *SYNT-1* (Fig. 1, top) has three layers with different community connectivity and fixed noise between communities. Connectivity probabilities of communities in each layer correspond to one cyclic permutation of the following probabilities: $p_{11} = 0.4, p_{22} = 0.2, p_{33} = 0.1$. Noise between communities is $p_{12} = p_{13} = p_{23} = 0.05$ and it is the same in all layers. In that way, in each layer we have only one community that is fully represented, while other two are partially represented. *SYNT-2* (Fig. 1, middle) has fixed probabilities of communities' connectivities (i.e., $p_{11} = 0.4, p_{22} = 0.2, p_{33} = 0.1$) but noise between blocks in each layer corresponds to one cyclic permutation of the following probabilities: $p_{12} = 0.05, p_{13} = 0.03, p_{23} = 0.01$. *SYNT-MIX* corresponds to the mixture of *SYNT-1* and *SYNT-2* and it is generated by permuting both p_{ii} and p_{ij} probabilities when generating different layers.

b) Real data description: We use three different real data multi-layer networks. Two of them are constructed from bibliographic datasets, and the third one is a mobile phone dataset. The first dataset is *CiteSeer* [3] dataset, consisting of 3312 papers belonging to 6 different research categories. We consider these categories as the ground truth classes. We construct two layers: *citation layer*, representing the citation relations between papers extracted from the paper citation records; and the paper *similarity layer*, constructed by extracting a vector of 3703 most frequent and unique words for each paper, and then computing the cosine similarity between each pair of papers. The spy plots of the adjacency matrices of these layers are represented in Fig. 2 (top). The second dataset is a subset of *CoRA* dataset [15] consisting of 1662 Machine Learning papers grouped into 3 different research categories. Namely, Genetic Algorithms, Neural Networks and Probabilistic Methods. We use the same approach as for *CiteSeer* dataset to construct the *citation* and *similarity* layers. The spy plots of the adjacency matrices of these layers are represented in Fig. 2 (bottom). The third dataset we adopt is the MIT Reality Mining Dataset (not shown), which consists of 87 mobile phone users on the MIT campus. The network consist of three layers constructed based on the three types of information: physical location, bluetooth scans and phone calls [7].

[3]http://linqs.umiacs.umd.edu/projects//projects/lbc/

B. Algorithms for clustering multi-layer networks

We adopt two baseline approaches, **SNMTF(e)** and **SNMTF(l)**, that are based on SNMTF algorithm. Since *SNMTF* is a single network method, in order to apply it on multi-layer networks, we consider the *early* and *late* fusion of network layers respectively. In addition to that, we also consider three state-of-the art techniques, namely Co-regularized Graph Clustering based on NMF (**CGC-NMF**) [4], tensor factorization approach based on PARAFAC decomposition with sparse latent factors (**GraphFuse**) [5] and a spectral method based on the subspace representation of multi-layer graphs (**SC-ML**) [8]. Below we briefly explain each of the method and provide the implementation details:

SNMTF(e): we first merge all the network layers into a single network described by the following adjacency matrix: $\mathbf{A} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{A}^{(i)}$. Then we apply *SNMTF* on $\mathbf{A}$. To initialize cluster indicator matrix, we use SVD approach for SNMTF [16]. To extract clusters from the obtained cluster indicator matrix, $\mathbf{H}$, we applied Ncut algorithm [17] on the similarity matrix $\mathbf{H}\mathbf{H}^T$.

SNMTF(l): we first apply *SNMTF* on each network layer, $\mathbf{A}^{(i)}$, separately, and obtain clustering assignment. Then we merge obtained individual clustering assignments into a consensus clustering assignment. We use the same strategy for initialization and clustering extraction as in the case of **SNMTF(e)**.

CGC-NMF: Co-regularized multi-domain Graph Clustering based on symmetric NMF proposed in [4]. In CGC-NMF each domain represents a single network layer. There is a set of parameters $\gamma_{ij} \geq 0$ that balance between single-domain and cross-domain clustering objective for each pair of layers ij. Given that in all our experiments the relationship between node labels for any pair of layers is *one-to-one*, we set $\gamma_{ij} = 1$ (as in [4]) for all pairs of layers and throughout all our experiments. We use the result of SNMTF applied on each network layer separately to initialize the cluster indicator matrices of network layers in CGC-NMF.

SC-ML: Spectral clustering on Multi-Layer graphs proposed in [8]. In SC-ML there is a single regularization parameter, α , that balances the trade-off between two terms in the SC-ML objective function. In all our experiments we choose the value of α that leads to the best clustering performance. In the case of *CiteSeer, CoRA* and *MIT*, the optimal parameters are $\alpha = 0.1$, 0.1 and 0.2, respectively.

GraphFuse: Tensor factorization approach using PARAFAC decomposition with sparse latent factors proposed in [5]. The single parameter, sparsity penalty factor λ, is chosen to lead to the best clustering performance ($\lambda = 0.10$, 0.01 and $1,000$ for *CiteSeer, CoRA* and *MIT*, respectively.

WSSNMTF and **NG-WSSNMTF**: we use the same strategy for matrix initialization and extraction of clusters as described in **SNMTF(e)**. The maximum number of iterations, $max\_iter = 100$, has been proven sufficient for reaching the convergence in all our experiments. The sparsity penalty parameter, η, for both methods, is chosen to lead to the best clustering performance (*WSSNMTF*: $\eta = 1.0$, 0.01 and 1.0); *NG-WSSNMTF*: $\eta = 1.0$, 10.0 and $1,000$ for *CiteSeer, CoRA* and *MIT*, respectively.

C. Clustering accuracy

To evaluate and compare the performance of our proposed methods with above described methods we use the following three widely used clustering accuracy measures: *Purity*, *Normalized Mutual Information (NMI)* and *Adjusted Rand Index (ARI)*. All three measures provide a quantitative way to compare the computed clusters $\Omega = \{\omega_1, \ldots, \omega_k\}$ with respect to the ground truth classes: $C = \{c_1, \ldots, c_k\}$. *Purity* represents percentage of the total number of nodes classified correctly, and it is defined as: $Purity(\Omega, C) = \frac{1}{N}\sum_k \max_j |\omega_k \cap c_j|$ [18], where N is the total number of objects, and $|\omega_k \cap c_j|$ represents the number of nodes in the intersection of ω_k and c_j. To trade-off the quality of the clustering against the number of clusters we use *NMI*. NMI is defined as: $NMI(\Omega, C) = \frac{I(\Omega;C)}{|H(\Omega)+H(C)|/2}$ [19], where I is the mutual information between node clusters Ω and classes C, while $H(\Omega)$ and $H(C)$ represent the entropy of clusters and classes respectively. Finally, *Rand Index* represents percentage of true positive (TP) and true negative (TN) decisions assigns that are correct (i.e., accuracy). It is defined as: $RI(\Omega, C) = \frac{TP+TN}{TP+FP+FN+TN}$, where, FP and FN represent false positive and false negative decisions, respectively. *ARI* is defined to be scaled in range $[0, 1]$ [19]. All three measures are in the range $[0, 1]$, and the higher their value, the better clustering quality is.

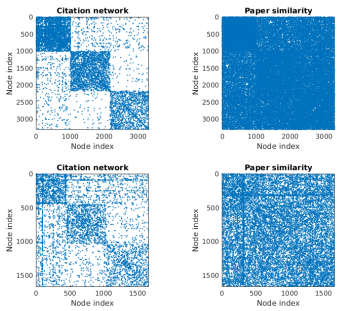

Fig. 2. (top) *CiteSeer* multi-layer network of 3312 papers. Two layers represent citation connections between papers and similarity of papers' 3703 most frequent words, respectively. (bottom) *CoRA* multi-layer network of 1662 papers. Two layers represent citation connections between papers and similarity of papers' 1443 most frequent words, respectively.

D. Results and discussion

The clustering accuracy of the described methods applied on synthetic multi-layer network is shown in Table I, with each column representing a different method. The results are shown in terms of the above described clustering evaluation measures. For each of the three synthetic network, the best result is highlighted in bold. We observe that both *WSSNMTF* and *NG-WSSNMTF* outperform other methods when applied on *SYNT-1*. Recall that *SYNT-1* is designed to have complementary layers; that is, one community is more present in one layer than in other layers. Given that, we can observe that both of our proposed methods are capable of successfully utilising complementary information contained in all layers and achieving the highest clustering results. Performance of our proposed methods on *SYNT-2*, with layers having different levels of nosy links, is also better than the performance of other factorization methods, and comparable to the performance of *SC-ML* spectral clustering method. On the other hand, baseline methods and other factorization methods are characterized with significantly lower clustering performance. The clustering result on the last synthetic network, *SYNT-MIX*, as being the hardest one to cluster, indicate the best performance of our *NG-WSSNMTF*. Comparable results are obtained when using *WSSNMTF* and *SC-ML* methods.

With respect to the real multi-layer networks, shown in Table II, both of our methods perform better than other methods on both *CoRA* and *CiteSeer* multi-layer networks. Specifically, performance of *NG-WSSNMTF* on *CiteSeer* dataset is slightly better than the performance of *WSSNMTF* and *SC-ML*, but it is significantly better than the performance of other factorization methods. As for *MIT* multi-layer networks, performance of both *WSSNMTF* is somehow disappointing in comparison to methods. However, performance of *NG-WSSNMTF* is significantly better than the performance of other methods. On the other hand, note that unlike *CoRA* and *CiteSeer* networks, clustering *MIT* networks is more challenging due to the approximative groundtruth information that is obtained by manual annotation [7].

IV. Conclusion

In this paper we address the problem of community detection in multi-layer networks by proposing two novel methods, *WSSNMTF* and *NG-WSSNMTF*, based on the symmetric non-negative matrix factorization. Both methods work by simultaneously decomposing adjacency matrices representing graph layers while keeping the low-dimensional latent factor matrices the same across the decomposition. In this way both methods ensure to take into account the correlated effect and shared information between network layers. Moreover, both methods are designed to take into account both missing, as well as, noisy links. In terms of clustering accuracy, we demonstrate the superior performance of both of our methods over some baseline and *state-of-the-art* methods on synthetic data. We show that simple averaging of adjacency matrices representing network layers usually leads to the worst

TABLE I
CLUSTERING ACCURACY MEASURES FOR THREE SYNTHETICALLY GENERATED MULTI-LAYER NETWORKS: *SYNT-1* (TOP), *SYNT-2* (MIDDLE) AND *SYNT-MIX* (BOTTOM).

		SNMtF (e)	SNMtF (l)	CGC-NMF	SC-ML	GraphFuse	WSSNMTF	NG-WSSNMTF
SYNT-1	Purity	0.878	0.645	0.798	0.993	0.912	**1.000**	**1.000**
	NMI	0.698	0.166	0.434	0.961	0.717	**1.000**	**1.000**
	ARI	0.705	0.150	0.435	0.983	0.709	**1.000**	**1.000**
SYNT-2	Purity	0.937	0.893	0.892	**1.000**	0.950	**1.000**	**1.000**
	NMI	0.810	0.692	0.699	**1.000**	0.872	**1.000**	**1.000**
	ARI	0.798	0.696	0.674	**1.000**	0.866	**1.000**	**1.000**
SYNT-M	Purity	0.963	0.623	0.663	0.993	0.912	0.998	**1.000**
	NMI	0.885	0.171	0.240	0.966	0.728	0.993	**1.000**
	ARI	0.891	0.178	0.247	0.985	0.707	0.996	**1.000**

TABLE II
CLUSTERING ACCURACY MEASURES FOR THREE REAL WORLD MULTI-LAYER NETWORKS: *CiteSeer* (TOP), *CoRA* (MIDDLE) AND *MIT* (BOTTOM).

		SNMtF (e)	SNMtF (l)	CGC-NMF	SC-ML	GraphFuse	WSSNMTF	NG-WSSNMTF
CiteSeer	Purity	0.558	0.212	0.566	0.630	0.446	0.629	**0.639**
	NMI	0.280	0.014	0.275	0.330	0.159	0.331	**0.337**
	ARI	0.249	0.000	0.248	0.308	0.107	0.316	**0.329**
CoRA	Purity	0.662	0.492	0.687	0.794	0.645	0.796	**0.809**
	NMI	0.345	0.009	0.388	0.496	0.193	0.497	**0.519**
	ARI	0.264	0.001	0.303	0.495	0.187	0.498	**0.526**
MIT	Purity	0.656	0.437	0.663	0.678	0.655	0.575	**0.689**
	NMI	0.440	0.184	0.443	0.460	0.463	0.281	**0.491**
	ARI	0.270	0.037	0.276	0.347	0.316	0.149	**0.395**

clustering performance. While our first method utilizes ordinary gradient to optimize the objective function, the second method uses natural gradient which forces the common low-dimensional latent matrix factors to be orthonormal which ultimately leads to better clustering results. This has been clearly indicated by significantly better clustering performance on real-world multi-layer networks.

ACKNOWLEDGMENT

We wish to thank Xiaowen Dong for providing us with the MIT data.

REFERENCES

[1] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3–5, pp. 75–174, 2010.

[2] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, "Multilayer networks," *Journal of Complex Networks*, 2014.

[3] J. Kim and J.-G. Lee, "Community detection in multi-layer graphs: A survey," *SIGMOD Rec.*, vol. 44, no. 3, pp. 37–48, dec 2015.

[4] W. Cheng, X. Zhang, Z. Guo, Y. Wu, P. F. Sullivan, and W. Wang, "Flexible and robust co-regularized multi-domain graph clustering," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '13. New York, NY, USA: ACM, 2013, pp. 320–328.

[5] E. E. Papalexakis, L. Akoglu, and D. Ienco, "Do more views of a graph help? community detection and clustering in multi-graphs," in *Proceedings of the 16th International Conference on Information Fusion, FUSION 2013, Istanbul, Turkey, July 9-12, 2013*, 2013, pp. 899–905.

[6] W. Tang, Z. Lu, and I. S. Dhillon, "Clustering with multiple graphs," in *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, ser. ICDM '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 1016–1021. [Online]. Available: http://dx.doi.org/10.1109/ICDM.2009.125

[7] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov, "Clustering with multi-layer graphs: A spectral perspective," *IEEE Transactions on Signal Processing*, vol. 60, no. 11, pp. 5820–5831, Nov 2012.

[8] ——, "Clustering on multi-layer graphs via subspace analysis on grassmann manifolds," *Trans. Sig. Proc.*, vol. 62, no. 4, pp. 905–918, feb 2014.

[9] D. Kuang, H. Park, and C. H. Ding, "Symmetric nonnegative matrix factorization for graph clustering." in *SDM*, vol. 12. SIAM, 2012, pp. 106–117.

[10] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix T-factorizations for clustering," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '06. New York, NY, USA: ACM, 2006, pp. 126–135.

[11] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[12] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *In NIPS*. MIT Press, 2000, pp. 556–562.

[13] Y. Panagakis, C. Kotropoulos, and G. R. Arce, "Non-negative multilinear principal component analysis of auditory temporal modulations for music genre classification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 576–588, March 2010.

[14] S.-I. Amari, "Natural gradient works efficiently in learning," *Neural Comput.*, vol. 10, no. 2, pp. 251–276, feb 1998.

[15] P. Sen, G. M. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective classification in network data," *AI Magazine*, vol. 29, no. 3, pp. 93–106, 2008.

[16] C. Boutsidis and E. Gallopoulos, "Svd based initialization: A head start for nonnegative matrix factorization," *Pattern Recognition*, vol. 41, no. 4, pp. 1350 – 1362, 2008.

[17] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000. [Online]. Available: http://dx.doi.org/10.1109/34.868688

[18] Y. Zhao and G. Karypis, "Empirical and theoretical comparisons of selected criterion functions for document clustering," *Mach. Learn.*, vol. 55, no. 3, pp. 311–331, jun 2004.

[19] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.