Do more Views of a Graph help? Community Detection and Clustering in Multi-Graphs

Evangelos E. Papalexakis*, Leman Akoglu†, Dino Ienco‡ §
*School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA
†Dept. of Computer Science, Stony Brook University, NY, USA
‡ IRSTEA, UMR TETIS, Montpellier, France
§Lirmm, Montpellier, France
Email: epapalex@cs.cmu.edu, leman@cs.stonybrook.edu, dino.ienco@irstea.fr

***Abstract*—Given a co-authorship collaboration network, how well can we cluster the participating authors into communities? If we *also* consider their citation network, based on the same individuals, is it possible to do a better job? In general, given a network with multiple types (or views) of edges (e.g., collaboration, citation, friendship), can community detection and graph clustering benefit? In this work, we propose MULTICLUS and GRAPHFUSE, two *multi-graph* clustering techniques powered by Minimum Description Length and Tensor analysis, respectively. We conduct experiments both on real and synthetic networks, evaluating the performance of our approaches. Our results demonstrate higher clustering accuracy than state-of-the-art baselines that do not exploit the multi-view nature of the network data. Finally, we address the fundamental question posed in the title, and provide a comprehensive answer, based on our systematic analysis.**

I. INTRODUCTION

Many data types, nowadays, can be represented by a network in which entities correspond to nodes and relationships between entities correspond to edges between nodes. However, as the data complexity increases the standard definition of a simple graph falls short to represent the complex semantics that reside in real world networks. More specifically, we can have multiple sources of information describing different types of relationships associated with the nodes in a network. For example, a set of users may have various communication channels (e.g. phone, email, messaging, etc.) or researchers in a field may have different dimensions of interaction (co-authorship, citations, using similar keywords). As a result different information networks, involving the same set of objects, can be inferred. In both scenarios users (researchers) are the nodes of the network while each relation (dimensions of interaction) represents a different semantic relationship between two objects in the graph. These multiple semantics (or dimensions) cannot be described with only one simple graph but they may be expressed by a set of different graphs sharing the same set of nodes. These multi-source networks are often referred to as multi-view graphs, multi-dimensional graphs, multi-layer graphs, or simply multi-graphs[18].

Due to the popularity of networks, mining network patterns has become an important task in different domains such as computer science, physics, economy, sociology, biology, and chemistry. One of the most important and challenging research problems that attracts much attention is graph clustering [9], [21], [22]. The goal of this task is to obtain groups of nodes that are similar w.r.t. some structural or node attribute information. Many approaches were proposed in the context of single graph clustering[1], [11], [20] while the problem of clustering multi-dimensional graphs has gained interest only recently[19].

Multi-graph clustering aims to fully exploit the interactions among different dimensions of a given network and is able to take into account the correlations among them, whereas standard approaches that manage each graph independently cannot leverage the correlated information coming from the different dimensions. Moreover, information coming from multiple sources may have different characteristics and value. For example, the citation information among papers is highly valuable for clustering, however it may be quite sparse. On the other hand, the co-term information are plenty, however it may be noisy as two papers having similar terms is not directly indicative that they belong to the same topic (e.g., the term *cluster* in the data mining field or in the cloud computing area). The motivation behind multi-graph clustering is exactly to combine and blend in informative-but-sparse and plenty-but-noisy information holistically to strengthen each other and improve the clustering performance.

In this paper we propose two new methods for clustering multi-view graphs. Our first proposed method, MULTICLUS, is based on an information theoretical approach, where the formulation aims to simultaneously "describe" all the views of the network using as few bits as possible. The developed algorithm strives to find the clustering that can best compress the multi-graph at all views. The advantage of our first method is that it requires no user-defined parameters, i.e. can determine the number of clusters automatically. On the other hand, it can work only with binary, i.e. unweighted, graphs. Our second proposed method, GRAPHFUSE, is based on a tensor factorization approach, which can handle weighted graphs and uses search heuristics to find the best number of clusters. We compare our methods against two baseline strategies on both synthetic and real-world multi-graphs, and show that MULTICLUS and GRAPHFUSE yield superior performance over competitors in all clustering tasks.

The rest of this paper: related work on multi-graph clustering (§II), our problem formulation (§III), proposed methods (§IV & §V), quantitative and qualitative results on both synthetic and real data (§VI), and concluding remarks (§VII).

II. Related Work

Multi-dimensional networks allow for the representation of complex data with different semantic relations between objects. For instance in the context of social network analysis, [3] introduces the problem of community detection over multi-dimensional graphs. The authors model the different relationships between two nodes using different types of edges. Based on this model, they introduce a new community detection algorithm.

In [18], the authors perform interaction analysis among communities over heterogeneous multi-dimensional social networks like Del.icio.us, Flickr, and YouTube. They show the usefulness of this rich representation to model real complex interaction between users. In order to extract interaction behavior from multi-graph data, [4] presents a graph mining approach to extract quasi-clique structures from multi-dimensional graphs. More specifically the work is devoted to extract multi-dimensional coherent quasi-cliques which define clusters of vertices that are densely connected by edges belonging to the same dimension.

A first approach that cope with the issue of clustering multi-dimensional networks is proposed in [23]. In this work a generalization of normalized cut for multi-dimensional graphs is developed. The framework leads to a mixture of Markov chains defined over each dimension of the multi-dimensional graph. In [15] a meta-clustering that deals with multi-dimensional networks is introduced. They do not focus on a specific clustering algorithm that directly deals with the multi-dimensionality, on the contrary they introduce a meta strategy that aggregates the independent clusterings derived by the different dimensions.

In [19] the authors propose a factorization method based on linked matrices to solve the multi-graph clustering problem. In this model, each graph is approximated by a graph-specific factor with a common factor shared by all the graphs. This common factor is used as a link among the different dimensions. In [16] a new variational Bayesian framework for clustering multi-graphs is proposed. This approach is based on a probabilistic generative model based on variational Bayesian estimation. The algorithm is mainly tested over biological networks in which different interaction networks associated with the same set of genes are built. As the method is based on a generative model, the approach requires extra parameters (e.g., hyper-parameter of the Dirichlet distribution).

Our proposed work, in contrast to heuristic approaches, is based on theoretical foundations of information theory and tensor decompositions, and is perfectly suitable for 3-mode multi-graph data (nodes × nodes × views).

III. Problem Definition

A multi-graph $\mathcal{G}$ is a set of m graphs defined over the same set of nodes. More formally, $\mathcal{G} = \{G_l\}_{l=1}^{m}$ where each graph $G_l = (V, E_l)$ consists of the set of nodes V and a set of edges $E_l : V \times V$. n denotes the number of nodes $|V|$.

In Fig. 1(a) a simple example is shown. The multi-graph in (a) is defined over 5 nodes $V = \{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}\}$ connected by 3 different types of edges represented by *solid*, *dotted*, and *dashed* lines. Each dimension represents one of the 3 different edge-semantics and can be associated with a standard adjacency matrix (Fig. 1(b), 1(c) and 1(d)).

Given the above notation, the multi-dimensional graph clustering problem can be stated as follows: Given a multi-graph $\mathcal{G}$, find a partitioning C of the nodes in V such that $\forall_{C_i, C_j \in C} C_i \cap C_j = \emptyset$, and $\bigcup_i C_i = V$. The primary goal of the partitioning is to often optimize an objective function that aims to minimize inter-cluster cross-edges while yielding well-connected dense clusters with high intra-cluster connectivity, at all graph dimensions.

In this work, we formulate (1) a description-length-based, and (2) a tensor-decomposition-based objective function to address this goal. We describe our proposed solutions in detail next.

IV. Our First Attempt: MultiCLUS

As a first attempt for multi-graph clustering, we generalize earlier work on automatic cross-associations [5] so that it can handle multiple graphs at the same time. [2] recently extended cross-associations to attributed graphs. Following similar ideas, we formulate the problem as a data compression task for multiple adjacency matrices, each defined w.r.t. a different view of the network.

More specifically, our solution is based on the Minimum Description Length (MDL) principle [14]. That is, we formulate an objective function based on the total number of bits required to "describe" the multiple adjacency matrices based on a common clustering. We define our formulation in detail next.

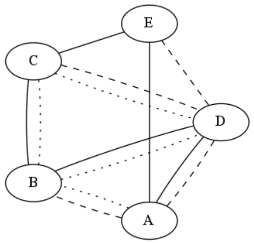
A. Objective Function Formulation

Simply put, MDL is a model selection principle which is based on lossless compression. When regarded as encoding the data by a "sender" to describe it to a "receiver", the formulation consists of (1) model, and (2) data description given the model, such that the "receiver" could fully decode the original data. The goal is to use as few bits as possible such that the description cost is minimized.

As we are dealing with graph clustering, our models consist of a set of possible clusterings. In addition, our data consists of "blocks" in each input adjacency matrix, defined by a given clustering that is the same for all matrices. Our goal then is to find *the* clustering that would minimize the total model and data description cost (in bits). We explain each description cost below.

Model Description Cost consists of encoding the number of node clusters as well as the corresponding assignment of nodes to their respective clusters:

- The number of nodes n requires $\log^* n$ bits, where $\log^*$ is the universal code length for integers [14].
- The number of node clusters k requires $\log^* k$ bits.
- The node cluster assignments with arithmetic coding requires $nH(P)$ bits, where H denotes the Shannon entropy function, P is a multinomial random variable with the probability $P_i = \frac{r_i}{n}$ and r_i is the size of the ith node cluster, $1 \geq i \geq k$. The cluster assignments for all views of the network is the same (shared clustering), and will be described once.



(a)

	A	B	C	D	E
A	0	0	0	0	1
B	0	0	1	1	0
C	0	1	0	0	1
D	0	1	0	0	0
E	1	0	1	0	0

(b)

	A	B	C	D	E
A	0	1	0	1	0
B	1	0	0	0	0
C	0	0	0	1	0
D	1	0	1	0	1
E	0	0	0	1	0

(c)

	A	B	C	D	E
A	0	0	0	0	0
B	0	0	1	1	0
C	0	1	0	1	0
D	0	1	1	0	0
E	0	0	0	0	0

(d)

Fig. 1. (a) Example multi-dimensional graph. Each different dimension (i.e. edge-type) is represented as a matrix: solid (-) (b), dashed (- -) (c) and dotted (..) edges (d).

Data Description Cost consists of encoding the matrix blocks, for each G_l:

- For each block $B^l_{i,j}$, $i, j = 1, ..., k$ and $l = 1, ..., m$, $n_1(B^l_{ij})$ is the number of 1s in the sub-matrix, which requires $\log^* n_1(B^l_{ij})$ bits.
- Having encoded the summary information about the rectangular blocks, we next encode the actual blocks B^l_{ij}. We can calculate the density $P_{ij}(1)$ of 1s in B^l_{ij} using the description code above as $P_{ij}(1) = \frac{n_1(B^l_{ij})}{n(B^l_{ij})}$, where $n(B^l_{ij}) = n_1(B^l_{ij}) + n_0(B^l_{ij}) = r_i \times r_j$, where $n_0(B^l_{ij})$ and $n_1(B^l_{ij})$ are the number of 0s and 1s in B^l_{ij}, respectively. Then the number of bits required to encode each block using arithmetic coding is: $E(B^l_{ij}) = -n_1(B^l_{ij}) \log_2(P_{ij}(1)) - n_0(B^l_{ij}) \log_2(P_{ij}(0)) = n(B^l_{ij}) H(P_{ij})$.

B. Our Objective Function: Total Encoding Cost (length in bits)

$$MDL_{objFunc} = \log^* n + \log^* k - \sum_{i=1}^{k} r_i \log_2(\frac{r_i}{n}) +$$
$$\sum_{l=1}^{m} \sum_{i=1}^{k} \sum_{j=1}^{k} \log^* n_1(B^l_{ij}) + E(B^l_{ij})$$

Our objective function defines the total description, i.e. encoding, cost of our input multiple graphs *given* a clustering. Our aim is to use an algorithm that will find the clustering that *minimizes* the total cost. Finding the optimal clustering with the minimum cost has been stated to be NP-hard in [17]. Therefore, we resort to a heuristic iterative algorithm based on a top-down clustering approach as in [5], with an extension to consider total encoding cost over all input matrices. The main idea is to iteratively increase the number of clusters and reassign each row and column, i.e. node, to the cluster for which the reduction in total cost is the most. As such, the algorithm is greedy and monotonic and often converges to a local optimum. In practice, however, it has been shown to perform quite well on both synthetic and real-world graphs. We refer to [2], [5], [17] for more details.

V. Proposed Method: GraphFuse

In this section, we introduce GraphFuse, a method that treats all different views of a multi-graph as a *tensor*; more specifically, we consider the adjacency matrix of each view of the graph as a different slice of a three-way tensor, and we introduce a method that is able to cluster the given graph, and additionally, identify the influence of each view on each cluster extracted.

Notation A scalar is denoted by a lowercase, italic letter, e.g. x. A column vector is denoted by a lowercase, boldface letter, e.g. $\mathbf{x}$. A matrix is denoted by an uppercase, boldface letter, e.g. $\mathbf{X}$. A tensor is denoted by an uppercase, boldface, underlined letter, e.g. $\underline{\mathbf{X}}$.

Brief introduction to tensors & tensor decompositions. An n-mode tensor is essentially a multidimensional matrix, indexed by n variables. In this work, we focus on three-way tensors, due to their immediate application to the concept of graphs with multiple views. Namely, each slice of a tensor can be viewed as the adjacency matrix of a different view of a particular graph.

Parafac decomposition [8] is a highly popular method for tensor analysis. Specifically, the Parafac decomposition of a tensor $\underline{\mathbf{X}}$ into F rank-one components is $\underline{\mathbf{X}} \approx \sum_{f=1}^{F} \mathbf{a}_f \circ \mathbf{b}_f \circ \mathbf{c}_f$, where $\mathbf{a} \circ \mathbf{b} \circ \mathbf{c}(i, j, k) = \mathbf{a}(i)\mathbf{b}(j)\mathbf{c}(k)$.

Recently, Parafac SLF [13] (SLF stands for sparse latent factors), a variation of the Parafac decomposition was proposed. This decomposition imposes sparsity constraints on the latent factors of the plain Parafac decomposition, i.e. the columns of matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$. By imposing sparsity, essentially one is able to do tensor co-clustering: The non-zeros of the i-th column of $\mathbf{A}$ select which elements of the first mode of the tensor belong to the i-th co-cluster, and so on. For a detailed survey of tensors and tensor decompositions, see [10].

Description of GraphFuse. In this section, we introduce our tensor based approach for multi-graph clustering, called GraphFuse. At the heart of our proposed method lies the Parafac decomposition with Sparse Latent Factors (SLF) [13], which we described above shortly.

Parafac SLF is specifically tailored to *soft* co-clustering, in which we seek to find (possibly overlapping) subsets of rows, columns, and fibers of a given tensor, possibly ignoring some “noisy” data. On the contrary, for the task at hand, we need to assign every node of the graph to one of the available clusters. This task definition, at first, makes the direct application of Parafac SLF to the problem seem inappropriate. However, in this section, we introduce a few modifications to Parafac SLF, in order to make it suitable for *hard* multi-graph clustering. The modifications we applied are the following:

1) PARAFAC SLF allows, by definition, overlapping, i.e. one row, column, or fiber of the tensor to belong to more than one cluster. For the first two modes of the tensor, which correspond to the nodes of the graph, this overlapping freedom needs to be restricted, such that a node belongs to *at most* one cluster. In order to do that, for each node that belongs to more than one clusters, we assign it to the one with the higher weight, i.e. retain the maximum element of each row of $\mathbf{A}$ and $\mathbf{B}$ (line 6 of Algorithm 1). We do not need to do the same for matrix $\mathbf{C}$, since it captures the influence of each graph view on each cluster, and we ideally *require* overlapping effects in this context.

2) Additionally, with PARAFAC SLF being a *soft* technique, some nodes might have been completely ignored in the result, as they may exhibit very low variation (and usually being ultimately noise). In this setting, however, every node has to be assigned to *exactly* one cluster. To this end, we first extract $R-1$ components (line 1 of Algorithm 1). This means that one arbitrary node of the graph either belongs to one of those $R-1$ components (i.e. clusters) or is not assigned anywhere. If the latter occurs, we create an R-th cluster, in which all "left-out" nodes are assigned.

In Algorithm 1 we provide the pseudo-code of our proposed algorithm, as thoroughly described in the previous lines. Vectors $\boldsymbol{\alpha}_I$ and $\boldsymbol{\alpha}_J$ indicate the clusters dictated by the first and the second modes of the tensor respectively. If the graph is undirected, then the two clustering results should be similar, if not identical, but if the tensor captures non-reciprocal relations, then it is natural to expect variations between $\boldsymbol{\alpha}_I$ and $\boldsymbol{\alpha}_J$.

Connection of GRAPHFUSE to LMF [19]. One of the recent existing approaches to multi-graph clustering is introduced in [19], where the authors propose a Linked Matrix Factorization (LMF) model; this approach approximates every view $\mathbf{X}_k$ of the graph as $\mathbf{X}_k \approx \mathbf{P}\mathbf{\Lambda}_k\mathbf{P}^T$ with Frobenius norm regularization on both $\mathbf{\Lambda}_k$ and $\mathbf{P}$. In this work, we show that LMF may, under certain conditions, be expressed as a tensor decomposition which bears certain similarities to our approach but is differentiated in some key points. Nevertheless, it is still of interest to investigate the theoretical similarities of the two approaches.

Lemma 1: If matrices $\mathbf{\Lambda}_k$ of LMF are diagonal, then the LMF model can be expressed as a regularized *symmetric* (in the first two modes) PARAFAC, or regularized INDSCAL [10] model.

Proof: The INDSCAL decomposition is simply a PARAFAC decomposition in which the matrices $\mathbf{A}$ and $\mathbf{B}$ are identical. For simplicity, we drop the regularization terms from all the equations discussed. If we express the k-th slice of the tensor as $\mathbf{X}_k$, then for INDSCAL we can write $\mathbf{X}_k \approx \mathbf{A}\text{diag}(\mathbf{C}(k,:))\mathbf{A}^T$ (where $\text{diag}(\mathbf{C}(k,:))$ creates a diagonal matrix using the k-th row of $\mathbf{C}$). Similarly, LMF approximates each graph view (or slice of the tensor) as $\mathbf{X}_k \approx \mathbf{P}\mathbf{\Lambda}_k\mathbf{P}^T$. If we rename $\mathbf{A}$ to $\mathbf{P}$ (simply a variable substitution), and in the case of matrices $\mathbf{\Lambda}_k$ being diagonal, the optimal solution of the two models is concluded to be the same. ■

Our approach, however, is based on an improvement of the PARAFAC model, which, by imposing sparsity promoting constraints, is able to perform better in terms of clustering quality, thus differentiating itself from the aforementioned models. In [19], the authors do not specify how often the $\mathbf{\Lambda}_k$ matrices are indeed diagonal, however, we deem interesting to point out this connection.

VI. EXPERIMENTS

In this section, we provide both quantitative and qualitative results for our proposed algorithms. For the quantitative evaluation, we compare the clustering quality of our approaches to that of two widely used baselines which do not take advantage of the multi-view nature of the data. To this end, we use 5 different datasets (3 synthetic, 2 real) which we describe next.

A. Data description

Synthetic data generation.: In order to study the performance of our algorithm on different types of graphs, we generated synthetic multi-graphs with various number of views, containing various density views, and with varying clustering quality. Our generative algorithm is based on the planted partitions model [6]. Simply put, given the desired number of nodes in each cluster we split the adjacency matrix into blocks defined by the partitioning. For each block B_{ij}, the user also provides a probability p_{ij}. Using a random process we assign a 1, i.e. an edge, for each possible entry in the block, and 0 otherwise. In other words, p_{ij} specifies the density of each block. We also add noise to all graphs; we use $p_{noise} = 0.05$ in our experiments.

Using the planted partitions model, we generated 3 different multi-graphs. In SYNTHETIC-1, we have 5 views and 5 clusters of various sizes. In view 1, clusters are dense with few cross edges, in views 2-3 the clusters are dense with many cross edges, and the views 4-5 have very sparse clusters with also sparse cross edges. Due to limited space, we show the spy-plots for the views of only SYNTHETIC-2 and SYNTHETIC-3 in Fig. 2. Notice that these two synthetic multi-graphs each have 5 views and 3 clusters, and they share the same clustering. The difference between them is the amount of cross edges, or noise, introduced. By construction, clustering SYNTHETIC-3 is expected to be harder; hence we refer to these multi-graphs as SIM for simple and DIF for difficult to cluster, respectively.

Real data description.: The two real datasets, DBLP-1 and DBLP-2 come from the DBLP online database[1]. More specifically, each of the views of these two datasets corresponds to an author-author graph. In the first view, each edge represents a citation from one author to the other. The second view connects two authors if they co-author at least one paper together. Finally, the third view connects two authors who share at least three terms in the title or abstract of their publications. Both datasets are portions of a larger dataset, manually extracted and labeled. In particular, DBLP-1 contains authors who published in venues STOC+FOCS, AAAI, SIGIR, TODS and DBLP-2 contains those published in venues ICDE, PODS, TKDE, CACM. That is, the ground truth clustering involves 4 author clusters for each of our real multi-networks.

In Fig.s 3 and 4 we illustrate the views for each one of our real datasets.

[1] http://dblp.uni-trier.de/

Algorithm 1: GRAPHFUSE

```
Input: Multi-graph $\mathcal{G}$ in tensor form $\underline{\mathbf{X}}$ of size $I \times J \times K$, number of clusters $R$, sparsity penalty factor $\lambda$.
Output: Assigments to clusters $\boldsymbol{\alpha}_I$ and $\boldsymbol{\alpha}_J$. Matrix $\mathbf{C}$ of size $K \times R$ that shows the contribution of each one of the $K$ views to each one of the $R$ clusters.
1: $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$ = PARAFAC SLF ($\underline{\mathbf{X}}$, $R-1$, $\lambda$).
2: for $i = 1 \cdots I$ do
3:     if $\mathbf{A}(i,:) = \mathbf{0}$ then
4:         $\boldsymbol{\alpha}_I(i) = R$
5:     else
6:         $\boldsymbol{\alpha}_I(i) = \arg\max \mathbf{A}(i,:)$
7:     end if
8: end for
9: Repeat iteration 2-8 for all $J$ rows of $\mathbf{B}$. Labels are output in $\boldsymbol{\alpha}_J$.
```

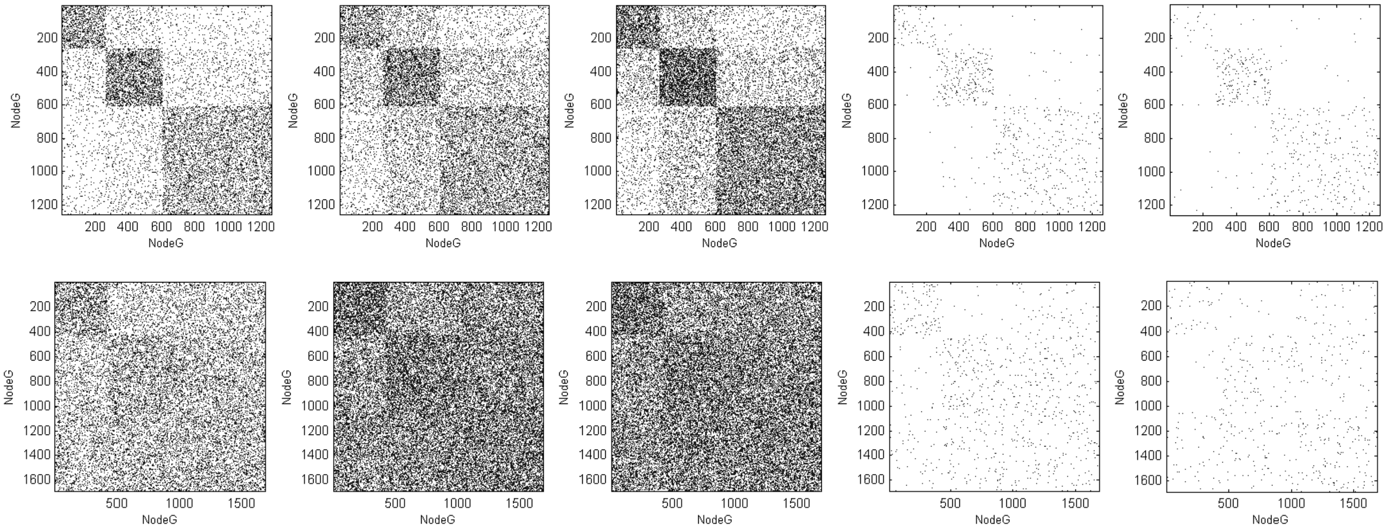
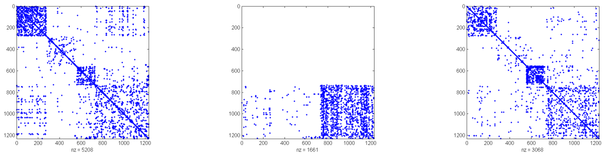
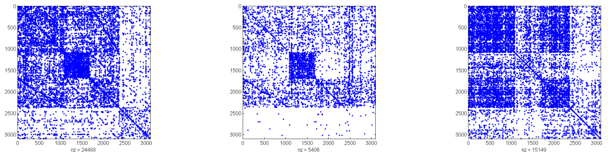


Fig. 2. (top) SYNTHETIC-2 SIM and (bottom) SYNTHETIC-3 DIF share the same clustering scheme, with different amount of cross edges and cluster densities. DIF multi-graph, by construction, is harder to cluster than SIM.



(a) citation (b) co-auth. (c) co-term

Fig. 3. Spy-plots of 3 views in DBLP-1



(a) citation (b) co-auth. (c) co-term

Fig. 4. Spy-plots of 3 views in DBLP-2

B. Clustering accuracy

In order to evaluate the performance of our proposed methods, we use the Normalized Mutual Information, a widely used metric for computing clustering accuracy of a method against the desired ground truth clustering [12]. Moreover, we compare our methods, in terms of NMI, with two baseline approaches, which we briefly describe in the sequel:

BASELINE-1 algorithm sums all the adjacency matrices of a multi-graph obtaining a new aggregate sum-matrix and applies a k-way spectral clustering over this aggregate [20]. The k-way spectral clustering is based on the k-means algorithm that is applied on the Laplacian of the sum-matrix.

BASELINE-2 algorithm first constructs the spectral kernel for each graph view and then sums the spectral kernels summarizing all the dimensions of the multi-graph. Successively, the k-means algorithm is applied to the matrix containing the sum of the kernels in order to obtain the final clustering. Details for this algorithm may be found in [19].

In Table I we show the NMI results on all datasets for all methods. We observe that MULTICLUS always outperforms baseline methods on all synthetic datasets. As for GRAPHFUSE, it has good performance over SYNTHETIC-1 and SYNT-2-SIM while, for SYNT-3-DIF, the results are on par with the baselines. Recall that by construction SYNT-3-DIF is difficult to cluster (see Fig.2 bottom), hence the drop in performance for all methods.

With respect to the real datasets, GRAPHFUSE obtains the best scores over both DBLP-1 and DBLP-2, while MULTICLUS has comparable behaviour with the baselines. We notice that NMI scores are overall lower on real datasets, as they have much less structure than the synthetic ones (see Fig.3) in addition to a lot more noise (see Fig.4). Nevertheless, GRAPHFUSE achieves significantly better accuracy compared to other methods. These encouraging results underline the merits of modeling the multi-graph clustering problem using tensors, as they seem to well exploit the interrelations of the views.

Dataset	Baseline-1	Baseline-2	MultiCLUS	GraphFuse-1	GraphFuse-2
Synthetic-1	0.77 ± 0.11	0.96 ± 0.06	**1** ± 0	**1** ± 0	**1** ± 0
Synt-2-sim	0.68 ± 0.12	0.97 ± 0.11	**1** ± 0	**1** ± 0	**1** ± 0
Synt-3-dif	0.54 ± 0.01	0.56 ± 0.02	**0.90** ± 0.01	0.51 ± 0.17	0.67 ± 0.12
Dblp-1	0.12 ± 0.00	0.08 ± 0.01	0.11 ± 0.01	**0.30** ± 0.02	0.29 ± 0.02
Dblp-2	0.08 ± 0.01	0.04 ± 0.00	0.04 ± 0.00	**0.12** ± 0.02	0.09 ± 0.02

TABLE I. NMI clustering accuracy of proposed methods and competitors on all datasets. Our proposed methods achieve superior performance for all clustering tasks.

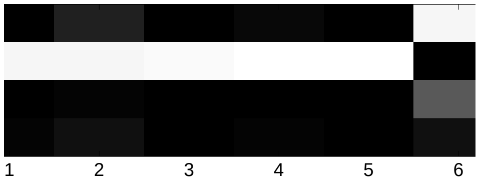


Fig. 5. View-by-cluster matrix **C** by GraphFuse; which essentially encodes the intensity of influence of each of the K views on each of the R clusters. The density of each view is clearly reflected here, but as an interesting future direction, we could potentially identify low quality views by observing how influential they are, according to **C**.

C. Do more graph views help?

This question is one of the fundamental motivations of this work. Simply put, we want to understand whether the addition of more, different views of a given multi-graph is beneficial to the overall clustering quality. As a convention, we assume that all views are given a fixed number from 1 to K.

First, we want to evaluate if the mere presence of more views itself is beneficial, *on average*, for the clustering accuracy. In order to do that, we simply iterate over all possible combinations of $r = 1 \cdots K$ views of varying number and measure the NMI based on GraphFuse. For Dblp-1 and Dblp-2, with only 2 views each, we measured average NMI respectively equal to 0.3037 and 0.0948, whereas adding a third view improved average accuracy to 0.3131 and 0.1208. Note that for this experiment, we report the maximum NMI of the two modes I and J of the tensor (although the trend is followed by both modes).

A second question we address is, how the clustering performance on a set of views R and another set of views C compare to each other, when $C \subset R$. Intuitively we would expect that the set R (the one with more views) allows us to obtain better results, in terms of NMI, than the set C. Our tests of the above hypothesis on Dblp-1 showed that NMI does *not* always increase monotonically with more views. For example, for view-1 and view-2 we obtained NMI=0.2844 where adding view-3 increased NMI to 0.3010. On the flipside, for a different ordering, we obtained NMI=0.3346 for two views and adding the third view caused the NMI to drop to 0.3009. Same behaviour was observed for Dblp-2. This demonstrates that while adding more views helps *on average*, adding a noisy view to a set of informative views might hurt the clustering accuracy for certain cases. A future research direction is to investigate how to carefully select the most informative views of a graph while downgrading the noisy ones.

D. **Data mining case study: RealityMining**

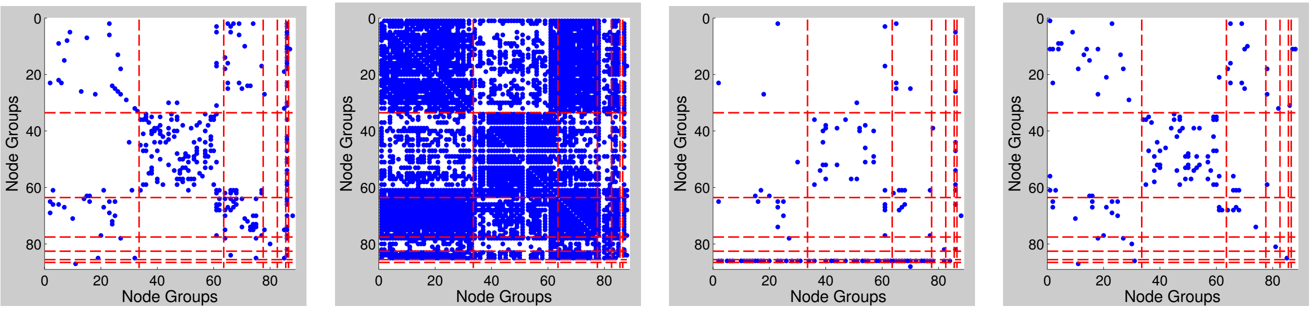
In this section, we provide a data mining case study on the RealityMining dataset. This dataset was introduced in [7] and contains data collected by the MIT Media Lab, including subjects (undergraduate and graduate CS and business students) whose interactions were monitored by a pre-installed piece of software on their mobile devices. The different views offered by the dataset pertain to the means of interaction between a pair of subjects. Namely, `CALL` view refers to subjects calling each other, `DEVICE` view contains Bluetooth device scans, `SMS` view is constructed based on text message exchanges, and `FRIEND` view contains friendship claims.

In Fig.6, we show all four views of the dataset as clustered by GraphFuse where $R = 6$. Qualitatively, we see that the algorithm's output concurs with the communities that appear to be strong on the spy-plots of each view. For example, cluster 2 is a community of business school students that are mostly isolated from the rest of the graph. Another example is cluster 6 of size 1, which contains a single subject with many incoming calls and many outgoing SMSs.

Fig.5 illustrates the view-by-cluster matrix **C** as provided by GraphFuse. The gray-color coding encodes the intensity of influence of each of the K views of the graph on each of the R clusters (lighter means higher influence). We observe that `DEVICE` view captures the most structure for clusters 1-5. On the other hand, views `CALL` and `SMS` respectively have the highest influence on 1-node cluster 6, which is in the same lines with the nature of that subject as we discussed above. All in all, the matrix **C** by GraphFuse could potentially be used in selecting high-quality views of a multi-graph that capture the most clustering structure.

VII. Concluding Remarks

In this paper we address the multi-graph clustering problem, where the goal is to find well-defined clusters across *all* the views (a.k.a. dimensions, layers) of a given graph. We propose two different solutions for clustering multi-graphs, based on Minimum Description Length and Tensor-based decomposition principles, respectively. We validate the effectiveness of our techniques over both synthetic and real Dblp networks, obtaining better clustering accuracy than two competitor methods that ignore the multi-view aspect of the networks. Our case study on the real RealityMining data reveals interesting clusters that agree with human intuition. Moreover, we show that our tensor-based method is a generalization of a recent approach [19], under appropriate conditions. Finally, we asses how the clustering process benefits from the existence of multiple views. In short, the presence of more views is beneficial *on average*, but for particular instances, addition of noisy views may deteriorate clustering quality.



(a) CALL (b) DEVICE (c) SMS (d) FRIEND

Fig. 6. Results on the four views of the REALITYMINING multi-graph. Red dashed lines outline the clustering found by GRAPHFUSE.

This outcome paves the way for interesting research questions, e.g. how to select only informative and non-redundant views of the multi-graph or how to weigh the different dimensions appropriately to obtain the best clustering accuracy.

ACKNOWLEDGEMENTS

The authors would like to thank Prof. Christos Faloutsos for his insightful comments and discussions about this work. We also thank the reviewers for their feedback. Research was sponsored by the Defense Threat Reduction Agency and was accomplished under contract No. HDTRA1-10-1-0120. Funding was provided by the U.S. Army Research Office (ARO) and Defense Advanced Research Projects Agency (DARPA) under Contract Number W911NF-11-C-0088. Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on. Leman Akoglu is partially supported by the Stony Brook University Office of the Vice President for Research.

REFERENCES

[1] Charu C. Aggarwal and Haixun Wang, editors. *Managing and Mining Graph Data*, volume 40 of *Advances in Database Systems*. Springer, 2010.

[2] Leman Akoglu, Hanghang Tong, Brendan Meeder, and Christos Faloutsos. PICS: Parameter-free identification of cohesive subgroups in large attributed graphs. In *SDM*, pages 439–450, 2012.

[3] Michele Berlingerio, Michele Coscia, and Fosca Giannotti. Finding redundant and complementary communities in multidimensional networks. In *CIKM*, 2011.

[4] Brigitte Boden, Stephan Günnemann, Holger Hoffmann, and Thomas Seidl. Mining coherent subgraphs in multi-layer graphs with edge labels. In *KDD*, 2012.

[5] Deepayan Chakrabarti. Autopart: Parameter-free graph partitioning and outlier detection. In *PKDD*, pages 112–124, 2004.

[6] Anne Condon and Richard M. Karp. Algorithms for graph partitioning on the planted partition model. *Random Struct. Algorithms*, 18(2):116–140, 2001.

[7] N. Eagle, A.S. Pentland, and D. Lazer. Inferring friendship network structure by using mobile phone data. *PNAS*, 106(36):15274–15278, 2009.

[8] R.A. Harshman. Foundations of the PARAFAC procedure: Models and conditions for an" explanatory" multimodal factor analysis. *UCLA Working Papers in Phonetics*, 16:1–84, 1970.

[9] Jianbin Huang, Heli Sun, Jiawei Han, Hongbo Deng, Yizhou Sun, and Yaguang Liu. Shrink: a structural clustering algorithm for detecting hierarchical communities in networks. In *CIKM*, pages 219–228, 2010.

[10] T.G. Kolda and B.W. Bader. Tensor decompositions and applications. *SIAM review*, 51(3), 2009.

[11] Jure Leskovec, Kevin J. Lang, and Michael W. Mahoney. Empirical comparison of algorithms for network community detection. In *WWW*, pages 631–640, 2010.

[12] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[13] E.E. Papalexakis and N.D. Sidiropoulos. Co-clustering as multilinear decomposition with sparse latent factors. In *IEEE ICASSP*, pages 2064–2067, 2011.

[14] J. Rissanen. A universal prior for integers and estimation by minimum description length. *The Annals of Statistics*, 11(2):416–431, 1983.

[15] Matthew Rocklin and Ali Pinar. Latent clustering on graphs with multiple edge types. In *WAW*, pages 38–49, 2011.

[16] Motoki Shiga and Hiroshi Mamitsuka. A variational bayesian framework for clustering with multiple graphs. *IEEE Trans. Knowl. Data Eng.*, 24(4):577–590, 2012.

[17] Jimeng Sun, Christos Faloutsos, Spiros Papadimitriou, and Philip S. Yu. Graphscope: parameter-free mining of large time-evolving graphs. In *KDD*, 2007.

[18] Lei Tang, Xufei Wang, and Huan Liu. Community detection via heterogeneous interaction analysis. *Data Min. Knowl. Discov.*, 25(1):1–33, 2012.

[19] Wei Tang, Zhengdong Lu, and Inderjit S. Dhillon. Clustering with multiple graphs. In *ICDM*, pages 1016–1021, 2009.

[20] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[21] Xiaowei Xu, Nurcan Yuruk, Zhidan Feng, and Thomas A. J. Schweiger. Scan: a structural clustering algorithm for networks. In *KDD*, pages 824–833, 2007.

[22] Zhiqiang Xu, Yiping Ke, Yi Wang, Hong Cheng, and James Cheng. A model-based approach to attributed graph clustering. In *SIGMOD*, pages 505–516, 2012.

[23] Dengyong Zhou and Christopher J. C. Burges. Spectral clustering and transductive learning with multiple views. In *ICML*, pages 1159–1166, 2007.