Mining Community Structures in Multidimensional Networks

OUALID BOUTEMINE and MOHAMED BOUGUESSA, University of Quebec at Montreal

We investigate the problem of community detection in multidimensional networks, that is, networks where entities engage in various interaction types (dimensions) simultaneously. While some approaches have been proposed to identify community structures in multidimensional networks, there are a number of problems still to solve. In fact, the majority of the proposed approaches suffer from one or even more of the following limitations: (1) difficulty detecting communities in networks characterized by the presence of many irrelevant dimensions, (2) lack of systematic procedures to explicitly identify the relevant dimensions of each community, and (3) dependence on a set of user-supplied parameters, including the number of communities, that require a proper tuning. Most of the existing approaches are inadequate for dealing with these three issues in a unified framework. In this paper, we develop a novel approach that is capable of addressing the aforementioned limitations in a single framework. The proposed approach allows automated identification of communities and their sub-dimensional spaces using a novel objective function and a constrained label propagation-based optimization strategy. By leveraging the relevance of dimensions at the node level, the strategy aims to maximize the number of relevant within-community links while keeping track of the most relevant dimensions. A notable feature of the proposed approach is that it is able to automatically identify low dimensional community structures embedded in a high dimensional space. Experiments on synthetic and real multidimensional networks illustrate the suitability of the new method.

CCS Concepts: • **Information systems → Clustering;**

Additional Key Words and Phrases: Data mining, social networks, community detection

ACM Reference Format:
Oualid Boutemine and Mohamed Bouguessa. 2017. Mining community structures in multidimensional networks. ACM Trans. Knowl. Discov. Data 11, 4, Article 51 (June 2017), 36 pages.
DOI: http://dx.doi.org/10.1145/3080574

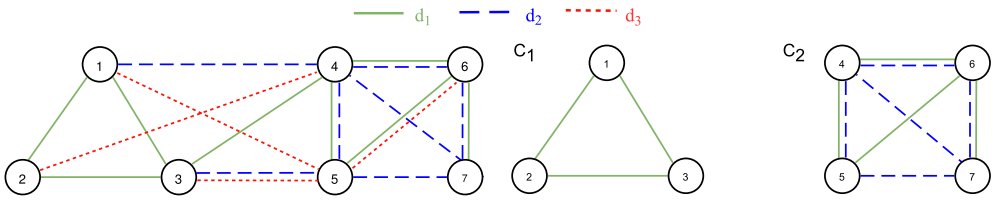
1. INTRODUCTION

A network is an abstract representation of complex interactions in a real world system. With nodes (or vertices) representing entities and edges (or links) mimicking their interactions, several systems, including social, biological, and technological ones could be described by means of a network. Drawing on tools from graph theory, data mining, and social sciences, the analysis of complex interaction patterns in real world systems has attracted increasing attention. One of the most important tasks in network analysis is community discovery where the goal is to find subsets of densely connected or highly interactive nodes [10].

Community discovery has received much attention in the last few years and many approaches have been proposed [16, 49]. A thorough survey about this topic can be

This work is supported by research grants from the Natural Sciences and Engineering Research Council of Canada (NSERC).
Authors' addresses: O. Boutemine and M. Bouguessa, Department of Computer Science, University of Quebec at Montreal, 201 President-Kennedy avenue, Montreal, Quebec, H2X 3Y7, Canada; emails: boutemine.oualid@courrier.uqam.ca, bouguessa.mohamed@uqam.ca.

DOI: http://dx.doi.org/10.1145/3080574



(a) A 3-dimensional network. (b) The multidimensional partition.

Fig. 1. An example of two communities lodging different sub-dimensional spaces.

found in [23] for example. The majority of the proposed approaches, however, are primarily designed for monodimensional networks where pairs of nodes connect through a single edge [53]. Although widely used, this standard representation may fail to capture the complex and rich interaction patterns of real world systems. In fact, entities might engage in different kinds of relations simultaneously. For instance, two friends in a social network might also work for the same company while sharing the same set of interests. Similarly, two authors could be tied by multiple connections, each representing a conference or a journal in which they coauthored a paper. Attempting to fit these complex patterns into a monodimensional network might result in a loss of crucial knowledge. A typical way to deal with this complexity is to qualify the relationships according to their type so that pairs of nodes are connected through multiple edges, in such a way that each edge describes an interaction of a specific kind. These types of networks are often referred to in the literature as multidimensional networks [5], multilayer networks [17], or multiplex networks [3]. For the purpose of convenience, we refer to these kinds of networks as multidimensional networks. As in [5], we also refer to the different interactions between two nodes as dimensions. Finally, we call the projection of the network nodes across one dimension a layer.

Detecting community structures in multidimensional networks is a challenging problem for which a number of approaches have been proposed [6, 17, 45, 53]. Whereas a universally accepted definition of a multidimensional community is far from established, a number of research works consider it as a densely connected group of nodes across all dimensions [1, 53]. Accordingly, a community must exist in every dimension of the network. However, in the multidimensional setting, such a definition appears to be less realistic. This is likely due to the variations in node activity across dimensions. The authors in [44] observed that, in real world networks, dimensions are often given unequal importance by the interacting entities. The involvement of nodes in activities across dimensions was shown to follow a power law distribution where a limited subset of nodes interacts through the whole dimensional space. Nicosia and Latora [44] show that, indeed, the activity of one node in a dimension is often correlated with its activity (or non-activity) in other dimensions. Hence, considering all dimensions on an equal basis might miss the true community structures of the multidimensional network.

Recent works [8, 45], suggested that, in multidimensional networks, a community may exist in a subset of dimensions instead of the whole dimensional space. Specifically, each community C_k is defined by a couple (V_k, D_k), where V_k denotes the nodes forming C_k and D_k denotes the subset of dimensions in which C_k exists. Nodes forming C_k tend to be more densely connected across all the dimensions in D_k than elsewhere. Dimensions in D_k are called the relevant dimensions of C_k, and constitute its subspace, while the remaining ones (that is, dimensions that do not belong to D_k) are called its irrelevant dimensions. A dimension can be relevant to zero, one, or more communities. For the purpose of illustration, consider the three-dimensional network depicted by Figure 1(a).

One would expect two communities to result from this multidimensional network. As shown by Figure 1(b), the first community $C_1 = (V_1, D_1) = (\{n_1, n_2, n_3\}, \{d_1\})$ while the second community $C_2 = (V_2, D_2) = (\{n_4, n_5, n_6, n_7\}, \{d_1, d_2\})$. Nodes forming C_2 are densely connected along dimensions d_1 and d_2. C_1, however, is only defined by d_1. On the other hand, the knowledgeable reader can observe from Figure 1 that dimension d_3 does not contribute to the formation of either community, making it a completely irrelevant dimension. To summarize, from this pictorial illustration, we can see that d_1 is relevant to both C_1 and C_2, d_2 is relevant to C_2 only, while d_3 is not relevant to any community.

In an attempt to discover community structures in multidimensional networks, a number of approaches [11, 30] rely on transformation strategies to learn a monodimensional representation that can be processed by a classical community detection algorithm. Other methods [4, 53], adopt various aggregation schemes to "flatten" the multidimensional network into a simple graph, in which each pair of nodes is connected by a single weighted edge. Community detection techniques for weighted graphs can thus be applied on the aggregated network. Some approaches, such as the one suggested in [53], exploit the idea of ensemble clustering [51] to detect multidimensional communities. In such an approach, the dimensions of the network under investigation are considered separately for community detection. Next, an ensemble strategy is implemented in order to find a consensus by aggregating the extracted communities from distinct layers. Different from these approaches, a few techniques were recently proposed to allow standard community detection algorithms to operate in a multidimensional setting. For instance, a WalkTrap-inspired algorithm was proposed in [31], while several tensor decomposition-based approaches have been described in [22, 37, 45]. A generalization of the well-known modularity measure was also presented in [42] to handle multidimensional networks.

In spite of these advanced techniques, identifying communities in multidimensional networks continues to pose challenges to existing algorithms in various ways. In fact, each of the existing methods suffers from one or even more of the following shortcomings:

(1) Many approaches encounter difficulties when dealing with networks with sparse or irrelevant dimensions, that is, dimensions where nodes are isolated or randomly wired without any apparent community structure. In fact, the presence of such "less informative" dimensions might affect the relevant structural features on other layers which might result in communities that are harder to detect. Referring back to Figure 1, links belonging to d_3 make the separation of the two communities less apparent. Several existing algorithms, especially those that adopt consensus or aggregation strategies, are affected when confronted with this situation. This effect is further aggravated when dealing with communities hiding in very low dimensional spaces, that is, when the number of relevant dimensions is much lower than that of the whole network. Hence, providing an effective way to disregard and alleviate sensitivity to irrelevant links is a critical requirement and a real challenge that needs to be addressed in modern community mining algorithms.
(2) There is a lack of systematic procedures to explicitly identify the relevant dimensions of each community. In fact, as discussed above, a number of algorithms operate under the assumption that the community structure is shared across all layers. In real world networks, however, it is not surprising to notice different patterns of activity across dimensions [3, 44]. Whereas few approaches [11, 45] can estimate the importance of dimensions to the formation of detected communities, a systematic way to discriminate between the relevant and non-relevant groups is still lacking in most proposals. Because it offers an additional level of knowledge, we believe

that finding the associated subspaces of the detected communities is crucial and of a great practical importance.

(3) The vast majority of approaches available in the literature thus far rely on some input parameters that require a proper tuning. In particular, the performance of dimensionality reduction [4, 28, 53], and consensus clustering [1, 51], depends on the selected standard community detection algorithm. Likewise, feature integration [21, 20, 53, 54], and tensor decomposition-based methods [37, 45], rely on various parameters such as the number of structural features or the regularization coefficients. These parameters can seriously affect the clustering accuracy if inappropriate values are provided. Furthermore, many multidimensional communities detection algorithms require the number of communities to be fixed ahead of time. In real world applications, however, it is rarely possible for users to supply accurate values as prior knowledge about the investigated data is not always available.

The aforementioned problems have been tackled separately, and specific approaches have been proposed in the literature, which tend to hardly fit the whole framework. The purpose of this article is, instead, to face the three issues in a unified framework. We introduce a simple, intuitive, yet effective local search technique based on labels propagation. The proposed approach (henceforth MDLPA, short for MultiDimensional Label Propagation Algorithm) is fully automated and requires neither parameters nor prior knowledge to recover the communities and their associated sub-dimensional spaces. MDLPA adopts a novel constrained propagation mechanism that allows it to track the most relevant dimensions while maximizing the number of within-community links across the relevant subspaces. The constraints are expressed through a weighting scheme which assigns a score to each neighbor based on the number of relevant dimensions connecting the pair of nodes. These weights act as gravitational pull, pushing densely connected nodes in a locally correlated subspace to regroup and acquire more neighbors gradually.

We summarize the significance of our work as follows:

(1) We view the task of identifying community structures in multidimensional networks from a label propagation principle perspective, based on which we devise a novel objective function that naturally involves the selection of dimensions in the optimization process. We then devise a constrained propagation mechanism to optimize the proposed quality function. The propagation is an iterative procedure which involves a scoring function that estimates the projected contribution of links from the relevant dimensions.

(2) The developed approach is parameter-free in the sense that it allows the identification of community structures and also their associated relevant dimensions in a fully automatic fashion, without asking the user to set any input parameter, such as an assumed number of communities.

(3) We conduct experiments on synthetic and various real world networks where we demonstrate the performance of our algorithm compared to baseline and other state-of-the-art techniques that have the advantage of using prior knowledge about the network under investigation, such as the number of communities. Furthermore, our experiments show that the proposed approach provides completive results in situations that involve the presence of communities hiding in very low dimensional spaces.

The reminder of this article is organized as follows: In Section 2, we provide a high level description of mainstream algorithms for community detection in multidimensional networks and discuss their strengths and weaknesses. Section 3 describes our

approach in detail. Experiments and performance results on both synthetic and real networks are reported in Section 4. Finally, Section 5 concludes this article.

2. RELATED WORK

In this section, we provide a high-level description of mainstream algorithms for detecting community structures in multidimensional networks. Although a complete survey is out of the scope of this article, we still provide a critical review to give a perspective on the position of this work vis-à-vis existing approaches. Further details and discussions on some existing methods can be found, for example, in [15, 39].

Considered as a baseline for more sophisticated techniques [39], several aggregation-based approaches have been proposed [4, 28, 53]. The key assumption underlying this class of algorithms is that the information of one layer of the multidimensional network reinforces the connectivity of other layers. Hence, one straightforward way to recover the communities is to collapse the multidimensional network into a weighted graph and then process it by a traditional community detection algorithm. Berlingerio et al. [4] presented two aggregation schemes: (1) binary aggregation, in which the set of edges connecting any pair of nodes is substituted by a single edge, and (2) frequency-based aggregation, in which the number of edges connecting two nodes is used as a weight. Tang et al. [53] introduced another strategy where the average weight of all possible edges is used. While generally simpler and computationally more efficient, aggregation-based approaches suffer from three issues: (1) the dependency on the monodimensional community detection algorithm, (2) the sensitivity to irrelevant dimensions which may yield an aggregated graph with less distinct communities, and (3) the structural information loss that might result from the compression.

Treating dimensions on an equal basis, as is performed by *ad-hoc* aggregation schemes, could lead to a final graph that may not reflect the true community structures. A more convenient way would be to discriminate between dimensions based on their relevance to the formation of communities. This is, for instance, the strategy adopted by graph representation learning [30] where the aim is to recover the combined representation that matches, at best, the structural features of the network's dimensions. From a community detection point of view, the goal is to estimate some coefficients that strengthen the ties among nodes belonging to the same community while making them weaker for inter-community pairs. For instance, the authors of [11] introduced a constrained linear regression-based algorithm that finds the optimal combined representation by optimizing the mean squared error of the distance separating the optimal partition from the ground truth partitions of individual layers. The major drawback of this approach lies in the need for the ground truth partitions. To cope with these limitations, a boosting inspired framework named LBGA (Locally Boosted Graph Aggregation) was recently proposed in [30]. LBGA uses a quality function and combines it with a clustering algorithm into a reward system that promotes the presence of within-community edges, allowing it to produce highly modular graphs. Note that LBGA suffers from its dependency on a number of parameters such as, the learning rates and the fixing threshold.

Recovering communities from separate layers and finding a consensus partition is another straightforward way to handle multidimensional networks [53]. This problem has been widely investigated in the literature as ensemble clustering and several appropriate strategies have been proposed, including Cluster-Based Similarity Partitioning Algorithm (CSPA), Hypergraph Partition Algorithm (HGPA) and Meta-Clustering Algorithm (MCLA) [51]. CPSA for example builds an object similarity matrix from each partition based on the memberships to the communities. Afterwards, the resulting matrices are combined according to a similarity metric. Next, a conventional community detection algorithm is applied on the final similarity matrix. Since they were originally

designed to address the problem of finding a consensus among partitions obtained from the same dataset (either using different algorithms or through several runs of the same algorithm with different input parameters), ensemble clustering approaches might perform less competitively when confronted to networks with irrelevant dimensions.

Berlingerio et al. [6] introduced a frequent closed itemsets mining-based approach called ABACUS (frequent pAttern mining-BAsed Community discovery in mUltidimensional networkS). The approach aims at finding possible overlaps between communities in the same or across different layers. ABACUS builds a transactional list of community memberships and processes it afterwards by a frequent closed itemsets mining algorithm [9]. In this model, nodes backing a frequent closed itemset forge a multidimensional community. The main drawback of such a model, however, is that the quality of communities depends on three important factors: (1) the standard community detection algorithm being used, (2) the frequent closed itemsets mining algorithm, and (3) the minimal support threshold which represents the minimum number of nodes for a multidimensional community to be considered.

Similar to consensus clustering, feature integration-based approaches consider dimensions on an individual basis. However, unlike former approaches [6, 53], which combine the partitions, the most important structural features are extracted and combined instead. Tang et al. [52] presented the PMM method (Principal Modularity Maximization) which was further investigated in [53]. The approach combines the spectral features of the individual modularity matrices, that is, the top-n eigenvectors with the largest positive eigenvalues, and then applies singular value decomposition to produce a lower-dimensional embedding, serving as an input to a k-means algorithm. Several related feature integration schemes have recently been proposed. For instance, Dong et al. [21] introduced the SC-ML method (Spectral Clustering on Multilayer Graphs) while a linked matrix factorization-based approach was proposed by Tang et al. [54]. Because they rely on k-means, feature integration-based approaches require the number of communities to be provided by the user.

Recently, several works were focused on the development of multidimensional alternatives from standard approaches. Mucha et al. [42] introduced a generalized definition of modularity named multi-slice modularity. The generalized definition is devised from a null model that is based on the Laplacian dynamics [32]. This definition would allow classical modularity maximization algorithms to operate in the multidimensional setting. For instance, Carchiolo et al. [12] designed a multi-slice modularity maximization approach that is inspired from the Louvain method [7].

More recently, De Domenico et al. [17] proposed a compression-based extension to the Infomap algorithm [48]. The information flow is modeled as a random walk where the best partition is recovered by minimizing a modified map equation. Another approach called LART (Locally Adaptive Random Transitions), which is inspired from the WalkTrap method [46], was proposed by Kuncheva and Montana [31]. LART is based on a discrete-time random walk where transition probabilities are updated according to the local topological similarities among layers. Vertices are then clustered in a hierarchical fashion according to a dissimilarity measure. The advantage of LART and Multiplex Infomap is that they do not require any prior knowledge about the number of communities. Besides, they can handle networks where communities exist in different subspaces. However, their major limitation lies in the input parameters' tuning, namely, the random walk length and the relax rate.

In [26], the authors introduced an extended seed-centric approach called Mux-Licod (Multiplex Leaders identification for community detection). The idea is to identify communities by looking for a special set of nodes, called seeds or leaders, which exhibit a central role in the formation of communities. Mux-Licod adopts an entropy-based

degree centrality for identification of seeds. Specifically, a node is retained as a seed if its centrality ranks first among a majority of similar neighbors. Next, seeds are further clustered according to their similarity with respect to common neighbors. Finally, communities are iteratively built around the final seeds according to the preference of nodes. It is worth noting that Mux-Licod can determine the number of communities automatically. However, this algorithm suffers from its dependence to other input parameters such as the similarity and the neighborhood thresholds.

Boden et al. [8] introduced MiMAG (Mining Multi-layered, Attributed Graphs), a best-first search algorithm partly based on the Quick algorithm [35]. MiMAG is specifically designed to recover 0.5-quasi-cliques, that is, communities with a density of links above 0.5 for each contributing dimension. The idea is to enumerate candidate subsets of nodes in an enumerated tree using a depth first traversal. Each visited subset is then tested for the quasi-clique property. In spite of its ability to detect the subspaces and handle weighted edges, MiMAG's quasi-clique constraint might prevent it from detecting sparser communities. Besides, the algorithm is limited to subspaces of two or more dimensions and only retains communities which are more than 8 nodes in size.

Multidimensional community discovery has also been investigated using tensor decomposition techniques [22, 34, 37, 45]. Naturally, an l-dimensional network corresponds to a third-order tensor of l slices capturing each, the adjacency matrix of one layer. Therefore, one could leverage tensor decompositions to recover community structures hiding in the different subsets of dimensions. A full survey of tensor decomposition methods and their applications is available in [29]. Papalexakis et al. [45] introduced GraphFuse, an approach which can produce a soft co-clustering by extracting overlapping rows, columns and fibers of a tensor. In spite of its ability to measure the relevance of dimensions to the detected communities, GraphFuse requires an adequate parameter tuning as it depends on the sparsity penalty factor and the number of communities.

Finally, it is worth noting that some algorithms such as ABACUS [6], Multislice Louvain [12], Multiplex Infomap [19], and LART [31], tend to produce a partition of the multidimensional network with overlapping community structures across multiple subsets of dimensions. The goal is to identify all densely connected nodes in all subspaces. The output of such algorithms is very large since the same node is allowed to belong to multiple communities in different subspaces at the same time. Although such an approach can discover interesting characteristics of the network, the interpretation of their output remains a challenging task. In this paper, we focus on algorithms that produce disjoint community structures that exist in different subspaces of dimensions. We believe that such a partitioning provides clearer interpretability of the results, as compared to reporting many overlapping communities with densely connected nodes across multiple subspaces.

3. THE MDLPA APPROACH

3.1. Problem Statement

Before describing our approach, let us first introduce some notation and definitions. As in [5], we use multigraphs to represent multidimensional networks. Specifically, let $G = (V, E, D)$ be an undirected and unweighted multigraph, where V is a set of n nodes; D is a set of l dimensions; E is a set of m edges, that is, the set of triplets (v, u, d) such that $v, u \in V$ are nodes and $d \in D$ is a dimension. The triplet (v, u, d) specifies that the two nodes v and u are connected by one edge that belongs to the dimension $d \in D$. Each pair of nodes in G can thus be connected by at most l possible edges.

In this paper, we focus on the problem of discovering a hard partition of G in such a way that every node $v \in V$ belongs to a single community $C_k = (V_k, D_k)$, $k = 1, \ldots, K$, where K is an unknown number of communities; V_k is a subset of V; $D_k \subseteq D$ is a

sub-dimensional space such that nodes in V_k are densely connected in D_k. Recall that the dimensions in D_k are called relevant dimensions for the community C_k. The remaining dimensions, that is, $D - D_k$, are called the irrelevant dimensions for C_k.

MDLPA is focused on discovering disjoint community structures which satisfy the following properties:

(1) A community $C_k = (V_k, D_k)$, $k = 1, \ldots, K$, is a non-empty subset of G.
(2) Within each community $C_k = (V_k, D_k)$, the set of nodes V_k must be more densely connected across all dimensions in D_k than elsewhere.
(3) Each set D_k of C_k should contain a sufficient number of relevant dimensions that distinguish members of the community C_k from other nodes of G.
(4) The subsets of dimensions $\{D_k\}_{k=1,\ldots,K}$ may or may not be disjoint and may have different cardinalities.

The reason for the first property is primarily that we aim to partition the multigraph G into a finite set of disjoint communities $\{C_k\}_{k=1,\ldots,K}$, where each community C_k is defined as a pair (V_k, D_k). The second property ensures that each community C_k is composed by a set of nodes V_k which are closely connected along each dimensions in D_k in comparison to other nodes not in C_k. In other words, nodes in C_k must exhibit a higher internal density of links across all dimensions in D_k. As specified in the third property, every D_k must contain all and only relevant dimensions that are helpful in distinguishing community members. Finally, the last property is based on the fact that communities might exist in different sub-dimensional spaces of the network. The community detection algorithm must support the cases where two or more communities lodge the same or different dimensions of a network. For the purpose of illustration, let us consider the partition in Figure 1. We can claim that both $C_1 = (V_1, D_1) = (\{n_1, n_2, n_3\}, \{d_1\})$ and $C_2 = (V_2, D_2) = (\{n_4, n_5, n_6, n_7\}, \{d_1, d_2\})$ satisfy the four properties.

3.2. Developing an Objective Function

We view the task of discovering communities in multidimensional networks from a label propagation principle, based on which, we devise a novel objective function that guides the search for the optimal partitioning of G. In what follows, we briefly review the label propagation principle and the associated objective function for community detection in monodimensional networks. Next, we describe our novel objective function and the reasoning behind it. A notable feature of the proposed objective function is that it considers both the dimension relevance and the community membership into a single label propagation-based optimization problem, to search community structures.

A well-studied class of approaches for community detection in monodimensional networks is label propagation-based algorithms [2, 33, 38, 47]. A label propagation algorithm (LPA) relies solely on the network structure to guide its community search process. The algorithm starts from an initial state where each node is assigned a unique numerical label that represents its community membership. Then, each node adopts the dominant label among its neighbors. The dominance is dictated by a propagation rule specific to the LPA variant. For instance, the basic LPA [47] selects the most frequent label. The relabeling process keeps repeating asynchronously until all nodes are assigned the dominant label in their neighborhoods. Communities are then extracted from the set of nodes bearing the same label. LPA is simple to implement, fast, and does not need any parameters, including the number of communities, which makes it practical in many contexts.

Barber and Clark [2] presented a mathematical formulation for the basic LPA [47] (hereafter LPAs) where communities are described as a result of the optimization of an objective function. The optimization procedure is based on label propagation whie the objective function measures the number of within-community edges. Formally, for an

undirected monodimensional network G', the objective function $\mathcal{F}$ for label propagation is defined as

$$\mathcal{F} = \frac{1}{2} \sum_{v,u \in V} A_{vu}\ \delta(l_v, l_u), \tag{1}$$

where the A_{vu} are components of the adjacency matrix of G'; l_v and l_u are the community membership labels of nodes v and u respectively; δ is the Kronecker delta, a function that returns 1 if l_v and l_u are equal, that is, when v and u are in the same community. Otherwise, the function returns zero. $\mathcal{F}$ was defined based on the fact that LPAs updates the membership labels so as to increase the number of edges inside a community. Formally, the LPAs optimization procedure is expressed as

$$l'_v = \arg\max_{l} \sum_{u \in V} A_{vu}\ \delta(l_u, l), \tag{2}$$

where l'_v denotes the new community membership label of node v. In case two or more values maximize the sum, arg max should keep the current label l_v of v if it satisfies (2) or otherwise take a random label that maximizes it.

To support the multidimensional setting, one straightforward way to redefine the objective function described by (1) is to sum over all within-community edges irrespective of their dimensions. For a multigraph G, $\mathcal{F}$, given by (1), can be redefined as

$$\mathcal{F}' = \frac{1}{2} \sum_{v,u \in V} \sum_{d \in D} A^d_{vu}\ \delta(l_v, l_u), \tag{3}$$

where the A^d_{vu} are components of the the adjacency matrix of the layer d of G.

In fact, the search for multidimensional communities could be achieved through the optimization of (3) using the label propagation rule in (2). By considering the number of links $\sum_{d \in D} A^d_{vu}$ connecting the pair (v, u) as a weight A_{vu} in (2), the same optimization procedure can be applied to recover the communities on a multidimensional network. This would correspond to the application of LPAs on the aggregated network using the frequency aggregation strategy [4].

Barber and Clark [2] state that identifying monodimensional communities by maximizing (1) does not guarantee a better partition since the global maximum corresponds to the trivial solution where all nodes are assigned to the same community. As Equation (2) produces local changes, the search for the global optimum of $\mathcal{F}$ as described by (1) is subject to being trapped at a local maximum, which corresponds to one possible partition and hence allows it to unfold the communities and eventually avoid the trivial maximum [2]. However, as the number of inter-community edges increases, the undesirable global maximum becomes much harder to avoid. For the multidimensional case, the same observations remain valid as the function described by (3) considers all links on an equal basis. Therefore, any procedure which attempts to maximize $\mathcal{F}'$ as defined by (3) is subject to inheriting the same drawbacks.

In order to avoid the undesirable global maximum of Equation (3), a common approach consists of introducing additional constraints which restrict the search space. Intuitively, since the density of links is higher within the relevant subspaces, nodes belonging to the same community would predominantly use a common subset of dimensions to reach out to each other. Hence, one can identify the relevant subspaces for the communities by looking for the most frequently used dimensions at the node level. Such a group would define the relevant dimensions to the node and can thus be leveraged as an additional constraint over the links connecting neighbors within the same community so that the relevance of the dimensions they represent is also considered. The goal is indeed to seek a partition where members of the same community

connect through a maximum number of links within a common subspace. Equation (3) can thus be redefined as follows:

$$\mathcal{F}_{mult} = \frac{1}{2} \sum_{v,u \in V} \sum_{d \in D} A^{d}_{vu} \, I_{D_v}(d) \, I_{D_u}(d) \, \delta(l_v, l_u), \tag{4}$$

where $D_v \subseteq D$ and $D_u \subseteq D$ denote the subset of relevant dimensions for nodes v and u respectively, and I is the indicator function of the subsets of D. I returns 1 if the dimension d of the link (v, u, d) is part of the relevant group D_v (or D_u) or zero otherwise. Therefore, a within-community link (v, u, d) is considered if and only if d is reciprocally relevant to v and u. Note that Equation (4) holds true for one-dimensional networks when $D = \{d\}$ and $D_v = \{d\}$ $\forall v \in V$, making it consistent with Equation (1).

A notable feature of the developed objective function $\mathcal{F}_{mult}$ is that its value is constituted more by relevant dimensions, which, in turn, facilitates the selection of dimensions based on the particular network properties of different communities and dimensions. Here, it is important to note that any procedure which attempts to maximize $\mathcal{F}_{mult}$ must provide a proper selection mechanism for the relevant dimensions. This brings the question of how to define the relevance to a node and in which sense, since there are several possible ways depending on the mining task at hand. The problem setting, according to the previous section, requires a definition that can locally discriminate between dimensions based on their contribution of links. One suitable metric is the degree centrality as it naturally captures the number of links at the node level. For a node v, we can see how relevant a dimension d is by measuring the degree centrality of v in d. Alternatively, one can use the fraction of neighbors reachable in d with respect to all possible neighbors. This is, for instance, the assumption used by the relevance metrics introduced in [5].

3.3. The Optimization Procedure

In this section, we devise a label propagation-based strategy that involves the selection of relevant dimensions in the optimization of the developed objective function $\mathcal{F}_{mult}$. The procedure adopts a weighting scheme which assesses the affinity of a node to its neighbor according to the number of relevant dimensions connecting a pair of nodes. The weights allow the proposed propagation rule to meet the constraints of Equation (4) by assigning labels so that the number of within-community edges is locally maximized along the relevant subspace. In a nutshell, MDLPA proceeds in two phases:

(1) In the first phase, MDLPA measures for each node v its affinity scores to each one of its neighbors u. These scores, which we also refer to as the attraction weights, are estimated based on the number of relevant dimensions connecting the pair (v, u). The relevance is initially considered from v's perspective, that is, based on the fraction of v's neighbors that can be exclusively reached within the same dimensions as u. The higher the score is, the higher is the number of relevant dimensions to v and the higher is the attraction of u on v. The relevant groups D_v are then selected by looking for subsets of dimensions backing the highest attraction weight on v. Next, the initially estimated scores are adjusted based on D_u so that the relevance to the attracting neighbor u is also considered.
(2) In the second phase, the community structure is recovered using an iterative propagation process. Each relabeling operation involves the previously estimated weights in the membership selection so that the number of within-community edges increases along the relevant subspaces. The relevant dimensions of the winning neighbors are then forwarded to the acquired node so that a consensus is reached over a common subspace. Finally, the attraction weights carried by the processed

node are adjusted once again so that the new relevant dimensions are considered. This process keeps repeating until no further membership updates are possible.

It is important to note that the optimization procedure that we propose is different from the standard label propagation strategies developed for monodimensional networks. Our two-phases optimization strategy is devised to involve dimension selection in the optimization process to automatically identify communities and their relevant dimensions without requesting the user to set any parameters. The proposed weighted scheme allows the search for community structures that may hide in low dimensional space and thus avoid the bias that may be caused by the presence of many irrelevant dimensions in the network under investigation. The experimental results corroborate our claim. The details of each phase of our optimization procedure are given in the next sections.

3.3.1. Phase 1: Initialization. The aim of this phase is to estimate for each node $v \in V$ its affinity to each one of its neighbors u. These scores, which we refer to as the attraction weights, are intended to serve a two-fold purpose: (1) measuring the likelihood of joining u's community, that is, the projected contribution of links within its relevant subspace that results from the addition of v, and (2) identifying the relevant dimensions at the node level, that is the groups D_v.

The main assumption underlying this phase is that nodes sharing memberships to the same community tend to connect more often through the same subset of dimensions. Hence, one would expect more same community neighbors within the relevant dimensions than with the irrelevant ones. Thus, it is possible to recover the relevant dimensions D_v to the node v and therefore its affinity to each neighbor u by simply using the fraction of nodes that can be directly reached within the dimensions connecting the pair (v, u). The more frequent the neighbors are, the more relevant the group is and the higher is the attraction of u on v and vice versa. Consequently, nodes exhibiting high attraction weights internally are likely to end up in the same community.

In what follows, we introduce the weighting scheme adopted by MDLPA to estimate the affinity scores and the corresponding strategy for the selection of the relevant dimensions D_v. The estimation of the attraction weight of a neighbor u on v involves the number and relevance of dimensions connecting the pair (v, u) in a two-step process. Initially, the weight is measured by considering the number of relevant dimensions to v only and later adjusted to reflect their relevance to the neighbor u based on the group D_u. The higher the score is, the higher is the contribution of the node v to u's community.

Initial estimation of the attraction weights: Berlingerio et al. [5] introduced the dimension relevance xOR metric (DR_{xOR}), a function that evaluates the relevance of a group of dimensions $S \subseteq D$ to a node v by measuring the ratio of the neighbors that can be exclusively reached in any subset of S. Formally, the DR_{xOR} is defined as

$$DR_{xOR}(v, S) = \frac{|\eta_{xOR}(v, S)|}{|\eta(v, D)|}, \tag{5}$$

where $\eta(v, D)$ denotes the set of neighbors of $v \in V$ across all dimensions in D and it is defined as $\eta(v, D) = \{u | \exists (v, u, d) \in E \wedge d \in D\}$ and $\eta_{xOR}(v, S)$ is the set of v's neighbors exclusively reachable in any subset of $S \subseteq D$ and it is defined as $\eta_{xOR}(v, S) = \{u | \exists (v, u, s) \in E \wedge s \in S \wedge \forall d \in D - S, \nexists (v, u, d) \in E\}$. The DR_{xOR} returns values in [0, 1] and achieves its maximum when all v's neighbors cannot be reached outside S. This metric offers a set of desirable qualities that makes it suitable for the estimation of the attraction weights and therefore the maximization of $\mathcal{F}_{mult}$ defined by Equation (4). First, the DR_{xOR} can measure the relevance of a group of dimensions simultaneously. In addition, since the η_{xOR} does not consider neighbors that can be reached outside S, the DR_{xOR} favors larger groups of relevant dimensions to v.

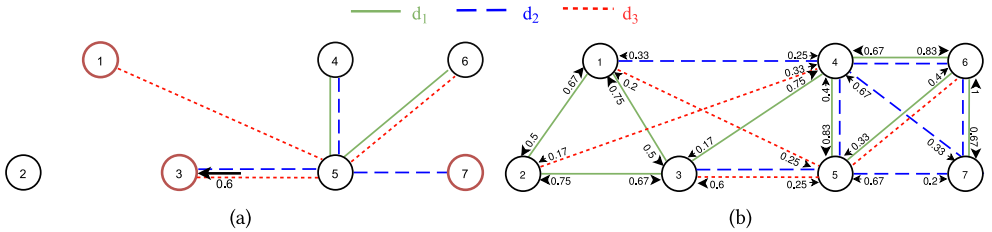


Fig. 2. (a) The initial attraction weight of node n_3 on node n_5. Only nodes $\{n_1, n_3, n_7\}$ can be exclusively reached within $\{d_2, d_3\}$. (b) The initial attraction weights w_0 of each node on each one of its neighbors. The direction indicates the affinity of the source to the destination.

Let $D(v, u) = \{d | \exists (v, u, d) \in E\}$ be the set of dimensions connecting the nodes v and u. We define $w_0(v, u)$, the initial attraction weight of u on v (which also reads the initial affinity score of v to u), by

$$w_0(v, u) = DR_{xOR}(v, D(v, u)) \tag{6}$$

A higher value of $w_0(v, u)$ indicates a higher number of relevant dimensions to v and initially, a higher contribution of links to the community of u. For the purpose of future updates, $w_0(v, u)$ are kept in a non-symmetric adjacency matrix. Figure 2(a) illustrates the initial attraction weight of node n_3 on node n_5. Neighbors of n_5 which respect the condition of η_{xOR} against $D(n_5, n_3)$, that is, that are exclusively reachable in $\{d_2, d_3\}$, are highlighted in bold (specifically n_1, n_3, and n_7).

The initially estimated scores w_0 provide a preliminary partition of the network in the sense that they allow neighbors with high affinity to each other to rapidly reach a consensus over a same community label. However, since the relevance to the attracting neighbors is not considered, w_0 might be biased and the reciprocity constraint of Equation (4) could be violated. For the purpose of illustration, let us consider the estimated weights in Figure 2(b). With the partition in Figure 1(b) in mind, we can clearly see that, except for node n_5, all nodes show more interest to join same community neighbors, as suggested by the highest values of w_0. In contrast, since $w_0(n_5, n_3)$ only reflects the relevance of $\{d_2, d_3\}$ to n_5, its high value would push n_5 to join the community of n_3. However, as both dimensions (that is, d_2 and d_3) are irrelevant to C_1, acquiring n_5 would violate the reciprocity constraint of Equation (4) by bringing links that do not contribute to its formation, which in turn leads to a lower density of links among the group. Consequently, w_0 must be adjusted so that the relevance to the attracting neighbors is also considered.

Adjustment of the attraction weights: As discussed in the previous section, $w_0(v, u)$ defines the contribution of v to u's community based on the number of relevant dimensions within $D(v, u)$ to v. However, as u could express a different distribution of links in its own neighborhood, $D(v, u)$ might be irrelevant to u and likely to its community. Therefore, in order to avoid the likely case in which two neighbors with dissimilar neighborhoods end up in the same community, the number of relevant dimensions within $D(v, u)$ to the neighbor u must also be reflected in $w_0(v, u)$. A possible way to achieve this is to penalize the initially estimated value w_0 by the distance separating $D(v, u)$ from the relevant dimensions D_u to u. Using the Jaccard coefficient, we formally define the adjusted attraction weight $w(v, u)$ of a neighbor u on v by

$$w(v, u) = w_0(v, u) \times \frac{|D_u \cap D(v, u)|}{|D_u \cup D(v, u)|}, \tag{7}$$

The higher the value of w, the higher is the number of relevant dimensions within $D(v, u)$ for v and u jointly. This would provide a relative measure in which the term

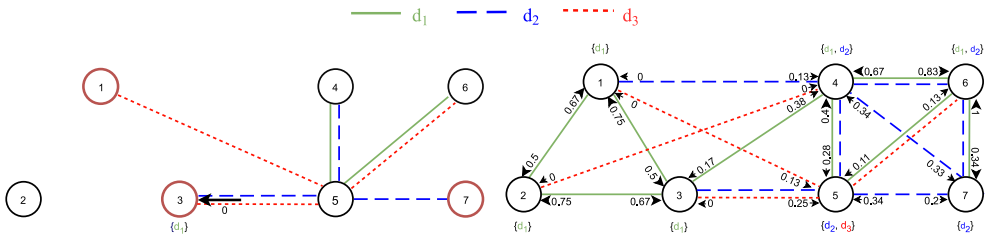


Fig. 3. (a) The attraction weight of node n_3 on node n_5 after the consideration of the relevance of $\{d_2, d_3\}$ to n_3. (b) The relevant groups D_v are leveraged to calculate w for further refinement of w_0. The direction indicates the contribution of the source to the destination's community. Node groups exhibiting high values internally are expected to forge a community.

$\sum_{d \in D} A^d_{vu}\, I_{D_v}(d)\, I_{D_u}(d)$ of Equation (4) is indirectly estimated. Figure 3(a) illustrates the adjusted attraction weight $w(n_5, n_3)$ after considering the relevance of $\{d_2, d_3\}$ to n_3, for which d_1 is the most frequently used dimension and thus its only relevant dimension.

It remains to be specified how to select the relevant dimensions D_u. Recall that w_0 reflects the number of relevant dimensions to v. Hence, one possible way to recover the relevant dimensions D_v is to select the group $D(v, u)$ for which w_0 is the highest. However, as the DR_{xOR} favors larger relevant sets, the selected group might still run a chance of catching irrelevant dimensions. To illustrate this case, consider node n_7 in Figure 2(b). $D(n_7, n_6)$ would be selected as the relevant group D_{n_7} since it defines the highest w_0 score among its neighbors. However, d_1 is less frequently used by n_7 which makes it irrelevant. To avoid this case, a possible solution consists of grouping the neighbors based on $D(v, u)$ and then picking the set backing the neighbors with the highest combined weight. Formally, the initial set of relevant dimensions D_v^{init} is defined as

$$D_v^{init} = \arg\max_{S} \sum_{u \in \eta(v,D)} w_0(v, u)\, \delta(D(v, u), S). \tag{8}$$

Note that in the case where two or more sets maximize the sum, arg max should take their union.

Besides providing a mechanism to distinguish between same community neighbors, the relevant groups D_v of the nodes forming a community C_k could also be leveraged to recover its relevant subspace D_k. The next section introduces a strategy to achieve a consensus over a subspace D_k from the relevant groups D_v. Figure 3(b) illustrates the relevant groups D_v according to (8), along with the corresponding attraction weights w. Communities C_1 (defined by n_1, n_2, and n_3) and C_2 (defined by n_4, n_5, n_6, and n_7) are better separated after the update, as depicted by the relatively low weight values on inter-community edges (Figure 3(b)). The major advantage of such a scheme is that it provides a relative measure in which the contribution of relevant dimensions is indirectly estimated, solely, using the local distribution of edges. The summary of the initialization phase is outlined in Algorithm 1.

3.3.2. Phase 2: Communities Discovery. Following the weighting strategy described in the first phase, we introduce a new label propagation rule which exploits the estimated weights w in the selection of the community membership labels so that Equation (4) is maximized. The rule searches for the optimal partition by changing the memberships of nodes to the neighboring community that carries the highest attraction weight w,

ALGORITHM 1: Initialization

Input: $G(V, E, D)$
Output: w_0, w and $\{D_v^{init}\}_{v \in V}$
begin
  foreach $v \in V$ **do**
    foreach $u \in \eta(v, D)$ **do**
      Calculate $w_0(v, u)$ according to (6);
    end
    Select D_v^{init} according to (8);
  end
  foreach $v \in V$ **do**
    foreach $u \in \eta(v, D)$ **do**
      Calculate $w(v, u)$ according to (7);
    end
  end
  Return w_0, w, and $\{D_v^{init}\}_{v \in V}$;
end

as it would define the highest value of the term $\sum_{d \in D} A_{vu}^{d}\, I_{D_v}(d)\, I_{D_u}(d)$ in Equation (4). Formally, the membership update rule can be expressed as

$$l'_v = \arg\max_{l} \sum_{u \in \eta(v,D)} w(v, u)\, \delta(l_u, l). \tag{9}$$

When two or more competing communities satisfy Equation (9), arg max should select a random label from the dominant group irrespective of the current label of v. Such a change of affiliation would allow the algorithm to search for better solutions by performing a random walk when a plateau of Equation (4) is reached.

Initially, we assign a unique community label l_v for each node v. Then, nodes are processed asynchronously according to Equation (9). With each relabeling step, we update the selected D_v groups and the attraction weights w. The goal is to select the dimensions that are relevant to the acquired node and its new community jointly by finding a shared relevant subspace between the groups. Such a consensus can be achieved by intersecting the relevant groups D_u of the winning neighbors u, with those connecting them to the acquired node, that is, $D(v, u)$, as they would already define the highest values of w and therefore the most relevant dimensions to v and its new community jointly. Consequently, irrelevant dimensions possibly caught in the first rounds will get pruned when more nodes join the community. Formally, the updated relevant dimensions D_v of v with respect to the winning neighbors are defined as

$$D_v = \left[\cup_{u \in \eta(v,D) | l'_v = l_u} D(v, u) \right] \cap \left[\cup_{u \in \eta(v,D) | l'_v = l_u} D_u \right]. \tag{10}$$

Following the update rule in Equation (10), the attraction weights w of the processed node v on its neighbors u are adjusted based on its new relevant group D_v. Once the whole network is processed, the algorithm checks whether every node is assigned to a dominating community label according to Equation (9). If it is not the case, a new propagation round starts and the update rules in Equations (7), (9), and (10) are applied iteratively. The communities start to build up from small regions of nodes with similar neighborhoods and then gradually acquire more neighbors from the same subspace, till the boundary of another competing community is reached. In such a case, the community which receives the highest number of links within its subspace would determine the membership of the node. Once the stop criterion is satisfied, the algorithm halts and the resulting communities are recovered from the vertices bearing

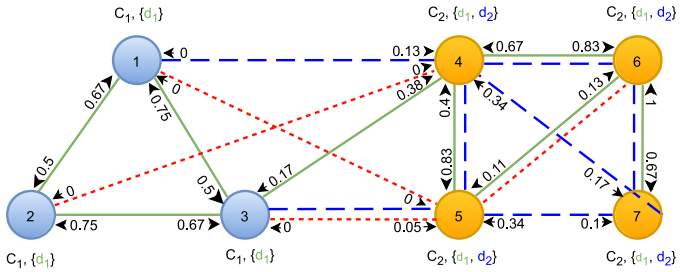


Fig. 4. The final partition as recovered by MDLPA with the refined attraction weights. Nodes $\{n_1, n_2, n_3\}$ are all bearing the same community label C_1 within dimension d_1. The same goes for nodes $\{n_5, n_6, n_7, n_8\}$ which form the community C_2 in $\{d_1, d_2\}$.

ALGORITHM 2: Communities discovery

Input: $G(V, E, D)$
Output: $C_k = (V_k, D_k),\ k = 1, \ldots, K$
begin
  $V_k = D_k = \emptyset$;
  Estimate w_0, w, $\{D_v^{init}\}_{v \in V}$ according to Algorithm 1;
  foreach $v \in V$ **do**
    Assign a unique community membership label l_v for v;
  end
  // Identification of l_v and D_v for each $v \in V$
  while $\exists v \in V$ *such that* l_v *is different from the dominant label in* $\eta(v, D)$ **do**
    Shuffle V;
    foreach $v \in V$ **do**
      Update l_v according to (9);
      Update D_v according to (10);
      foreach $u \in \eta(v, D)$ **do**
        Update $w(u, v)$ according to (7);
      end
    end
  end
  Based on the estimated l_v, regroup nodes into K subsets $\{V_k\}_{k=1,\ldots,K}$;
  // Identification of the set relevant dimensions D_k associated to each V_k
  foreach V_k **do**
    foreach $v \in V_k$ **do**
      $D_k = D_k \cup D_v$
    end
  end
  Return $C_k = (V_k, D_k)$;
end

same community labels. The sub-dimensional spaces D_k defining each community C_k can thus be obtained by merging the relevant dimensions D_v of all members. Figure 4 illustrates the final partition of the network depicted in Figure 1(a) along with the community labels and the adjusted weights after two rounds of propagation. We can see that the relevant dimensions of nodes n_5 and n_7 have been updated to reflect those adopted by their same community neighbors. The steps of the MDLPA approach are depicted in Algorithm 2.

3.4. Complexity Analysis

The computational complexity of MDLPA depends mainly on the initialization phase and the community discovery phase. In the following, we discuss the complexity of each phase.

3.4.1. Complexity Analysis of Phase 1. The running time of this phase is mainly affected by the total number of neighbors in the multigraph G. For each node v, the maximum possible number of neighbors $|\eta(v, D)|$ is equal to the sum of v's degrees at distinct dimensions $deg(v) = \sum_{d \in D} deg_d(v)$, that is, when each direct neighbor is reachable through a single dimension.

For each $u \in \eta(v, D)$, in order to compute $w_0(v, u)$, we need to perform $deg(v)$ verifications against $D(v, u)$ and hence a worst-case time of $O(deg(v))$. Since we have $deg(v)$ neighbors, the maximum possible number of dimension groups $D(v, u)$ is equal to $deg(v)$, which yields a worst-case time of $O(deg^2(v))$ for w_0 to be computed for all neighbors. Therefore, the overall worst-case time for a connected multigraph G is equal to $O(M_1)$, where $M_1 = \sum_{v \in V} deg^2(v)$ denotes the first Zagreb index [25]. For simple graphs, several upper bounds for M_1 have been proposed [36], including $2nm - n^2 + n$. Since we assume a worst-case of a single edge to each neighbor, the inequality $M_1 \leq 2nm - n^2 + n$ holds for G as $m = \frac{1}{2} \sum_{v \in V} \sum_{d \in D} deg_d(v)$ and denotes the total number of edges. Therefore, computing w_0 is performed in $O(mn)$ at most.

Initializing the community membership labels l_v is achieved in $O(n)$ while selecting the relevant dimensions D_v requires $O(m)$. At each node v, the neighbors u are first grouped according to $D(v, u)$ which requires $O(deg(v))$. Next, the group with the maximum combined attraction weight is selected and its dimensions group is assigned to v, which requires a worst-case time of $O(deg(v))$ and an overall time of $O(m)$. Finally, computing w for each node v requires $O(deg(v))$ and hence an overall time of $O(m)$. As a result, the labels initialization and the estimation of w is performed in $O(m + n)$. Therefore, the whole initialization phase requires a worst-case time of $O(mn)$.

3.4.2. Complexity Analysis of Phase 2. In this phase, each propagation iteration is performed in a near linear time to the number of edges m. At each node v, we first group the neighbors u according to their community label. We then pick the group with the maximum combined attraction weight and assign its community label to v, requiring, hence, a worst-case time of $O(deg(v))$. The same complexity holds when updating D_v and w and therefore a worst-case time complexity of $O(m)$. Although the number of iterations is unknown *a priori*, our experiments show that more than 97% of nodes are correctly classified (assigned to the dominant community label) by the 3rd iteration, regardless of the number of nodes or dimensions (for $l > 1$), which is consistent with the results of LPAs on simple graphs [47]. Shuffling the nodes list requires $O(n)$ while making the convergence check needs a worst-case time of $O(m)$. Hence each iteration requires a worst-case time of $O(m + n)$.

Consequently, the overall complexity of MDLPA in its current implementation is $O(mn)$, making it linearly scalable with the number of edges m, irrespective of the increasing number of dimensions l, provided that the number of nodes n is constant. Also note that when l is constant, the way w_0 are estimated can be modified so that the worst-case time becomes $O(m)$. With l dimensions, we obtain a maximum of $2^l - 1$ possible dimension groups between any pair of adjacent nodes. Thus, computing the initial attraction weights w_0 on v for all possible groups requires $O(2^l deg(v))$. It straightforwardly follows that the complexity on the whole multigraph is $O(m)$ for constant l. Consequently, the new worst-case time of the algorithm becomes $O(m + n)$. Thus, when the number of dimensions l is fixed, the algorithm scales almost linearly

in the number of edges m. The favorable computational efficiency of label propagation-based algorithms is thus preserved.

4. EMPIRICAL STUDY

In this section, we conduct a series of experiments to evaluate the suitability of MDLPA. The aim is to study the performance of the proposed approach in comparison to other competitor algorithms in different settings. To this end, we use various synthetic and real world networks. In what follows, we first describe the experimental setting and the selected performance metrics. Next, we report the results and provide discussions.

4.1. Experimental Setting

4.1.1. Compared Algorithms. To demonstrate the suitability of MDLPA, we compared it with various approaches belonging to three different categories of algorithms: (1) aggregation-based, (2) feature integration-based, and (3) ensemble clustering. For the aggregation-based side, we implemented two aggregation strategies, namely, the binary-based and the frequency-based aggregation scheme [4, 28]. We elected to use the basic label propagation algorithm proposed in [47] to recover the community structures on the aggregated network. For the feature integration-based camp, we selected the PMM[1] method [53] and the SC-ML[2] algorithm [21]. For the third category, we considered the ensemble clustering techniques proposed in [51][3]. As a base algorithm, we used the Louvain method [7], a parameter-free approach which detects the communities by maximizing the modularity [43], in order to uncover community structures (partitions) in each layer of the multidimensional network under investigation. To obtain the final partition, we considered the three consensus methods suggested in [51], namely, CSPA, HGPA, and MCLA. We believe that our choice of algorithms covers a variety of community detection approaches. Finally, it is worth noting that for tensor decomposition-based approaches, we initially considered GraphFuse [45] but later disregarded it as the running times were not reasonable. For a 100-dimensional network with 3,000 nodes, the algorithm takes more than a week to recover the partition.

4.1.2. Parameters Tuning and Execution. PMM as well as SC-ML require the number of communities to be specified by the user. In our experiments, the target number of communities for PMM and SC-ML was set to the number of real communities. Furthermore, since PMM and SC-ML are both parameters-laden, we tried various values. Specifically, for PMM [52, 53], we selected the number of structural features in $[5, 14]$ with 1 graded increments. For SC-ML [21], the values of the regularization parameter were taken from $[0, 1]$ with 0.1 graded increments. Recall that, as mentioned in Section 2, PMM as well as SC-ML use k-means to detect community structures. Therefore, in order to avoid initialization bias (random initialization of k-means centroids), we run PMM and SC-ML, for each selected parameter value, 10 times. The combination of parameters that leads to the best average result is kept and the corresponding best, mean, and worst results are also reported.

We have also run the remaining algorithms, that is, aggregation-based, MDPLA, and ensemble clustering, 10 times and reported the worst, the average and the best results. We have performed such repetitive runs mainly due to the nondeterministic nature of these algorithms. In fact, the two aggregation methods (binary-based and frequency-based) considered in the comparison apply the basic LPA [47] to the aggregated network to detect communities. It is known that LPA is nondeterministic and

[1]The implementation of PMM is available from: http://leitang.net/heterogeneous_network.html.
[2]The implementation of SC-ML is available from: http://lts4.epfl.ch/xdong/code.
[3]The implementation of the clustering ensemble approach is available from: http://strehl.com/soft.html.

therefore its application to the aggregated networks may not deliver a unique solution. Our approach suffers also from the same problem since our optimization procedure is based on the label propagation principle. However, as the experimental results will show, the impact that poses this problem to the proposed approach is not significant. Finally, as discussed above, we used the Louvain method as a based detector for ensemble clustering. The Louvain approach is also nondeterministic and may lead to different solutions since, as mentioned in [7], the output of this approach depends on the order in which the nodes are considered. Therefore, we run the ensemble clustering method 10 times. For each run, the final partition is identified using the best results reported by one of the three consensus techniques (CSPA, HGPA, and MCLA) proposed in [51]. Similar to previous approaches, we reported the worst, the average, and the best result of ensemble clustering. We believe that the execution scheme that we have adopted contributes to ensure fairness in comparison.

4.2. Evaluation Criteria

In order to assess the community detection effectiveness, we considered both external and internal criteria. Internal criteria are used when the community detection results are evaluated according to a predefined structure. On the other hand, external criteria are used when the structures of the identified communities are evaluated in terms of the quantities that are computable from the available data. In our experiments, we used internal criteria when the ground truth was available (especially with synthetic networks). In the absence of the ground truth (which is generally the case in real networks) we used external criteria.

4.2.1. Internal Criteria. Among existing measures, we chose to use the Normalized Mutual Information (NMI) [41], the Adjusted Rand Index (ARI) [27] and the Fowlkes-Mallows Index (FMI) [24]. The NMI, ARI, and FMI evaluate the community detection results by calculating the correspondence with the reference partition R (that is, the ground truth) and the obtained partition O (that is, the partition generated by a community detection algorithm). The NMI is defined as

$$NMI = \frac{2I(R, O)}{H(R) + H(O)}, \tag{11}$$

where $I(R, O) = H(R) - H(R|O)$ is the mutual information between R and O, $H(R)$, and $H(O)$ are the Shannon entropy of R and O respectively, and $H(R|O)$ is the conditional entropy of R given O.

ARI and FMI are defined as

$$ARI = \frac{2(ad - bc)}{(a + b)(b + d) + (a + c)(c + d)}, \tag{12}$$

$$FMI = \frac{a}{\sqrt{(a + b)(a + c)}}, \tag{13}$$

where a, b, c, and d denote, respectively, the number of node pairs that are in the same community in both R and O, in the same community in R but not O, in the same community in O but not R, and in a different community in both R and O. The more similar the two partitions, the larger the values of NMI, ARI, and FMI. When R and O are identical, the indices' values will be one. When O is only as good as a random partition, the indices' values will be close to zero. We believe that the comparison of the values given by these three different metrics gives us an objective criterion for evaluating the results of competing algorithms.

4.2.2. External Criteria. In our experiments, we considered local and global external metrics to evaluate the community structures. In a nutshell, as described in [39], local metrics are based on the assumption that a community has weak interactions with their neighboring nodes. In this setting, the evaluation of a community is isolated from the rest of the network. On the other hand, global measures consider the quality of the communities and their interactions among communities. In the following, we describe two local metrics and one global measure that we have considered in the experiments.

The first local metric that we have implemented is the redundancy proposed by Berlingerio et al. [4]. For a given community C_k, the redundancy $\rho(C_k)$ is defined as

$$\rho(C_k) = \frac{1}{|DF_k||SP_k|} \sum_{(v,u)\in SP'_k} \sum_{d\in DF_k} A^d_{vu}, \tag{14}$$

where DF_k denotes the set of dimensions found in C_k, SP_k designates the set of pairs (v, u) connected by at least one dimension in C_k, $SP'_k \subseteq SP_k$ represents the set of pairs connected by at least two dimensions. $\rho(C_k)$ measures the ratio of the number of edges connecting adjacent nodes, in at least two dimensions, to the theoretical maximum number of edges between all pairs of nodes. $\rho(C_k)$ returns values in $[0, 1]$ and reflects the redundancy of connections: the more dimensions that connect each pair of nodes within a community, the higher the redundancy will be [4]. For a given partition $P = \{C_1, C_2, \ldots, C_k\}$, the redundancy can be obtained by measuring the average redundancy among communities $\{C_k\}_{k=1,\ldots,k}$ in P [26]. Formally, $\rho(P)$ is defined as

$$\rho(P) = \frac{1}{|P|} \sum_{C_k\in P} \rho(C_k). \tag{15}$$

The second selected local metric is the multidimensional community density which measures the connectedness within communities [6]. This metric corresponds to the number of edges in a community normalized by the maximum possible for that community. Formally, for a given community C_k, the multidimensional community density $MCD(C_k)$ is defined as:

$$MCD(C_k) = \frac{\frac{1}{2}\sum_{v,u\in V_k}\sum_{d\in DF_k} A^d_{vu}}{|DF_k||V_k|\frac{|V_k|-1}{2}}. \tag{16}$$

$MCD(C_k)$ returns values in $[0, 1]$ such that the largest values indicate high connectedness within C_k. For a given partition $P = \{C_1, C_2, \ldots, C_k\}$, the multidimensional density of P can be obtained by measuring the average multidimensional community density among communities $\{C_k\}_{k=1,\ldots,k}$ in P. Formally, $MCD(P)$ can be defined as

$$MCD(P) = \frac{1}{|P|} \sum_{C_k\in P} MCD(C_k). \tag{17}$$

It is important to make the distinction between the set DF_k, used in (14) and in (16), and the set D_k that our approach aims to discover. In fact, as mentioned above, DF_k simply designates the set of all the dimensions found in a community C_k, while D_k corresponds to the set of relevant dimensions of C_k. Accordingly, $D_k \subseteq DF_k$. To illustrate the difference, consider again the network depicted by Figure 1, which contains two communities. As mentioned previously, $C_1 = (V_1, D_1) = (\{v_1, v_2, v_3\}, \{d_1\})$, while $C_2 = (V_2, D_2) = (\{v_4, v_5, v_6, v_7\}, \{d_1, d_2\})$. On the other hand, as can be seen from Figure 1, the set of dimensions found in C_1 is $DF_1 = \{d_1\}$, while the set of dimensions found in C_2 is $DF_2 = \{d_1, d_2, d_3\}$. In our experiments, we used DF_k to calculate $\rho(C_k)$ and $MCD(C_k)$ for the communities generated by all algorithms considered in the comparison (including

ours). Furthermore, since our algorithm has the advantage of automatically identifying the relevant dimensions of each community and in order to illustrate the impact of the selection of relevant dimensions on the quality of results, we considered also D_k in the calculation of $\rho(C_k)$ and $MCD(C_k)$. In this case, we simply replaced DF_k by D_k in (14) and in (16) and show the results only for MDLPA since the other compared algorithms are not able to identify relevant dimensions of the detected communities.

Relying solely on local metrics, such as the redundancy and multidimensional community density, might fail in providing a fair assessment of the identified community structures. In fact, a partition where communities are made out of a single adjacent pair of nodes would result in higher scores for both the redundancy and the multidimensional community density. Thus, one must also consider a global measure that considers both the quality of the communities and the interactions between them. Mucha et al. [42] proposed a generalized modularity metric Q for multidimensional networks which they call the multi-slice modularity. Formally, for a given multigraph G, Q is defined as

$$Q = \frac{1}{2\mu} \sum_{v,u \in V} \sum_{d,r \in D} \left[\left(A^d_{vu} - \gamma_d \frac{k^d_v k^d_u}{2m_d} \right) \delta(d,r) + \delta(v,u) \sigma^{d,r}_u \right] \delta\left(g^d_v, g^r_u\right), \tag{18}$$

where μ denotes the normalization factor, γ_d is the resolution parameter for dimension d, δ is the Kronecker delta, $\sigma^{d,r}_u$ is the coupling parameter of node u in dimension d to itself in dimension r, k^d_v and k^d_u denote, respectively, the degrees of nodes v and u in dimension d, m_d is the total number of edges in dimension d, and finally g^d_v and g^r_u designates the communities of nodes v and u in dimensions d and r respectively. Note that, in our experiment, we fixed the value of the resolution parameter γ_d as well as the value of the coupling parameter $\sigma^{d,r}_u$ to 1 which corresponds to their default value suggested in [42]. The multi-slice modularity Q returns values in $[0, 1]$ such that highest values suggest that the detected community structure is good.

4.3. Experiments on Synthetic Networks

The goal of the experiments conducted in this section is to evaluate the suitability of MDLPA in terms of (1) accuracy – the aim is to test whether our algorithm, in comparison with other existing approaches, is able to correctly identify multidimensional communities, and (2) efficiency – the aim is to determine how the running time scales with the size and the dimensionality of the multidimensional network. For this purpose, we generated a variety of synthetic networks to simulate various situations, using the network generation model described below.

4.3.1. Network Generation Model. The synthetic networks were generated according to the planted partitions model [14]. The generation process was made parametric to the number of nodes n, the number of communities K, the number of dimensions l, the average community dimensionality l_r, the range of the intra-community density of links $[\vartheta_{intra\_min}, \vartheta_{intra\_max}]$ and the range of the inter-community density of links $[\vartheta_{inter\_min}, \vartheta_{inter\_max}]$. Based on the provided parameters, the communities are planted randomly across dimensions as follows:

(1) First, a shared community structure is planted across a subset of $\frac{l_r}{2}$ relevant dimensions (D_{r1}), randomly selected from D. For each $d \in D_{r1}$, the corresponding adjacency matrix is split into K blocks B_{kd} of different sizes each of which represent a single community. Each generated block B_{kd} keeps the same size regardless of d in D_{r1}. Each block's nodes are wired randomly according to a probability, which is uniformly sampled from $[\vartheta_{intra\_min}, \vartheta_{intra\_max}]$. Inter-community edges are

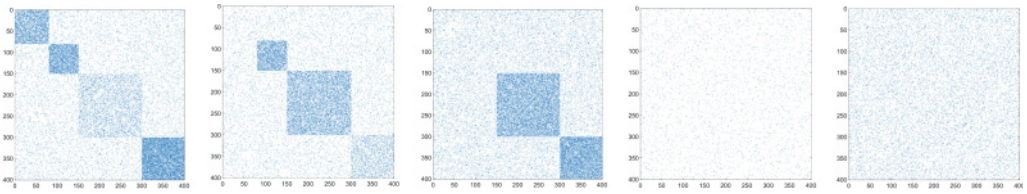


Fig. 5. An example of a five-dimensional synthetic network. Subfigures illustrate the associated adjacency matrices for (a) d_1, (b) d_2, (c) d_3, (d) d_4, and (e) d_5. Each block describes a community projected along a specific dimension.

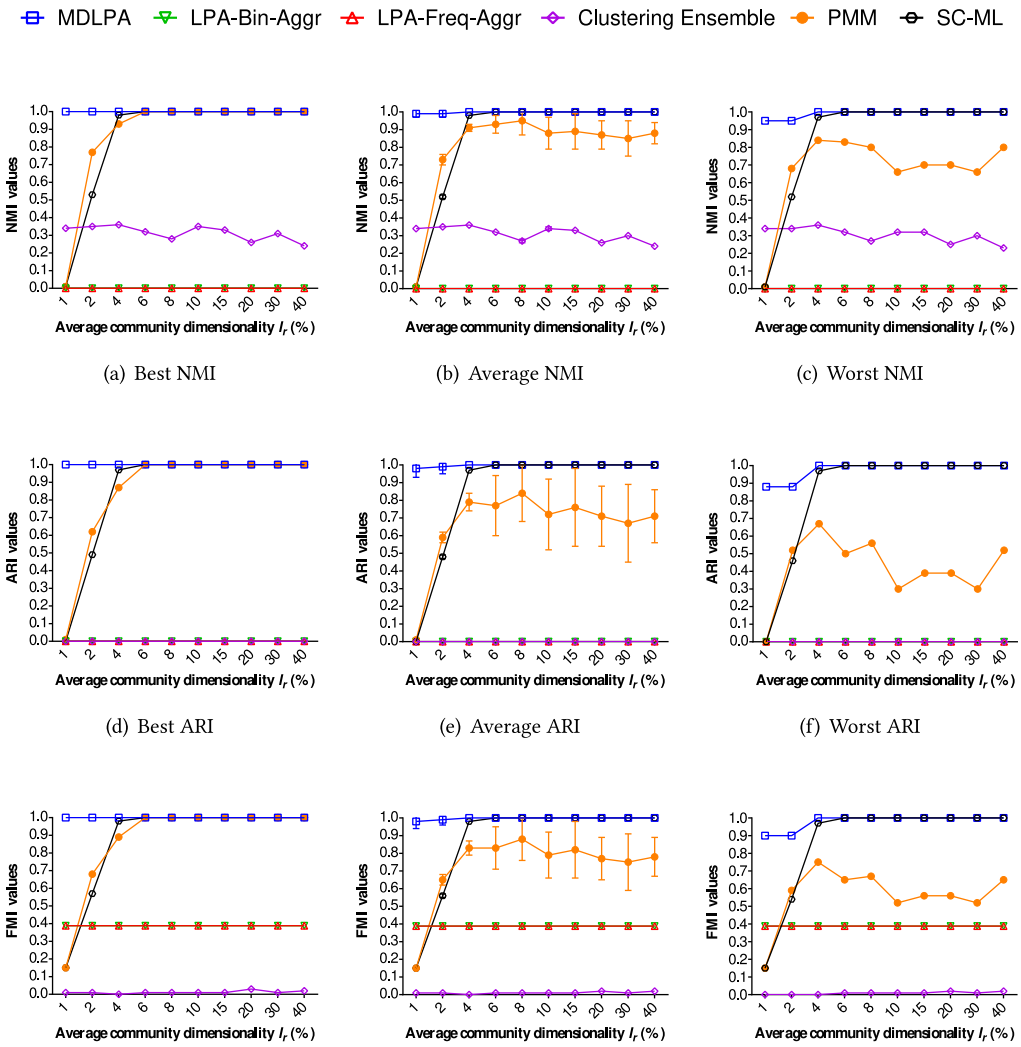
then generated according to another probability, which is uniformly drawn from $[\vartheta_{inter\_min}, \vartheta_{inter\_max}]$.

(2) Next, another subset D_{r2} of l_r relevant dimensions is randomly selected from $D - D_{r1}$. The same community blocks are then generated and randomly planted across the subsets of D_{r2}, such that the average dimensionality among the generated communities is close to l_r. Finally, the adjacency matrices of the remaining irrelevant dimensions are constructed following the Erdös–Rényi model with an edge generation probability uniformly sampled from $[\vartheta_{inter\_min}, \vartheta_{inter\_max}]$ for each layer.

With such a model, the interaction patterns of nodes would differ depending on the dimension and the community memberships in the resulting network. We believe that the synthetic network generation model allows us to simulate various situations, which in turn makes it possible to perform an objective experimental evaluation of MDLPA as well as compared algorithms.

For the purpose of illustration, Figure 5 depicts an example of a five-dimensional synthetic network with 400 nodes split into four communities of different sizes. The number of relevant dimensions of each community varies from one to three. Note that each figure in this pictorial illustration represents the adjacency matrix of the graph corresponding to each layer of the generated multidimensional network. Blocks within each matrix correspond to a community in a specific dimension. As we can see from Figure 5, the four communities exist in different subspaces. For example, the first community (defined by the first block in Figure 5(a)) exists in dimension d_1 only, while the second community (defined by the second block in Figure 5(a) and the first block in Figure 5(b)) exists in dimensions d_1 and d_2. Finally, as depicted by Figure 5(d) and Figure 5(e), no community structures exist in dimensions d_4 and d_5. This example is simple and provided here just for the purpose of illustration. The generated networks used for the evaluation, however, are more complex. The following section describes the networks used in the experiments and the results of compared algorithms.

4.3.2. Quality of Results. The main concern of this set of experiments is to analyze the impact of the community dimensionality on (1) the community detection accuracy and (2) the ability of MDLPA to identify the real relevant dimensions of the detected communities. To this end, we generated 10 different networks with $n = 3{,}000$ nodes and number of dimensions $l = 100$. The average community dimensionality l_r varied from 1 to 40 percent of the whole dimensionality l. The community structure was planted using the following parameters: the range of the intra-community density of links $[\vartheta_{intra\_min}, \vartheta_{intra\_max}]$ was fixed to $[0.2, 0.6]$ while the inter-community density of links $[\vartheta_{inter\_min}, \vartheta_{inter\_max}]$ were taken from $[0, 0.022]$; the number of communities was fixed to seven with a size varying between 10 and 20 percent of the network size n. Finally, note that since, with synthetic networks, the community label of each node is



(a) Best NMI (b) Average NMI (c) Worst NMI

(d) Best ARI (e) Average ARI (f) Worst ARI

(g) Best FMI (h) Average FMI (i) Worst FMI

Fig. 6. Results on synthetic networks.

known beforehand, we have, obviously, ignored these labels when applying competing algorithms, but we have used them together with NMI, ARI and FMI to evaluate the output of each algorithm.

Figure 6 illustrates the results of compared algorithms, evaluated with NMI, ARI, and FMI. Note that in this figure, LPA-Bin-Aggr and LPA-Freq-Aggr denote the performance of LPAs on the aggregated networks using, respectively, the binary-based and the frequency-based aggregation strategy. Recall that, as discussed in Section 4.1.2, for each network, we run competing algorithms 10 times and then we report the worst, the average, and the best results.

Overall, the experiments revealed that aggregation-based methods and ensemble clustering were not capable of discovering the real community structure of the networks. For the former, the performance of LPAs was greatly affected by the high

dimensionality of the networks. All the time, the algorithm yields the large connected component irrespective of the adopted aggregation scheme or the average community dimensionality of the network l_r. This could be attributed to the impact of the aggregation which leads to a higher proportion of inter-community edges in the aggregated network.

The accuracy of ensemble clustering, on the other hand, did not improve despite the increasing number of relevant dimensions. We argue that the consensus partitions were mainly affected by the sparse layers, for which the Louvain algorithm tends to overestimate the number of communities. We have observed that, for $l_r = 40\%$, ensemble clustering generates around 800 communities ranging in size from 1 to 7 nodes. The accuracy was even worse for $l_r \leq 6\%$, for which ensemble clustering produces around 2,500 communities, mostly consisting of a single node. The fact that dimensions are given equal importance makes the recovery of a meaningful consensus partition quite challenging, as the real latent community structure will be hidden by the noise coming from the irrelevant dimensions. Based on such result, we can claim that consensus clustering would only achieve good performance if the network exhibits a correlated community structure across all the dimensions.

As can be seen from Figure 6, MDLPA achieves better and more consistent results despite the varying values of l_r. Our algorithm is able to locate the dense regions in the multidimensional space, enabling it to focus the community search process on the relevant subset of layers. The results depicted by Figure 6 suggest that MDLPA is more resistant to the structural variations of communities across different dimensions compared to the competition especially when the average community dimensionality is below 6 percent, which is considered a challenging case. In such a setting, the difference between compared algorithms becomes apparent and the impact of irrelevant dimensions is much more evident. Although slightly affected, MDLPA's results confirm its immunity to the presence of a large number of irrelevant dimensions, thanks to the update rules' ability to appropriately select and update the sets of relevant dimensions $\{D_k\}$. Besides, MDLPA is more stable than competing algorithms, as demonstrated by the very low standard deviation (see the error bars in Figures 6(b), (e), and (h)). Overall, from Figure 6, the reader can observe that there are few differences between the worst and the best results of MDPLA. Although, the optimization strategy of MDPLA is based on the label propagation principle, the algorithm tends to be more stable than its competitors. This is mainly due to the incorporation of dimension relevance in the proposed objective function of MDLPA.

For $l_r \geq 6\%$, both PMM and SC-ML achieve their best performance. While SC-ML is more robust compared to PMM, both algorithms can successfully recover the original ground truth memberships when provided the correct number of communities. This is expected as both approaches were originally designed to handle the structural features' variations. In contrast, when $l_r = 1\%$, both SC-ML and PMM fail regardless of the provided parameter values. Whereas ensemble clustering achieves a higher NMI score through a high number of small communities (around 2,880 communities), SC-ML and PMM tend to assign nodes in a random way, as reported by the close to zero values of the three metrics, which indicates a significant disparity between the original partitioning and the identified community structures. To summarize, the results of this experiment suggest that MDLPA is effective in detecting multidimensional communities in complex situations that involve the presence of low dimensional community structures.

Let us now evaluate how similar the dimensions selected by MDLPA are to the real relevant dimensions. Note that, in this evaluation, we did not consider compared algorithms, as they do not offer a mechanism to determine the subspace associated with each community. In order to evaluate the similarity between the selected dimensions and the real relevant dimensions, we rely on precision and recall metrics. Specifically,

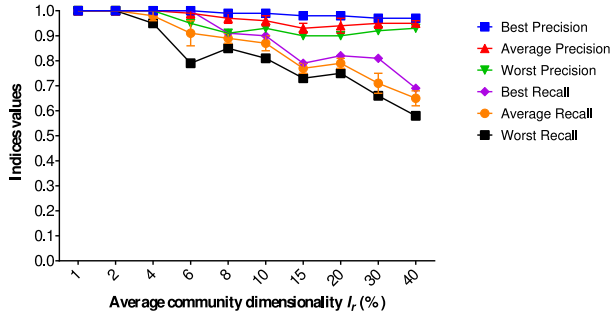


Fig. 7. Accuracy of selected dimensions.

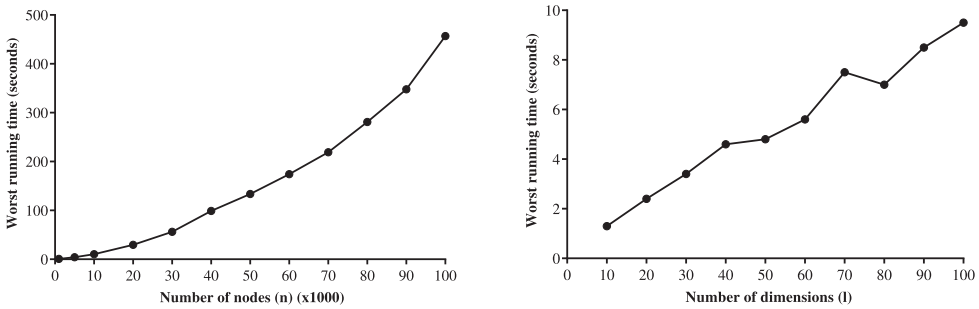


Fig. 8. (a) Scalability with the network size. (b) Scalability with dimensionality of the network.

for each community, precision measures the ratio of the number of real relevant dimensions being selected to the number of selected dimensions. Recall is the number of selected real relevant dimensions divided by the actual number of real relevant dimensions. The reported value of a resulting partition is the average of all detected communities.

Figure 7 shows the precision and recall of the selected dimensions for the results produced by our algorithm. As can be seen, the high precision values confirm MDLPA's ability in disregarding irrelevant dimensions even for very low values of l_r. In contrast, the recall of relevant dimensions decreases linearly with the increasing size of the subspaces. We argue that this is mainly caused by the selection mechanism which tends to disregard dimensions whose links are not dominating the neighborhoods. We observed that, indeed, relevant dimensions with relatively low density, compared to the other dimensions, were sometimes ignored in favor of the denser group, which explains the constant decrease in the recall. When l_r is less than 6 percent, however, MDLPA successfully recovers the real subspace.

4.3.3. Scalability Results. In this section, we study the scalability of MDLPA with increasing network size and dimensionality. Note that in all of the following experiments, the quality of the results returned by MDLPA is similar to that presented in the previous subsection.

Scalability with the network size: Figure 8(a) shows the results for scalability with the size of the network. In this experiments we generated 12 different networks with 5 dimensions and varied the number of nodes n from 1,000 to 100,000. The number of communities in each network is equal to $n/100$. We fixed the number of relevant

dimensions to two, the intra-community density of links to 0.3, and, finally, the inter-community density of links to 0.005. Figure 8(a) illustrates the worst running time of our algorithm. Note that since the number of dimensions l was fixed, the running time of the algorithm would thus be affected by the increasing number of nodes n and edges m of the network. In Figure 8(a), the curve exhibits a quadratic behavior as the size of the network increases. Such a behavior is consistent with the complexity of MDLPA which is $O(mn)$ as discussed in subsection 3.4. Overall, we believe that the running time of MDLPA is reasonable. For medium-sized networks ($n \approx 50,000$) the algorithm performs multidimensional community detection, in a full automatic manner without any parameters setting, in less than 3 minutes.

Scalability with dimensionality of the network: To study the scalability of our approach with respect to the dimensionality of the network, we generated 10 different networks each of which contains 1,000 nodes grouped into 10 communities. The dimensionality of the generated networks l varies from 10 to 100, whereas the average community dimensionality l_r was fixed to 20 percent of the whole dimensionality l. We also fixed the intra-community density of links to 0.3 and the inter-community density of links to 0.005. Figure 8(b) illustrates the worst running time of our algorithm with the increasing number of dimensions l. As can be seen from this figure, our algorithm exhibits a linear behavior with respect to the number of edges m (m increases faster than l). Such a result corroborate our complexity analysis described previously. When the number of nodes n is constant, the complexity is $O(m)$ which suggests that the number of dimensions l does not affect the scalability of the algorithm.

4.4. Experiments on Real Networks

In this section, we put our approach to work using four real world networks: (1) Aarhus computer science department network, (2) Pierre Auger observatory network, (3) Foursquare multidimensional network, and (4) *Drosophila melanogaster* protein-protein interactions network. Whereas synthetic networks offer a controlled environment where the number of communities is known in advance, a prior knowledge about community structures in real world networks is often missing. Thus, in the absence of ground truth, we have only retained algorithms that do not require the number of communities as an input parameter and we have only considered external criteria for the purpose of evaluation. A description of each network used in the experiment and the analysis of the MDLPA results follow.

4.4.1. Aarhus Computer Science Department Network. This is an unweighted and undirected five-dimensional social network representing interactions between the employees of the computer science department at the Aarhus University [40]. The original network consists of 61 employees (admin staff, professors, associates, Ph.D. students, and post-doctoral researchers) belonging to eight workgroups and interacting in five different dimensions. To make the experiment more convenient, we sampled a subset of 52 nodes from the original network by eliminating: 6 nodes with unknown workgroup memberships, 2 nodes belonging to multiple workgroups simultaneously, and a special node making a singleton workgroup. Figure 9 records the statistics about the different layers of the network.

Figure 10 reports the results for compared algorithms, evaluated with the redundancy ρ, the multidimensional community density MCD, and the multi-slice modularity Q. Note that, as discussed in Section 4.2.1 , in order to estimate ρ and MCD for our algorithm, we used DF_k, the set of dimensions found in C_k, and also D_k the set of relevant dimension of C_k identified by MDLPA. The goal is to illustrate the impact of relevant dimensions on the quality of results. In Figure 10, MDLPA (DF) indicates that ρ and MCD are calculated using the set of dimensions found in each community

Dimension	Type	#Links	Density
1	Lunch	162	0.1222
2	Facebook	96	0.0724
3	Coauthors	21	0.0158
4	Leisure	87	0.0656
5	Work	114	0.0860

Fig. 9. The Aarhus computer science department network's dimensions.

	Redundancy ρ			MCD			Multi-slice modularity Q			Avg number of communities
Algorithm	worst	mean ± std.dev	best	worst	mean ± std.dev	best	worst	mean ± std.dev	best	
MDLPA (DF)	0.41	0.44 ± 0.04	0.51	0.42	0.47 ± 0.04	0.55	0.63	0.65 ± 0.01	0.66	7
MDLPA (RD)	0.48	0.57 ± 0.09	0.72	0.53	0.60 ± 0.06	0.71				
LPA-Bin-Aggr	0.25	0.27 ± 0.04	0.38	0.07	0.15 ± 0.09	0.28	0.30	0.34 ± 0.07	0.53	2
LPA-Freq-Aggr	0.37	0.50 ± 0.09	0.55	0.28	0.42 ± 0.10	0.48	0.53	0.55 ± 0.00	0.55	5
Ensemble Clustering	0.76	0.80 ± 0.03	0.83	0.67	0.69 ± 0.01	0.71	0.35	0.36 ± 0.02	0.41	30

Fig. 10. Results on the Aarhus computer science department network.

returned by our algorithm, while MDLPA (RD) refers to the fact that the calculation of ρ and *MCD* is based on the set of relevant dimensions of each community identified by MDLPA. On the other hand, we only used the set of dimensions found in each community returned by competing algorithms in the calculation of ρ and *MCD*, since these algorithms are not able to detect the relevant dimensions of each community.

From Figure 10, we can see that MDLPA provides the best modularity values. In contrast to competing algorithms, the combination of the values of the three metrics reported by our algorithm, suggest that MDLPA discovers meaningful community structures. In addition, from Figure 10, we can see the improvement of the values of ρ and *MCD* when we consider the relevant dimensions instead of the set of all dimensions found in a community (see the results of MDLPA (DF) versus the results MDLPA (RD)). The improvements in the redundancy and the multidimensional density scores on the selected subspaces by MDLPA suggest that our algorithm can effectively disregard dimensions with insignificant contribution to the community structure. In order to have an idea of the output of our algorithm, Figure 11 illustrates a partition identified by MDLPA, with seven communities and the corresponding detected relevant dimensions. The network was projected on a single layer for better rendering. In fact, for the sake of simplicity, we dropped all edges connecting the same pair of nodes and replaced them with a single one bearing the IDs of the original dimensions connecting the pair. As we can see from this pictorial illustration, MDLPA discovers meaningful community structures that exist in different sub-dimensional spaces.

We have also compared the performance of competing algorithms by considering workgroup memberships as a possible reference partition of the network. The comparison is solely based on the assumption that employees working within the same group tend to develop more ties internally than with members of other workgroups. The aim is to evaluate whether the original workgroups could be recovered by examining the latent interaction patterns. Although it does not imply any formal consideration, still, interesting conclusions can be drawn from this assumption. Figure 12 reports the performance of competing algorithms with respect to the presumed latent workgroups community structure. Note that for this special case, we have considered both PMM and SC-ML in the comparison, as the number of workgroups (communities) is available. In this experiment, we have set the number of communities to be identified by PMM and SC-ML equal to seven, which corresponds to the number of workgroups.

As we can see from Figure 12, MDLPA as well as PMM and SC-ML perform well and achieve quite comparable results. On the other hand, the results of aggregation-based approaches and ensemble clustering are less competitive than those of MDLPA, PMM,

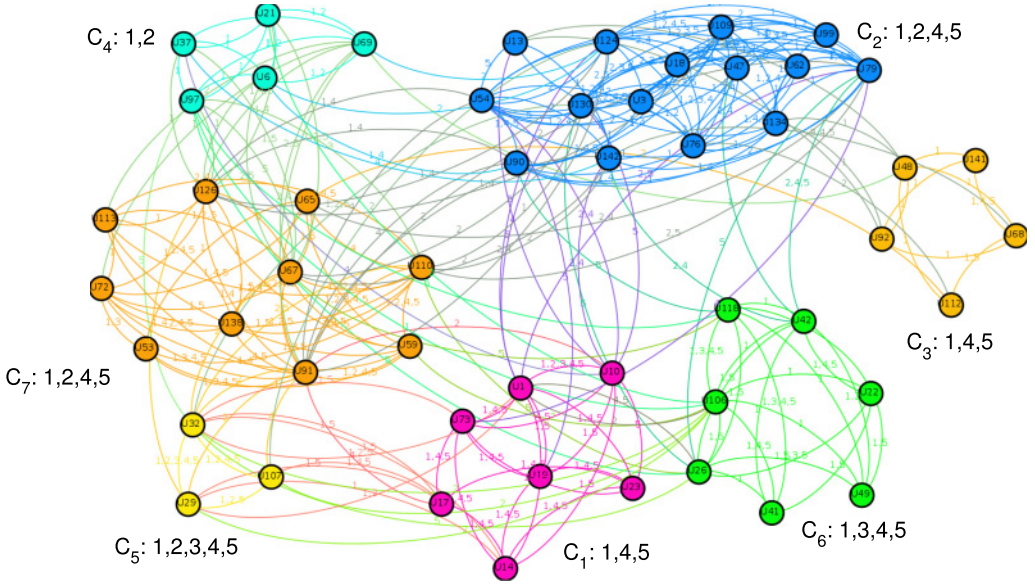


Fig. 11. The community structures of the Aarhus computer science department network as identified by MDLPA. Numbers on edges designate dimension IDs (1: Lunch, 2: Facebook, 3: Coauthors, 4: Leisure and 5: Work). Selected relevant dimensions are reported nearby each community.

	NMI			ARI			FM		
Algorithm	**worst**	**mean ± std.dev**	**best**	**worst**	**mean ± std.dev**	**best**	**worst**	**mean ± std.dev**	**best**
MDLPA	0.83	0.84 ± 0.01	0.86	0.65	0.70 ± 0.04	0.75	0.72	0.75 ± 0.03	0.79
LPA-Bin-Aggr	0.00	0.13 ± 0.18	0.57	0.00	0.04 ± 0.08	0.25	0.37	0.39 ± 0.04	0.50
LPA-Freq- Aggr	0.60	0.64 ± 0.03	0.66	0.25	0.31 ± 0.03	0.33	0.50	0.52 ± 0.01	0.53
Ensemble Clustering	0.67	0.69 ± 0.01	0.71	0.15	0.19 ± 0.04	0.31	0.28	0.32 ± 0.04	0.42
PMM	0.75	0.84 ± 0.05	0.90	0.47	0.67 ± 0.12	0.78	0.56	0.72 ± 0.10	0.82
SC-ML	0.80	0.84 ± 0.02	0.86	0.66	0.75 ± 0.04	0.78	0.71	0.79 ± 0.03	0.81

Fig. 12. Performance results on the Aarhus computer science department network with respect to the presumed latent community structure.

and SC-ML. The results of MDLPA, PMM, and SC-ML suggest that, in general, the identified community structures are not much different from the original workgroups of the employees. Although PMM and SC-ML achieve higher results when provided the number of communities, MDLPA can systematically identify the latent community structure without any additional information. Our algorithm automatically identifies communities within each of which nodes are well connected across different subsets of dimensions. For instance, referring back to Figure 11, community C_6 corresponds to the workgroup G_6 while community C_1 corresponds to workgroup G_1 with the exception of node U_{17} which originally belongs with workgroup G_5. We noticed that MDLPA assigns U_{17} to G_1 instead of its original workgroup G_5, because it socializes and works more frequently with members of G_1, making its contribution of links much higher than if it were to join G_5. The same observation remains valid for nodes unassigned to their original workgroups, as it is the case, for example, with G_3 and G_4 (C_3 and C_4, respectively) which "lost" some members in favor of larger workgroups G_2 and G_7 (C_2 and C_7, respectively). Besides, the selected relevant dimensions offer an additional insight about the main drivers of interaction within communities. For instance, members of G_1 coauthor papers less often and completely avoid interactions in Facebook. The same goes for G_6's members who tend to avoid direct contact on Facebook. In contrast, the

Dimension	Task	#Links	Density	Dimension	Task	#Links	Density
1	Neutrinos	60	0.0005	9	Spectrum	80	0.0006
2	Detector	550	0.0042	10	Photons	21	0.0002
3	Enhancements	5433	0.0412	11	Atmospheric	51	0.0004
4	Anisotropy	76	0.0006	12	SD reconstruction	211	0.0016
5	Point source	105	0.0008	13	Hadronic interactions	53	0.0004
6	Mass composition	191	0.0014	14	Exotics	18	0.0001
7	Horizontal	61	0.0005	15	Magnetic	38	0.0003
8	Hybrid reconstruction	184	0.0014	16	Astrophysical scenarios	21	0.0002

Fig. 13. The Pierre Auger observatory network's dimensions.

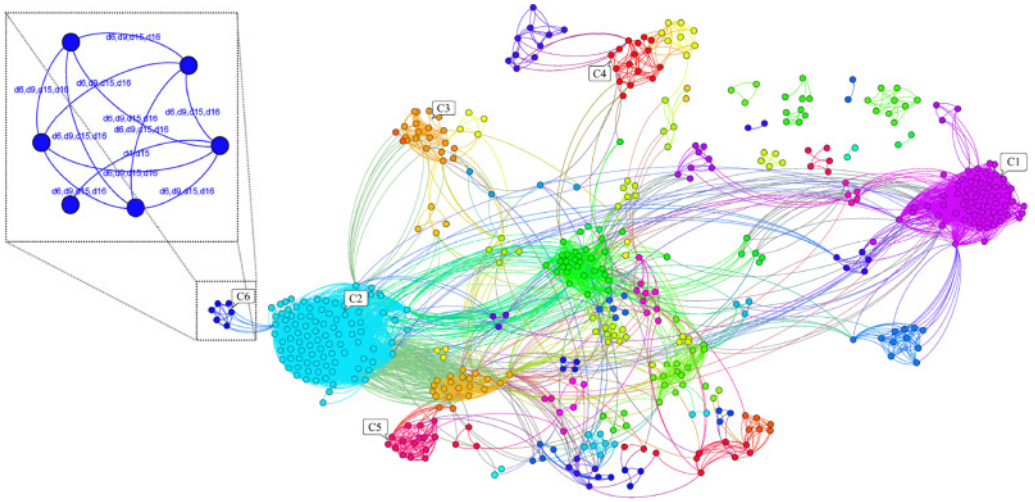


Fig. 14. The community structures of the Pierre Auger observatory network as identified by MDLPA. For the purpose of illustration, we zoom the community C_6 which represents a multidisciplinary team of six researchers working on four different tasks.

employees of G_5 are implicated in all kinds of interactions, which possibly indicates strong and well-established professional and social relationships.

4.4.2. Pierre Auger Observatory Network. We analyzed the Pierre Auger observatory network [17], a 16-dimensional collaboration network of 514 scientists working on different tasks at the Pierre Auger observatory for ultra-high-energy cosmic rays. Each task represents a collaboration dimension that connects researchers if they coauthored a related report. The network is characterized by the presence of many small connected components and relatively many sparse dimensions (that is, dimensions with low density values) as shown in Figure 13. The partition of the network as identified by MDLPA is illustrated in Figure 14. For the purpose of clarity, we projected the network by substituting the set of dimensions connecting any pair by a single edge.

The partition identified by MDLPA reveals a specialized organization of scientists, in which every group focuses on a limited, and often distinct, subset of research tracks. Overall, we found that the subspace size of the detected communities varies between 1 and 6 dimensions, which corresponds to a dimensionality of 6 percent to 38 percent of the whole space, with a majority of communities involved in a single task. For instance, communities C_1 and C_2 in Figure 14 correspond to teams working on the enhancements task, while C_3 and C_4 represent groups specializing in, respectively, detectors and hybrid reconstruction. Note that the size of these four communities (that

	Redundancy ρ			MCD			Multi-slice modularity Q			Avg number of communities
Algorithm	**worst**	**mean ± std.dev**	**best**	**worst**	**mean ± std.dev**	**best**	**worst**	**mean ± std.dev**	**best**	
MDLPA (DF)	0.39	0.41 ± 0.02	0.44	0.69	0.70 ± 0.01	0.72	0.83	0.84 ± 0.01	0.85	66
MDLPA (RD)	0.63	0.67 ± 0.02	0.69	0.78	0.81 ± 0.01	0.83				
LPA-Bin-Aggr	0.30	0.36 ± 0.03	0.39	0.60	0.65 ± 0.03	0.69	0.77	0.77 ± 0.00	0.77	46
LPA-Freq-Aggr	0.35	0.40 ± 0.03	0.42	0.62	0.66 ± 0.02	0.68	0.77	0.77 ± 0.00	0.77	45
Ensemble Clustering	1.00	1.00 ± 0.00	1.00	0.52	0.65 ± 0.07	0.76	0.68	0.68 ± 0.00	0.68	493

Fig. 15. Results on the Pierre Auger observatory network.

is, C_1, C_2, C_3, and C_4) represents 38 percent of the network size. On the other hand, we found that several groups engage in various tasks simultaneously. For example, community C_5 corresponds to an interdisciplinary team of 14 researchers collaborating on detectors and hybrid reconstruction while members of community C_6 specialize in mass composition, spectrum, magnetic, and astrophysical scenarios at the same time. MDLPA also reports the presence of several communities completely disconnected from the giant connected component, as illustrated in Figure 14, of which, some are as small as two nodes.

Figure 15 reports the results for compared algorithms, evaluated with the redundancy ρ, the multidimensional community density *MCD*, and the multi-slice modularity Q. As we can see, ensemble clustering fails in detecting meaningful community structures as the number of communities was nearing the size of the network. With few exceptions, we found that most recovered communities correspond to a single researcher while the remaining ones have a maximum of two researchers. In addition, among the few remaining communities, several correspond to completely disconnected pairs. This explains the difference between the values of the density and redundancy which, theoretically, must be equal to one for single pair communities. We argue that this is mainly due to the sparse nature of the dimensional space. In fact, for this particular network, the average community dimensionality size, as identified by MDLPA, was around 10 percent of the dimensional space which corresponds to an average of 1.6 tasks per team. Since researchers tend to specialize in small well defined tasks, the absence of involvement on other layers causes the Louvain method to produce a large number of small communities.

Based on the results of LPA-Bin-Aggr and LPA-Freq-Aggr, we can surmise that, on the aggregated network, LPAs achieve good results as the aggregation (binary-based or frequency-based) produces a highly modular representation. In addition to limiting the structural information loss caused by the aggregation, the overlapping nature of the communities in the Pierre Auger observatory network [17] reinforces their separation, which explains the close results between the binary-based and the frequency-based schemes. On the other hand, as can be seen from Figure 15, the performance of MDLPA, as measured by the three quality metrics considered in our experiments, is better than that of LPA-Bin-Aggr and LPA-Freq-Aggr. Besides, the improvements in the redundancy and the multidimensional community density on the selected relevant dimensions by MDLPA confirm its ability to provide an additional insight about the interests that discriminate between the different groups.

4.4.3. Foursquare Multidimensional Network. Foursquare is one of the largest geo-social networks where users share their locations with friends and followers through check-ins at specific places, also called venues. Venues represent physical locations such as restaurants, monuments, hotels or airports and are identified through dedicated pages where users can interact and leave their feedback. We constructed a 4-dimensional network of social interactions among 3,488 users of the Foursquare platform. In a first step, we used the Twitter streaming API to retrieve a random tweet with check-in information at a random place in New York City. We then used the publicly available

Dimension	Type	#Links	Density
1	Friendship on Foursquare	19865	0.0033
2	Follower or following relation on Twitter	2133	0.0004
3	Likes on the same venue	6994	0.0012
4	Feedback on a visited place	2380	0.0004

Fig. 16. The Foursquare multidimensional network's dimensions.

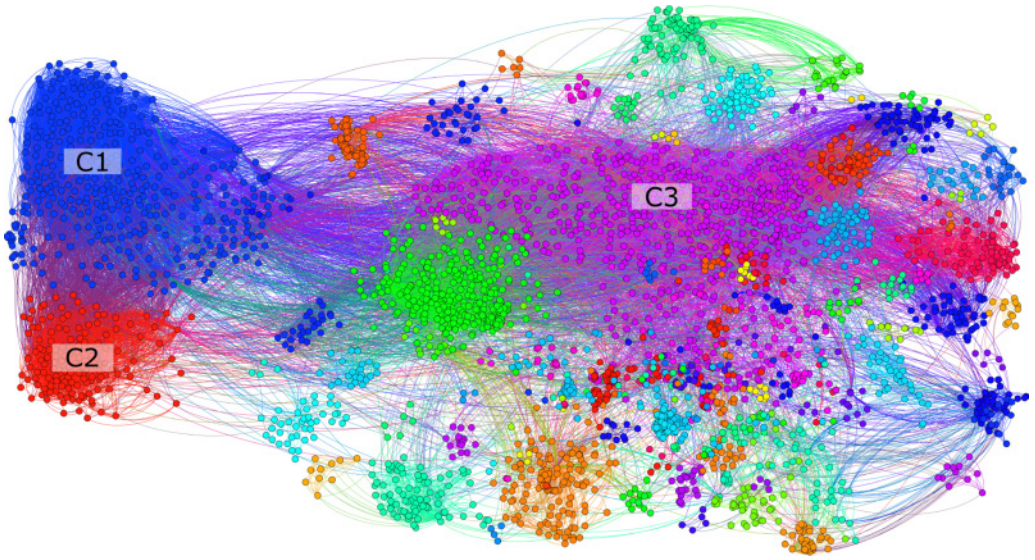


Fig. 17. A partition of the Foursquare network as reported by MDLPA.

data from the Twitter profiles to access the corresponding Foursquare accounts. The Foursquare profile information is then used to retrieve the friends' list, the venue likes and the tips (reviews about venues) of the account holder. For each friend, we collected the same set of information following a breadth-first search for three successive hops.

To build the network, we considered four different kinds of interactions: (1) *Friendship on Foursquare*: as the name implies, the Foursquare friendship dimension connects two users (nodes) if they are friends on Foursquare; (2) *Follower or following relation on Twitter*: by means of this dimension, we aim to create an additional link between two friends in Foursquare if they have either a follower or a following relationship on Twitter. Obviously, here, we only considered users with a linked Twitter account to Foursquare; (3) *Likes on the same venue*: through this dimension, two users are connected if they share a like for the same venue; and finally (4) *Feedback on a visited place*: users are connected along this dimension if they leave a feedback on a place which they both visited. Figure 16 provides basic statistics about these four dimensions.

We applied our algorithm to detect community structures in the Foursquare network. Figure 17 depicts the partition identified by MDLPA. We found that dimension 2 (follower or following relation on Twitter) and dimension 4 (feedback on a visited place) do not offer any apparent community structure. This could be attributed to the fact that both dimensions are sparse since, as can be seen from Figure 16 , they are characterized by low-density values. On the other hand, we found that dimension 3 (likes on the same venue) contributes to the formation of the largest community identified by MDLPA. This community is labeled as C_3 in Figure 17. Finally, we found that dimension 1 (friendship on Foursquare) dominates the contribution to the community structures on this network compared to the three remaining dimensions. Indeed, most

	Redundancy ρ			MCD			Multi-slice modularity Q			Avg number of communities
Algorithm	**worst**	**mean ± std.dev**	**best**	**worst**	**mean ± std.dev**	**best**	**worst**	**mean ± std.dev**	**best**	
MDLPA (DF)	0.29	0.32 ± 0.03	0.37	0.39	0.40 ± 0.02	0.45	0.54	0.57 ± 0.01	0.57	105
MDLPA (RD)	0.52	0.56 ± 0.04	0.61	0.43	0.44 ± 0.02	0.49				
LPA-Bin-Aggr	0.10	0.17 ± 0.04	0.23	0.57	0.62 ± 0.03	0.65	0.32	0.34 ± 0.01	0.35	68
LPA-Freq-Aggr	0.29	0.32 ± 0.03	0.36	0.60	0.63 ± 0.03	0.66	0.35	0.36 ± 0.01	0.38	73
Ensemble Clustering	0.80	0.86 ± 0.05	0.94	0.13	0.14 ± 0.01	0.15	0.25	0.25 ± 0.00	0.26	2639

Fig. 18. Results on the Foursquare network.

Dimension	Type	#Links	Density
1	Direct interaction	14142	0.0002
2	Suppressive genetic interaction	1733	0.0000
3	Additive genetic interaction	1330	0.0000
4	Physical association	6573	0.0001
5	Colocalization	33	0.0000
6	Association	4	0.0000
7	Synthetic genetic interaction	4	0.0000

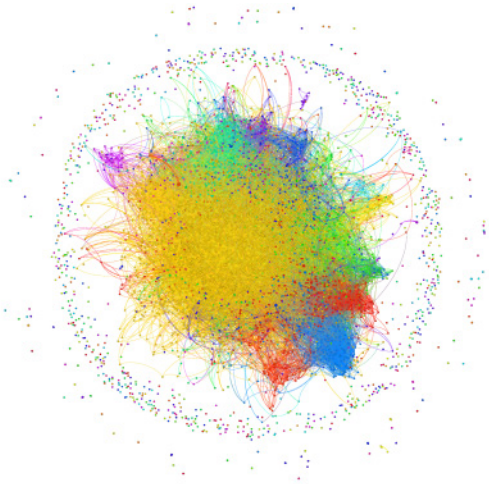
Fig. 19. The *Drosophila melanogaster* network's dimensions.

recovered communities represent Foursquare friends. For instance, community C_1 in Figure 17 represents 359 Foursquare friends, while community C_2 corresponds to a group of 151 Foursquare friends.

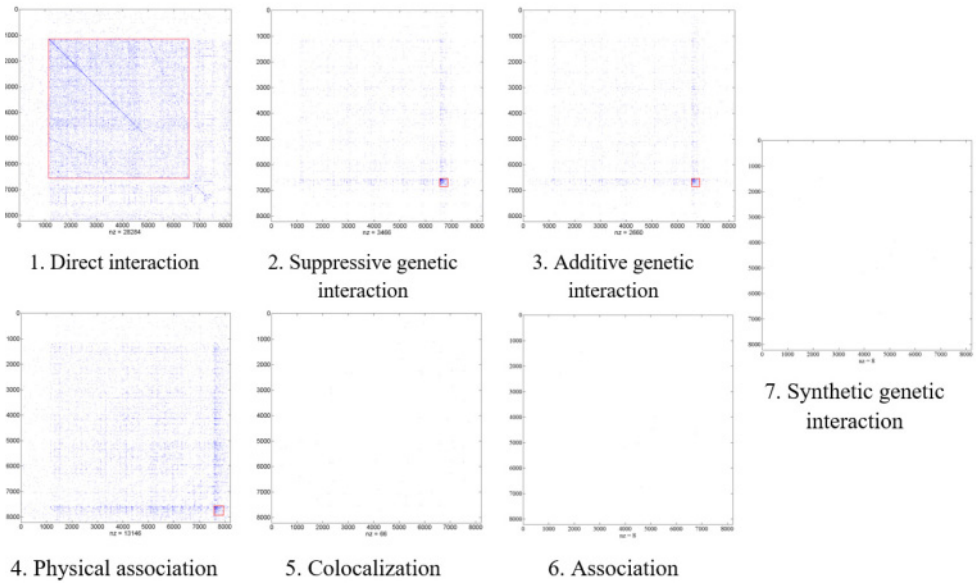
In Figure 18, we show the performance of competing algorithms. As we can see, ensemble clustering generates a significant number of communities with small sizes (one to three nodes maximum). Similar to the previous results, the fact that ensemble clustering generates a very large number of small communities, contributes to boost the value of the redundancy and the multidimensional community density. On the other hand, ensemble clustering reports the worst modularity values. We found also that the two aggregation-based strategies tend to generate partitions mostly dominated by one large community and many very small communities containing two nodes. This explains the relatively high values of the multidimensional community density achieved by aggregation-based methods. Finally, as illustrated by Figure 18, our algorithm reports the highest modularity values and achieves fairly competitive ρ and *MCD* values. The reported improvements in the redundancy and the density metrics also confirm the suitability of the subspace selection mechanism of MDLPA.

4.4.4. Drosophila Melanogaster Protein–Protein Interactions Network. Protein–protein interactions provide a fundamental condition for life sustainability in living organisms. The analysis of such interaction networks has become an area of active research among various fields as they provide a crucial tool for understanding the processes, functions, and organization at the cell level [55]. The discovery of biological function among highly interactive proteins or genes is one possible application of community detection algorithms [13]. In fact, it has been shown in the literature that, indeed, molecules that interact more frequently together generally ensure the same, or a similar, cellular role. Pairs sharing common interaction partners have a higher chance of ensuring a common function. These functional modules can be captured by a separate community. The investigated network concerns 8,215 proteins (nodes) of the *D. melanogaster* fruit fly [18, 50]. The network was built according to seven different types of interactions (dimensions). Figure 19 presents the main characteristics of each dimension of this network.

The identified community structures are depicted in Figure 20(a). The belt around the giant connected component represents small or completely isolated protein modules. Combined, the small communities account for 1,099 proteins which corresponds to



(a) The partition identified by MDLPA.



(b) The associated adjacency matrices of the identified partition.

Fig. 20. The community structures of the *Drosophila melanogaster* network as identified by MDLPA.

13 percent of the total size of the network. This explains the relatively high number of communities discovered by MDLPA, LPA-Bin-Aggr, and LPA-Freq-Aggr as shown in Figure 21. This is mainly due to the fact that our algorithm MDLPA, as well as the basic LPA, both consider isolated nodes as separate communities. In contrast, ensemble clustering seems to be more affected by the sparse layers and mostly produces singleton communities only (among 8,215 nodes, the algorithm returns, on average, 8,211 communities). In fact, the adjacency matrix of each dimension of the network depicted in Figure 20(b), suggests the presence of three sparse dimensions (dimension 5: colocalization, dimension 6: association, and dimension 7: synthetic genetic interaction) that do not reflect any community structures. Here, it is worth mentioning that these

	Redundancy ρ			MCD			Multi-slice modularity Q			Avg number of communities
Algorithm	**worst**	**mean ± std.dev**	**best**	**worst**	**mean ± std.dev**	**best**	**worst**	**mean ± std.dev**	**best**	
MDLPA (DF)	0.24	0.27 ± 0.02	0.29	0.67	0.68 ± 0.01	0.69	0.76	0.76 ± 0.00	0.77	1143
MDLPA (RD)	0.55	0.58 ± 0.04	0.63	0.69	0.70 ± 0.01	0.70				
LPA-Bin-Aggr	0.06	0.07 ± 0.02	0.09	0.63	0.68 ± 0.01	0.76	0.74	0.78 ± 0.02	0.81	1893
LPA-Freq-Aggr	0.06	0.12 ± 0.03	0.15	0.64	0.64 ± 0.01	0.65	0.79	0.81 ± 0.00	0.81	2097
Ensemble Clustering	0.84	0.87 ± 0.05	0.91	0.41	0.43 ± 0.03	0.45	0.67	0.67 ± 0.00	0.67	8211

Fig. 21. Results on the *Drosophila melanogaster* network.

three sparse dimensions contribute to a total of 41 links only across the whole network. Also note that the blank region which surrounds the shaded regions of the adjacency matrices depicted in Figure 20(b) corresponds to the belt in Figure 20(a).

Figure 20 illustrates the results of MDLPA on the *D. melanogaster* network.

For the sake of discussion, it is worth noting that MDLPA discovered three major communities spanned in different subspaces. The knowledgeable reader can observe the three community structures through a careful visual inspection of the first four adjacency matrices (associated to dimensions 1, 2, 3, and 4) depicted by Figure 20(b). In this figure, the largest block of the adjacency matrix associated to the direct interaction dimension illustrates the biggest community, which contains 5,019 proteins. The two small blocks in the two adjacency matrices associated to dimension 2 and dimension 3 (blocks in the lower right hand side of the suppressive and the additive genetic interaction matrices) are related to another community which contains 122 proteins. Finally, the small dense block in the lower right hand side of the physical association matrix is related to a community of 275 proteins.

With respect to the external criteria, we observe from Figure 21 that MDLPA clearly outperforms LPA-Bin-Aggr and LPA-Freq-Aggr on the redundancy and obtains comparable results on the multidimensional community density and the multi-slice modularity. Ensemble clustering, on the other hand, exhibits a similar behavior with respect to the previous networks as the number of communities was nearing the number of nodes. Despite being a central aspect in protein–protein interactions analysis, we did not investigate the biological meaning of the discovered communities since the main focus of our work is to identify regions of high density in the different subspaces of a network. Still, MDLPA discovers statistically relevant groups of highly interactive proteins, as reported in Figure 21. Besides, the relevant dimensions selected by the algorithm provide key information about the most important channels of interaction among the proteins of the same module. We believe that the discovered groups may characterize some biological phenomenon or suggest new hypotheses that can be verified using existing biological expertise. So, it requires further research to deepen the understanding of the obtained modules on this and similar networks.

5. CONCLUSION

In this paper, we have addressed the problem of community detection in multidimensional networks. We presented some shortcomings of several existing algorithms, namely, the parameter tuning, the sensitivity to irrelevant dimensions, and the inability to recover the most relevant dimensions. To address these limitations, we first reviewed the structural properties of the communities we are targeting and described the challenges they present to some existing approaches. We then proposed our algorithm which, in addition to eliminating any need for user-supplied parameters, can automatically discriminate between the relevant and the non-relevant dimensions. The experimental evaluation supports our claim that the approach is capable of identifying meaningful communities that reflect hidden functional and organizational characteristics of real world systems that would not have been discovered from single connectivity sources.

In light of the experimental results, we believe that the recovery of a latent community structure might only be possible through an efficient fusion of the multidimensional information. Therefore, unless having a shared community structure, applying consensus clustering on partitions recovered from structurally dissimilar dimensions is likely to end up in a meaningless clustering. On the other hand, unfolding communities from an aggregated representation can lead to different outcomes, as the same algorithm might behave differently depending on the amount of preserved information from the structural features of participating dimensions. In contrast, MDLPA can combine and exploit the heterogeneous connectivity information more efficiently, allowing it to identify regions of high density and their locations in the multidimensional space. Furthermore, in addition to improving the clustering accuracy, the ability to disregard non-informative sources contributes valuable input that helps in understanding the key interaction drivers. Moreover, we believe that the selected relevant dimensions would benefit other tasks like network compression, representation learning and collaborative filtering, making it a promising tool for many real world scenarios.

MDLPA generates hard partitions. One possible direction for further research is to support soft clustering by allowing communities to share nodes at different dimensions. Another course for further work is to extend the relevance metric and the update rules in order to support weighted multigraphs. Finally, we believe that our approach can successfully be applied for the discovery of time evolving communities in dynamic networks, that is, networks where interactions develop over time. By considering snapshots (or alternatively time intervals) from the network's time-ordered sequence as distinct dimensions, we might be able to recover communities along with the most significant time spans of their formation. Further investigation is needed in this direction.

ACKNOWLEDGMENTS

The authors gratefully thank Dr Matteo Magnani for providing the Aarhus computer science department network and the associated metadata, Dr Manlio De Domenico for providing the Pierre Auger observatory and the *Drosophila melanogaster* protein–protein interactions networks and Dr Peter J. Mucha for making available the code of the generalized modularity metric. The authors also would like to thank the reviewers and associate editor for their valuable comments and important suggestions.

REFERENCES

[1] Alessia Amelio and Clara Pizzuti. 2014. A cooperative evolutionary approach to learn communities in multilayer networks. In *Proceedings of Parallel Problem Solving from Nature - PPSN XIII*. Lecture Notes in Computer Science, vol. 8672, Springer, 222–232.

[2] Michael J. Barber and John W. Clark. 2009. Detecting network communities by propagating labels under constraints. *Physical Review E* 80, 2 (2009), 026129.

[3] Federico Battiston, Vincenzo Nicosia, and Vito Latora. 2014. Structural measures for multiplex networks. *Physical Review E* 89, 3 (2014), 032804.

[4] Michele Berlingerio, Michele Coscia, and Fosca Giannotti. 2011. Finding and characterizing communities in multidimensional networks. In *Proceedings of the IEEE International Conference on Advances in Social Networks Analysis and Mining (ASONAM'2011)*. 490–494.

[5] Michele Berlingerio, Michele Coscia, Fosca Giannotti, Anna Monreale, and Dino Pedreschi. 2013. Multidimensional networks: Foundations of structural analysis. *World Wide Web* 16, 5–6 (2013), 567–593.

[6] Michele Berlingerio, Fabio Pinelli, and Francesco Calabrese. 2013. ABACUS: Frequent pAttern mining-based community discovery in multidimensional networks. *Data Mining and Knowledge Discovery* 27, 3 (2013), 294–320.

[7] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008, 10 (2008), P10008.

[8] Brigitte Boden, Stephan Günnemann, Holger Hoffmann, and Thomas Seidl. 2012. Mining coherent subgraphs in multi-layer graphs with edge labels. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'2012)*. 1258–1266.

[9] Christian Borgelt. 2003. Efficient implementations of apriori and eclat. In *Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI'2003)*.

[10] Mohamed Bouguessa, Shengrui Wang, and Benoit Dumoulin. 2010. Discovering knowledge-sharing communities in question-answering forums. *ACM Transactions on Knowledge Discovery from Data* 5, 1, Article 3 (Dec 2010), 49 pages.

[11] Deng Cai, Zheng Shao, Xiaofei He, Xifeng Yan, and Jiawei Han. 2005. Mining hidden community in heterogeneous social networks. In *Proceedings of the 3rd ACM SIGKDD International Workshop on Link Discovery (LinkKDD'2005)*. 58–65.

[12] Vincenza Carchiolo, Alessandro Longheu, Michele Malgeri, and Giuseppe Mangioni. 2011. Communities unfolding in multislice networks. In *Complex Networks*. Communications in Computer and Information Science. Springer, 187–195.

[13] Jingchun Chen and Bo Yuan. 2006. Detecting functional modules in the yeast protein-protein interaction network. *Bioinformatics* 22, 18 (2006), 2283–2290.

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

[15] Michele Coscia, Fosca Giannotti, and Dino Pedreschi. 2011. A classification for community discovery methods in complex networks. *Statistical Analysis and Data Mining: The ASA Data Science Journal* 4, 5 (2011), 512–546.

[16] Michele Coscia, Giulio Rossetti, Fosca Giannotti, and Dino Pedreschi. 2014. Uncovering hierarchical and overlapping communities with a local-first approach. *ACM Transactions on Knowledge Discovery from Data* 9, 1, Article 6 (Aug. 2014), 27 pages.

[17] Manlio De Domenico, Andrea Lancichinetti, Alex Arenas, and Martin Rosvall. 2015. Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems. *Physical Review X* 5, 1 (2015), 011027.

[18] Manlio De Domenico, Vincenzo Nicosia, Alexandre Arenas, and Vito Latora. 2015. Structural reducibility of multilayer networks. *Nature Communications* 6 (2015).

[19] Manlio De Domenico, Albert Sole, Sergio Gomez, and Alex Arenas. 2014. Navigability of interconnected networks under random failures. *Proceedings of the National Academy of Sciences of the United States of America* 111, 23, 8351–8356.

[20] Xiaowen Dong, Pascal Frossard, Pierre Vandergheynst, and Nikolai Nefedov. 2012. Clustering with multi-layer graphs: A spectral perspective. *IEEE Transactions on Signal Processing* 60, 11 (2012), 5820–5831.

[21] Xiaowen Dong, Pascal Frossard, Pierre Vandergheynst, and Nikolai Nefedov. 2014. Clustering on multilayer graphs via subspace analysis on grassmann manifolds. *IEEE Transactions on Signal Processing* 62, 4 (2014), 905–918.

[22] Daniel M. Dunlavy, Tamara G. Kolda, and W. Philip Kegelmeyer. 2011. Multilinear algebra for analyzing data with multiple linkages. In *Graph Algorithms in the Language of Linear Algebra*. J. Kepner and J. Gilbert (Eds.), Fundamentals of Algorithms, SIAM, Philadelphia (2011), 85–114.

[23] Santo Fortunato. 2010. Community detection in graphs. *Physics Reports* 486, 3 (2010), 75–174.

[24] Edward B. Fowlkes and Colin L. Mallows. 1983. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association* 78, 383 (1983), 553–569.

[25] A. Graovac, I. Gutman, N. Trinajstić, and T. Živković. 1972. Graph theory and molecular orbitals. *Theoretica Chimica Acta* 26, 1 (1972), 67–78.

[26] Manel Hmimida and Rushed Kanawati. 2015. Community detection in multiplex networks: A seed-centric approach. *Networks and Heterogeneous Media* 10, 1 (2015), 71–85.

[27] Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of Classification* 2, 1 (1985), 193–218.

[28] Przemyslaw Kazienko, Katarzyna Musial, Elzbieta Kukla, Tomasz Kajdanowicz, and Piotr Brodka. 2011. Multidimensional social network: Model and analysis. In *Computational Collective Intelligence. Technologies and Applications*. Lecture Notes in Computer Science, vol. 6922, Springer, 378–387.

[29] Tamara G. Kolda and Brett W. Bader. 2009. Tensor decompositions and applications. *SIAM Review* 51, 3 (2009), 455–500.

[30] Jeremy Kun, Rajmonda Caceres, and Kevin Carter. 2014. Locally boosted graph aggregation for community detection. arXiv preprint arXiv:1405.3210.

[31] Zhana Kuncheva and Giovanni Montana. 2015. Community detection in multiplex networks using locally adaptive random walks. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM'2015)*. ACM, 1308–1315.

[32] Renaud Lambiotte, J.-C. Delvenne, and Mauricio Barahona. 2014. Random Walks, Markov Processes and the Multiscale Modular Organization of Complex Networks. *IEEE Transactions on Network Science and Engineering* 1, 2 (2014), 76–90.

[33] Ian X. Y. Leung, Pan Hui, Pietro Lio, and Jon Crowcroft. 2009. Towards real-time community detection in large networks. *Physical Review E* 79, 6 (2009), 066107.

[34] Xutao Li, Michael K. Ng, and Yunming Ye. 2014. Multicomm: Finding community structure in multi-dimensional networks. *IEEE Transactions on Knowledge and Data Engineering* 26, 4 (2014), 929–941.

[35] Guimei Liu and Limsoon Wong. 2008. Effective pruning techniques for mining quasi-cliques. In *Machine Learning and Knowledge Discovery in Databases*. Lecture Notes in Computer Science, Springer, 33–49.

[36] Muhuo Liu and Bolian Liu. 2009. New sharp upper bounds for the first Zagreb index. *Communications in Mathematical and in Computer Chemistry* 62, 3 (2009), 689–698.

[37] Xinhai Liu, Shuiwang Ji, Wolfgang Glanzel, and Bart De Moor. 2013. Multiview partitioning via tensor methods. *IEEE Transactions on Knowledge and Data Engineering* 25, 5 (2013), 1056–1069.

[38] Xin Liu and Tsuyoshi Murata. 2010. Advanced modularity-specialized label propagation algorithm for detecting communities in networks. *Physica A: Statistical Mechanics and Its Applications* 389, 7 (2010), 1493–1500.

[39] Chuan Wen Loe and Henrik Jeldtoft Jensen. 2015. Comparison of communities detection algorithms for multiplex. *Physica A: Statistical Mechanics and Its Applications* 431 (2015), 29–45.

[40] Matteo Magnani, Barbora Micenkova, and Luca Rossi. 2013. Combinatorial analysis of multiple networks. arXiv preprint arXiv:1303.4986.

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

[42] Peter J. Mucha, Thomas Richardson, Kevin Macon, Mason A. Porter, and Jukka-Pekka Onnela. 2010. Community structure in time-dependent, multiscale, and multiplex networks. *Science* 328, 5980 (2010), 876–878.

[43] Mark E. J. Newman. 2006. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences* 103, 23 (2006), 8577–8582.

[44] Vincenzo Nicosia and Vito Latora. 2015. Measuring and modelling correlations in multiplex networks. *Physical Review E* 92, 3 (2015), 032805.

[45] Evangelos E. Papalexakis, Leman Akoglu, and D. Ience. 2013. Do more views of a graph help? Community detection and clustering in multi-graphs. In *Proceedings of the16th IEEE International Conference on Information Fusion (FUSION'2013)*. 899–905.

[46] Pascal Pons and Matthieu Latapy. 2005. Computing communities in large networks using random walks. In *Computer and Information Sciences-ISCIS 2005*. Lecture Notes in Computer Science, vol. 3733, Springer, 284–293.

[47] Usha Nandini Raghavan, Rèka Albert, and Soundar Kumara. 2007. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E* 76, 3 (2007), 036106.

[48] Martin Rosvall and Carl T. Bergstrom. 2008. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences* 105, 4 (2008), 1118–1123.

[49] Sucheta Soundarajan and John E. Hopcroft. 2015. Use of local group information to identify communities in networks. *ACM Transactions on Knowledge Discovery from Data* 9, 3, Article 21 (April 2015), 27 pages.

[50] Chris Stark, Bobby-Joe Breitkreutz, Teresa Reguly, Lorrie Boucher, Ashton Breitkreutz, and Mike Tyers. 2006. BioGRID: A general repository for interaction datasets. *Nucleic Acids Research* 34, suppl 1 (2006), D535–D539.

[51] Alexander Strehl and Joydeep Ghosh. 2003. Cluster ensembles—A knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research* 3 (2003), 583–617.

[52] Lei Tang, Xufei Wang, and Huan Liu. 2009. Uncoverning groups via heterogeneous interaction analysis. In *Proceedings of the 9th IEEE International Conference on Data Mining (ICDM'2009)*. 503–512.

[53] Lei Tang, Xufei Wang, and Huan Liu. 2012. Community detection via heterogeneous interaction analysis. *Data Mining and Knowledge Discovery* 25, 1 (2012), 1–33.

[54] Wei Tang, Zhengdong Lu, and Inderjit S. Dhillon. 2009. Clustering with multiple graphs. In *Proceedings of the 9th IEEE International Conference on Data Mining (ICDM'2009)*. 1016–1021.

[55] Aidong Zhang. 2009. *Protein Interaction Networks: Computational Analysis*. Cambridge University Press.

Received December 2015; revised October 2016; accepted April 2017