Noname manuscript No.
(will be inserted by the editor)

Ensemble-based Community Detection in Multilayer Networks

Andrea Tagarelli · Alessia Amelio · Francesco Gullo

Abstract The problem of community detection in a multilayer network can effectively be addressed by aggregating the community structures separately generated for each network layer, in order to infer a consensus solution for the input network. To this purpose, clustering ensemble methods developed in the data clustering field are naturally of great support. Bringing these methods into a community detection framework would in principle represent a powerful and versatile approach to reach more stable and reliable community structures. Surprisingly, research on consensus community detection is still in its infancy. In this paper, we propose a novel modularity-driven ensemble-based approach to multilayer community detection. A key aspect is that it finds consensus community structures that not only capture prototypical community memberships of nodes, but also preserve the multilayer topology information and optimize the edge connectivity in the consensus via modularity analysis. Empirical evidence obtained on seven real-world multilayer networks sheds light on the effectiveness and efficiency of our proposed modularity-driven ensemble-based approach, which has shown to outperform state-of-the-art multilayer methods in terms of modularity, silhouette of community memberships, and redundancy assessment criteria, and also in terms of execution times.

1 Introduction

Multilayer networks are increasingly used as a powerful model to represent the organization and relationships of complex data in a wide range of scenarios [8,17].

A great deal of attention has recently been devoted to the problem of *community detection* in a multilayer network [23,16]. Solving this problem is important in order to unveil meaningful patterns of node groupings into communities, by taking into account the different interaction types that involve all the entity nodes in a complex network.

A. Amelio
University of Calabria, Rende (CS), Italy, E-mail: aamelio@dimes.unical.it
A. Tagarelli
University of Calabria, Rende (CS), Italy, E-mail: andrea.tagarelli@unical.it
F. Gullo
UniCredit, R&D Dept., Rome, Italy, E-mail: gullof@acm.org

Existing approaches can broadly be classified into three main categories: *flattening methods*, *aggregation methods*, and *direct methods*. Flattening methods determine a single-layer network from the multilayer one, whereupon any conventional community detection algorithm is applied on that network. Aggregation methods detect a community structure separately for each network layer, after that an aggregation mechanism is used to obtain the final community structure. Finally, direct methods aim to compute a community structure directly on the input multilayer network, by optimizing some multilayer quality-assessment criterion.

In this work, we focus on the aggregation approach to multilayer community detection, however from a perspective still poorly explored in the literature, where the aggregation of layer-specific community structures is performed by resorting to a *clustering ensemble* approach. Clustering ensemble methods have been successfully used as an advanced clustering tool to exploit the availability of a set, or *ensemble*, of multiple clustering solutions generated on the same set of objects, by using different clustering algorithms or settings [11, 29, 22, 25, 12]. Given an ensemble of clusterings, the goal is to compute a *consensus clustering*, i.e., a single, prototypical clustering solution that optimizes a certain objective function properly defined over information available from the clusterings in the ensemble. Intuitively, this allows the generation of a stable and more reliable solution out of a set of clusterings, each of which has its own bias and might have been delivered by non-deterministic and parametric processes. In a sense, an available ensemble is a source of knowledge from which one tries to infer a single, meaningful solution that is representative of the initially gained knowledge.

Our motivations for this work stem from the opportunity of bringing the clustering ensemble paradigm into the problem of multilayer community detection. This way, not only the requirement for a unique method or setting for determining a multilayer community structure is relaxed, but also the availability of multiple community structures can be profitably exploited to learn a consensus for the multilayer network.

We believe that adopting a consensus-clustering-based aggregation approach to address the community detection problem in multilayer networks provides a number of benefits compared to direct approaches. Direct methods for multilayer community detection need to find a community structure in the multilayer network from scratch, using information from the multiple layers while inferring the communities, which is likely to be more difficult than inferring a community structure from a single graph. By contrast, an aggregation approach based on consensus clustering, like the one proposed in this work, will benefit from exploiting an available set of solutions (perhaps from the same method) each focused on a particular relation/dimension/layer, then will infer a final structure at a desired "degree of consensus".

It should also be noted that the idea of using consensus clustering to solve the problem of multilayer community detection is not novel in the literature [31, 19]; however, to the best of our knowledge, the concept of *consensus community structure* has never been formally defined and exploited so far, neither has been embedded in a modularity optimization problem.

In this paper, we address the problem of community detection on multilayer networks by proposing a novel *modularity-driven ensemble-based* framework. The input is a multilayer network and an ensemble of community structures over it. Each of the community structures in the ensemble is a non-overlapping partition of a particular layer graph of the network, and obtained by applying any (single-graph) community detection algorithm. The general objective is to compute a community structure solution that, by optimizing some property (i.e., modularity) based on information from all of the layer-specific community structures, is inferred as a consensus community structure for the multilayer network. The proposed approach follows the clustering ensemble paradigm as the consensus solution is computed by ignoring any prior knowledge about the community detection method(s) that originally determined the layer-specific community structures; in other terms, the communities in the ensemble are considered as they are. However, our problem differs from the conventional clustering ensemble one, whereby the consensus partition is derived without accessing the original features of the objects in the data collection, thus not preserving the relationships among the objects. *Our notion of consensus community structure, instead, is designed to preserve the multilayer network topology, thus taking into account not only the community memberships of nodes but also the amount and types of links among nodes.*

Our first contribution is the definition of two baseline methods that rely on a *co-association-based consensus clustering* scheme [11, 29, 12],[1] suitably defined over the multiple layers of a network.

Our defined baseline methods have the property of discovering a consensus community structure whose underlying graph can be seen as a *topological upper-bound* and a *topological lower-bound*, respectively, of the input multilayer network, for a given co-association threshold. However, if we consider the general desiderata for community detection tasks, i.e., high within-community connectivity and low inter-community connectivity, the solutions of both baseline methods are in principle not optimal: in fact, the "topological-lower-bound" solution may be poorly descriptive in terms of multilayer edges that characterize the internal connectivity of the communities, whereas the "topological-upper-bound" solution may be redundant in terms of multilayer edges connecting different communities.

To overcome these issues, we define a well-principled theoretical framework by formulating the problem of *modularity-optimization-driven ensemble-based multilayer community detection*. Our defined *consensus objective function* is optimized to discover a consensus solution with maximum modularity, subject to the constraint of being searched over a hypothetical space of consensus community structures that are valid w.r.t. the input ensemble and topologically bounded by the baseline solutions.

We have developed a hill-climbing algorithm for the modularity-based multilayer community detection problem. The algorithm starts with the consensus solution provided by the topological-lower-bounded baseline, then iteratively seeks a better solution in terms of modularity by incrementally refining the within-community connectivity and the inter-community connectivity, until no further improvements can be found.

We evaluated our proposed methods over seven real-world multilayer networks. Results have shown that our modularity-driven approach to multilayer community detection produces consensus communities that have far better multilayer modularity and quality of community memberships w.r.t. the ensemble-based baseline methods. Our main method also outperforms the competitors in terms of both effectiveness and efficiency aspects.

The rest of the paper is organized as follows. Section 2 briefly overviews related work. Section 3 introduces background concepts for the proposed approach, which is described in Section 4. Sections 5 and 6 present experimental methodology and results. Section 7 concludes the paper and outlines future research directions.

[1] Consistently with classic literature on ensemble clustering, in this paper we will use term *co-association* rather than a more intuitive *co-occurrence*.

2 Related Work on Multilayer Community Detection

Flattening methods. These methods first flatten the input multilayer network into a single-layer one, then apply any conventional community detection method. Berlingerio *et al.* [3] derive a single-layer network from a multilayer one by drawing an edge between any two vertices that are connected in at least one layer, and assigning a proper weight to each edge. Edge weights are defined according to criteria defined over structural multilayer properties of edges. Rocklin and Pinar [26] focus on the problem of deriving the more appropriate function to aggregate edge weights coming from different layers given a predefined community structure. Tang *et al.* [31] define a general framework for community detection on a multilayer network that relies on four blocks of aggregation, so that the multilayer communities are computed by aggregating the output of any of these blocks. The categorization of the method by Tang *et al.* therefore depends on the block at the end of which aggregation is performed. Performing aggregation on the first block (i.e., network aggregation) gives a flattening method [31].

Aggregation methods. These methods adopt an opposite approach w.r.t. flattening methods, thus avoiding to loose useful information from each of the layers: they first detect a community structure for each layer separately, then aggregate information from such structures. Methods falling into this category differ from each other by the way how community-structure aggregation is performed.

Berlingerio *et al.* [4] propose *ABACUS*, an aggregation method based on frequent pattern mining. Each vertex is associated with a transaction as a list of pairs given by layer identifier and identifier of the community which that vertex belongs to in that layer. The aggregate community structure is found by applying frequent closed itemset mining on the set of transactions.

Principal Modularity Maximization (PMM) [30] aims to find a concise representation of features from the various layers (dimensions) through two main steps: structural feature extraction and cross-dimension integration. Structural features from each dimension are first extracted via modularity maximization, then concatenated and subjected to PCA to select the top eigenvectors, which represent possible community partitions. Using these eigenvectors, a low-dimensional embedding is computed to capture the principal patterns across all the dimensions of the network, finally a simple k-means on this embedding is carried out to find out the discrete community assignment.

The aforementioned framework by Tang *et al.* [31] introduces the utility integration criterion for computing utility matrices of a community detection method for each layer separately. Then it optimizes an objective function for the aggregated multilayer utility matrix. Also, the framework includes a partition integration block, which consists in applying a *clustering ensemble* based approach (cf. Sect. 3.2) over the set of clusterings of the set of nodes identified in each layer. Analogously, Lancichinetti and Fortunato [19] introduce a framework that combines multiple solutions of the same clustering algorithm into a *consensus* matrix, then iteratively applies the clustering algorithm over it until the matrix turns into a block diagonal one. The obtained blocks correspond to the final community solution. Burgess *et al.* [5] adopt a similar approach upon recovering missing information on networks.

The latter two methods are examples of ensemble based approaches for multilayer community detection. However, unlike our proposed approach, they use a clustering ensemble method as a black-box tool for the problem at hand, mainly focusing on the community membership of nodes, and in some cases trying to refine inter-community connectivity based on pruning to unveil connected components for the consensus generation; furthermore, they do not generate the consensus by optimizing modularity or any other quality criterion.

Direct methods. These are methods that work directly on the input multilayer network. They usually define ad-hoc community-quality assessment criteria, and search for multilayer community structures that optimize such criteria [23,9,2,6,13,18]. We focus the following discussion on representative methods that use modularity optimization.

Generalized Louvain (GL) [23] extends the classic Louvain method using multislice modularity, so that each node-layer tuple is assigned separately to a community.

MultiGA [1] and *MultiMOGA* [2] are both based on genetic algorithms. MultiGA exploits a fitness function that combines the modularity values computed for each layer. The best-fit individual is selected from the final population, and the community structure of layer corresponding to the maximum modularity is returned. Upon this solution, a label assignment and a local search strategy are employed to refine the community structure in terms of modularity. MultiMOGA utilizes a multiobjective genetic approach, depending on a predetermined ordering of the layers, which optimizes the modularity for the current layer and the similarity with the solution found on previously considered layers.

Locally Adaptive Random Transitions (LART) [18] is a random-walk based method. It first runs a different random walk for each layer, then a dissimilarity measure between nodes is obtained leveraging the per-layer transition probabilities. Finally, a hierarchical clustering method is used to produce a hierarchy of communities which is eventually cut at the level corresponding to the best value of multislice modularity [23].

Multiplex-Infomap [6] is an extension to multiplex networks of the classic *Infomap* algorithm [27]. Infomap is a search algorithm that minimizes the flow-based *map equation* model, which relies on the principle that communities are detected as groups of nodes among which the flow, based on a random walk model, persists for a long time once entered.

In Sects. 5–6, we shall include GL, LART, Multiplex-Infomap, MultiGA and MultiMOGA, along with the aggregation methods PMM and ABACUS, in our experimental evaluation.

3 Background

3.1 Modularity

Modularity quantifies the difference between the expected number of edges linking nodes inside a community and the actual number of edges linking nodes inside a community. Given a set of nodes $\mathcal{V}$, for any $v \in \mathcal{V}$ we use symbol $d(v)$ to denote the degree of v, and symbol $d(\mathcal{V})$ to denote the total degree of nodes over the entire graph, i.e., $d(\mathcal{V}) = \sum_{v \in \mathcal{V}} d(v)$. Let $\mathcal{C}$ denote a community structure, which corresponds to a partition of the input graph into disjoint sets of nodes. For any community $C \in \mathcal{C}$, we denote with $d(C)$ the sum of degrees of nodes within C; moreover, we use symbol $d_{int}(C)$ to denote the internal degree of C, i.e., the portion of $d(C)$ which corresponds to the number of edges that link nodes in C to other nodes in C (i.e., twice the number of links internal to C). Modularity is defined as follows [24,21]:

$$Q(\mathcal{C}) = \frac{1}{d(\mathcal{V})} \sum_{C \in \mathcal{C}} \left(d_{int}(C) - \frac{d(C)^2}{d(\mathcal{V})} \right) \tag{1}$$

The value of Q varies from -0.5 to 1.0. In particular, it reaches the minimum of -0.5 when all edges link nodes in different communities, and the maximum of 1.0 when all edges link nodes in the same community.

3.2 Ensemble clustering

Given a set of objects D, a *clustering* solution $\mathcal{C} = \{C_1, \ldots, C_k\}$ defined over D is a partition of D into k disjoint groups (clusters). An *ensemble* of clustering solutions is a set $\mathcal{E} = \{\mathcal{C}_1, \ldots, \mathcal{C}_m\}$, where each $\mathcal{C}_i$ (with $i = 1..m$) is a clustering solution defined over D. Intuitively, an ensemble can be obtained in various ways, such as applying different clustering methods over the same set D, varying one or more model parameters of the clustering method(s), using different subspaces of object features, or varying the measure of distance/similarity used in the clustering method(s). Given a clustering ensemble $\mathcal{E}$, a *consensus clustering* derived from $\mathcal{E}$ is a clustering solution $\mathcal{C}^*$ that optimizes a given consensus function by exploiting information available from $\mathcal{E}$.

Main approaches to clustering ensemble are divided into three main categories: *instance-based*, *cluster-based*, and *hybrid clustering* [29,12]. The instance-based approach is the basic one, since the hybrid approach is a combination of the other two, and the cluster-based approach employs instance-based schemes upon meta-clusters, i.e., clusters of clusters that compose each clustering of the ensemble.

In this work, we follow an approach to multilayer community detection that exploits the instance-based clustering ensemble scheme, which is here briefly recalled. An instance-based clustering ensemble method takes as input a co-occurrence (or co-association) matrix $\mathbf{M}$ defined over D and $\mathcal{E}$, such that the (i, j)-th entry of the matrix stores the number of clustering solutions in $\mathcal{E}$ in which the i-th and j-th objects appear in the same cluster, divided by the size of $\mathcal{E}$. Following a majority voting approach, $\mathbf{M}$ is pruned, such that all the objects whose corresponding entry in the matrix is above a certain threshold θ are joined into the same cluster. This approach has been proved to be equivalent to an agglomerative hierarchical clustering with single-linkage on $\mathbf{M}$, cutting the resulting dendrogram according to θ [11,12].

4 Ensemble-based Multilayer Community Detection

4.1 Problem statement

We are given a *multilayer network* graph $G_{\mathcal{L}} = (V_{\mathcal{L}}, E_{\mathcal{L}}, \mathcal{V}, \mathcal{L})$, with set of layers $\mathcal{L} = \{L_1, \ldots, L_\ell\}$ and set of entities $\mathcal{V}$. Each layer corresponds to a given type of entity relation, or edge-label. According to the general multilayer network model described in [17], for each choice of entity in $\mathcal{V}$ and layer in $\mathcal{L}$, we denote with $V_{\mathcal{L}} \subseteq \mathcal{V} \times \mathcal{L}$ the set containing the entity-layer combinations in which an entity is present in the corresponding layer. The set $E_{\mathcal{L}} \subseteq V_{\mathcal{L}} \times V_{\mathcal{L}}$ contains the undirected links between such entity-layer tuples. For every layer $L_i \in \mathcal{L}$, let $V_{L_i} = \{v \in \mathcal{V} \mid (v, L_i) \in V_{\mathcal{L}}\} \subseteq \mathcal{V}$ be the set of nodes in the graph of L_i, and $E_{L_i} \subseteq V_{L_i} \times V_{L_i}$ be the set of edges in L_i. To simplify notations, we will also refer to V_{L_i} and E_{L_i} as V_i and E_i, respectively. Note that while entities (i.e., elements of $\mathcal{V}$) are not required to participate in all layers, each entity has to appear in at least one layer, i.e., $\bigcup_{i \in 1..\ell} V_{L_i} = \mathcal{V}$. Moreover, the only inter-layer edges are regarded as "couplings" of nodes representing the same entity between different layers.

A key concept in this work is the *community structure ensemble* for a given multilayer network.

Definition 1 (Ensemble of community structures) Given a multilayer network $G_{\mathcal{L}} = (V_{\mathcal{L}}, E_{\mathcal{L}}, \mathcal{V}, \mathcal{L})$, with $\ell = |\mathcal{L}|$ layers, an *ensemble of layer-specific community structures* for $G_{\mathcal{L}}$ is a set $\mathcal{E} = \{\mathcal{C}_1, \ldots, \mathcal{C}_\ell\}$, such that each $\mathcal{C}_h$ (with $h = 1..\ell$) is a partitioning of the layer graph G_h (i.e., a community structure for G_h). □

We assume the availability of an ensemble of community structures for any given multilayer network. The community structures in the ensemble might be obtained by applying any community detection algorithm to each layer graph. We do not require dependency or correlation between the layer-specific community structures. Moreover, we ignore any information relating to the particular community detection method or configuration that was employed to produce the community structures. Also, according to the above definition, we remark that each community structure in the ensemble is regarded as a partitioning of a layer-specific graph, i.e., communities are disjoint in terms of node membership.

Given an ensemble of community structures for a multilayer network, our general objective is to compute a *consensus community structure* as a set of communities that are representative of how nodes were grouped and topologically-linked together over the layer community structures in the ensemble.

Definition 2 (Consensus community structure (meta-definition)) Given a multilayer network $G_{\mathcal{L}} = (V_{\mathcal{L}}, E_{\mathcal{L}}, \mathcal{V}, \mathcal{L})$ and an ensemble of community structures $\mathcal{E} = \{\mathcal{C}_1, \ldots, \mathcal{C}_\ell\}$ (with $\ell = |\mathcal{L}|$) defined over $G_{\mathcal{L}}$, a *consensus community structure* for $\mathcal{E}$ is a partitioning of a graph with nodes in $\mathcal{V}$ and edges in $E_{\mathcal{L}}$, which is representative of the community structures in $\mathcal{E}$. □

Definition 3 (Ensemble-based Multilayer Community Detection (EMCD) (meta-problem)) Given a multilayer network and an ensemble of layer-specific community structures for it, determine a consensus community structure from the ensemble. □

Note that the above meta-definition captures only the intuition that the consensus should *agree* with the community structures in the ensemble, but no hint is provided about the *quality* the consensus should have. In fact, according to the general desiderata in community detection, each community in the consensus should have high internal connectivity and low external connectivity. Moreover, given the multiplexity of the input graph, we seek to identify communities whose nodes are internally connected by many edges possibly of different types, and are externally connected by few edges of different types.

Nevertheless, we first need to define how to determine the community membership of nodes in the consensus structure. For this purpose, our general approach is to start from a *co-association*-based scheme defined over the layers, which resembles a major strategy in research on ensemble clustering (cf. Sect. 3.2) to infer a clustering solution (i.e., the consensus) that agrees most with the input clusterings. In the following Sect. 4.2, we describe two baseline approaches that rely on the co-association of nodes over the layer-specific community structures.; these baselines are called *cluster-induced EMCD* since they use co-association to derive a consensus clustering of nodes which is eventually used to compute a consensus community structure. Subsequently, in Sect. 4.3, we provide our main formulation of the EMCD problem, whereby we refine the definition of consensus community structure by integrating the requirement of quality of consensus via modularity optimization.

Table 1 Main symbols.

symbol	*description*
$\mathcal{V}$	Set of entities
L; $\mathcal{L}$; ℓ	Layer; set of layers; number of layers
$G_{\mathcal{L}}$	Multilayer graph
$V_{\mathcal{L}}, E_{\mathcal{L}}$	Set of nodes and set of edges in $G_{\mathcal{L}}$
C; $\mathcal{C}$	Community; community structure
$\mathcal{E}$	Ensemble of community structures
$\mathbf{M}$; θ	Co-association matrix; co-association threshold
$\mathcal{C}^*$	Consensus community structure
Q	Modularity function
$d(V_{\mathcal{L}})$	Total degree of $G_{\mathcal{L}}$
$d_L(C)$, $d_L^{int}(C)$	Degree of C and internal degree of C in graph of layer L
$d_{L,L'}^{ext}(C)$	External degree of C (i.e., twice the sum of inter-layer edges) between graphs of layers L and L'
γ_L	Resolution factor (specific for layer L)
β	Bit to enable inter-layer coupling factor
V_i, E_i	Set of nodes and set of edges of the i-th layer graph (G_i)
$\mathcal{C}_i$	Community structure of the i-th layer graph (G_i)
$E(\mathcal{C})$	Set of edges from the multilayer graph underlying consensus $\mathcal{C}$
$E_i(C)$	Set of edges of the subgraph of layer L_i induced from the set of nodes in community C
$E_{i,\mathcal{C}}(C)$	Set of edges of layer L_i from the multilayer graph underlying consensus $\mathcal{C}$ that are internal to C
$E_i(C_j, C_h)$	Set of edges of the subgraph of layer L_i induced from the set of nodes in communities C_j, C_h that link C_j to C_h
$E_{i,\mathcal{C}}(C_j, C_h)$	Set of edges of layer L_i from the multilayer graph underlying consensus $\mathcal{C}$ that link C_j to C_h
$Q_{i,j}^{+n_e}(\mathcal{C})$	Update-modularity function for adding n_e edges of layer L_i within community $C_j \in \mathcal{C}$
$Q_{i,jh}^{+n_e}(\mathcal{C})$	Update-modularity function for adding n_e edges of layer L_i to link communities C_j and C_h
$Q_{i,jh}^{-n_e}(\mathcal{C})$	Update-modularity function for removing n_e edges of layer L_i that link communities C_j and C_h

4.2 Baseline approaches

4.2.1 Direct Cluster-induced EMCD

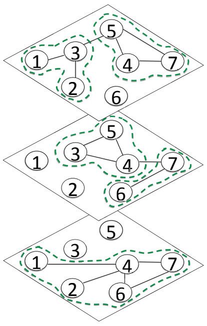
Given a multilayer network $G_{\mathcal{L}}$ and an ensemble $\mathcal{E}$ for it, we define the *co-association matrix* $\mathbf{M}$, with size $|\mathcal{V}| \times |\mathcal{V}|$, such that the (i,j)-th entry stores the number of communities shared by $v_i, v_j \in \mathcal{V}$, subject to the condition that the two nodes are linked to each other, divided by the number of layers (i.e., the size of the ensemble): $M(i,j) = \frac{|m_{ij}|}{\ell}$, where $m_{ij} = \{h \mid L_h \in \mathcal{L} \wedge \exists C \in \mathcal{C}_h, \mathcal{C}_h \in \mathcal{E},\ s.t.\ v_i, v_j\ in\ C \wedge (v_i, v_j) \in E_h\}$. Note that, since each node in a layer is assigned to only one community, the number of communities shared by any two nodes corresponds to the number of layers in which the two nodes are assigned the same community.

Moreover, in the definition of $\mathbf{M}$, we have introduced a *constraint of linkage* between nodes sharing a community in order to ensure that consensus communities will not contain disconnected components in the multilayer graph. It should be noted the linkage constraint is consistent with the requirement of having as high density as possible within any (consensus) community.

Our first proposed baseline, called *direct cluster-induced EMCD* (C-EMCD), requires matrix $\mathbf{M}$ to infer a clustering of $\mathcal{V}$, denoted as $\mathcal{S}$. More specifically, we wish to retain only meaningful co-association values, by dropping the lowest ones which reflect unlikely consensus memberships, and hence are due to noise. Therefore, $\mathbf{M}$ is subjected to a filtering step based on a user-specified parameter of *minimum co-association* $\theta \in [0,1]$. It should also be noted that, without this pruning, $\mathbf{M}$ would be a very dense matrix, which would make any clustering process computationally expensive.

By "cutting" $\mathbf{M}$ according to θ, the row (or column) projections corresponding to the entries greater than or equal to θ, identify a clustering of $\mathcal{V}$, where each cluster contains nodes that are ensured to be (directly or indirectly) linked together in $G_{\mathcal{L}}$. Finally, a consensus community structure, $\mathcal{C}^*$, is obtained where each community cor-

$\theta \geq 2/3$



	1	2	3	4	5	6	7
1	0	2/3	1/3	1/3	0	1/3	1/3
2	2/3	0	1/3	1/3	0	1/3	1/3
3	1/3	1/3	0	1/3	1/3	0	0
4	1/3	1/3	1/3	0	2/3	1/3	2/3
5	0	0	1/3	2/3	0	0	1/3
6	1/3	1/3	0	1/3	0	0	2/3
7	1/3	1/3	0	2/3	1/3	2/3	0

$C_1 = \{1,2\}$
$C_2 = \{3\}$
$C_3 = \{4,5,6,7\}$

	1	2	3	4	5	6	7
1	0	0	1/3	1/3	0	0	0
2	0	0	1/3	1/3	0	0	0
3	1/3	1/3	0	1/3	1/3	0	0
4	1/3	1/3	1/3	0	2/3	1/3	2/3
5	0	0	1/3	2/3	0	0	1/3
6	0	0	0	1/3	0	0	2/3
7	0	0	0	2/3	1/3	2/3	0

$C_1 = \{1\}$ $C_2 = \{2\}$
$C_3 = \{3\}$ $C_4 = \{4,5,6,7\}$

Fig. 1 Example multilayer network (with layer-specific community structures marked by green dashed curves) and corresponding co-association matrix and consensus community structure without (on the left) and with (on the right) the constraint of node linkage.

responds to the multilayer subgraph of $G_{\mathcal{L}}$ induced from each of the clusters in $\mathcal{S}$. Note that any consensus community will correspond to a connected subgraph, but not necessarily to a maximal complete subgraph of the multilayer network.

Example 1 Figure 1 shows how the co-association matrix $\mathbf{M}$ *would vary if the constraint of linkage between nodes was not considered. We set* $\theta \geq 2/3$ *to derive the consensus communities. If the constraint of linkage is discarded, then nodes 1 and 2 are included in the same consensus community* C_1*, because they belong to the same community in two out of three layers. On the contrary, when the constraint is considered, nodes 1 and 2 are assigned to different consensus communities* C_1 *and* C_2*, because they are disconnected in the graph. This occurs since, in their communities shared in the first and in third layer, nodes 1 and 2 were connected to nodes (i.e., 3 in the first layer, and 4 in the third layer) that will be assigned to different consensus communities as well. By contrast, nodes 5 and 6 belong to the same consensus community also in the case of linkage constraint enabled, even though they were never connected in the layer graphs.* ■

4.2.2 Constrained Cluster-induced EMCD

Each of the consensus communities produced by C-EMCD corresponds to the subgraph of $G_{\mathcal{L}}$ induced from the set of nodes belonging to that community. This method, however, discards the actual contribution of the different layers in determining the node co-associations, as specified in m_{ij}, for every $v_i, v_j \in V$. This limitation is overcome by the following alternative method, we call cluster-induced EMCD method with co-association constraints, for short *constrained cluster-induced EMCD*, and hereinafter denoted as CC-EMCD.

In CC-EMCD, the computation of consensus communities is refined in such a way that the structure of a consensus community accounts only for those specific layers that allow any two nodes to be linked in the shared community. Specifically, for each cluster $S \in \mathcal{S}$, we derive a community as the subgraph $C = \langle V, E \rangle$ of $G_{\mathcal{L}}$ with a set of nodes $V = S$ and a set of edges $E = \{(v_i, v_j, h) \in E_{\mathcal{L}} \mid v_i, v_j \in V \wedge h \in m_{ij}\}$.

CC-EMCD also differs from C-EMCD in the definition of the inter-community link structure of the consensus solution. Analogously to the approach used in the within-community link structure formation, the idea is to link any two consensus communities by using only the fraction of the multilayer graph that actually involves the connection of nodes from one community to another. In this case, however, we account for the layer contribution in an inverse way w.r.t. the within-community link structure. Specifically, we select only edges that correspond to those layers in which any two nodes do *not* appear in the co-association matrix, i.e., given two communities $C^{(1)}, C^{(2)} \in \mathcal{C}^*$, with node sets $V^{(1)}, V^{(2)}$, we compute the set of edges connecting them as $E(C^{(1)}, C^{(2)}) = \{(v_i, v_j, h) \in E_{\mathcal{L}} \mid v_i \in V^{(1)}, v_j \in V^{(2)} \wedge h \notin m_{ij}\}$.

Example 2 Consider a network with ten layers, and two nodes v_i, v_j linked to each other through six edges in $G_{\mathcal{L}}$. Also, the two nodes co-occur in four layer communities, thus $M(i,j) = 4/10$. Suppose $\theta = 0.5$, then the two nodes will be assigned to different communities in the consensus structure; moreover, v_i will be linked to v_j through 6-4=2 inter-community edges in the consensus structure. By contrast, in the case the two nodes are not directly connected in $G_{\mathcal{L}}$, then regardless of θ, they will be assigned to different clusters and will not be linked to each other. ■

4.3 Modularity-driven EMCD

In this section we formulate our main proposal to solve the EMCD problem. This stems from the observation that the previously discussed baselines C-EMCD and CC-EMCD produce a consensus community structure whose underlying graph can be seen as a *topological upper-bound* and a *topological lower-bound* of $G_{\mathcal{L}}$, respectively, for a given co-association threshold θ. Intuitively, while being a topological upper-bound, the solution provided by C-EMCD is not necessarily optimal in the sense that it might be redundant in terms of multilayer edges connecting different communities; by contrast, the solution provided by CC-EMCD is topologically minimal for $G_{\mathcal{L}}$, but it may loose important layer information, i.e., the CC-EMCD consensus communities may be poorly descriptive in terms of multilayer edges that characterize their internal connectivity.

In the respect of the general desiderata for community detection tasks, i.e., high within-community connectivity and low inter-community connectivity, our key idea is to formulate the EMCD problem as an optimization problem in which the consensus solution is optimal in terms of modularity, and is to be discovered within a hypothetical space of consensus community structures topologically bounded by CC-EMCD and C-EMCD solutions.

Definition 4 (Modularity-driven Ensemble Multilayer Community Detection problem) Given a multilayer network $G_{\mathcal{L}} = (V_{\mathcal{L}}, E_{\mathcal{L}}, \mathcal{V}, \mathcal{L})$, an ensemble of community structures $\mathcal{E} = \{\mathcal{C}_1, \ldots, \mathcal{C}_\ell\}$ for $G_{\mathcal{L}}$, and a co-association threshold θ, the *modularity-driven ensemble multilayer community detection* problem (M-EMCD) is to find a consensus community structure $\mathcal{C}^*$ for $G_{\mathcal{L}}$ by solving the following:

$$
\begin{aligned}
\mathcal{C}^* = \underset{\widehat{\mathcal{C}}}{\operatorname{argmax}} \quad & Q(\widehat{\mathcal{C}}) \\
\text{subject to} \quad & \mathcal{C}^{\perp} \sqsubseteq \widehat{\mathcal{C}} \sqsubseteq \mathcal{C}^{\top} \\
\text{with} \quad & \mathcal{C}^{\perp} := \textsf{CC-EMCD}(G_{\mathcal{L}}, \mathcal{E}, \theta) \\
& \mathcal{C}^{\top} := \textsf{C-EMCD}(G_{\mathcal{L}}, \mathcal{E}, \theta)
\end{aligned}
\tag{2}
$$

where, for any community structures $\mathcal{C}', \mathcal{C}''$ of $G_{\mathcal{L}}$, $\mathcal{C}' \sqsubseteq \mathcal{C}''$ holds iff $E(\mathcal{C}') \subseteq E(\mathcal{C}'')$, and $Q(\cdot)$ is the modularity function for multilayer networks. □

In the above definition, the relation of community structure "containment" (denoted by symbol $\sqsubseteq$) hints at searching for a consensus community structure over a multilayer graph whose topology might be refined to ensure better modularity of the community structure, paying particular attention to the enrichment of within-community structures and possibly to the simplification of the inter-community structures. As stated in the problem, the structure refinement is to be accomplished to preserve the topology of the input multilayer graph, according to the lower-bound and upper-bound consensus solutions.

It is worth emphasizing that the solution of the M-EMCD problem satisfies the two expected requirements. In fact, (i) the consensus $\mathcal{C}^*$ complies with the community structures in the input ensemble, because it is discovered from a space of candidates delimited by two community structures that are designed to be representative of the ensemble, and (ii) the consensus $\mathcal{C}^*$ is optimal w.r.t. a quality criterion that holds independently of the particular ensemble in input.

4.3.1 Multilayer modularity

Here we formally specify the modularity function, Q, required in our previously defined M-EMCD problem.

We propose an extension of modularity to multilayer networks by accounting for the layer-specific contributions of edges in the internal and external connectivity of the communities. Two key ingredients in multilayer modularity are the *resolution* and *inter-layer coupling* factors. The former models a notion of layer-specific relevance, thus helps mitigating the effect on the size distribution of community due to the *resolution limit* known in modularity [10]. The inter-layer coupling factor quantifies the strength of linkage between layers. In order to deal with scenarios in which a particular ordering among layers is required, we generalize the inter-layer coupling factor by admitting a *partial order relation* $\prec_{\mathcal{L}}$ over the layers.

Definition 5 (Modularity of Multilayer Network) Let $G_{\mathcal{L}} = (V_{\mathcal{L}}, E_{\mathcal{L}}, \mathcal{V}, \mathcal{L})$ be a multilayer network graph and, optionally, let $\prec_{\mathcal{L}}$ be a partial order relation over the set of layers $\mathcal{L}$. Given a community structure $\mathcal{C}$ for $G_{\mathcal{L}}$, we define the *multilayer modularity* of $\mathcal{C}$ as follows:

$$Q(\mathcal{C}) = \sum_{C \in \mathcal{C}} Q(C) = \frac{1}{d(V_{\mathcal{L}})} \sum_{C \in \mathcal{C}} \sum_{L \in \mathcal{L}} \left(d_L^{int}(C) - \gamma_L \frac{(d_L(C))^2}{d(V_{\mathcal{L}})} + \beta \sum_{L' \in \mathcal{P}(L)} d_{L,L'}^{ext}(C) \right) \tag{3}$$

where, for any community $C \in \mathcal{C}$, $d_L(C)$ and $d_L^{int}(C)$ are respectively the degree of C and the internal degree of C, by considering only the edges of layer L, $d(V_{\mathcal{L}})$ is the total degree of the entire graph, i.e., $d(V_{\mathcal{L}}) = \sum_{L \in \mathcal{L}} \sum_{v \in V_L} d(v)$, γ_L is a resolution parameter for edges of layer L, $d_{L,L'}^{ext}(C)$ is the external degree of C, i.e., twice the sum of inter-layer edges involving nodes that belong to C, $\beta \in \{0, 1\}$, and $\mathcal{P}(L)$ is the set of valid pairings with L defined as:

$$\mathcal{P}(L) = \begin{cases} \{L' \in \mathcal{L} \mid L \prec_{\mathcal{L}} L'\}, & \text{if } \prec_{\mathcal{L}} \text{ is defined} \\ \mathcal{L} \setminus \{L\}, & \text{otherwise} \end{cases}$$

□

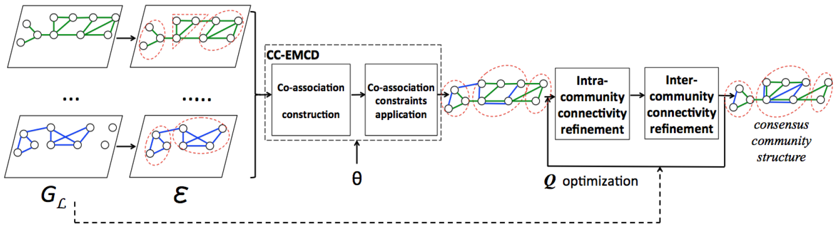


Fig. 2 Overview of the modularity-based EMCD framework

Throughout the rest of this work we set the value of γ_L (for each $L \in \mathcal{L}$) to the default one. Note this is a reasonable choice, which avoids introducing any bias in the evaluation of how far the actual amount of interactions deviates from the expected random connections, over each layer. In particular, setting each γ_L to a value different from one will affect not only the modularity value but, more importantly in our context, also the rate of refinement of the within-community and inter-community connectivity, which will be discussed next. Concerning the $d^{ext}(C)$ term, we expect that accounting for inter-layer edges would lead to complex forms of consensus communities, which deserve attention that cannot be ensured in this paper due to space limitations; therefore, we shall omit the inter-layer edges contribution by setting $\beta = 0$. We will leave the study of both types of parameters as future work (cf. Sect. 7).

4.3.2 An approximation algorithm for the M-EMCD *problem*

Overview of the algorithm. Figure 2 sketches an overview of the proposed M-EMCD method.[2] Given an input multilayer graph $G_{\mathcal{L}}$ and an ensemble $\mathcal{E}$ for it, and a co-association threshold θ, the CC-EMCD method is first employed to produce the "lower-bound" consensus community structure. In the core execution of the M-EMCD method, this consensus is iteratively refined through two main steps, respectively within-community and across-community, until modularity is optimized to return the final consensus community structure. Note that the refinement is performed to preserve the topology of $G_{\mathcal{L}}$, according to the "upper-bound" consensus.

Modularity update functions. As above anticipated, the M-EMCD method requires a stage of refinement over an initial structure of consensus communities (which corresponds to the θ-based lower-bound consensus over the input $G_{\mathcal{L}}$ and $\mathcal{E}$). In operational terms, this refinement is accomplished by insertion/removal of edges, possibly from different layers, inside a community and/or between communities. Each step of refinement requires evaluation of the update occurring in our multilayer modularity. In this regard, we manipulate Eq. 3 in order to formulate modularity update functions that reflect the change to the community structure due to some structural modifications in a particular community (or pairs of communities) and for a particular layer. Let us first rewrite Eq. 3 as follows:

[2] We will make the source code available upon decision on this article.

$$
\begin{aligned}
d^2(V_{\mathcal{L}})Q(\mathcal{C}) &= \sum_{C\in\mathcal{C}}\sum_{L\in\mathcal{L}} d(V_{\mathcal{L}})d_L^{int}(C) - (d_L(C))^2 = \\
&= d(V_{\mathcal{L}})\underbrace{\sum_{C\in\mathcal{C}}\sum_{L\in\mathcal{L}} d_L^{int}(C)}_{D_{L,C}^{int}} - \underbrace{\sum_{C\in\mathcal{C}}\sum_{L\in\mathcal{L}} (d_L(C))^2}_{D_{L,C}^{2}} \Rightarrow \\
\Rightarrow Q(\mathcal{C}) &= \frac{1}{d(V_{\mathcal{L}})} D_{L,C}^{int} - \frac{1}{d^2(V_{\mathcal{L}})} D_{L,C}^2
\end{aligned} \tag{4}
$$

Insertion of edges within a community. Suppose n_e edges of type corresponding to layer L_i are added between nodes (i.e., $2n_e$ nodes) belonging to community C_j. The resulting modularity, denoted as $Q_{i,j}^{+n_e}(\mathcal{C})$, is expressed by the following update function:

$$
Q_{i,j}^{+n_e}(\mathcal{C}) = \frac{D_{L,C}^{int} + 2n_e}{d(V_{\mathcal{L}}) + 2n_e} - \frac{D_{L,C}^2 + 4n_e \times (n_e + d_{L_i}(C_j))}{(d(V_{\mathcal{L}}) + 2n_e)^2} \tag{5}
$$

It follows that, to calculate modularity of a community structure $\mathcal{C}$ subject to an update involving the structure of any C_j according to edges of L_i, it is enough to store the quantities $d_{L_i}(C_j)$ ($\forall L_i \in \mathcal{L}, C_j \in \mathcal{C}$), and the cumulated counts $D_{L,C}^{int}$ and $d(V_{\mathcal{L}})$.

Note that we do not consider removal of within-community edges: this is explained since, as it will be clarified later in this section, M-EMCD exploits the consensus solution generated by CC-EMCD, which represents the topological lower-bound for a given θ, and as such it does not require further pruning of within-community edges.

Insertion of edges between communities. Suppose $N_e = n_{e_1} + \ldots + n_{e_K}$ edges of a selected type $L_i \in \mathcal{L}$ are added to link a selected community C_j to any of its neighbors in the set $N(C_j) = \{C_{j_1}, \ldots, C_{j_K}\}$, i.e., n_{e_k} edges are inserted between nodes in C_j and nodes in any of C_{j_k}. The resulting modularity, denoted as $Q_{i,j}^{+N_e}(\mathcal{C})$, is expressed by the following update function:

$$
Q_{i,j}^{+N_e}(\mathcal{C}) = \frac{D_{L,C}^{int}}{d(V_{\mathcal{L}}) + 2N_e} - \frac{D_{L,C}^2 + N_e^2 + \sum_k n_{e_k}^2 + 2N_e d_{L_i}(C_j) + 2\sum_k n_{e_k} d_{L_i}(C_{j_k})}{(d(V_{\mathcal{L}}) + 2N_e)^2} \tag{6}
$$

In case of insertion of n_e edges to link community C_j with only one of its neighbors, say C_h, the above equation is rewritten as:

$$
Q_{i,jh}^{+n_e}(\mathcal{C}) = \frac{D_{L,C}^{int}}{d(V_{\mathcal{L}}) + 2n_e} - \frac{D_{L,C}^2 + 2n_e^2 + 2n_e(d_{L_i}(C_j) + d_{L_i}(C_h))}{(d(V_{\mathcal{L}}) + 2n_e)^2} \tag{7}
$$

Removal of edges between communities. Analogously, in case of removal of N_e edges from a selected community C_j and any of its neighbors in the set $N(C_j) = \{C_{j_1}, \ldots, C_{j_K}\}$, the resulting modularity, denoted as $Q_{i,j}^{-N_e}(\mathcal{C})$, is expressed by the following update function:

$$
Q_{i,j}^{-N_e}(\mathcal{C}) = \frac{D_{L,C}^{int}}{d(V_{\mathcal{L}}) - 2N_e} - \frac{D_{L,C}^2 + N_e^2 + \sum_k n_{e_k}^2 - 2N_e d_{L_i}(C_j) - 2\sum_k n_{e_k} d_{L_i}(C_{j_k})}{(d(V_{\mathcal{L}}) - 2N_e)^2} \tag{8}
$$

In case of removal of n_e edges linking community C_j and only one of its neighbors, say C_h, the above equation is rewritten as:

$$Q^{-n_e}_{i,jh}(\mathcal{C}) = \frac{D^{int}_{L,C}}{d(V_{\mathcal{L}}) - 2n_e} - \frac{D^2_{L,C} + 2n_e^2 - 2n_e(d_{L_i}(C_j) + d_{L_i}(C_h))}{(d(V_{\mathcal{L}}) - 2n_e)^2} \quad (9)$$

The M-EMCD algorithm. Algorithm 1 shows our proposed algorithmic solution for the M-EMCD problem. According to the previously presented overview, Algorithm 1 is a a hill-climbing algorithm for the modularity-based multilayer community detection problem. The algorithm starts with the consensus solution provided by the topological-lower-bounded baseline, then iteratively seeks a better solution in terms of modularity by incrementally refining the within-community connectivity and the inter-community connectivity, until no further improvements can be found.

The algorithm starts by invoking the CC-EMCD method to obtain an initial consensus community structure (lines 1-2), then the optimization is performed in two main stages, by examining one layer at a time:

- In the first stage (lines 6-12), the algorithm seeks the community C_j in the current consensus whose refinement C'_j corresponds to the maximum modularity in the consensus if this would contain C'_j in place of C_j; moreover, if this leads to an increment in the current value of modularity, the consensus is actually updated with C'_j (lines 10-12).
- In the second stage (lines 13-20), the algorithm attempts to refine the connectivity between C'_j and any its neighbor communities, updating the consensus at each step of modularity improvement.

The two stages are repeated iteratively until a maximum of modularity is reached (lines 3-22) and a final consensus community structure is produced.

The within-community refinement is carried out by function update_community (line 7). For any layer L_i and community C_j, selected from the current consensus $\mathcal{C}$, it adds to C_j as many edges from the graph of L_i as possible, i.e., the set obtained from the difference between the set of edges of the subgraph of layer L_i induced from the set of nodes in community C (denoted as $E_i(C_j)$ in line 25) and the set of edges of layer L_i from the multilayer graph underlying $\mathcal{C}$ that are internal to C (i.e., $E_{i,\mathcal{C}}(C_j)$ in line 25). The modified C_j and its modularity are returned.

The inter-community refinement is carried out by function update_community_structure (line 14). For any layer L_i and adjacent communities C_j, C_h, selected from the current consensus $\mathcal{C}$, it performs the following operations and evaluates the corresponding modularity: (i) it removes all edges of layer L_i from the multilayer graph underlying consensus $\mathcal{C}$ that link C_j to C_h (denoted as $E_{i,\mathcal{C}}(C_j, C_h)$ in line 30); (ii) it adds all edges between nodes in C_j and nodes in C_h from G_i that are not contained in the set previously removed (line 32); (iii) it performs the previous operations jointly. The best modularity over the three contingencies and the corresponding modified inter-connectivity between C_j, C_h are returned.

4.4 Example of execution of EMCD methods

Figures 3–4 illustrate an example of multilayer community detection with the outcomes of our proposed EMCD methods. Given the network with 10 nodes, 31 total edges and 3 layers shown in Fig. 3(left), suppose the layer-specific community structures shown in

Algorithm 1 Modularity-based Ensemble Multilayer Community Detection

Input: Multilayer graph $G_{\mathcal{L}} = (V_{\mathcal{L}}, E_{\mathcal{L}}, \mathcal{V}, \mathcal{L})$, ensemble of community structures $\mathcal{E} = \{\mathcal{C}_1, \ldots, \mathcal{C}_\ell\}$ (with $\ell = |\mathcal{L}|$), co-association threshold $\theta \in [0, 1]$.
Output: Consensus community structure $\mathcal{C}^*$ for $G_{\mathcal{L}}$.

$\mathcal{C}_{lb} \leftarrow \mathsf{CC\text{-}EMCD}(G_{\mathcal{L}}, \mathcal{E}, \theta)$
$\mathcal{C}^* \leftarrow \mathcal{C}_{lb}$
repeat
    for $L_i \in \mathcal{L}$ **do**
        $Q \leftarrow Q(\mathcal{C}^*)$
        {*Refine intra-community connectivity of* C_j}
        for $C_j \in \mathcal{C}^*$ **do**
            $\langle C'_j, Q'_j \rangle \leftarrow \mathsf{update\_community}(\mathcal{C}^*, C_j, L_i)$
        end for
        $j^* \leftarrow \mathrm{argmax}\, Q'_j$
        if $Q'_{j^*} > Q$ **then**
            $\mathcal{C}^* \leftarrow \mathcal{C}^* \setminus C_j \cup C'_{j^*}$
        end if
        {*Refine inter-community connectivity between* C_{j^*} *and each of its neighbors*}
        for $C_h \in N(C_{j^*})$ **do**
            $\langle \mathcal{C}'_h, Q'_h \rangle \leftarrow \mathsf{update\_community\_structure}(\mathcal{C}^*, C_{j^*}, C_h, L_i)$
        end for
        $h^* \leftarrow \mathrm{argmax}\, Q'_h$
        if $Q_{h^*} > Q$ **then**
            $\mathcal{C}^* \leftarrow \mathcal{C}'_{h^*}$
            $Q \leftarrow Q_{h^*}$
        end if
    end for
until $Q(\mathcal{C}^*)$ cannot be further maximized
return $\mathcal{C}^*$

function $\mathsf{update\_community}(\mathcal{C}, C_j, L_i)$
$E \leftarrow E_i(C_j) \setminus E_{i,\mathcal{C}}(C_j)$
$C'_j \leftarrow addEdges(E, C_j)$
$Q'_j \leftarrow Q^{+|E|}_{i,j}(\mathcal{C})$ {*Update modularity through Eq. (5)*}
return $\langle C'_j, Q'_j \rangle$

function $\mathsf{update\_community\_structure}(\mathcal{C}, C_j, C_h, L_i)$
$E \leftarrow E_{i,\mathcal{C}}(C_j, C_h)$
$Q^- \leftarrow Q^{-|E|}_{i,jh}(\mathcal{C})$ {*Update modularity through Eq. (9)*}
$E' \leftarrow E_i(C_j, C_h) \setminus E$
$Q^+ \leftarrow Q^{+|E'|}_{i,jh}(\mathcal{C})$ {*Update modularity through Eq. (7)*}
$\mathcal{C}^+ \leftarrow addEdges(E', \mathcal{C})$
$\mathcal{C}^- \leftarrow delEdges(E, \mathcal{C})$
$Q^{\pm} \leftarrow Q^{-|E|}_{i,jh}(\mathcal{C}^+)$ {*Update modularity through Eq. (9)*}
$\mathcal{C}^{\pm} \leftarrow delEdges(E, \mathcal{C}^+)$
$\langle \mathcal{C}, Q \rangle \leftarrow \mathrm{argmax}\{Q^+, Q^-, Q^{\pm}\}$
return $\langle \mathcal{C}, Q \rangle$

Fig. 3(right) have been separately provided by some community detection scheme. Starting from this ensemble of community structures, Fig. 4 shows the consensus community structures computed by C-EMCD*,* CC-EMCD *and* M-EMCD *methods on the example network for three settings of parameter θ — note that these correspond to regimes within which the assignment of nodes to communities do not change. While the lower range of θ is not meaningful (since all three methods generate a single community), it is in-*

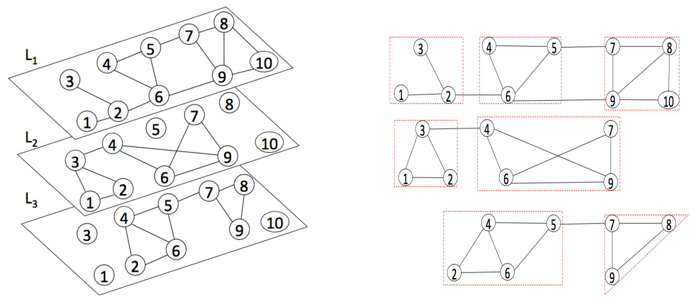
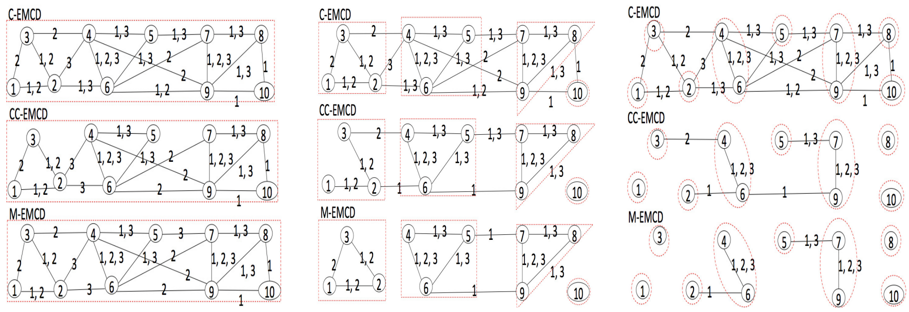


Fig. 3 Example multilayer network (on the left) and community structures identified on each layer graph (on the right). Communities are marked by red dashed polygons.



(a) (b) (c)

Fig. 4 Consensus community structures obtained by EMCD methods (from top: C-EMCD, CC-EMCD, M-EMCD) for three settings of θ: (a) $\theta \in [0, \frac{1}{3}]$, (b) $\theta = 0.5$, and (c) $\theta \in [\frac{2}{3}, 1]$. Numbers on the edges correspond to the layers on which two nodes are linked together.

teresting to note that edges from some layer are discarded in CC-EMCD*, whereas they are partially recovered by* M-EMCD *for improving the modularity of the solution. In the case of $\theta = 0.5$,* M-EMCD *and* C-EMCD *produce three communities, whereas in the* CC-EMCD *solution the singleton community composed by node 10 is added. Note that nodes $1, 2, 3$ are assigned to the same community, which is expected since they were originally located in the same community in the ensemble community structure of two layers out of three; the same happens for nodes $4, 5, 6$ and for $7, 8, 9$. Again, the same assignment of nodes is provided by* CC-EMCD *and* M-EMCD*, however the former solution is composed of a lower number of edges; part of them are recovered in the solution by* M-EMCD*, while other edges disappear, which result in an increase in modularity. Finally, for the highest range of $\theta \in [2/3, 1]$, as expected the methods tend to produce more and smaller consensus communities. This is particularly evident in the* M-EMCD *solution, in which a quite severe pruning in the edge structure is performed in order to reach a consensus that reflects the tough constraint of co-association imposed by θ. Comparing the consensus community structure found by* M-EMCD *and* C-EMCD*, we observe that* M-EMCD *performs a pruning of the inter-community edges, while it mostly preserves the within-community edges, eventually increasing the modularity of the consensus structure; this is particularly evident in the comparison of* M-EMCD *and* C-EMCD *solutions corresponding to $\theta = 0.5$ and $\theta \in [0, 1/3]$.* ■

4.5 Computational complexity aspects

We discuss here the computational complexity of our proposed EMCD methods.

Let us first consider the C-EMCD and CC-EMCD methods. The computational cost of C-EMCD is mainly due to the construction of the co-association matrix $\mathbf{M}$. This can be incrementally performed, by requiring a single scan of the adjacency list of the multilayer graph. More in detail, we maintain a consensus structure index to store the consensus community membership of each entity in $\mathcal{V}$, and, for one layer at a time, the corresponding adjacency list and community structure. For every entity, we iterate over its neighbors on each layer graph to check if the two nodes belong to the same community in that layer, and to update the count of communities shared by the two nodes. The cost of update of this count is constant as it requires direct accesses to the consensus index. Therefore, the time complexity of C-EMCD is $\mathcal{O}(|E_{\mathcal{L}}|)$.

The CC-EMCD method has the same time and spatial complexity as C-EMCD, with one difference. This corresponds to a hash index on a data structure that stores, for every pair of entities, the set of layers on which the two entity-nodes are adjacent and assigned to the same community and the set of layers the two entity-nodes are adjacent but not assigned to the same community. Since accessing this hash table has cost $\mathcal{O}(1)$, the time complexity of CC-EMCD is $\mathcal{O}(|E_{\mathcal{L}}|)$.

Let us now discuss the complexity of the main method, M-EMCD. This has at least the same complexity as CC-EMCD, because the latter is performed in the initial step of Algorithm 1 (line 1). The cost of the execution of the main loop depends on the number I of iterations needed to converge at a local optimum. In every iteration, M-EMCD searches for the best community to refine internally and externally (with its neighbors), through the routines update_community and update_community_structure, respectively; recall that both routines exploit appropriate modularity-update rules, which are performed efficiently with spatial cost $\mathcal{O}(\ell \times |\mathcal{C}^*|)$, since they require to store the quantities $d_{L_i}(C_j)$, plus few more global counts (e.g., $d(V_{\mathcal{L}})$, $D^{int}_{L,C}$). The cost of a single evaluation of update_community is comprised of the cost of two manipulations in the community structure (line 25 and line 26), both bounded by the number of edges within a particular community and of a particular layer, plus the $\mathcal{O}(1)$ cost of modularity update (line 27). Therefore, performing update_community over all layers and communities (lines 4–8) is $\mathcal{O}(|E_{\mathcal{L}}| + \ell \times |\mathcal{C}^*|)$. Selecting the best-modularity community takes $\mathcal{O}(\ell \times |\mathcal{C}^*|)$, over all layers and communities (lines 9–11). The inter-community refinement stage (lines 13–21) also costs $\mathcal{O}(|E_{\mathcal{L}}| + \ell \times |\mathcal{C}^*|)$. In fact, a single evaluation of update_community_structure requires operations whose cost is either bounded by the number of layer-specific internal edges (lines 30, 32, 34, 35, 37) or constant (lines 31, 33, 36, 38). Overall, the time complexity of M-EMCD is $\mathcal{O}(I \times (|E_{\mathcal{L}}| + \ell \times |\mathcal{C}^*|))$.

5 Experimental Evaluation

5.1 Datasets

Our experimental evaluation was mainly conducted on seven real-world multilayer network datasets. This selection is motivated by (i) diversification in terms of data domain (i.e., transportation networks, mixed online/offline relations, single-platform and multi-platform relations in social media, co-authorships, classroom relations), and (ii) public availability, which enables reproducibility.

AUCS [16] describes relationships among university employees: work together, lunch together, off-line friendship, friendship on Facebook, and coauthorship. *DBLP* [16]

Table 2 Main characteristics of our evaluation network datasets. Mean and standard deviation over the layers are reported for degree, average path length, and clustering coefficient statistics.

	#entities ($\lvert\mathcal{V}\rvert$)	#edges	#layers	node set coverage	edge set coverage	degree	avg. path length	clust. coeff.
AUCS	61	620	5	0.73	0.20	10.43 ± 4.91	2.43 ± 0.73	0.43 ± 0.097
DBLP	1 314 050	7 647 677	44	0.06	0.02	7.46 ± 3.06	8.59 ± 1.39	0.69 ± 0.13
EU-Air	417	3588	37	0.13	0.03	6.26 ± 2.90	2.25 ± 0.34	0.07 ± 0.08
FF-TW-YT	6407	74836	3	0.58	0.33	9.97 ± 7.27	4.18 ± 1.27	0.13 ± 0.09
Higgs-Twitter	456 631	16 070 185	4	0.67	0.25	18.28 ± 31.20	9.94 ± 9.30	0.003 ± 0.004
London	369	441	3	0.36	0.33	2.12 ± 0.16	11.89 ± 3.18	0.036 ± 0.032
VC-Graders	29	518	3	1.00	0.33	17.01 ± 6.85	1.66 ± 0.22	0.61 ± 0.89

represents co-authorships over different time slices, which correspond to the publication years in the period 1971-2014. EU-Air transport network [16] (*EU-Air*, for short) represents European airport connections considering different airlines. FF-TW-YT (stands for FriendFeed, Twitter, and YouTube) [8] was built by exploiting the feature of FriendFeed as social media aggregator to align registered users who were also members of Twitter and YouTube. *Higgs-Twitter* [16] represents friendship, reply, mention, and retweet relations among Twitter users. London transport network [33] (*London*, for short) models three types of connections of train stations in London: underground lines, overground, and DLR. 7thGraders [33] (*VC-Graders*, for short) represents students involved in three relationships: friendship, work together, and affinity in the class.

Table 2 reports for each dataset, the size of set $\mathcal{V}$, the number of edges in all layers, the average coverage of node set (i.e., $1/|\mathcal{L}| \sum_{L_i \in \mathcal{L}}(|V_i|/|\mathcal{V}|)$), and the average coverage of edge set (i.e., $1/|\mathcal{L}| \sum_{L_i \in \mathcal{L}}(|E_i|/\sum_{L_i}|E_i|)$). The table also shows basic, monoplex structural statistics, such as degree, average path length, and clustering coefficient, for the layer networks of each dataset.

We also resorted to a synthetic multilayer network generator, *mLFR Benchmark*,[3] mainly for our evaluation of efficiency of the M-EMCD method (cf. Section 6.1.5). mLFR extends the tool proposed in [20] for multilayer networks. We used mLFR to create a multilayer network with 1 million of nodes, setting other available parameters as follows: 10 layers, average degree 30, maximum degree 100, mixing at 20% , layer mixing 2. We hereinafter refer to this synthetic network as *mLFR-1M*.

5.2 Competing methods

We resorted to state-of-the-art methods for community detection, which cover all of the main categories of existing approaches, namely *flattening*, *aggregation* and *direct* methods (cf. Sect. 2).

As representative of the category of flattening methods, we define a baseline method that applies a community detection method on the *flattened graph* of the input multilayer network, that is, a weighted multigraph having $\mathcal{V}$ as set of nodes, the set of edges $\{(u,v) \mid \exists u,v \in \mathcal{V} \wedge L \in \mathcal{L} \ \wedge \ ((u,L),(v,L)) \in E_{\mathcal{L}}\}$, and edge weights that express the number of layers on which two nodes are connected. In this work, we chose to use the serial version of the *Nerstrand* algorithm, recently developed by La Salle and Karypis [21]. Our choice is motivated since Nerstrand has shown to be both an extremely efficient and effective method to discover non-overlapping communities in (single-layer) weighted graphs via modularity optimization based on the multilevel paradigm *coarsening-initial clustering-uncoarsening*.

[3] http://www.ii.pwr.edu.pl/brodka/mlfr.php

We also comparatively evaluate our approach with the following multilayer community detection methods, previously discussed in Section 2: *ABACUS* [4], *Principal Modularity Maximization* (PMM) [30],[4] *Generalized Louvain* (GL) [23],[5] *Multiplex-Infomap* [6],[6] *MultiGA* [1], *MultiMOGA* [2], and *Locally Adaptive Random Transitions* (LART) [18]. Recall that PMM and ABACUS are representative methods of the category of *aggregation* approaches (the same to which our EMCD methods belong), while the latter five are *direct* methods. Apart from ABACUS, all methods were selected because, while having different characteristics, they all use modularity either as optimization criterion or as evaluation criterion (LART) to produce the final community structure.

5.3 Assessment criteria

We use both internal and external validation criteria to assess the consensus community structure solution provided by EMCD methods.

Internal criteria include, besides evaluation of the multilayer modularity, the *redundancy* measure and our defined *multilayer silhouette*. The redundancy measure is based on the assumption that a high quality community should have many "redundant" connections, i.e., pairs of nodes connected through edges of different layers [3]. For each community, it is defined as the actual number of redundant connections divided by the theoretical maximum (i.e., total number of layers times total number of node pairs in the community); a global redundancy is finally obtained averaging the redundancy values over all communities. Note that, while ranging between 0 and 1, redundancy is not defined for singleton communities.

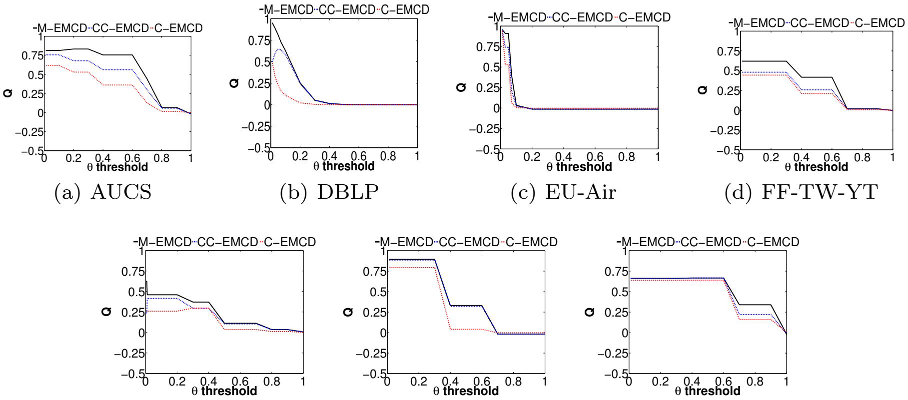
While the redundancy accounts for the coverage of layers within each community, we also consider the quality of cluster assignment, i.e., how well each node fits its assigned community. In this respect, silhouette measure [28] is a suitable criterion, however it is originally designed for single-layer graphs. We introduce a twofold modification in the definition, in that (i) the distance computation terms are linearly combined over all layers, and (ii) the distance between two nodes is computed as one minus the Jaccard coefficient defined over the layer-specific sets of neighbors. Silhouette may range from -1 to 1 (the higher, the better).

As for the external criteria, we use the *normalized mutual information* (NMI), in its two versions by Strehl and Ghosh [29] and by Dhillon *et al.* [7]. NMI determines the alignment in terms of community memberships of nodes between a community structure and another one used as reference, so that the higher the NMI value the better the alignment — NMI ranges between 0 and 1. In our setting, for any given multilayer network, the reference will correspond to the solution obtained by Nerstrand on the flattened multilayer graph (cf. Sect. 5.2), or alternatively to the layer-specific community structure solutions obtained by Nerstrand on each of the layer graphs.

5.4 Experimental settings

The main parameter of EMCD methods, θ, was varied in its full range of admissible values, at a fine-grain step (0.001). We shall present results corresponding to values of θ that determined meaningful variations in terms of multilayer modularity (Eq. 3),

[4] `http://leitang.net/heterogeneous_network.html`
[5] http://netwiki.amath.unc.edu/GenLouvain/GenLouvain
[6] http://muxviz.net/



(a) AUCS (b) DBLP (c) EU-Air (d) FF-TW-YT

(e) Higgs-Twitter (f) London (g) VC-Graders

Fig. 5 Consensus solutions obtained by EMCD methods, for varying θ: modularity values. *(Best viewed in color)*

specifically the values in the set $\{0.01, 0.03, 0.05, 0.07\}$ and from 0.1 to 0.9 with step of 0.1. Moreover, to generate the ensemble from each of the evaluation network datasets, we applied Nerstrand on the individual layer-specific graphs — note that, by default, it does not require an input number of communities.

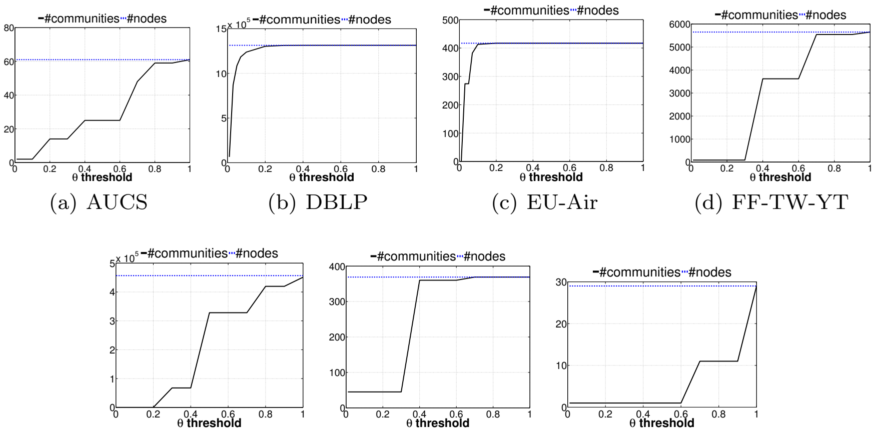
As far as the competing methods, GL determines a community structure for each layer of a network, therefore a final solution was derived by assigning each node to the community which lays on most of the layers. PMM requires an input number of communities. We devised two configurations for this method: the one in which we conducted an exhaustive search for the number of communities corresponding to the best performance in terms of modularity, on every dataset; and the other one in which the input parameter was set to the number of communities determined by our method; we will use notation PMM$^{k^*}$ to refer to the latter configuration of PMM. Moreover, we set to 50 the number of runs of the k-means clustering method, whose application is required by PMM to obtain the consensus solution. As concerns ABACUS, this method utilizes the *eclat* frequent-pattern mining method to generate the transactional representation of the ensemble. As by default configuration, the main model parameter in ABACUS (i.e., the minimum support threshold) was kept quite low on each dataset, typically in the range from three to ten. For the genetic approaches (i.e., MultiGA and MultiMOGA), LART, and Multiplex-Infomap, we referred to the default parameters as specified in their respective works.

6 Results

6.1 Evaluation of EMCD methods

6.1.1 Modularity

Figures 5 and 6 show the modularity and the size of the consensus community structure, respectively, obtained by each of the EMCD methods, by varying θ. The methods generate consensus solutions of the same size, for any particular dataset and θ, therefore the number of consensus communities is plotted once; also, the number of nodes for



(a) AUCS (b) DBLP (c) EU-Air (d) FF-TW-YT

(e) Higgs-Twitter (f) London (g) VC-Graders

Fig. 6 Consensus solutions obtained by EMCD methods, for varying θ: number of communities. *(Best viewed in color)*

Table 3 Best-modularity consensus solutions obtained by M-EMCD: modularity value with corresponding θ regime and number of communities (with percentage of singletons), and gains in modularity w.r.t. the other EMCD methods.

network	θ range	modularity	#communities (% singletons)	gain w.r.t. CC-EMCD	gain w.r.t. C-EMCD
AUCS	[0.2, 0.4)	0.863	14 (21.4%)	+0.15	+0.33
DBLP	[0.01, 0.03)	0.952	64 779 (4.0%)	+0.46	+0.46
EU-Air	[0.027, 0.07)	0.910	274 (76.6%)	+0.17	+0.38
FF-TW-YT	(0, 0.34)	0.620	86 (3.5%)	+0.14	+0.18
Higgs-Twitter	(0, 0.01)	0.625	86 (0%)	+0.39	+0.37
London	(0, 0.34)	0.895	45 (0%)	+0.01	+0.10
VC-Graders	[0.67, 1)	0.340	11 (0%)	+0.12	+0.18

each network graph is reported as a constant, blue dashed line, which corresponds to the upper bound of the community number.

First, the modularity value, for all methods, tends to follow a non-increasing trend as the threshold value increases. On the contrary, the number of communities tends to increase as the threshold value becomes higher (until it eventually reaches the number of nodes in the graph); this is expected, since it is clear that a high θ value will penalize the assignment of two nodes to the same community.

Among the three methods, M-EMCD turns out to be the absolute winner, reaching the highest modularity over all datasets. Moreover, the M-EMCD solution has as good as or better modularity than that obtained by the other two methods for the same θ.

Table 3 summarizes the M-EMCD consensus configurations corresponding to the best modularity performances for each dataset, focusing on non-trivial solutions (i.e., consensus structures with at least two communities but less than the total number of nodes). It highlights the evident superiority of M-EMCD against the other EMCD methods. Note also that, with the exception of Higgs-Twitter and DBLP, CC-EMCD tends to prevail against C-EMCD in terms of modularity.

The table also provides indications about the fraction of singleton communities in the consensus, i.e., disconnected components comprised of a single node of the graph.

Table 4 Silhouette and global redundancy corresponding to the best consensus solutions obtained by M-EMCD, and gains w.r.t. the other EMCD methods.

network	silhouette			redundancy		
	M-EMCD	gain w.r.t. CC-EMCD	gain w.r.t. C-EMCD	M-EMCD	gain w.r.t. CC-EMCD	gain w.r.t. C-EMCD
AUCS	0.366	+0.13	+0.81	0.910 ± 0.097	+0.04	0.0
DBLP	0.084	+0.20	+0.10	0.512 ± 0.290	0.0	-0.001
EU-Air	0.093	+0.44	+0.44	0.620 ± 0.103	+0.02	0.0
FF-TW-YT	0.041	+1.03	+0.04	0.615 ± 0.282	+0.03	0.0
Higgs-Tw.	0.052	+0.33	+0.36	0.658 ± 0.247	+0.40	0.0
London	0.179	+0.03	+0.03	0.533 ± 0.328	+0.03	0.0
VC-Graders	0.288	+0.05	+0.18	0.945 ± 0.064	+0.03	0.0

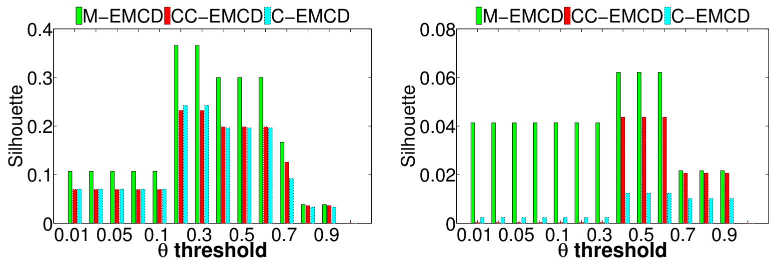


Fig. 7 Silhouette by EMCD methods for varying θ, on AUCS (left) and FF-TW-YT (right).

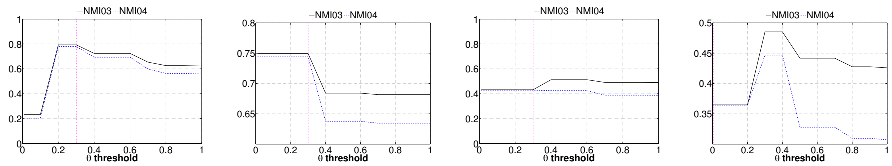
Since these communities correspond to nodes that did not satisfy the θ-based co-association constraint, this can be seen as related to the ability of M-EMCD to detect outliers in the consensus solution. We observe that, with the exception of EU-Air, the best-modularity consensus includes zero or a small fraction of singletons, which indicates that results are not biased by the presence of a large number of singleton communities.

6.1.2 Community membership

We evaluated the quality of EMCD consensus solutions also from the viewpoint of community membership of nodes. In this regard, we took two perspectives, corresponding to an internal criterion approach, based on silhouette, and to an external criterion approach, based on NMI, respectively.

Silhouette evaluation. The M-EMCD method behaves substantially better than the other EMCD methods also in terms of silhouette. As shown in Table 4, M-EMCD gains 0.32 (up to 1.03 on FF-TW-YT) w.r.t. CC-EMCD and 0.28 (up to 0.81 on AUCS) w.r.t. C-EMCD— recall that silhouette may range from -1 to 1. Figure 7 provides details on the silhouette of the consensus community structure obtained by EMCD methods, for varying θ. We observe that the silhouette of M-EMCD is higher (i.e., better) than CC-EMCD and C-EMCD over the various θ values, and in most cases M-EMCD outperforms the other methods. Interestingly, the latter occurs consistently with the best-modularity performance, i.e., the largest gain in silhouette is obtained by M-EMCD over the same θ range that leads to the best modularity.

NMI evaluation. Figure 8 shows the NMI performances obtained by comparing the EMCD consensus solutions with the corresponding ones by Nerstrand on the flattened graph, by varying θ, for some selected datasets. Note that since EMCD methods



(a) AUCS (b) London (c) FF-TW-YT (d) Higgs-Twitter

Fig. 8 NMI03 [29] and NMI04 [7] performances by EMCD methods w.r.t. Nerstrand on the flattened graphs, for varying θ. The vertical colored line on each plot refers to the θ value corresponding to the best-modularity consensus by M-EMCD. *(Best viewed in color)*

Table 5 Mean and standard deviation of NMI results for each dataset, between the best-modularity consensus solution by M-EMCD and the community structure by Nerstrand on each layer graph, averaged over the various layers.

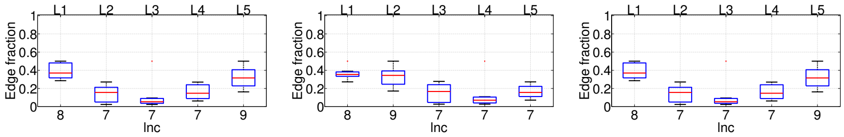
	AUCS	DBLP	EU-Air	FF-TW-YT	Higgs-Twitter	London	VC-Graders
NMI03 [29]	0.752 ± 0.063	0.510 ± 0.003	0.853 ± 0.019	0.418 ± 0.040	0.305 ± 0.133	0.779 ± 0.013	0.707 ± 0.011
NMI04 [7]	0.741 ± 0.059	0.481 ± 0.003	0.847 ± 0.019	0.358 ± 0.050	0.259 ± 0.123	0.766 ± 0.024	0.667 ± 0.014

obtained very similar values of NMI, a single series, which corresponds to M-EMCD, is reported for each of the NMI measures. We first observe that the two NMI measures behave similarly, possibly by a scaling factor, on most θ regimes. One general remark relevant for the comparison between M-EMCD and the baseline is that the highest NMI values do not necessarily correspond to the θ value by which the best-modularity consensus was obtained (indicated with a colored vertical line in each of the plots): in fact, while on AUCS and London the maximum NMI (about 0.8 and 0.75, respectively) is reached for their respective best-modularity θ, on Higgs-Twitter and FF-TW-YT (along with the remaining datasets, not shown), the best-performing θ does not match the θ corresponding to the best NMI for the particular dataset. In general, this result indicates that the community membership in the solution by Nerstrand on the flattened graph can be quite different from that in the modularity-based optimal structure of consensus obtained by M-EMCD. Note also that the average NMI values over θ are usually in mid-low ranges, which means that the similarity between M-EMCD and Nerstrand-flattened community structures is moderately low. Overall, taking into account the joint contribution of the layers for the modularity optimization in the consensus solution differentiates from a community structure solution on the flattened graph where the relative contribution of each of the layers is discarded.

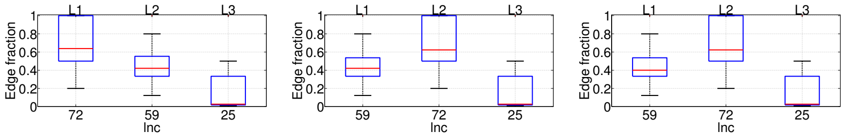
The above remarks on the community membership alignment between the solutions of the two methods complement with results shown in Table 5. This reports on NMI values between the best-modularity consensus by M-EMCD and the community structure that Nerstrand obtained on each layer graph; finally, the NMI values were averaged over the various layers. In the table, we observe indeed that NMI (mean) values range from about 0.3 on Higgs-Twitter to 0.85 on EU-Air, with averages over all datasets of 0.62 NMI03 and 0.59 NMI04. This indicates that the community membership of nodes in the consensus keeps a moderate similarity with the community memberships over each layer on average.

6.1.3 Layer coverage

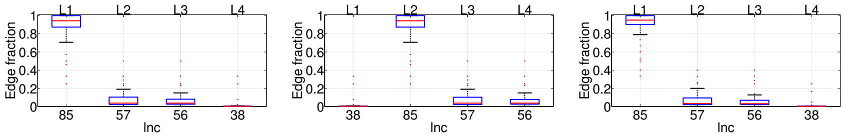
Table 4 summarizes the global redundancy associated with the best consensus solution obtained by M-EMCD. Two main remarks stand out here. First, M-EMCD is able



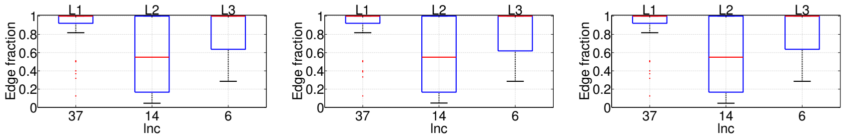
(a) M-EMCD, AUCS (b) CC-EMCD, AUCS (c) C-EMCD, AUCS



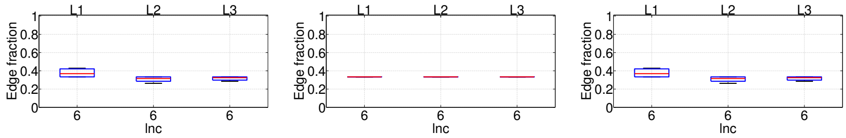
(d) M-EMCD, on FF-TW-YT (e) CC-EMCD, on FF-TW-YT (f) C-EMCD, on FF-TW-YT



(g) M-EMCD, Higgs-Tw. (h) CC-EMCD, Higgs-Tw. (i) C-EMCD, Higgs-Tw.



(j) M-EMCD, London (k) CC-EMCD, London (l) C-EMCD, London



(m) M-EMCD, VC-Graders (n) CC-EMCD, VC-Graders (o) C-EMCD, VC-Graders

Fig. 9 Per-layer distribution of edges over the consensus communities obtained by EMCD methods. Each EMCD solution is taken at the θ value for which M-EMCD reaches the maximum modularity. The bottom x-axis indicates, for each layer, the number of communities which contain only edges from that layer. *(Best viewed in color)*

to produce consensus communities whose internal connectivity is, on average, characterized by most of the layers. Second, M-EMCD has also the same ability in terms of redundancy as C-EMCD, whose solution indeed represents the topological upper bound, for a given θ, of the communities being identified.

To deepen our understanding on the impact of the different layers on the structure of the consensus communities, we also analyzed the per-layer distributions of the fraction of edges specific of any particular layer, over the consensus communities, as shown in Fig. 9. In the Nerstrand case, for each dataset the algorithm is applied on the flattened graph, then information on the community membership is projected over the multilayer network, finally the redundancy distribution is computed over the multilayer-projected communities. The figure also shows, for each layer in a network, the number of communities that only contain edges from that layer. At a first glance, we observe that the per-layer boxplots for M-EMCD are quite similar to those for C-EMCD. This result is indeed consistent with what we observed in the redundancy evaluation. Furthermore, coupling redundancy results from Table 4 and results shown in this figure, it should be noted that the highest values of redundancy of M-EMCD, observed in AUCS (0.91) and VC-Graders (0.95), correspond to situations in which the distribution of layer-characteristic communities is more uniform. However, unlike redundancy, evaluating the per-layer edge distribution allows us to know more about the role taken by each layer in the composition of the consensus communities. For instance, on Higgs-Twitter (results not shown), there is one layer predominant on the others; Conversely, on DBLP (results not shown), all layers participated almost equally in the edge distribution of the consensus communities. Yet, on London, the mid value of redundancy (0.533) should be reconsidered as actually all three layers participate well in the composition of the communities (the first and third layers are highly characteristic for all communities, and the second one corresponds to a distribution with median of 0.6; cf. Fig. 9-j).

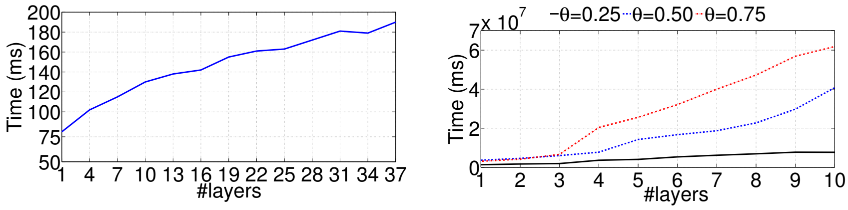
6.1.4 Robustness against ensemble perturbations

Our methods are parametric to a single parameter, θ, for any input multilayer network and ensemble for it. Here we investigate how robust the M-EMCD method is against perturbations in the ensemble used as input.

To this purpose, while maintaining Nerstrand as core method to generate the communities in the ensemble, we configured it by specifying the number of desired communities as input parameter, rather than leaving Nerstrand free to automatically determine the number of communities. For a given dataset network, we generated multiple (e.g., 50) ensembles, by varying each time the setting of the number of communities to obtain on each layer of the network. More in detail, if we indicate with $k_1, \ldots, k_\ell$ the number of communities Nerstrand would automatically detect, we selected the number of communities to obtain at the i-th layer graph ($i = 1..\ell$) by picking it in the interval $[k_i - \epsilon, k_i + \epsilon]$ uniformly at random, where ϵ is an offset selected empirically.

For this analysis, here we report results on EU-Air. We selected this dataset to appreciate at best the effect of ensemble perturbations in the performance of M-EMCD—this choice is justified since it has much more layers than the other datasets but DBLP, however unlike the latter, there is no excessive proliferation in the number of consensus communities (cf. Table 3). We carried out 50 runs and the analyzed the distribution of performance scores corresponding to the 50 ensembles. We perturbed the size of each layer in the ensemble at 5% of the size of the consensus solution obtained by M-EMCD (with the default configuration of Nerstrand), i.e., we set $\epsilon = 0.05 \times |\mathcal{C}^*| \approx 15$.

Results revealed a good robustness of M-EMCD to variations in the size of the ensemble clusterings available for a given dataset network. The resulting boxplots of the distributions of modularity, silhouette, redundancy and average per-ensemble number of consensus communities were all very short. In particular, the size of the consensus solutions obtained by M-EMCD varied from 359 to 365 (with mean 362) over the 50 runs; more interestingly, modularity further increased w.r.t. the performance reported in Table 3, with the following summary: 0.942 (min), 0.962 (mean), 0.964 (median), 0.963 (1st quartile), 0.965 (3rd quartile), 0.968 (max), with standard deviation of just 0.0048.



(a) (b)

Fig. 10 Time performance of M-EMCD on (a) EU-Air and (b) mLFR-1M.

6.1.5 Efficiency evaluation

We analyzed the time performance of M-EMCD,[7] mainly to investigate how well the method scales over networks as they increase in size.

To this purpose, we focused our evaluation on two networks: EU-Air (already selected for the previously reported analysis on robustness) and mLFR-1M (cf. Sect. 5.1). For each of the two network datasets, we ordered the layer graphs by increasing size, then we derived several subsets by grouping the layer graphs according to their size order. More specifically, the first subset contained the smallest layer graph, and the n-th subset ($n > 1$) contained the portion of the original network that corresponds to the first n smallest layer graphs. For every subset considered, the ensemble corresponded to the community structures of the layer graphs belonging to the subset.

Figure 10 shows execution times obtained by M-EMCD in the two evaluation scenarios. In the EU-Air case, we reported for each subset the execution time corresponding to the best-modularity θ setting, whereas in the mLFR-1M scenario, we reported execution times for three selected settings of θ, keeping one value of θ at a time fixed over the various subsets. From the figure, we observe clear evidence in both scenarios that the time performance trend grows linearly with the size (in terms of layers, hence edge set) of the network under consideration. Therefore, our M-EMCD method scales well by increasing the size of the network. Note also that in Fig. 10(b) the slope of the trend line tends to increase with θ, which might imply an increase in the number of consensus communities.

All the above remarks are consistent with our findings from the computational complexity analysis discussed in Sect. 4.5. To complete our understanding on this, it should also be noted that the number of iterations, required by M-EMCD to converge, turns out to be small. For instance, considering the best-performing runs of M-EMCD, the number of iterations varied from few units (less than ten on London, VC-Graders and AUCS) to few tens (23 on EU-Air, 38 on FF-TW-YT, 70 on DBLP, 75 on Higgs-Twitter).

6.2 Comparison with competing methods

In this section we present performance results obtained by the competing methods, and compare them w.r.t. M-EMCD (Table 6).

Looking at modularity results, M-EMCD outperformed all competing methods, with the following gains averaged over the datasets: 0.63 vs. LART, 0.60 vs. PMM$^{k^*}$, 0.36 vs. Infomap, 0.32 vs. GL, 0.30 vs PMM, 0.27 vs. MultiMOGA, 0.23 vs. Nerstrand, 0.17

[7] Experiments were run on an Intel Core i7-3960X CPU @3.30GHz, 64GB RAM machine.

Table 6 Gains of M-EMCD w.r.t. the competing methods, in terms of modularity, silhouette, global redundancy, and number of communities corresponding to best-modularity consensus solution. *na* means that the competing method did not terminate due to memory issues.

method	criterion	AUCS	DBLP	EU-Air	FF-TW-YT	Higgs-Tw.	London	VC-Graders
Nerstrand	modularity	+0.34	+0.17	+0.62	+0.24	+0.02	+0.07	+0.17
	silhouette	+0.15	+0.001	+0.01	+0.02	-0.02	+0.11	+0.01
	redundancy	+0.11	-0.12	+0.29	-0.09	-0.36	+0.17	+0.02
	#communities	+9	+13 466	+268	+43	+63	+29	+9
ABACUS	modularity	+0.10	*na*	+0.02	+0.16	+0.32	+0.10	-0.30
	silhouette	+0.38	*na*	+0.12	+0.04	+0.20	+0.24	-0.71
	redundancy	+0.20	*na*	+0.27	+0.13	+0.95	+0.39	+0.12
	#communities	+12	*na*	+250	+84	+36	+29	+10
PMM$^{k^*}$	modularity	+0.67	*na*	+0.89	+0.52	+0.60	+0.69	+0.24
	silhouette	+0.22	*na*	+0.23	+0.05	-0.02	+0.11	+0.13
	redundancy	-0.003	*na*	-0.07	+0.04	-0.36	-0.18	+0.003
PMM	modularity	+0.15	*na*	+0.54	+0.39	+0.26	+0.27	+0.16
	silhouette	+0.37	*na*	+0.25	+0.69	+0.20	+0.43	+0.24
	redundancy	+0.14	*na*	+0.06	+0.10	+0.79	-0.19	+0.06
	#communities	+12	*na*	+269	+76	+76	+5	+9
GL	modularity	+0.21	*na*	+0.46	+0.20	*na*	+0.62	+0.13
	silhouette	+0.17	*na*	+0.04	+0.11	*na*	+0.14	+0.10
	redundancy	+0.11	*na*	+0.32	-0.12	*na*	-0.03	+0.07
	#communities	+8	*na*	+262	-626	*na*	-212	+9
Infomap	modularity	+0.50	*na*	+0.30	+0.29	*na*	+0.45	+0.26
	silhouette	+0.53	*na*	+0.20	+0.88	*na*	+0.33	+0.45
	redundancy	+0.15	*na*	+0.06	-0.33	*na*	-0.48	+0.00
	#communities	+3	*na*	+272	-117	*na*	+43	+1
LART	modularity	+0.58	*na*	+0.91	*na*	*na*	+0.89	+0.12
	silhouette	+0.50	*na*	+0.14	*na*	*na*	+0.18	+0.32
	redundancy	+0.13	*na*	+0.37	*na*	*na*	+0.53	+0.06
	#communities	-13	*na*	-107	*na*	*na*	-294	+5
MultiGA	modularity	+0.17	*na*	+0.25	*na*	*na*	+0.10	+0.16
	silhouette	+0.37	*na*	+0.06	*na*	*na*	+0.23	+0.24
	redundancy	+0.10	*na*	+0.34	*na*	*na*	-0.07	+0.06
	#communities	+9	*na*	+269	*na*	*na*	+16	+9
MultiMOGA	modularity	+0.29	*na*	+0.27	+0.40	*na*	+0.39	+0.00
	silhouette	+0.34	*na*	+0.14	+0.74	*na*	+0.21	+0.43
	redundancy	+0.08	*na*	+0.35	-0.03	*na*	+0.04	+0.01
	#communities	+7	*na*	+269	-129	*na*	+32	+7

vs. MultiGA, and 0.07 vs. ABACUS. This remarkably hints that our approach is able to produce multilayer communities that are substantially better in modularity than those obtained by existing flattening, aggregation, or direct methods.

Also in terms of silhouette, M-EMCD tends to outperform all competing methods, with the following average gains: 0.48 vs. Multiplex-Infomap, 0.37 vs. MultiMOGA, 0.36 vs. PMM, 0.29 vs. LART, 0.23 vs. MultiGA, 0.12 vs. PMM$^{k^*}$, 0.11 vs. GL, 0.05 vs. ABACUS, and 0.04 vs. Nerstrand. Note that the least gains by M-EMCD are those against Nerstrand, which is not surprising as M-EMCD consensus solutions are derived by an ensemble of community structures obtained by using Nerstrand on each of the layers in a network. Overall, the consensus solutions by M-EMCD show better quality of community memberships of the nodes in a network.

Considering global redundancy values, M-EMCD generally shows higher values than those of competitors over the various networks, with average gains of 0.34 vs. ABACUS, 0.27 vs. LART, 0.16 vs. PMM, 0.11 vs. MultiGA, 0.09 vs. MultiMOGA, 0.07 vs. GL, and 0.003 vs. Nerstrand. While consistently yielding higher global redundancy w.r.t. ABACUS and LART, M-EMCD consensus communities may have lower redundancy than communities produced by the other methods (e.g., Multiplex-Infomap and PMM$^{k^*}$). Nevertheless, coupled with modularity and silhouette results, this suggests that M-EMCD can utilize less information from the various layers than other methods to obtain higher quality consensus community structures.

We also observe that M-EMCD tends to produce much more communities than Nerstrand, ABACUS, PMM, MultiGA and MultiMOGA, while different relative behaviors correspond to comparison with the other methods on some networks.

On the efficiency viewpoint, one remark that stands out from the table is that all methods but Nerstrand incurred memory issues on some datasets. (Experimental platform corresponded to the same as specified in Sect. 6.1.5.) In this regard, it should be noted that some of our competitors methods inherently suffer from efficiency and scalability issues. For instance, the two genetic methods MultiGA and MultiMOGA have high computational complexity, which not only depends on the (high) numbers of generations and of individuals, but it is also at least quadratic in the number of nodes in the networks. Also, LART requires the computation of similarity matrix from the pair-wise transition probabilities, and hence could not scale well with large multilayer networks.

In general, by comparing the runtimes obtained by the competing methods with those obtained by M-EMCD, we found that M-EMCD outperforms the competing methods in terms of efficiency as well. For instance, on EU-Air, our method took about 0.190 seconds to produce the consensus solution (cf. Fig. 10(a)), while the following runtimes were achieved by the competing methods (in seconds): 2 by PMM, 23 by GL, 20 by Multiplex-Infomap, 475 by LART, 1026 by MultiGA, 12375 by MultiMOGA. Overall, the following orders of magnitude of percentage increase, averaged over the evaluation datasets, were obtained by the competing methods: about 1000% by PMM, 1.0E+4 % by Multiplex-Infomap, GL, and ABACUS, 1.0E+5 % by LART, 5.0E+5 % by MultiGA, and 5.0E+6 % by MultiMOGA.

6.3 Summary of findings

Several remarks stand out from our evaluation of EMCD methods in terms of different assessment criteria. A first important finding is that the modularity-based approach to the EMCD problem is highly effective in producing consensus communities with improved modularity w.r.t. the CC-EMCD and C-EMCD methods. M-EMCD also outperforms CC-EMCD and C-EMCD in terms of silhouette of community membership, whereby in most cases the highest gain occurs for the same θ range corresponding to the highest modularity value. Internal connectivity of the M-EMCD consensus communities is characterized by the presence of most of the layers, in general retaining the same ability in terms of redundancy as C-EMCD, whose consensus represents the topological upper bound for the communities being identified, for a given θ. M-EMCD has shown to be relatively robust to the presence of disconnected components in a multilayer graph, as its solutions tend to have a small number of singleton communities. Our method has also shown to be relatively robust against perturbations in the input ensemble, in terms of size of its constituting clusterings. Yet, M-EMCD scales well with the size of a multilayer network, in accordance to its computational cost that is linear in the number of edges.

M-EMCD was compared with state-of-the-art methods for community detection, which cover all of the main categories of existing approaches for multilayer networks, namely flattening, aggregation and direct methods. In our evaluation, the first category was represented by Nerstrand on the flattened graph, the second by PMM and ABACUS, and the third by GL, LART, Multiplex-Infomap, MultiGA, and MultiMOGA. Remarkably, M-EMCD consensus communities have shown to be substantially better than those generated by the competing methods, in terms of both modularity and silhouette of community membership. This outperforming behavior of M-EMCD is further strengthened by the fact the method tends to use less information from the layers of the network than the competing methods, while producing better consensus community structures.

7 Conclusion

We presented the first well-principled formulation of the consensus-based aggregation problem for multilayer community detection. The main innovation aspect of our framework is that, unlike the few existing aggregation-based community detection approaches, both the aggregation and consensus inference steps are not naive: our discovered consensus community structures are not only designed to capture prototypical community memberships of nodes, but they also account for intra-community and inter-community connectivity, whereas the consensus function is learned via modularity-based optimization instead of being simply based on the sharing of a certain minimum percentage of clusters in the ensemble. Results have shown significance as well as outperforming behavior of our approach over state-of-the-art methods in terms of modularity, silhouette of community memberships, and redundancy assessment criteria.

Limitations and Future Work

Our ensemble-based approach to multilayer community detection has one main model parameter, i.e., the co-association threshold (θ). This is typical in the ensemble clustering paradigm and allows the user to control the level of consensus in the output community structure. However, since the parameter is data-dependent, choosing a fixed value is not recommended for networks with different characteristics. In this regard, it would be worth investigating a way for automatically determine the "best" or "preferred" co-association setting for an input multilayer network.

Concerning the ensemble, while this is regarded as an input data, in this study we have considered only the scenario of non-overlapping community detection. It is certainly of interest to extend our approach to handle overlapping consensus solutions as well.

Our defined multilayer modularity measure involves two important terms, i.e., the layer-specific resolution factors and the inter-layer coupling factors. We plan to empirically evaluate the impact of the two terms on the modularity optimization of the consensus. In this respect, one major direction of future research is the evaluation of the EMCD problem in time-evolving network scenarios; in particular, the implications of enabling the time-aware inter-layer coupling factor in the computation of our defined multilayer modularity, and hence in the discovery of the consensus communities, deserve attention.

It will be furthermore interesting to push forward our research to address ensemble-based community evolution problems under a dynamic or online learning framework, by leveraging recent literature on online consensus clustering (e.g., [32, 15, 14]).

On another side, our modularity-optimization-based approach would allow for an integration with a direct method for multilayer community detection; in effect, our approach could be seen as a refinement for any direct approach, since M-EMCD can in principle take as input a multilayer community structure, possibly provided by one of the existing direct methods. Within this view, our approach might be referred to as a "hybrid" approach to solving the multilayer community detection problem. This would further pave the way for the development of novel community detection frameworks for multilayer networks.

References

1. A. Amelio and C. Pizzuti. A Cooperative Evolutionary Approach to Learn Communities in Multilayer Networks. In *Proc. PSSN*, pages 222–232, 2014.
2. A. Amelio and C. Pizzuti. Community detection in multidimensional networks. In *Proc. ICTAI*, pages 352–359, 2014.
3. M. Berlingerio, M. Coscia, and F. Giannotti. Finding and characterizing communities in multidimensional networks. In *Proc. ASONAM*, pages 490–494, 2011.
4. M. Berlingerio, F. Pinelli, and F. Calabrese. ABACUS: frequent pattern mining-based community discovery in multidimensional networks. *Data Min. Knowl. Discov.*, 27(3):294–320, 2013.
5. M. Burgess, E. Adar, and M. Cafarella. Link-Prediction Enhanced Consensus Clustering for Complex Networks. *PLoS ONE*, 11(5):e0153384, 2016.
6. M. De Domenico, A. Lancichinetti, A. Arenas, and M. Rosvall. Identifying Modular Flows on Multilayer Networks Reveals Highly Overlapping Organization in Interconnected Systems. *Phys. Rev. X*, 5, 011027, 2015.
7. I. S. Dhillon, Y. Guan, and B. Kulis. Kernel k-means: spectral clustering and normalized cuts. In *Proc. ACM KDD*, pages 551–556, 2004.
8. M. E. Dickison, M. Magnani, and L. Rossi. *Multilayer social networks.* 2016.
9. X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov. Clustering With Multi-Layer Graphs: A Spectral Perspective. *IEEE Trans. Signal Process.*, 60(11):5820–5831, 2012.
10. S. Fortunato and M. Barthelemy. Resolution limit in community detection. *Proc. Natl. Acad. Sci.*, 104(36), 2007.
11. A. Fred. Finding consistent clusters in data partitions. In *Proc. Work. on Multiple Classifier Systems*, pages 309–318, 2001.
12. F. Gullo, A. Tagarelli, and S. Greco. Diversity-based weighting schemes for clustering ensembles. In *Proc. SDM*, pages 437–448, 2009.
13. M. Hmimida and R. Kanawati. Community detection in multiplex networks: A seed-centric approach. *Networks and Heterogeneous Media*, 10(1):71–85, 2015.
14. D. Katselis, C. L. Beck, and M. van der Schaar. Ensemble Online Clustering through Decentralized Observations. In *Proc. IEEE Conf. on Decision and Control*, pages 910–915, 2014.
15. I. Khan, J. Z. Huang, and K. Ivanov. Incremental density-based ensemble clustering over evolving data streams. *Neurocomputing*, 191:34–43, 2016.
16. J. Kim and J.-G. Lee. Community detection in multi-layer graphs: A survey. *SIGMOD Record*, 44(3):37–48, 2015.
17. M. Kivela, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, 2014.
18. Z. Kuncheva and G. Montana. Community detection in multiplex networks using locally adaptive random walks. In *Proc. ASONAM*, pages 1308–1315, 2015.
19. A. Lancichinetti and S. Fortunato. Consensus clustering in complex networks. *Sci. Rep.*, 2:336, 2012.
20. A. Lancichinetti, S. Fortunato, and F. Radicchi Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78(4): 046110, 2008.
21. D. LaSalle and G. Karypis. Multi-threaded modularity based graph clustering using the multilevel paradigm. *J. Parallel Distrib. Comput.*, 76:66–80, 2015.
22. T. Li, C. Ding, and M. I. Jordan. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In *Proc. ICDM*, pages 577–582, Oct 2007.
23. P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela. Community structure in time-dependent, multiscale, and multiplex networks. *Science*, 328(5980):876–878, 2010.
24. M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69(2):026113, 2004.
25. N. Nguyen and R. Caruana. Consensus clusterings. In *Proc. ICDM*, pages 607–612, 2007.
26. M. Rocklin and A. Pinar. On clustering on graphs with multiple edge types. *Internet Mathematics*, 9(1):82–112, 2013.
27. M. Rosvall and C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci. USA*, 105:1118–1123, 2008.
28. P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

29. A. Strehl and J. Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3:583–617, 2003.
30. L. Tang, X. Wang, and H. Liu. Uncovering groups via heterogeneous interaction analysis. In *Proc. ICDM*, pages 503–512, 2009.
31. L. Tang, X. Wang, and H. Liu. Community detection via heterogeneous interaction analysis. *Data Min. Knowl. Discov.*, 25:1–33, 2012.
32. Z. Yu, P. Luo, J. You, H.-S. Wong, H. Leung, S. Wu, J. Zhang, and G. Han. Incremental Semi-Supervised Clustering Ensemble for High Dimensional Data Clustering. *IEEE Trans. Knowl. Data Eng.*, 28(3):701–714, 2016.
33. H. Zhang, C.-D. Wang, J.-H. Lai, and P. S. Yu. Modularity in Complex Multilayer Networks with Multiple Aspects: A Static Perspective. *CoRR*, abs/1605.06190, 2016.