

Consensus Community Detection in Multilayer Networks using Parameter-free Graph Pruning

Domenico Mandaglio, Alessia Amelio, and Andrea Tagarelli

DIMES - University of Calabria, 87036 Rende (CS), Italy
{d.mandaglio,a.amelio,tagarelli}@dimes.unical.it

Abstract. The clustering ensemble paradigm has emerged as an effective tool for community detection in multilayer networks, which allows for producing consensus solutions that are designed to be more robust to the algorithmic selection and configuration bias. However, one limitation is related to the dependency on a co-association threshold that controls the degree of consensus in the community structure solution. The goal of this work is to overcome this limitation with a new framework of ensemble-based multilayer community detection, which features parameter-free identification of consensus communities based on generative models of graph pruning that are able to filter out noisy co-associations. We also present an enhanced version of the modularity-driven ensemble-based multilayer community detection method, in which community memberships of nodes are reconsidered to optimize the multilayer modularity of the consensus solution. Experimental evidence on real-world networks confirms the beneficial effect of using model-based filtering methods and also shows the superiority of the proposed method on state-of-the-art multilayer community detection.

1 Introduction

Multilayer networks are pervasive in many fields related to network analysis and mining [8, 2]. Particularly, *community detection in multilayer networks* (ML-CD) has attracted lot of attention in the past few years, as witnessed by a relatively large corpus of studies (see, e.g., [7] for a survey).

An effective approach to ML-CD corresponds to *aggregation methods*, whose goal is to infer a community structure by combining information from community structures separately obtained on each of the layers [17, 18, 16]. A special class of such methods resembles theory on *clustering ensemble* [15, 6]: given a set of clusterings as different groupings of the input data, a *consensus* criterion function is optimized to induce a single, meaningful solution that is representative of the input clusterings. A key advantage of using a consensus clustering approach is that the inconvenience of guessing the “best” algorithm selection and parametrization is avoided, and hence consensus results will be more robust and show higher quality when compared to single-algorithm clustering.

Despite the well-recognized benefits of using the consensus/ensemble clustering paradigm, its exploitation to ML-CD is, surprisingly, relatively new in

the literature [18, 9, 16]; actually, to the best of our knowledge, only the most recent of these works goes beyond the use of a clustering ensemble approach as a black-box tool for ML-CD, by proposing the first well-principled formulation of the *ensemble-based community detection* (EMCD) problem. Indeed, in [16], aggregation is not limited at node membership level, but it also accounts for intra-community and inter-community connectivity; moreover, the consensus function is optimized via *multilayer modularity* analysis, instead of being simply based on the sharing of a certain minimum percentage of clusters in the ensemble.

The EMCD method proposed in [16] relies on a *co-association-based consensus clustering scheme*, i.e., the consensus clusters are derived from a co-association matrix built to store the fraction of clusterings in which any two nodes are assigned to the same cluster. Low values in this matrix would reflect unlikely consensus memberships, i.e., noise, and hence should be removed; to this purpose, the matrix is subjected to a filtering step based on a user-specified parameter of minimum co-association, θ . Unfortunately, setting an appropriate θ for a given input network is a challenging task, since too low values will lead to few, large communities, while too high values will lead to many, small communities. Moreover, this approach generally fails to consider properties related to node distributions and linkage in the network.

In this work, we aim to overcome the above issue, by proposing a new EMCD framework featuring a *parameter-free identification of consensus clusters* from which the *consensus community structure* will be induced. Our idea is to exploit a recently developed class of *graph-pruning methods based on generative models*, which are designed to filter out “noisy” edges from weighted graphs. A key advantage of these pruning models is that they do not require any user-specified parameter, since they enable edge-removal decisions by computing a statistical p -value for each edge based on a null model defined on the node degree and strength distributions. We originally introduce these models to multilayer community detection and propose an adaptation to multilayer networks.

Another limitation of EMCD is that the community membership of nodes remains the same through the process of detecting the modularity-driven consensus community structure. In this work, we also address this point, by defining a three-stage process in the EMCD scheme, which iteratively seeks to improve the multilayer modularity of the consensus community structure based on intra-community connectivity refinement, community partitioning, and relocation of nodes from a community to a neighboring one.

Two main findings are drawn from experimental results obtained on real-world multiplex networks: (i) some of the model-filters are effective in simplifying an input multilayer network to support improved community detection, and (ii) our proposed framework outperforms state-of-the-art multilayer community detection methods according to modularity and silhouette quality criteria.

In the rest of the paper, we provide background on generative-model-based filters and on the existing EMCD method (Section 2). Next, we present our proposed framework (Section 3). Experimental evaluation and results are discussed in Sections 4 and 5. Section 6 concludes the paper.

2 Background

2.1 Generative models for graph pruning

Pruning is a graph simplification task aimed at detecting and removing irrelevant or spurious edges in order to unveil some hidden property/structure of the network, such as its organization into communities. A simple technique adopted in weighted graphs consists in removing all edges having weight below a pre-determined, global threshold. Besides the difficulty of choosing a proper threshold for the input data, this approach tends to remove all ties that are weak at network level, thus discarding local properties at node level.

A relatively recent corpus of study addresses the task of filtering out “noisy” edges from complex networks based on *generative null models*. The general idea is to define a null model based on node distribution properties, use it to compute a *p*-value for every edge (i.e., to determine the statistical significance of properties assigned to edges from a given distribution), and finally filter out all edges having *p*-value above a chosen significance level, i.e., keep all edges that are least likely to have occurred due to random chance.

Methods following the above general approach have been mainly conceived to deal with weighted networks, so that the node degree and/or the node strength (i.e., the sum of the weights of all incident edges) are used to generate a model that defines a random set of graphs resembling the observed network. One of the earliest methods is the *disparity* filter [14], which evaluates the strength and degree of each node locally. This filter however introduces some bias in that the strength of neighbors of a node are discarded. By contrast, a global null model is defined with the *GloSS* filter [13], as it preserves the whole distribution of edge weights. The null model is, in fact, a graph with the same topological structure of the original network and with edge weights randomly drawn from the empirical weight distribution. Unlike disparity and GloSS, the null model proposed by Dianati [1] is maximum-entropy based and hence unbiased. Upon it, two filters are defined: the *marginal likelihood filter (MLF)*, which is a linear-cost method that assigns a significance score to each edge based on the marginal distribution of edge weights, and the *global likelihood filter*, which accounts for the correlations among edges. While performing similarly, the latter filter is more costly than MLF; moreover, both consider the strength of nodes, but not their degrees. Recently, Gemmetto et al. [5] proposed a maximum-entropy filter, *ECM*, for keeping only *irreducible edges*, i.e., the filtered network will retain only the edges that cannot be inferred from local information. ECM employs a null model based on the canonical maximum-entropy ensemble of weighted networks having the same degree and strength distribution as the real network [11]. Due to space limits, we report details of the MLF, GloSS and ECM filters in the *Online Appendix* available at <http://people.dimes.unical.it/andreatagarelli/emcd/>.

2.2 Ensemble-based Multilayer Community Detection

Let $G_{\mathcal{L}} = (V_{\mathcal{L}}, E_{\mathcal{L}}, \mathcal{V}, \mathcal{L})$ be a *multilayer network* graph, with set of layers $\mathcal{L} = \{L_1, \dots, L_{\ell}\}$ and set of entities \mathcal{V} . Each layer corresponds to a given type of

entity relation, or edge-label. For each pair of entity in \mathcal{V} and layer in \mathcal{L} , let $V_{\mathcal{L}} \subseteq \mathcal{V} \times \mathcal{L}$ be the set of entity-layer pairs representing that an entity is located in a layer. The set $E_{\mathcal{L}} \subseteq V_{\mathcal{L}} \times V_{\mathcal{L}}$ contains the undirected links between such entity-layer pairs. For every layer $L_i \in \mathcal{L}$, V_i and E_i denote the set of nodes and edges, respectively. Also, the inter-layer edges connect nodes representing the same entity across different layers (monoplex assumption).

Given a multilayer network $G_{\mathcal{L}}$, an *ensemble of community structures* for $G_{\mathcal{L}}$ is a set $\mathcal{E} = \{\mathcal{C}_1, \dots, \mathcal{C}_{\ell}\}$, such that each \mathcal{C}_h (with $h = 1..{\ell}$) is a community structure of the layer graph G_h . This ensemble could be obtained by applying any non-overlapping community detection algorithm to each layer graph.

Given an ensemble of community structures for a multilayer network, the problem of ensemble-based multilayer community detection (EMCD) 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. In order to determine the community membership of nodes in the consensus structure, a *co-association*-based scheme is defined over the layers, to detect a clustering solution (i.e., the consensus) that conforms most to the input clusterings. Given $G_{\mathcal{L}}$, and \mathcal{E} for $G_{\mathcal{L}}$, the *co-association matrix* \mathbf{M} is a matrix with size $|\mathcal{V}| \times |\mathcal{V}|$, whose (i, j) -th entry is defined as $|m_{ij}|/\ell$, where m_{ij} is the set of communities shared by $v_i, v_j \in \mathcal{V}$, under the constraint that the two nodes are linked to each other [16].

EMCD is modeled in [16] as an optimization problem in which the *consensus community structure* solution is optimal in terms of *multilayer modularity*, and is to be discovered within a hypothetical space of consensus community structures that is delimited by a “topological-lower-bound” solution and by a “topological-upper-bound” solution, for a given *co-association threshold* θ . Intuitively, 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 contain superfluous multilayer edges connecting different communities. The modularity-optimization-driven consensus community structure produced by the method in [16], dubbed M-EMCD, hence produces a solution that is ensured to have higher modularity than both the topologically-bounded solutions.

3 EMCD and parameter-free graph pruning

As previously discussed, the EMCD framework has one model parameter, i.e., the co-association threshold θ , which allows the user to control the degree of consensus required to every pair of nodes in order to appear in the same consensus community. Given a selected value for θ and any two nodes v_i, v_j , we say that their community linkage, expressed by $M(v_i, v_j)$, is considered as meaningful to put the nodes in the same consensus community iff $M(v_i, v_j) \geq \theta$.

However, choosing a fixed value of θ equally valid for all pairs of nodes raises a number of issues. First, there is an intrinsic difficulty of guessing the “best” threshold — since too low values will lead to few, large communities, while too

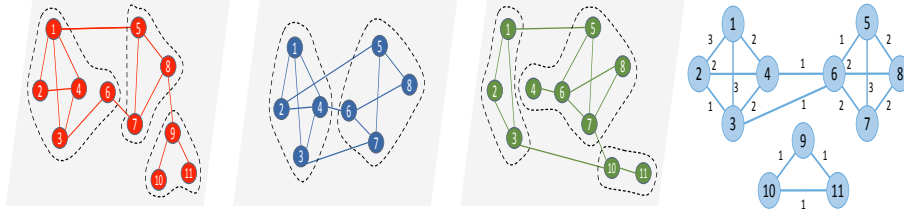


Fig. 1. Community structures (denoted by dotted curves) on a 3-layer network, and corresponding co-association graph.

high values will lead to many, small communities. Second, the approach ignores any property of the input network, and consequently a single-shot choice of θ may fail to capture the natural structure of communities. Of course, to overcome the two issues in practical cases, one could always try different choices of the parameter and finally select the best-performing one (e.g., in terms of modularity, as done in [16]), but it is clear that the approach does not scale for large networks.

It would instead be desirable to evaluate the significance of the co-associations by taking into account the topology of the multilayer network, so that a relatively low value of co-association might be retained as meaningful provided that it refers to node relations that make sense only for certain layers, while on the contrary, a relatively high value of co-association could be discarded if it corresponds to the linkage of nodes that have high degree and co-occur in the same community in many layers — in which case, the co-association could be considered as superfluous in terms of community structure.

In order to fulfill the above requirement, we define a *parameter-free approach to EMCD* that exploits the previously discussed pruning models. Since such models are only designed to work with (monoplex) weighted graphs, our key idea is to first infer a *weighted graph representation of the co-association matrix* associated to a multilayer network and its ensemble of community structures, and then apply a pruning model on it to retain only meaningful co-associations.

Definition 1 (Co-association graph). *Given a multilayer graph $G_{\mathcal{L}}$, an ensemble \mathcal{E} of community structures defined over it, and associated co-association matrix \mathbf{M} , we define the co-association graph $G_M = \langle V_M, E_M, w \rangle$ as an undirected weighted graph such that $V_M = \mathcal{V}$, $E_M = \{(v_i, v_j) \mid m_{ij} \neq \emptyset, w_{ij} = |m_{ij}|\}$.*

Below is an example of how the pruning of the co-association graph based on a user-specified threshold could lead to poorly meaningful consensus communities.

Example 1. Consider the 3-layer network and associated co-association graph in Figure 1. Focusing on the community membership of nodes, consider the following settings of a cutting threshold θ . For any $\theta \leq 1/3$, all edges will be kept (as the minimum valid weight is 1) and hence the co-association graph will be partitioned into the two communities corresponding to its two connected components, i.e., $\{1, \dots, 8\}$ and $\{9, 10, 11\}$; setting $1/3 < \theta \leq 2/3$ will lead to

Algorithm 1 Co-association matrix filtering

Input: Multilayer graph $G_{\mathcal{L}} = (V_{\mathcal{L}}, E_{\mathcal{L}}, \mathcal{V}, \mathcal{L})$, ensemble of community structures $\mathcal{E} = \{\mathcal{C}_1, \dots, \mathcal{C}_{\ell}\}$ (with $\ell = |\mathcal{L}|$), generative model for graph pruning WGP .
Output: Filtered co-association matrix \mathbf{M} for $G_{\mathcal{L}}$ and \mathcal{E} .
1: Let α be a statistical significance level (i.e., $\alpha = 0.05$) *{Co-association matrix initialization}*
2: $\mathbf{M} \leftarrow \text{matrix}(|\mathcal{V}|, |\mathcal{V}|)$
3: **for** $(i, j) \in \mathbf{M}$ **do**
4: $m_{ij} \leftarrow \{h \mid L_h \in \mathcal{L} \wedge \exists C \in \mathcal{C}_h, C_h \in \mathcal{E}, \text{ s.t. } v_i, v_j \text{ in } C \wedge (v_i, v_j) \in E_h\}$
5: $M(i, j) \leftarrow |m_{ij}|/\ell$
6: **end for**
7: $G_M = (V_M, E_M, w) \leftarrow \text{build_coassociation_graph}(G_{\mathcal{L}}, \mathbf{M})$ *{Using Def. 1}*
8: $(e, \gamma_{ij})_{e=(v_i, v_j) \in E_M} \leftarrow \text{compute_pValues}(G_M, WGP)$ *{Using Def. 2}*
9: **for** $(v_i, v_j) \in E_M$ **do**
10: **if** $\gamma_{ij} \geq \alpha$ **then** $M(i, j) \leftarrow 0$ *{Null hypothesis cannot be rejected}*
11: **return** \mathbf{M}

$\{1, \dots, 4\}$, $\{5, \dots, 8\}$, and $\{9\}, \{10\}, \{11\}$; finally, for $2/3 < \theta \leq 1$, the communities will be $\{1, 2, 3\}$, $\{5, 7\}$ and all the other nodes as singletons. It should be noted that no setting of θ can enable the identification of the three “natural” consensus communities, i.e., $\{1, \dots, 4\}$, $\{5, \dots, 8\}$, and $\{9, 10, 11\}$.

Definition 2 (Co-association hypothesis testing). *Given a co-association graph $G_M = (V_M, E_M, w)$, let WGP denote a statistical inference method whose generative null model is parametric w.r.t. node degree and strength distributions in G_M . We define the co-association hypothesis testing as a parametric testing based on WGP , whose null hypothesis for every observed edge is that its weight has been generated by mere chance, given the empirical strength and degree distributions, and the associated p-value is the probability that the null model produces a weight equal to or greater than the observed edge weight. If the p-value is lower than a desired significance level, then the null hypothesis can be rejected, which implies that the co-association of the two observed nodes is considered as statistically meaningful.*

Algorithm 1 shows the general scheme of creation of the co-association matrix, for a given multilayer network and associated ensemble of community structures, and its filtering based on the co-association hypothesis testing.

Enhanced M-EMCD (M-EMCD*). We propose an enhanced version of M-EMCD that has two main advantages w.r.t. the early M-EMCD method in [16]: (1) it incorporates parameter-free pruning of the co-association matrix described in Algorithm 1, and (2) it fixes the inability of the early M-EMCD in reconsidering the community memberships of nodes during the consensus optimization.

Algorithm 2 shows the pseudo-code of our proposed enhanced M-EMCD, dubbed M-EMCD*. Initially, the filtered co-association matrix computed by a selected model-filter WGP is provided as input to CC-EMCD, which computes the initial (i.e., lower-bound) consensus community structure (Line 2) [16]. This is iteratively improved in a three-stage modularity-optimization process: (i) refinement of connectivity internal to a selected community, (ii) refinement of connectivity between the community and its neighbors also involving relocation of nodes, and (iii) partitioning of the community.

Algorithm 2 Enhanced Modularity-driven Ensemble-based Multilayer Community Detection (M-EMCD*)

Input: Multilayer graph $G_{\mathcal{L}} = (V_{\mathcal{L}}, E_{\mathcal{L}}, \mathcal{V}, \mathcal{L})$, ensemble of community structures $\mathcal{E} = \{C_1, \dots, C_{\ell}\}$ (with $\ell = |\mathcal{L}|$), generative model for graph pruning *WGP*.
Output: Consensus community structure C^* for $G_{\mathcal{L}}$.

- 1: $M \leftarrow \text{co-associationMatrixFiltering}(G_{\mathcal{L}}, \mathcal{E}, \text{WGP})$ {Algorithm 1}
- 2: $C_{lb} \leftarrow \text{CC-EMCD}(G_{\mathcal{L}}, M)$ {Compute topological-lower-bound consensus community structure}
- 3: $C^* \leftarrow C_{lb}$
- 4: **repeat**
- 5: **for** $L_i \in \mathcal{L}$ **do**
- 6: $Q \leftarrow Q(C^*)$
 {Refine intra-community connectivity of C_j }
- 7: **for** $C_j \in C^*$ **do**
- 8: $\langle C'_j, Q'_j \rangle \leftarrow \text{update_community}(C^*, C_j, L_i)$
- 9: $j^* \leftarrow \text{argmax } Q'_j$
- 10: **if** $Q'_{j^*} > Q$ **then** $Q \leftarrow Q'_{j^*}$, $C^* \leftarrow C^* \setminus C_j \cup C'_{j^*}$
 {Refine inter-community connectivity between C_{j^*} and each of its neighbors}
- 11: **for** $C_h \in N(C_{j^*})$ **do**
- 12: $\langle C_h^{IC}, Q_h^{IC} \rangle \leftarrow \text{update_community_structure}(C^*, C_{j^*}, C_h, L_i)$
- 13: $\langle C_h^R, Q_h^R \rangle \leftarrow \text{relocate_nodes}(C^*, C_{j^*}, C_h)$
- 14: $\langle C_h, Q_h \rangle \leftarrow \text{argmax}\{Q_h^{IC}, Q_h^R\}$
- 15: $h^* \leftarrow \text{argmax } Q_h$
- 16: **if** $Q_{h^*} > Q$ **then**
- 17: $Q \leftarrow Q_{h^*}$, $C^* \leftarrow C_{h^*}$
- 18: **if** $Q_{h^*} = Q_h^R$ **then** $\langle C_h, Q_h \rangle \leftarrow \text{update_community_structure}(C^*, C_{j^*}, C_{h^*}, L_i)$
- 19: **else** $\langle C_h, Q_h \rangle \leftarrow \text{relocate_nodes}(C^*, C_{j^*}, C_{h^*})$
- 20: **if** $Q_h > Q$ **then** $Q \leftarrow Q_h$, $C^* \leftarrow C_h$
 {Evaluate partitioning of C_{j^*} into smaller communities}
- 21: $\langle C'_s, Q'_s \rangle \leftarrow \text{partition_community}(C^*, C_{j^*})$
- 22: **if** $Q'_s > Q$ **then** $Q \leftarrow Q'_s$, $C^* \leftarrow C^* \setminus C_{j^*} \cup C'_s$
- 23: **end for**
- 24: **until** $Q(C^*)$ cannot be further maximized
- 25: **return** C^*

The within-community connectivity refinement step (Lines 7-10) consists in seeking in the current solution C^* the community C_{j^*} whose internal connectivity modification leads to the best modularity gain. The internal refinement of a community C_j , applied to the layer L_i , is performed by function `update_community` (Line 8) which tries to add as many edges of type L_i as possible between nodes belonging to C_j , i.e., the set of edges in E_i whose end-nodes are both in C_j and are not present in the current solution C^* . The function then returns the modified C_j and the updated modularity.

Once identified the community C_{j^*} at the previous step, the algorithm tries to relocate nodes from C_{j^*} to its neighbor communities $N(C_{j^*})$ and/or to refine its external connectivity with them (Lines 11-20). The inter-community connectivity refinement is carried out by function `update_community_structure` (Line 12) which, for any layer L_i and neighbor communities C_j, C_h , evaluates the resulting modularity of adding and/or removing edges of type L_i in the current consensus C^* between C_j, C_h , compatibly with the set of edges of L_i in the original graph. The relocation of one node at a time from C_{j^*} to a neighbor community C_h is evaluated by `relocate_nodes` (Line 13) until there is no further improvement in modularity. The ordering of node examination is determined by a priority queue

Table 1. Main features of real-world multiplex network datasets used in our evaluation.

	#entities ($ \mathcal{V} $)	#edges	#layers (ℓ)		#entities ($ \mathcal{V} $)	#edges	#layers (ℓ)
AUCS [7]	61	620	5	FF-TW-YT [2]	6 407	74 836	3
EU-Air [7]	417	3 588	37	London [19]	369	441	3
FAO-Trade [4]	214	318 346	364	VC-Graders [19]	29	518	3

that gives more importance to nodes having more edges (of any type) towards C_h than edges linking them to nodes in their current community in \mathcal{C}^* .

The step of partitioning of C_{j^*} into smaller communities is carried out by function `partition_community` (Line 21). While this can in principle refer to the use of any (multilayer) modularity-optimization-based community detection method, we choose here to focus on the membership of nodes, and hence to devise this step in the simplified scenario of flattened representation of the consensus community C_{j^*} , i.e., a weighted monoplex graph with all and only the nodes belonging to C_{j^*} and weights expressing the number of layers on which two nodes are linked in \mathcal{C}^* . Upon this representation, we apply a graph partitioning method based on modularity optimization (cf. Sect. 4) and finally maintain the resulting partitioning only if it led to an improvement in modularity.

4 Evaluation methodology

Datasets. We used six networks for our evaluation (Table 1), which are among the most frequently used in relevant studies in multilayer community detection.

Competing methods. We selected four of the most representative methods for multilayer community detection: *Generalized Louvain* (GL) [12], *Multiplex Infomap* (M-Infomap) [3], *Principal Modularity Maximization* (PMM) [17], and the consensus clustering approach in [9] (hereinafter denoted as **ConClus**). Note that the latter two are aggregation-based methods; in particular, **ConClus** is a simple approach for consensus clustering in weighted networks.

Assessment criteria and setting. We employed the *multilayer modularity* defined in [16], the *multilayer silhouette* defined in [16], and *NMI* [15].

To generate the ensemble for each evaluation network, following the lead of the study in [16], we used the serial version of the **Nerstrand** algorithm [10], a very effective and efficient method for discovering non-overlapping communities in (single-layer) weighted graphs via modularity optimization. We also used **Nerstrand** for the community-partitioning step in our M-EMCD*.

As concerns the competing methods, we used the default setting for **GL** and **M-Infomap**. We varied the number of communities in **PMM** from 5 to 100 with increments of 5, and finally selected the value corresponding to the highest modularity. Also, we equipped **ConClus** with **Nerstrand** (for the generation of the clusterings), set n_p to the number of layers, and varied θ in the full range (with step 0.01) to finally select the value that determined the consensus clusters with the highest average NMI w.r.t. the initial ensemble solutions.

More details about the evaluation networks and the competing methods can be found at <http://people.dimes.unical.it/~andreatagarelli/emcd/>.

Table 2. Size and modularity (upper table) and silhouette (bottom table) of lower-bound (CC-EMCD) and M-EMCD* consensus (in brackets, when applicable, the increments over M-EMCD), with or without model-filters.

	CC-EMCD modularity				M-EMCD* modularity				M-EMCD* #communities			
	θ -based	MLF	ECM	GloSS	θ -based	MLF	ECM	GloSS	θ -based	MLF	ECM	GloSS
AUCS	0.60	0.68	0.66	0.21	0.86 (+0.03)	0.91	0.91	0.25	14	13	18	52
EU-Air	0.73	0.60	0.60	0.07	0.91	0.91	0.90	0.09	274	39	45	397 (-2)
FAO-Trade	0.74	0.59	0.30	0.20	1.00	1.00	0.99 (+0.29)	0.99 (+0.56)	41 (+1)	1 (-2)	11 (+3)	40 (-17)
FF-TW-YT	0.48	0.44	0.44	0.05	0.73 (+0.12)	0.94	0.94	0.05	119 (+33)	115	133	5134
London	0.89	0.85	0.85	0.41	0.90	0.97	0.97	0.49 (+0.06)	45	46	46	340 (-3)
VC-Graders	0.22	0.33	0.27	-0.01	0.88 (+0.54)	0.44	0.43	0.03 (-0.01)	3 (-8)	16	17	26 (-1)

	CC-EMCD silhouette				M-EMCD* silhouette			
	θ -based	MLF	ECM	GloSS	θ -based	MLF	ECM	GloSS
AUCS	0.07	0.23	0.28	0.14	0.37 (+0.01)	0.38	0.40	0.15
EU-Air	0.01	0.16	0.18	-0.05	0.09	0.27	0.30	0.04 (-0.02)
FAO-Trade	-0.06	0.01	0.02	0.01	0.08	1.00 (+0.91)	0.06 (-0.05)	0.06 (-0.05)
FF-TW-YT	0.00	0.06	0.06	0.03	0.00 (-0.04)	0.15	0.12	0.03
London	0.14	0.06	0.06	0.03	0.18	0.20	0.20	0.12 (+0.04)
VC-Graders	0.24	0.20	0.21	0.05	0.52 (+0.23)	0.24	0.28	0.83 (+0.77)

5 Results

5.1 Impact of model-filters on M-EMCD*

For every network, we analyzed size, modularity and silhouette of the consensus solution obtained before (i.e., at lower-bound CC-EMCD) and at convergence of the optimization performed by M-EMCD*, when using either global threshold θ pruning or one among MLF, ECM, and GloSS; in the former case, the value of modularity refers to the consensus solution corresponding to the best-performing θ value. Results are reported in Table 2 and discussed next. At the end of this section, we also mention aspects related to time performance evaluation.

Size of consensus solutions. MLF and ECM tend to produce similar number of communities. By contrast, GloSS is in general much more aggressive than the other models, which causes proliferation of communities in the co-association graph. Also, the final solution by M-EMCD* can differ in size from the initial consensus by CC-EMCD, due to the optimization of modularity.

Modularity analysis. Looking at the modularity results, besides the expected improvement by M-EMCD* over CC-EMCD in all cases, the following remarks stand out. First, MLF and ECM again behave similarly in most cases, while GloSS reveals to be much weaker; this is clearly also dependent on the tendency by GloSS of heavily pruning the co-association graph, as discussed in the previous analysis on the size of consensus solutions. Second, using MLF or ECM leads to higher modularity w.r.t. the best-performing global threshold, in all networks but VC-Graders. This would support the beneficial effect deriving from the use of a model-filter for the co-association graph matrix; note however that such results should be taken with a grain of salt, since modularity is computed on differently prunings of the same network. Also, FAO-Trade deserves a

special mention, since its much higher multigraph density (13.97) and dimensionality (i.e., number of layers) (cf. Table 1) also caused a densely connected co-association graph, with average degree of 74, average path length of 1.67, clustering coefficient of 0.64, and 1 connected component. This makes FAO-Trade a difficult testbed for a community detection task, which explains the outcome reported in Table 2: 11 consensus communities are produced when using ECM, 41 and 40 with θ -based approach and GloSS, respectively, with most of them singletons and disconnected, and even 1 community for MLF.

It is worth noting that most of the performance gains by M-EMCD* over M-EMCD are obtained for θ -based pruning, but not for model-filter pruning. This would suggest the ability of M-EMCD* of achieving high quality consensus even when a refined model-filter would not be used.

Silhouette and NMI analysis. In terms of silhouette, the use of model-filter pruning is beneficial to both CC-EMCD and M-EMCD* consensus solutions, where the latter achieve significantly higher silhouette in most cases. Among the filters, again MLF and ECM tend to perform closely — with a slight prevalence of ECM — and better than GloSS (except for VC-Graders, where the number of communities is close to the number of nodes in the co-association graph).

We also measured the NMI of M-EMCD and M-EMCD* model-filter consensus solutions vs. the corresponding solutions obtained by θ -based pruning (results not shown). NMI was found very high (above 0.8, up to 1.0) in EU-Air, AUCS, and VC-Graders, around 0.60-0.70 in FF-TW-YT and London, and around 0.40-0.50 in FAO-Trade. Overall, this indicates that the model-filter pruning has similar capabilities as the best θ -based pruning in terms of community membership, though with the advantage of not requiring parameter selection.

Time performance analysis. Considering the execution time of model-filter pruning (results not shown), ECM is in general more costly than GloSS, and this in turn more costly than MLF. This gap — at least one order of magnitude — of ECM against the other two filters can be explained since its higher requirements due to its capability of preserving both degree and strength distributions. Details are reported at <http://people.dimes.unical.it/andreatagarelli/emcd/>.

5.2 Evaluation with competing methods

Table 3 summarizes the increments in terms of size, modularity, silhouette (Table 2), and NMI of M-EMCD* solutions w.r.t. the corresponding solutions obtained by each of the competitors, by varying model-filters. For the NMI evaluation, we distinguished two cases: the one, valid for GL, PMM, or M-Infomap, whereby the reference community structure is the solution obtained by the method in case of θ -based pruning, with θ selected according to the best-modularity performance; the other one, valid for ConClus, whereby we computed the average NMI over the layer-specific community structures.

Table 3. Increments of number of communities, modularity, silhouette and NMI of M-EMCD* solutions, by varying model-filters, w.r.t. corresponding solutions obtained by GL, PMM, M-Infomap, and ConClus.

	Gains by M-EMCD* vs. GL											
	#communities			Modularity			Silhouette			NMI w.r.t. θ -based pruning		
	MLF	ECM	GloSS	MLF	ECM	GloSS	MLF	ECM	GloSS	MLF	ECM	GloSS
AUCS	+6	+8	+48	+0.09	+0.08	-0.39	+0.11	+0.10	-0.01	+0.06	+0.21	+0.47
EU-Air	-23	-27	+364	+0.12	+0.11	-0.23	+0.26	+0.29	+0.08	+0.51	+0.48	+0.3
FAO-Trade	-5	+4	+30	+0.53	+0.60	+0.70	+0.97	+0.07	+0.07	-0.55	-0.28	+0.21
FF-TW-YT	+111	+130	+5131	+0.29	+0.27	-0.29	-0.07	-0.10	-0.05	+0.02	+0.05	+0.4
London	+23	+23	+318	+0.05	+0.05	-0.42	+0.08	+0.08	-0.30	-0.14	-0.13	-0.06
VC-Graders	0	+2	+18	-0.23	-0.26	-0.40	+0.15	+0.21	+0.71	+0.3	+0.31	+0.08
	Gains by M-EMCD* vs. PMM											
	#communities			Modularity			Silhouette			NMI w.r.t. θ -based pruning		
	MLF	ECM	GloSS	MLF	ECM	GloSS	MLF	ECM	GloSS	MLF	ECM	GloSS
AUCS	-1	+4	+38	+0.43	+0.29	0.00	+0.12	+0.13	-0.04	+0.24	+0.26	+0.18
EU-Air	-47	-41	+311	+0.66	+0.65	+0.04	+0.30	+0.33	+0.12	+0.61	+0.61	+0.47
FAO-Trade	-39	-29	0	+0.91	+0.90	+0.90	+1.02	+0.06	+0.07	-0.61	-0.4	+0.06
FF-TW-YT	+104	+122	+5123	+0.66	+0.60	-0.03	-0.14	-0.15	-0.12	-0.1	-0.11	-0.13
London	+1	+1	+295	+0.26	+0.28	0.00	+0.03	+0.03	-0.02	+0.06	+0.07	+0.16
VC-Graders	+1	+2	+11	-0.05	-0.01	-0.13	+0.25	+0.27	+0.95	+0.24	+0.2	-0.29
	Gains by M-EMCD* vs. M-Infomap											
	#communities			Modularity			Silhouette			NMI w.r.t. θ -based pruning		
	MLF	ECM	GloSS	MLF	ECM	GloSS	MLF	ECM	GloSS	MLF	ECM	GloSS
AUCS	+4	+4	+45	+0.18	+0.23	-0.12	+0.17	+0.11	+0.11	+0.48	+0.46	+0.38
EU-Air	-255	-251	+167	+0.38	+0.37	-0.20	+0.35	+0.37	+0.18	+0.74	+0.74	+0.56
FAO-Trade	0	+10	+39	+1.00	0.00	+0.99	+2.00	+1.06	+1.06	0	+0.22	+0.66
FF-TW-YT	+113	+130	+5132	+0.20	+0.24	-0.53	-0.15	-0.15	-0.23	+0.4	+0.3	+0.23
London	+37	+38	+338	+0.52	+0.52	+0.05	+0.21	+0.20	+0.12	+0.39	+0.4	+0.84
VC-Graders	+15	+16	+25	-0.49	-0.50	-0.58	+1.24	+1.28	+1.83	+0.66	+0.64	+0.47
	Gains by M-EMCD* vs. ConClus											
	#communities			Modularity			Silhouette			avg NMI of ensemble		
	MLF	ECM	GloSS	MLF	ECM	GloSS	MLF	ECM	GloSS	MLF	ECM	GloSS
AUCS	+5	+9	+42	+0.33	+0.38	-0.26	+0.13	+0.17	-0.11	-0.03	+0.00	+0.03
EU-Air	-25	-18	+323	+0.71	+0.71	-0.07	+0.23	+0.27	+0.06	-0.05	-0.04	+0.20
FAO-Trade	-16	-11	+21	+0.59	+0.77	+0.74	+0.92	-0.02	-0.01	-0.55	-0.27	+0.01
FF-TW-YT	+17	+74	+4885	+0.48	+0.47	-0.33	+0.15	+0.12	+0.02	-0.06	-0.04	+0.18
London	+16	+21	+298	+0.15	+0.14	-0.30	+0.09	+0.10	-0.01	+0.01	+0.02	+0.12
VC-Graders	+10	+10	+20	+0.21	+0.24	-0.20	+0.09	+0.11	+0.68	+0.02	-0.04	-0.14

This comparative analysis was focused on the impact of using the various model-filters on the methods' performance. To this end, for every network and model-filter, we first generated an ensemble of layer-specific community structures via *Nerstrand*, then we built the co-association graph and applied the filter, finally we removed from the original multilayer network the edges pruned by the model-filter, before providing it as input to each of the competing methods.

One general remark is that M-EMCD* equipped with MLF or ECM outperforms all competing methods in terms of both modularity and silhouette, and tends to produce more communities, with very few exceptions. Concerning NMI results for the first three methods, again the increments by M-EMCD* are mostly positive, thus implying that model-filter pruning appears to be more beneficial, w.r.t. a global threshold based pruning approach, for M-EMCD* than GL, followed by PMM and M-Infomap. Also, it is interesting to observe that, with the exception of FAO-Trade for MLF and ECM, M-EMCD* has average NMI of ensemble comparable to or even better than ConClus, whose performance values are optimal in terms of NMI (i.e., the parameter threshold corresponded to the best NMI over each network).

6 Conclusion

We proposed a new framework for consensus community detection in multilayer networks. This is designed to enhance the modularity-optimization process w.r.t. existing EMCD method. Moreover, by exploiting parameter-free generative models for graph pruning, our framework overcomes the dependency on a user-specified threshold for the global denoising of the co-association graph.

References

1. Dianati, N.: Unwinding the hairball graph: Pruning algorithms for weighted complex networks. *Phys. Rev. E* 93, 012304 (2016)
2. Dickison, M.E., Magnani, M., Rossi, L.: *Multilayer Social Networks*. Cambridge University Press, UK (2016)
3. Domenico, M.D., Lancichinetti, A., Arenas, A., Rosvall, M.: Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems. *Phys. Rev. X* 5, 011027 (2015)
4. Domenico, M.D., Nicosia, V., Arenas, A., Latora, V.: Structural reducibility of multilayer networks. *Nature communications* 6, 6864 (2015)
5. Gemmetto, V., Cardillo, A., Garlaschelli, D.: Irreducible network backbones: unbiased graph filtering via maximum entropy. *arXiv* (June 2017).
6. Gullo, F., Tagarelli, A., Greco, S.: Diversity-based weighting schemes for clustering ensembles. In: *Proc. SIAM Data Mining*, pp. 437–448 (2009)
7. Kim, J., Lee, J.: Community detection in multi-layer graphs: A survey. *SIGMOD Record* 44(3), 37–48 (2015)
8. Kivela, M., Arenas, A., Barthelemy, M., Gleeson, J.P., Moreno, Y., Porter, M.A.: Multilayer networks. *J. Complex Networks* 2(3), 203–271 (2014)
9. Lancichinetti, A., Fortunato, S.: Consensus clustering in complex networks. *Sci. Rep.* 2, 336 (2012)
10. LaSalle, D., Karypis, G.: Multi-threaded modularity based graph clustering using the multilevel paradigm. *J. Parallel Distrib. Comput.* 76, 66–80 (2015)
11. Mastrandrea, R., Squartini, T., Fagiolo, G., Garlaschelli, D.: Enhanced reconstruction of weighted networks from strengths and degrees. *New J. Physics* 16 (2014)
12. Mucha, P.J., Richardson, T., Macon, K., Porter, M.A., Onnela, J.P.: Community structure in time-dependent, multiscale, and multiplex networks. *Science* 328(5980), 876–878 (2010)
13. Radicchi, F., Ramasco, J.J., Fortunato, S.: Information filtering in complex weighted networks. *Phys. Rev. E* 83, 046101 (2011)
14. Serrano, M.A., Boguna, M., Vespignani, A.: Extracting the multiscale backbone of complex weighted networks. *PNAS* 106(16), 6483–6488 (2009).
15. Strehl, A., Ghosh, J.: Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.* 3, 583–617 (2003)
16. Tagarelli, A., Amelio, A., Gullo, F.: Ensemble-based community detection in multilayer networks. *Data Min. Knowl. Discov.* 31(5), 1506–1543 (Sep 2017)
17. Tang, L., Wang, X., Liu, H.: Uncovering groups via heterogeneous interaction analysis. In: *Proc. IEEE ICDM*, pp. 503–512 (2009)
18. Tang, L., Wang, X., Liu, H.: Community detection via heterogeneous interaction analysis. *Data Min. Knowl. Discov.* 25, 1–33 (2012)
19. Zhang, H., Wang, C., Lai, J., Yu, P.S.: Modularity in Complex Multilayer Networks with Multiple Aspects: A Static Perspective. *CoRR* abs/1605.06190 (2016)