Discovering Community Structure in Multilayer Networks

Soumajit Pramanik[1], Raphael Tackx[2], Anchit Navelkar[1], Jean-Loup Guillaume[3] and Bivas Mitra[1]
[1]Department of Computer Science & Engineering, IIT Kharagpur, India
[2]Sorbonne Universités, UPMC Université Paris 06, CNRS, LIP6 UMR 7606, 4 place Jussieu 75005 Paris, France
[3]L3I, University of La Rochelle, France
soumajit.pramanik@cse.iitkgp.ernet.in, raphael.tackx@lip6.fr, anchitn@gmail.com
jean-loup.guillaume@univ-lr.fr, bivas@cse.iitkgp.ernet.in

***Abstract*—Community detection in single layer, isolated networks has been extensively studied in the past decade. However, many real-world systems can be naturally conceptualized as multilayer networks which embed multiple types of nodes and relations. In this paper, we propose algorithms for detecting communities in multilayer networks. The crux of the algorithm is based on the multilayer modularity index Q_M, developed in this paper. The proposed algorithm is parameter-free, scalable and adaptable to complex network structures. More importantly, it can simultaneously detect communities consisting of only single type, as well as multiple types of nodes (and edges). We develop a methodology to create synthetic networks with benchmark multilayer communities. We evaluate the performance of the proposed community detection algorithm both in the controlled environment (with synthetic benchmark communities) and on the empirical dataset (Yelp and Meetup dataset); in both cases, the proposed algorithm outperforms the competing state-of-the-art algorithms.**

***Keywords*-Multilayer Network, Modularity, Community Detection**

I. Introduction

Communities are defined as groups of nodes that are more densely connected to each other than to the rest of the network. The goal of the community detection algorithms, consequently, is to partition the networks into groups of nodes; large body of work exists on community detection in single and isolated network [1]. Recently, many real networks, including communication, social, infrastructural and biological ones, are often represented as multilayer networks [2]. A multilayer network is comprised of multiple interdependent networks, where each network layer represents one aspect of interaction. Moreover, the functionality of a node in one network layer is dependent on the role of nodes in other layers. For instance, a location based social networks (say, Yelp) can be represented as a multilayer network (see Fig. 1) where in one layer customers (visitors) are connected via social links and in the other layer location nodes are connected through proximity links. The (coupling) link connecting a customer with a location node represents the visit of a customer to a location.

Community detection in complex multilayer networks is an important research problem. The communities in multilayer networks help to identify functionally cohesive sub-units and reveal complex interactions between multi-type nodes and heterogeneous links. They are also found to be beneficial for different data mining tasks such as context-sensitive search, prediction and recommendation etc [3]. Community detection in multilayer network is challenging as the detected communities have possibility to contain only single or multiple types of nodes. Most of the recent endeavors concentrated on the multiplex networks [4], [5] where all layers share the identical set of nodes but may have multiple types of interactions. In multiplex network, some of the approaches propose new quality metrics [4] to measure the goodness of the detected communities whereas a few other approaches utilize random walk [5] or frequent-pattern mining techniques [6] to obtain structurally similar components. In principle, most of the aforementioned algorithms transform the problem to the classical community detection in a monoplex network leveraging on the fact that in multiplex network, one-to-one cross layer links connect the copies of the same nodes in multiple layers. Unfortunately, the presence of heterogeneous nodes across multiple layers and cross layer dependency links make the aforementioned solutions inadequate for multilayer networks.

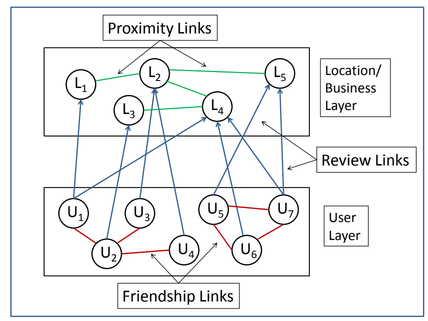


Fig. 1. A sample multilayer (Yelp) network.

Attempts have been made in bits and pieces to detect communities in multilayer networks; novel methodologies have been introduced such as Dirichlet process [7], tensor factorization [3], subspace clustering [8], non-negative matrix factorization [9] etc. However, most of these approaches suffer from several limitations. First of all, some of the aforesaid algorithms only work on a specific type of multilayer networks

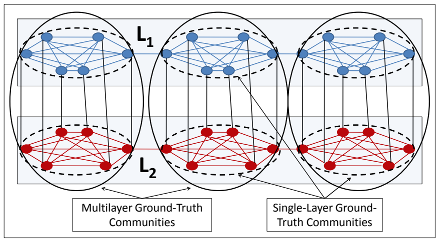


Fig. 2. Network configurations with two different types of ground truth communities.

(say, star-type [7] etc.). Secondly, some of them are forced to detect communities comprising only multiple types of nodes [3], hence introducing bias. Third, the desired number of communities are required to be fixed apriori for most of them [3], [8], limiting their capability to discover the true set of communities. Finally, a proper framework to generate benchmark communities for generic multilayer network is not available in any of them.

Recent endeavors directed towards development of modularity index for heterogeneous networks. For instance, composite modularity [10] calculates the modularity of a multi-relational network as the integration of modularities calculated for each single-relational subnetwork. However, due to the deficiency in definition, the composite modularity can only produce communities with single type of nodes. On the other side, modularity proposed in [11] in the context of gene-chemical interaction network fails to conceive the role of coupling links in communities characterization. The detailed exploration of prior art reveals the importance of the multilayer community detection algorithm which is free from (a) any external parameter, say total number of communities (b) any bias towards communities with only single type or only multiple types of nodes. Developing a suitable modularity index should be the first step towards this direction.

In this paper, we propose a community detection algorithm for multilayer networks which is able to detect communities comprising both single type as well as multiple types of nodes, depending on the network structure. First we represent the multilayer network with proper notations and define the problem of community detection (sec. II). Next, we develop a methodology to construct synthetic multilayer network with ground truth communities and evaluate it rigorously (sec. III). The major contribution of this paper is to propose a modularity index Q_M for characterizing communities in multilayer networks. Subsequently, we develop the multilayer community detection algorithms **GN-**Q_M and **Louvain-**Q_M incorporating the modularity index Q_M. We present the convergence proof for both the proposed algorithms along with their complexity analysis (sec. IV). We first evaluate the performance of the proposed modularity as a community scoring metric (sec. V) and then tested the performance of the developed algorithms against the competing algorithms. Controlled experiments, performed on the synthetic network, exhibit the ability of the **GN-**Q_M and **Louvain-**Q_M algorithms to efficiently detect communities comprising both single types and multiple types on nodes (sec. VI). Finally, we evaluate the proposed multilayer algorithms on the empirical dataset (Yelp and Meetup) and demonstrate that they outperform the state of the art baselines in correctly discovering the communities (sec. VII).

II. Representation & Problem statement

We start with formally representing the multilayer network and defining the respective communities. Next, we state the problem of detecting communities in multilayer network and the key challenges.

A. Representation

We represent a multilayer network as a tuple $\mathcal{G} = (\mathcal{G}_{\mathcal{U}}, \mathcal{G}_{\mathcal{B}})$ where $\mathcal{G}_{\mathcal{U}} = \{L_i : i \in \{1, 2, \ldots, M\}\}$ is a family of M uni-partite graphs (called layers of $\mathcal{G}$) and $\mathcal{G}_{\mathcal{B}} = \{L_{ij} : i, j \in \{1, 2, \ldots, M\}, i \neq j\}$ is the family of bipartite graphs containing nodes from individual layers and the cross layer interconnections among them. We denote each layer $L_i = (V_i, E_i)$ where V_i and E_i are respectively the set of nodes and intra-layer edges present in L_i. In the same line, we can represent L_{ij} as a triplet (V_i, V_j, E_{ij}) where $\{E_{ij} \subseteq \{V_i \times V_j\} : \ i, j \in \{1, 2, \ldots, M\}, i \neq j\}$ is the set of coupling edges between nodes of layers L_i & L_j.

Definition: A community C in a multilayer network $\mathcal{G}$ is defined as a cohesive module $(\mathcal{C}_{\mathcal{U}}, \mathcal{C}_{\mathcal{B}})$ of $\mathcal{G}$ containing a subset of nodes from one or more layers and all the edges having both endpoints incident on them. Mathematically, $\mathcal{C}_{\mathcal{U}}$ and $\mathcal{C}_{\mathcal{B}}$ can be defined as $\mathcal{C}_{\mathcal{U}} = \{L_i^C = (V_i^C, E_i^C) : V_i^C \subseteq V_i, E_i^C = \{E_i \cap (V_i^C \times V_i^C)\}, i \in \{1, 2, \ldots, M\}\}$ and $\mathcal{C}_{\mathcal{B}} = \{L_{ij}^C = (V_i^C, V_j^C, E_{ij}^C) : V_i^C \subseteq V_i, V_j^C \subseteq V_j, E_{ij}^C = \{E_{ij} \cap (V_i^C \times V_j^C)\}, i, j \in \{1, 2, \ldots, M\}, i \neq j\}$.

Importantly, communities of a multilayer network $\mathcal{G}$ can be divided into two types (see Fig. 2) (a) cross layer communities (containing multiple types of nodes) for which $|\mathcal{C}_{\mathcal{B}}| \neq \Phi$; (b) single layer communities (containing only single type of nodes) for which $|\mathcal{C}_{\mathcal{B}}| = \Phi$.

B. Problem Statement

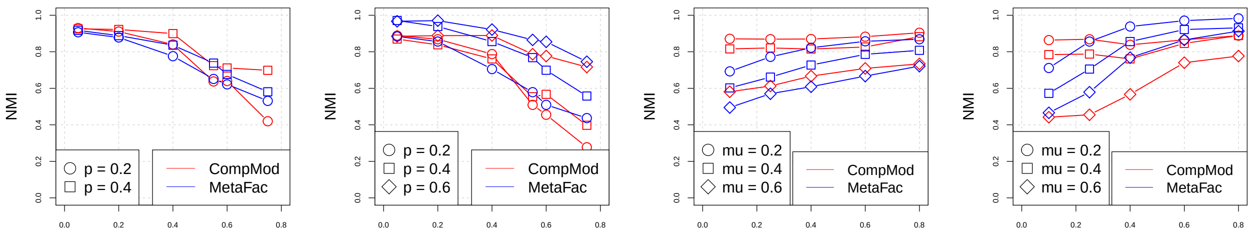
The problem of multilayer community detection algorithm is to divide the network $\mathcal{G}$ into a set of disjoint cohesive modules $C_1, C_2, \ldots, C_K$ which is a cover of the nodes in $\mathcal{G}$ such that each module C_i is comprised of a group of nodes densely connected inside & loosely connected outside the community.

The key challenges of this problem are two-fold - (a) deals with multilayer network which contains multiple types of links (of different densities) & nodes and (b) detects both cross layer & single layer communities simultaneously without any additional parameter.

III. Dataset

A. Synthetic Dataset Generation

In this section, we propose a methodology to generate benchmark multilayer networks with ground truth communities. The parameter α regulates the proportion of cross layer vs single layer communities in the benchmark. The network contains M number of different layers where each layer L_i



(a) mu, alpha = 0.4 (b) mu, alpha = 1.0 (c) p, alpha = 0.4 (d) p, alpha = 1.0

Fig. 3. Change of NMI values with μ and p for 'CompMod' and 'MetaFac' on 2-layer networks with 100 nodes in each layer, generated with maximum degree $k^i_{max} = 10$, average degree $\langle k_i \rangle = 6$ & coupling link density $d = 0.07$.

contains N_i nodes ($N_i = |V_i|$) with average degree $\langle k_i \rangle$. The methodology contains the following three steps:

Step 1. Single layer communities: First, we apply the LFR benchmark algorithm [12] to generate communities at each layer L_i with N_i nodes where both degree and community size distributions follow power law distribution with exponents γ_i and β_i respectively. We fix the mixing parameter as μ_i to construct $\mathcal{C}_i$ single layer communities in layer L_i.

Step 2. Cross layer communities: We combine the community $x_i \in \mathcal{C}_i$ of layer L_i with community $x_j \in \mathcal{C}_j$ of layer L_j to create the cross layer community x_{ij}. Assuming $|\mathcal{C}_i|$ and $|\mathcal{C}_j|$ as the number of communities in layers L_i and L_j respectively, $|\mathcal{C}_c| = min\{|\mathcal{C}_i|, |\mathcal{C}_j|\}$ denotes the maximum possible number of cross layer communities. We construct $(|\mathcal{C}_c| \times \alpha)$ cross layer communities by randomly combining single-layer communities from both the layers L_i & L_j respectively; notably each cross layer community x_{ij} may contain one or multiple single layer communities.

Step 3. Coupling links: Finally, we create the coupling links between the layers L_i and L_j with density d_{ij}. Fraction p denotes the mixing parameter for the cross layer communities. We first distribute $(N_i \times N_j \times d_{ij})$ coupling links randomly between the layers L_i and L_j where each link has one end in L_i and another end in L_j. Next, we rewire the coupling links such that p fraction of links stay inside the cross layer communities and the remaining $1-p$ fraction of links connect different cross layer communities.

B. Synthetic Dataset Evaluation

In the following, we evaluate the performance of the synthetic network generation algorithm; we examine whether the generated networks behave consistently with our expectation. In order to accomplish the task, we adhere to the following approach - **(a)** we generate synthetic networks varying different model parameters μ, α & p, which intrinsically regulate the quality of the community structures present in the generated networks. **(b)** we apply state-of-the-art multilayer community detection algorithms [3], [10] on this synthetic networks and evaluate the quality of the detected communities with respect to the ground truth communities. **(c)** we conclude that the synthetic network possesses desired properties, only if the quality of the detected communities in step (b) is consistent with the quality of ground truth communities specified in step (a).

1) Experimental Setup: We implement the following state-of-the-art multilayer community detection algorithms to detect the communities in step (b).

(i) MetaFac [3]: This algorithm detects communities based on the tensor factorization and requires the number of communities to be specified apriori[1]. It detects communities in a way such that each of them contains at least one node from every layer (i.e. only cross layer communities).

(ii) CompMod [10]: This algorithm detects communities by maximizing *composite modularity*, which is a combination of the modularities of different subnetworks.

We compute normalized mutual information (NMI) [13] index to compare the detected communities with the ground truth communities (step (b)). NMI is a measure of similarity of communities, which attains a high value if the ground truth and detected communities exhibit good agreement.

2) Evaluation: Finally, we accomplish the step (c) by demonstrating that synthetic network possesses the desired properties, in the following way.

Varying μ: In Fig. 3(a) and Fig. 3(b), we increase the mixing parameter μ (fraction of edges going out of single layer communities) which degrades the qualities of the communities. Clearly, as μ increases, NMI drops for both the state of the art algorithms, irrespective of α and p values. This points to the fact that with increasing μ, the ground truth communities degrade independent of the type of communities, which meets our expectation.

Varying p: Similarly, Fig. 3(c) and Fig. 3(d) focuses on p, which intuitively regulates the cohesiveness of the cross layer communities. Evidently, the obtained NMI rises with p for both the community detection algorithms. The slope is observed to be relatively steeper for the higher α, as it indicates the presence of more number of cross layer communities (magnifying the effect of p).

IV. Developing Multilayer Community Detection Algorithm

In this section, first we develop the modularity index Q_M to characterize the quality of multilayer communities. Next, we

[1]For our experiment, we vary the input number of communities from two to fifty and report the one exhibiting highest overlap with the ground truth.

show that simple adaptation of Q_M with classical methodologies may lead to the development of multilayer community detection algorithms.

A. Desired properties of multilayer communities: Intuition

We start this section highlighting the desired properties of the communities in a multilayer network. As introduced in section II, in multilayer networks, we observe two types of communities (see Fig. 2) (a) cross layer communities (containing multiple types of nodes) and (b) single layer communities (containing only single type of nodes).

For an 'ideal' cross layer community $C = (\mathcal{C_U}, \mathcal{C_B})$ of $\mathcal{G}$ (where $|\mathcal{C_B}| \neq \Phi$), the desired properties are the following,

$Property^{X1}$: The group of nodes in each uni-partite and bipartite layer (L_i^Cs and L_{ij}^Cs respectively) should be highly cohesive.

$Property^{X2}$: The (coupling) edges in the bipartite layer $L_{ij}^C \in \mathcal{C_B}$ should connect most of the nodes from L_i^C (i.e. V_i^C) with most of the nodes from L_j^C (i.e. V_j^C).

Similarly, for an 'ideal' single layer community $C = (\mathcal{C_U}, \mathcal{C_B})$ of $\mathcal{G}$ (where $\mathcal{C_B} = \Phi$), the desired properties are enumerated below,

$Property^{S1}$: The community should be highly cohesive within the layer L_i to which it belongs (i.e. $\mathcal{C_U} = L_i^C \subseteq L_i$).

$Property^{S2}$: The nodes in L_i^C (i.e. V_i^C) should be very loosely connected with nodes of other layers L_j.

B. Multilayer Modularity Index

Following the aforesaid intuitions, we propose multilayer modularity for a two-layer network $\mathcal{G} = \{\{L_1, L_2\}, \{L_{12}\}\}$ where $L_1 = (V_1, E_1)$ & $L_2 = (V_2, E_2)$ are the individual layers and $L_{12} = (V_1, V_2, E_{12})$ is the bipartite graph connecting nodes of layer L_1 and L_2.

We start with the basic notion of Newman-Girvan modularity [14], $Q = \frac{1}{2m}\sum_{i,j}(A_{ij} - P_{ij})\delta(\psi_i, \psi_j)$ where m is the total number of edges in the network, $i, j \in V_1 \cup V_2$ are any pair of nodes in the network, ψ_i indicates the community membership of node i and $\delta(\psi_i, \psi_j)$ is the Kronecker delta function which is 1 only if $\psi_i = \psi_j$ i.e. i and j belong to the same community (and 0 otherwise). A_{ij} represents the classical adjacency matrix of $\mathcal{G}$. The penalty term P_{ij} (often referred as null model) is the expected probability of having an edge between nodes i and j if edges are placed at random. Notably, in multilayer network the edges in one layer L_i are intrinsically different from another layer L_j, which gets reflected in the property of the corresponding layers as well (say diverse edge densities across the layers). This observation motivates us to introduce layer-wise discriminating null models, and propose the following multilayer modularity,

$$Q_M = \frac{1}{2m}\sum_{ij}\{(A_{ij} - P_{ij})\delta(\psi_i, \psi_j)\} \; where \tag{1}$$

$$P_{ij} = \begin{cases} P_{ij}^1 \;\; if \; i \in V_1 \;\&\; j \in V_1 \\ P_{ij}^2 \;\; if \; i \in V_2 \;\&\; j \in V_2 \\ P_{ij}^{12} \;\; if \; (i \in V_1 \;\&\; j \in V_2) \; or \; (i \in V_2 \;\&\; j \in V_1) \end{cases}$$

In the following, we compute the null model terms P_{ij}^1, P_{ij}^2 and P_{ij}^{12} separately for both types of communities of multilayer network and finally derive the multilayer modularity index Q_M.

1) Cross Layer Communities: Any cross layer community C is composed of three submodules - two intra layer (L_1^C, L_2^C) and one inter layer (L_{12}^C). The vanilla null model proposed in [14] directly derives the P_{ij}^1, P_{ij}^2 for intra layer submodules. The expected number of edges between any two nodes i and j (with intra-layer degrees h_i and h_j respectively) within the community C can be calculated as $P_{ij}^1 = (h_i * h_j)/2\,|E_1|$ (for submodule L_1^C) and $P_{ij}^2 = (h_i * h_j)/2\,|E_2|$ (for submodule L_2^C). For inter layer or bipartite submodule L_{12}^C, the probability of an edge between node i and j depends on their respective coupling degrees c_i and c_j. The probability of having a coupling edge between i, j can be estimated as $P_{ij}^{12} = (c_i * c_j)/|E_{12}|$ (similar to [15]). The aforementioned null models satisfy the desired requirements of $Property^{X1}$ introduced in section IV-A.

Each cross layer community C is represented as $\{\{L_1^C, L_2^C\}, \{L_{12}^C\}\}$ where L_1^C and L_2^C are the submodules with edges from E_1 and E_2 respectively and L_{12}^C is from E_{12}; we substitute P_{ij} in Eq. 1 to define its modularity as

$$\begin{aligned} Q_M^C = \forall i, j \in C \; \Bigg[\frac{1}{3}\Bigg\{ & \frac{1}{2\,|E_1|}\sum_{i,j \in V_1}\Big(A_{ij} - \frac{(h_i * h_j)}{2\,|E_1|}\Big) \\ & + \frac{1}{|E_{12}|}\sum_{i \in V_1, j \in V_2}\Big(A_{ij} - \frac{(c_i * c_j)}{|E_{12}|}\Big) \\ & + \frac{1}{2\,|E_2|}\sum_{i,j \in V_2}\Big(A_{ij} - \frac{(h_i * h_j)}{2\,|E_2|}\Big)\Bigg\}\Bigg] \end{aligned} \tag{2}$$

In case of inter layer submodule, if node i in C is not connected with any other layer (hence, coupling degree c_i zero), we use its intra layer degree h_i as a proxy of c_i for computing P_{ij}^{12}. This allows us to penalize those nodes in cross layer community C which are only connected with nodes within the same layer and *not* with the nodes in C of different layer. This amendment in the null model P_{ij}^{12} satisfies the desired requirements of $Property^{X2}$. Therefore with this modification, the Eq. 2 can be written as,

$$\begin{aligned} Q_M^C = \forall i, j \in C \; \Bigg[\frac{1}{3}\Bigg\{ & \frac{1}{2\,|E_1|}\sum_{i,j \in V_1}\Big(A_{ij} - \frac{(h_i * h_j)}{2\,|E_1|}\Big) + \\ & \frac{1}{2\,|E_1| + 2\,|E_2| + |E_{12}|}\sum_{i \in V_1, j \in V_2}\Big(A_{ij} - \frac{(c'_i * c'_j)}{2\,|E_1| + 2\,|E_2| + |E_{12}|}\Big) \\ & + \frac{1}{2\,|E_2|}\sum_{i,j \in V_2}\Big(A_{ij} - \frac{(h_i * h_j)}{2\,|E_2|}\Big)\Bigg\}\Bigg] \end{aligned} \tag{3}$$

where for any node i, $c'_i = c_i$ if $c_i > 0$ and $c'_i = h_i$ otherwise.

2) Single Layer Communities: In any single layer community C, all the constituent nodes belong to either L_1 or L_2. The null models can be directly derived as $P_{ij}^1 = (h_i * h_j)/2\,|E_1|$ and $P_{ij}^2 = (h_i * h_j)/2\,|E_2|$ for layer L_1 and L_2 respectively from the vanilla null model [14]. These null models satisfy the $Property^{S1}$ introduced in section IV-A. Hence, for each single layer community C represented as $\{L_1^C\}$, we substitute P_{ij} in Eq. 1 to compute the modularity as,

$$Q_M^C = \forall i,j \in C \left[\frac{1}{3} \left\{ \frac{1}{2\,|E_1|} \sum_{i,j \in V_1} \left(A_{ij} - \frac{(h_i * h_j)}{2\,|E_1|} \right) \right\} \right] \quad (4)$$

Importantly, there can be many nodes in C which are connected to other layer nodes via coupling edges, violating $Property^{S2}$ of desired single layer community. In order to penalize those nodes in a community C with coupling degree c_i, we add c_i along with h_i to estimate the null model. Subsequently, the modularity of C becomes

$$Q_M^C = \forall i,j \in C \left[\frac{1}{3} \left\{ \frac{1}{2\,|E_1| + |E_{12}|} \sum_{i,j \in V_1} \left(A_{ij} - \frac{(h_i + c_i) * (h_j + c_j)}{2\,|E_1| + |E_{12}|} \right) \right\} \right] \quad (5)$$

Finally, combining both types of communities, the overall modularity of the network can be represented as

$$Q_M = \frac{1}{3} \sum_{k=1}^{n_C} \Bigg[\forall i,j \in C_k \Bigg\{ \frac{1}{2\,|E_1| + \theta_{C_k} * |E_{12}|} \sum_{i,j \in V_1} \left(A_{ij} - \frac{(h_i + \theta_{C_k} * c_i) * (h_j + \theta_{C_k} * c_j)}{2\,|E_1| + \theta_{C_k} * |E_{12}|} \right) + \frac{1}{2\,|E_1| + 2\,|E_2| + |E_{12}|} \sum_{i \in V_1, j \in V_2} \left(A_{ij} - \frac{(c'_i * c'_j)}{2\,|E_1| + 2\,|E_2| + |E_{12}|} \right) + \frac{1}{2\,|E_2| + \theta_{C_k} * |E_{12}|} \sum_{i,j \in V_2} \left(A_{ij} - \frac{(h_i + \theta_{C_k} * c_i) * (h_j + \theta_{C_k} * c_j)}{2\,|E_2| + \theta_{C_k} * |E_{12}|} \right) \Bigg\} \Bigg] \quad (6)$$

where n_C is the total number of apriori communities and θ_{C_k} is a variable denoting type of the community C_k. θ_{C_k} is 1 if C_k is a single layer community and 0 if C_k is a cross layer community. Notably, for any single layer community, maximum one of the single layer terms in Eq. 6 can be non-zero; the other two terms will always be zero.

Algorithm 1: GN-Q_M

Input : A multilayer network $\mathcal{G}$ as defined in section IV-B where $E = E_1 \cup E_2 \cup E_{12}$.
Output: Maximum Q_M of $\mathcal{G}$ and detected communities.

Calculate the betweenness centralities of all edges
while $|E| > 1$ **do**
    Remove the edge $e \in E$ with maximum betweenness centrality from E
    $currQ$ = current Q_M of $\mathcal{G}$
    Save the current partition along with $currQ$
    Update the betweenness centralities of remaining edges
end
Display partition with maximum Q_M
return

Although, we show the derivation of Q_M for a two layer network, it can be easily extended for more than two layers. For instance, if the network has L layers with $L-1$ coupling relationships between them, there would be L single layer and $L-1$ bipartite modularity terms in the corresponding Q_M equation.

C. Community detection algorithm

We leverage on the single layer community detection algorithms Girvan-Newman [16] & Louvain [17] which detect communities by maximizing Girvan-Newman modularity [14].

Algorithm 2: Louvain-Q_M

Input : A multilayer network $\mathcal{G}$ as defined in section IV-B where $V = V_1 \cup V_2$.
Output: Maximum Q_M of $\mathcal{G}$ and detected communities.

while *True* **do**
    Place each node of $\mathcal{G}$ into a single community
    Save Q_M for this decomposition
    while *there are moved nodes* **do**
        foreach *node* $n \in V$ **do**
            c = neighboring community of n maximizing Q_M increase
            if *c results in a strictly positive increase* **then**
                move n from its community to c
            end
        end
    end
    if Q_M *reached is higher than the initial* Q_M **then**
        Display the partition found
        Transform $\mathcal{G}$ into the network between communities
    else
        break
    end
end
return

We substitute the Girvan-Newman modularity by our proposed modularity index Q_M and develop **GN-**Q_M (Algo. 1) and **Louvain-**Q_M (Algo. 2) algorithms respectively for multilayer networks. Although vanilla algorithms are intrinsically incapable of distinguishing different types of edges and nodes in the multilayer network, however, due to the adaptability of Q_M, **GN-**Q_M and **Louvain-**Q_M should be able to detect both cross layer and single layer communities. Essentially Q_M works as a patch on top of any single layer community detection algorithm to detect multilayer communities.

D. Convergence & Complexity

Finally we show that algorithms **GN-**Q_M and **Louvain-**Q_M converge and they are tractable in terms of time complexity.

1) ***GN-**Q_M:* At every step we remove one edge from the network and hence, the algorithm certainly stops after the removal of all the $|E_1| + |E_{12}| + |E_2|$ edges. Clearly, the theoretical worst case complexity of the algorithm is $O((|V_1|+|V_2|) \times (|E_1|+|E_{12}|+|E_2|)^2)$ as finding betweenness centrality in unweighted graphs costs $O((|V_1|+|V_2|) \times (|E_1| + |E_{12}| + |E_2|))$ operations [18].

2) ***Louvain-**Q_M:* At each iteration of every pass, each node is placed into one of its neighbouring community only if the movement leads to a strictly positive gain in modularity Q_M. Computing this gain both proves that the algorithms converges and gives an upper bound on its complexity.

Suppose, at any particular iteration the node x is to be moved from its own community C_1 to another community C_2. Without loss of generality, let us assume that $x \in V_1$ and x is connected with h_x nodes in V_1 & c_x nodes in V_2. For simplicity, we perform this movement in two steps: first, we remove x from C_1 and keep it as an isolated community; second, we insert x into C_2. C_1 and C_2 can be either cross layer or single layer communities, independently. Below we derive the gain assuming C_1 is cross layer and C_2 is single layer. The other three cases can be derived in a similar manner. Note that since the modularity is an independent sum over all communities, the contribution of other communities than C_1 and C_2 is not affected by the movement of x. Therefore we will only compute the change of modularity of C_1 and C_2.

The modularity $Q_M^{C_1,C_2}$, restricted to C_1 and C_2, before any movement can be simply expressed as, $Q_M^{C_1,C_2} = Q_M^{C_1} + Q_M^{C_2}$, where $Q_M^{C_1}$ follows from Eq. 3 (as C_1 is cross layer) and $Q_M^{C_2}$ follows from Eq. 5 (as C_2 is single layer).

Once node x is removed from C_1 and kept as an isolated community, the modularity sum of the C_1, C_2 and x becomes,

$$Q_M^{C_1,C_2,\{x\}} = \frac{1}{3}\left[\frac{1}{m_1}\sum_{i,j\in V_1, C_1-\{x\}}(A_{ij} - \frac{h_i h_j}{m_1}) + \frac{1}{m}\sum_{i,j\in C_1-\{x\}, i\in V_1, j\in V_2}(A_{ij} - \frac{c'_i c'_j}{m}) + \frac{1}{m_2}\sum_{i,j\in V_2, C_1-\{x\}}(A_{ij} - \frac{h_i h_j}{m_2})\right] + \frac{1}{3}\left[\frac{1}{m_{21}}(A_{xx} - \frac{(h_x+c_x)^2}{m_{21}})\right] + Q_M^{C_2}$$

where $m_1 = 2|E_1|$, $m_2 = 2|E_2|$, $m_{12} = 2|E_1| + |E_{12}|$, $m_{21} = 2|E_2| + |E_{12}|$ and $m = 2|E_1| + 2|E_2| + |E_{12}|$.

Hence, the change in modularity $\Delta_r = Q_M^{C_1,C_2,\{x\}} - Q_M^{C_1,C_2}$ due to this removal (assuming $A_{xx} = 0$) is the following,

$$\Delta_r = \frac{h_x^2}{3{m_1}^2} - \frac{1}{3}\left[\frac{1}{m_1}\sum_{i\in V_1, C_1-\{x\}}(A_{ix} - \frac{(h_i h_x)}{m_1}) + \frac{1}{m}\sum_{i\in V_2, C_1-\{x\}}(A_{xi} - \frac{c'_x c'_i}{m})\right] - \frac{(h_x+c_x)^2}{3m_{21}^2}$$

Similarly, the change in modularity Δ_i due to the insertion of x in C_2 can be computed as,

$$\Delta_i = \frac{1}{3m_{12}}\sum_{i\in V_1, C_2}(A_{ix} - \frac{(h_i+c_i)(h_x+c_x)}{m_{12}}) + \frac{(h_x+c_x)^2}{3m_{21}^2}$$

Finally, the overall improvement due to this movement of node x from community C_1 to community C_2 is[2],

$$\Delta_r + \Delta_i = \Theta(\frac{1}{m^2})$$

Since the minimum gain for every move is of the order of $1/m^2$, in the worst case we need $O(m^2)$ iterations to maximize Q_M (as it is comprised between -1 and 1). We apply the same technique as in [17] to find the best neighboring community, so the cost for one vertex is proportional to its degree. As each iteration considers every vertex once, it leads to $O(m)$ operations per iteration. The theoretical worst case complexity of **Louvain-Q_M** is therefore $O(m^3)$. In practice, this worst case complexity bound is quite loose. For instance, in Fig. 4(a), we show the fraction of nodes moved along with the gain in modularity in each iteration of the first pass of the **Louvain-Q_M** algorithm while running it on a synthetic two layer network (generated following section III-A). The network has 600 nodes (300 in each layer) with $29,738$ edges ($m^2 = 1,104,232,900$) and the theoretical number of iterations could be of the order 1 billion but it takes just 5 iterations to end the first pass.

[2]Same order of magnitude can be derived for other types of C_1 and C_2.

V. Evaluation: Modularity Index Q_M

In this section, we evaluate the proposed multilayer modularity Q_M against baseline indices. In this experiment, we implement different configurations of multilayer network following section III-A and regulate the tuning parameters to obtain the desired topology.

A. Baseline Multilayer Modularities

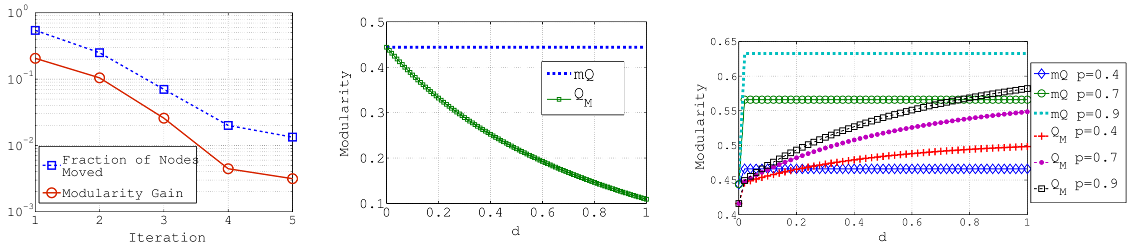
In literature, very few modularity indices are proposed for multilayer networks, illustrated in [10] & [11]. Out of this two, 'CompMod' proposed in [10] works only for communities with single type of nodes (i.e. single layer communities), leaving us with only the modularity 'mQ' proposed in [11] to compare with Q_M. For a two-layer network $\mathcal{G} = \{\{L_1, L_2\}, \{L_{12}\}\}$ where L_1 = (V_1, E_1) & L_2 = (V_2, E_2) are the individual layers and L_{12} = (V_1, V_2, E_{12}) is the bipartite graph connecting nodes of layer L_1 and L_2, mQ can be defined as,

$$mQ = \frac{1}{3}\sum_{k=1}^{n_C}\Bigg\{ \underbrace{(\frac{\left|E_1^{C_k}\right|}{|E_1|} - (\frac{h_1^{C_k}}{2|E_1|})^2)}_{Term\ for\ Single\ Layer\ L_1} + \underbrace{(\frac{\left|E_{12}^{C_k}\right|}{|E_{12}|} - \frac{r_{12}^{C_k} * s_{12}^{C_k}}{|E_{12}|^2})}_{Term\ for\ Coupling\ Edges} + \underbrace{(\frac{\left|E_2^{C_k}\right|}{|E_2|} - (\frac{h_2^{C_k}}{2|E_2|})^2)}_{Term\ for\ Single\ Layer\ L_2} \Bigg\} \quad (7)$$

where each community C_k is represented as $\{\{L_1^{C_k}, L_2^{C_k}\}, \{L_{12}^{C_k}\}\}$; $L_1^{C_k}$ and $L_2^{C_k}$ are the submodules with $E_1^{C_k}$ & $E_2^{C_k}$ edges from E_1 and E_2 respectively whereas $L_{12}^{C_k}$ contains $E_{12}^{C_k}$ edges from E_{12}; n_C is the number of apriori communities; $h_i^{C_k}$ is the sum of degrees of all $L_i^{C_k}$ nodes in L_i layer; $r_{12}^{C_k}$ is the sum of degrees of $L_1^{C_k}$ nodes in subnetwork L_{12} and $s_{12}^{C_k}$ is the sum of degrees of $L_2^{C_k}$ nodes in subnetwork L_{12}.

B. Network configurations

In order to compare mQ and Q_M, we construct a synthetic two-layer network $\{\{L_1, L_2\}, \{L_{12}\}\}$ (see Fig. 2) with three cohesive groups of same type of nodes (cliques) at each layer L_1 and L_2 respectively. Each clique contains 100



(a) Drop in modularity gain with number of iterations.

(b) mQ vs. Q_M for Config A while adding coupling links.

(c) mQ vs. Q_M for Config B while adding coupling links with varying p.

Fig. 4. (a) Drop in modularity gain along with fraction of modes moved in the first pass of **Louvain-Q_M** for a 300×300 synthetic multilayer network; (b) & (c) Comparative results of Q_M & mQ on the configurations in Fig. 2.

nodes (hence, 300 nodes per layer) and each layer contains $14,852$ intra layer edges. We consider two typical ground truth community configurations as shown in Fig. 2 - (i) **Config A:** single-layer communities comprising cohesive groups of single type of nodes only (six communities), (ii) **Config B:** cross layer communities, each comprising one group of nodes from Layer L_1 with another group from layer L_2 (three communities). Importantly, coupling edges, connecting two types of nodes, are the key characteristics of the multilayer network, which makes it strikingly different from multiplex network. Hence in this evaluation, we primarily concentrate on the coupling edges as a regulating topological parameter. The coupling edges between layers L_1 and L_2 can be tuned by varying the density parameters d and p (as discussed in section III-A).

C. Experimental Results

In our experiment, we increase the coupling edge density d from 0 to 1, for both the ground truth configurations A & B, with different p values. Intuitively, addition of coupling edges should dilute the single layer community structures in Config A, decreasing the modularity of Config A, whereas in Config B, it should make the cross layer communities more cohesive, increasing the modularity.

1) Config A: In Fig. 4(b) the plot corresponding to mQ reveals that it is completely insensitive to the increase in coupling edge density d. Precisely, in case of Config A the coupling edges do not have any contribution in Eq. 7 (coupling edge term vanishes for single layer communities), hence mQ remains invariant against addition of coupling edges. On the contrary, in case of Q_M, we penalize for the coupling edges connected with single layer communities (see $(h_i + \theta_{C_k} * c_i)$ terms in Eq. 6) which allows us to achieve the drop in Q_M values with increasing d (see Fig. 4(b)). This result concurs with the desired $Property^S$, introduced in section IV-A.

2) Config B: In case of Config B, mQ is unable to capture the desired increasing behavior (see $Property^X$, introduced in section IV-A) with the increase in coupling edges (except at the beginning when d becomes non-zero for the first time). Rather, it remains constant throughout the edge addition regime irrespective of the p values (see Fig. 4(c)). In Config B, as observed from Eq. 7, the coupling edge addition only affects the term for coupling edges of mQ; however, addition of edges influences the numerator and denominator almost equally, neutralizing the overall effect on mQ. On the other hand, in all the plots corresponding to Q_M, the modularity increases gracefully with coupling edge addition. This is achieved by suitably penalizing the null model in Eq. 3. As expected, the absolute value of modularity increases with p for both mQ and Q_M, since higher p improves the cohesiveness of cross layer communities.

In a nutshell, the aforesaid experiments clearly demonstrate the elegance of Q_M with respect to mQ, as a community quality index.

VI. Community Evaluation: Synthetic Network

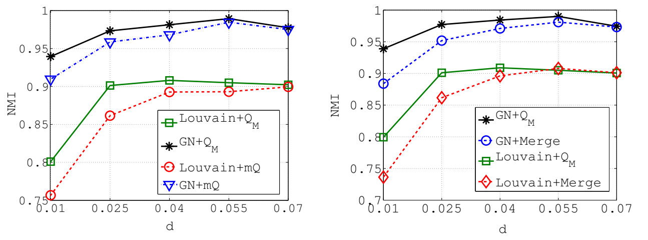
We evaluate the performance of the proposed multilayer community detection algorithms **GN-Q_M** and **Louvain-Q_M** in a controlled environment. First we present the experimental setup explaining the generation of synthetic multilayer network with ground truth communities and the evaluation metric. Next, we present the competing baseline algorithms and finally, we exhibit the elegance of the proposed algorithms over baselines from different perspectives.

A. Experimental setup

We generate the synthetic two layer networks ($\mathcal{G} = \{\{L_1, L_2\}, \{L_{12}\}\}$) with planted communities following the model proposed in section III-A. We fix the number of nodes $|V_i|$ in each layer L_i at 100 with maximum degree $k^i_{max} = 10$ & average degree $\langle k_i \rangle = 6$. The power-law exponents for degree distribution (γ_i) and community size distribution (β_i) for each layer are fixed at 2 and 1 respectively. The other model parameters (μ, α, p & d) are regulated and adjusted according to the requirement. We apply the normalized mutual information (NMI) [13] as the yardstick to quantify the similarity between the detected and ground truth communities. The synthetic network contains 30 single layer (& no cross layer) ground truth communities when $\alpha = 0$ and 15 cross layer (& no single layer) ground truth communities when $\alpha = 1$.

B. Competing algorithms

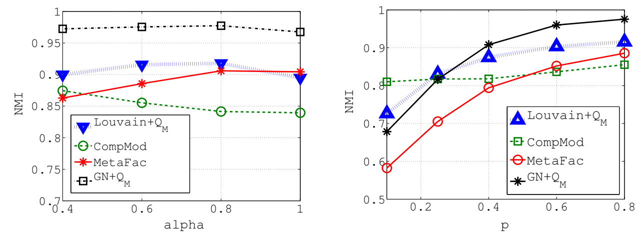
We introduce the following three classes of competing algorithms to evaluate the performance of **GN-Q_M** and **Louvain-Q_M**.



(a) Varying d values for $p = 0.4$, $\alpha = 0.6$, $\mu = 0.05$ for mQ based baselines

(b) Varying d values for $p = 0.4$, $\alpha = 0.6$, $\mu = 0.05$ for merging based baselines

Fig. 5. NMI of obtained and ground truth communities for various d values.



(a) Varying α values for $p = 0.8$, $\mu = 0.4$ and $d = 0.04$

(b) Varying p values for $\mu = 0.4$, $\alpha = 0.6$ and $d = 0.04$

Fig. 6. NMI of obtained and ground truth communities for various α & p values.

1) Baselines with mQ: We induce the multilayer modularity mQ proposed in [11] with the standard Girvan-Newman and Louvain algorithms [16], [17]. We refer these baseline algorithms as **GN-mQ** and **Louvain-mQ** respectively.

2) Merging based baselines: We apply standard Louvain algorithm [17] to detect communities at the individual layers and then attempt to merge those communities across the layers. We merge one top layer community C_T with one bottom layer community C_B with which it is maximally connected if the connection density between them is above a threshold[3]; otherwise keep C_T and C_B as a single layer community.

3) State-of-the-art algorithms: We implement 'MetaFac' [3] and 'CompMod' [10] as state of the art multilayer community detection algorithms, already introduced in section III-B.

C. Evaluation

1) Comparison with baselines with mQ: The proposed **GN-Q_M** and **Louvain-Q_M** algorithms perform quite closely with their respective benchmark **GN-mQ** and **Louvain-mQ** However, the improvement gets manifested when we reduce the density of coupling links d. In Fig. 5(a), we observe that **GN-Q_M** and **Louvain-Q_M** outperform the baselines in the lower link density regime. This comes from the fact that the modularity mQ fails to reflect the cohesiveness of multilayer communities for low d values, as explained in Fig. 4(c).

2) Comparison with merging based baselines: This baseline performs pretty close to aforesaid **GN-mQ** and **Louvain-mQ** since here also we optimize the modularities at top, bottom and bipartite layers separately. Evidently, the proposed **GN-Q_M** and **Louvain-Q_M** algorithms outperform this baseline in the low coupling link density (d) regimes (see Fig. 5(b)).

3) Comparison with state-of-the-art algorithms: **(a) Effect of α:** The model parameter α regulates the proportion of ground truth cross layer (against single layer) communities in the synthetic multilayer network. In Fig. 6(a), we observe that **GN-Q_M** and **Louvain-Q_M** do not exhibit high sensitivity with α; this points to the fact that performance of these two algorithms (specially GN-Q_M) does not depend on the proportion of single layer and cross layer communities present in the network. However, the performance of $MetaFac$ monotonically improves with increasing α whereas $CompMod$ exhibits the opposite behaviour. In fact, $MetaFac$ algorithm is intrinsically biased towards detecting cross layer communities whereas $CompMod$ is more suitable for detecting single layer communities. **(b) Effect of p:** Model parameter p realizes the cohesiveness of the coupling links in the apriori cross layer communities. We observe (see Fig. 6(b)) that our algorithms outperform $MetaFac$ and $CompMod$ in the presence of moderate to cohesive cross layer communities (say $p > 0.3$). However, for $p < 0.3$ $CompMod$ performs relatively better due to the degradation of the cross layer communities, since $CompMod$ intrinsically favors the single layer communities.

In a nutshell, we claim that (a) the proposed algorithms show pretty balanced behaviour across different range of model parameters (d, p etc) and (b) importantly, they can simultaneously detect both single layer and cross layer communities without any specific bias towards anyone of them (invariant to α). Though **GN-Q_M** and **Louvain-Q_M** perform relatively poorly in the lower p regions, notably, they never rank as the worst in the batch. $CompMod$ performs decently in lower p regions due to its intrinsic bias towards cross layer communities, where $MetaFac$ fails miserably.

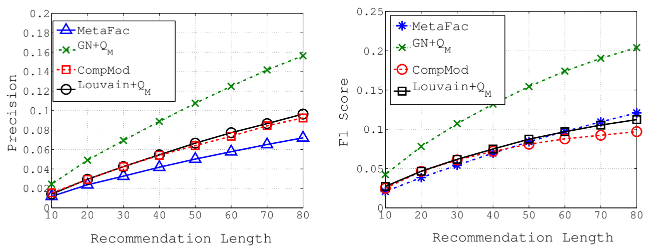
VII. Community Evaluation: Empirical Network

In this section, we demonstrate the performance of the **GN-Q_M** and **Louvain-Q_M** algorithms on the empirical dataset. First we construct the multilayer networks based on the information extracted from the 'Yelp' and 'Meetup' datasets. Next, we explain the evaluation procedure and show that the proposed algorithms outperform the competing baselines.

A. Yelp dataset

1) Data description: The dataset obtained from Yelp [19], a popular location based social network (LBSN) platform, contains detailed information regarding Yelp customers/visitors (including their social connections & residence), locations and the tips & reviews posted by the customers. We assume that a customer v visits a location L if v writes a tip/review for L. We concentrate on the most popular city "Las Vegas" containing $13,601$ locations and $173,697$ customers visiting those

[3]Merging is performed if the ratio of the number of coupling links between C_T & C_B and the total number of coupling links connected with C_T & C_B is at least Th. We vary Th from 0.1 to 1.0 and report the best obtained result.



(a) Precision (b) F1 Score

Fig. 7. Precision & F1 Score (avg. over all visitors) for communities obtained from different algorithms for various recommendation lengths on Yelp N/W.

locations. To keep the network tractable, we first consider only the customers visiting locations within 5KM radius from the center of the city. Then, we detect the maximally connected component in the friendship network of those visitors. This yields a set of 244 connected customers. Furthermore, we consider only the 1627 locations which are at least visited once by anyone of them.

2) Construction of multilayer network $\mathcal{G}_{yelp}$: We construct a multilayer network $\mathcal{G}_{yelp} = \{\{L_U, L_L\}, \{L_{UL}\}\}$ for Yelp with two layers where L_U = (V_U, E_U) is the customer layer containing customer nodes and their social connections; L_L = (V_L, E_L) is the location layer containing location nodes and their proximity connections (any two locations within 200 meters of each other are connected) and L_{UL} = (V_U, V_L, E_{UL}) is the bipartite graph connecting customer node $c \in V_U$ with location node $l \in V_L$ if customer c visits the location l (see Fig. 1) .

3) Evaluation procedure: Unlike synthetic dataset, obtaining ground truth communities for the empirical network is challenging. We evaluate the performance of the community detection algorithms on this network indirectly using a location recommendation framework. In the LBSN platform, location recommendation is a standard problem [19] where for one visitor $v \in V_U$, a set of potential locations $L \subseteq V_L$ are recommended for her future visit, based on the interest profile of v. First, we apply location similarity based collaborative filtering [20] on the empirical dataset to obtain a set of K recommended locations for each visitor v and consider it as our ground truth. Next, we apply the multilayer community detection algorithms on $\mathcal{G}_{yelp}$ and obtain K' disjoint communities $C_1, C_2, \ldots, C_{K'}$. As explained in section II, each C_i can be expressed as $\{\{L_U^{C_i}, L_L^{C_i}\}, \{L_{UL}^{C_i}\}\}$ where $L_U^{C_i}$ = ($V_U^{C_i}, E_U^{C_i}$), $L_L^{C_i}$ = ($V_L^{C_i}, E_L^{C_i}$) and $L_{UL}^{C_i}$ = ($V_U^{C_i}$, $V_L^{C_i}$, $E_{UL}^{C_i}$). We claim that for a visitor $v \in V_U^{C_i}$, the set $V_L^{C_i}$ is the recommended locations to visit, following the community detection algorithms. We evaluate this set $V_L^{C_i}$ against collaborative filtering based ground truth.

4) Evaluation metrics & Performance: Suppose, for a visitor v, L_C is the set of locations recommended by the multilayer community detection algorithms and L_R is the set of locations recommended by collaborative filtering. In such a scenario, we calculate the (a) Precision and (b) F1 Score of the recommendations as $\frac{|L_C \cap L_R|}{|L_C|}$ and $\frac{2|L_C \cap L_R|}{|L_C|+|L_R|}$ respectively.

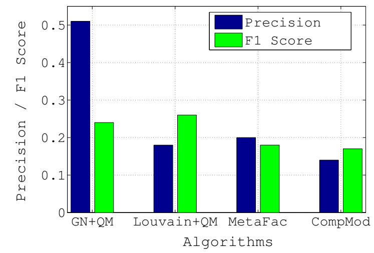


Fig. 8. Precision & F1 Score (averaged over all groups) for communities obtained from different algorithms on Meetup N/W.

Notably, we do not use Recall ($\frac{|L_C \cap L_R|}{|L_R|}$) here to avoid bias towards large communities. In Fig. 7, we plot precision and F1 score (averaged over all visitors) with varying recommendation length K for our algorithms along with the state of the art competing algorithms. Increasing K usually increases the numerator ($|L_C \cap L_R|$) for both the metrics, resulting in an increasing pattern of precision and F1 score. Clearly, both **GN-**Q_M & **Louvain-**Q_M (especially, **GN-**Q_M) outperforms the competing algorithms for almost all K values, demonstrating the elegance of the proposed algorithms.

B. *Meetup dataset*

1) Data description: Meetup, a popular event based social networking (EBSN) platform, facilitates similar minded people to form online groups and organize offline events. We develop a crawler to collect the Meetup [21] data for the city Chicago during a period of 20 months (from August 2015 to March 2017). The crawled dataset contains the details of 5727 Meetup groups, 342, 773 members and 31, 719 events hosted by those groups. The dataset contains the exact time, when a Meetup user joins a specific Meetup group. Additionally, we collect the profile for each Meetup group and its members which are characterized by suite of predefined tags (20 tags for members; 56 for groups) such as ‘web design’, ‘foodie’, ‘cycling’ etc. reflecting their respective preferences. In order to keep the network tractable, first we filter out all the Meetup groups possessing more than 30 members. Next, we select only the groups having at least 10 members joined before the 30^{th} November, 2016 (i.e. within the first 80% of the crawling period) and at least 5 members joining in next 4 months (last 20% of the crawling period). Finally, we obtain 49 Meetup groups and their corresponding 1194 members.

2) Construction of multilayer network $\mathcal{G}_{meetup}$: We construct a multilayer network $\mathcal{G}_{meetup} = \{\{L_M, L_G\}, \{L_{MG}\}\}$ containing all groups and members collected during the entire crawling period. The network contains the following two layers; L_M = (V_M, E_M) is the member layer containing user nodes and their similarity based connections and L_G = (V_G, E_G) denotes the group layer containing Meetup groups as nodes and their respective similarity based connections. In both the layers, similarities between node pairs are computed based on the Jaccard coefficient of their respective tags overlap; we connect the top 33^{rd} percentile of node pairs in L_M & L_G. L_{MG} = (V_M, V_G, E_{MG}) is the bipartite graph connecting a member node $x \in V_M$ with a group node $g \in V_G$ if x is a member of Meetup group g before November 30, 2016.

3) Evaluation procedure: We evaluate the performance of the community detection algorithms with the help of a group recommendation framework. In the EBSN platform, recommending suitable groups to Meetup users is a well studied problem [22], where for user x, a set of groups $g_S \subseteq V_G$ are recommended. To perform the same, we first apply the multilayer community detection algorithms on $\mathcal{G}_{meetup}$ and obtain K disjoint communities $C_1, C_2, \ldots, C_K$. As explained in section II, each C_i can be expressed as $\{\{L_M^{C_i}, L_G^{C_i}\}, \{L_{MG}^{C_i}\}\}$ where $L_M^{C_i} = (V_M^{C_i}, E_M^{C_i})$, $L_G^{C_i} = (V_G^{C_i}, E_G^{C_i})$ and $L_{MG}^{C_i} = (V_M^{C_i}, V_G^{C_i}, E_{MG}^{C_i})$. Suppose, for a group $g \in V_G$, we define $B_g \subseteq V_M$ and $A_g \subseteq V_M$ as the set of members joining g during the training (before Nov. 30) and test period (after Nov. 30) respectively. We claim that for a group $g \in V_G^{C_i}$, $(V_M^{C_i} - B_g)$ is the set of users recommended to join the Meetup group g in the test period.

4) Evaluation metrics & Performance: Following the aforesaid procedure, let M_g be the set of recommended users (by the community detection algorithms), to join the Meetup group g in the test period. A_g provides the ground truth information of group membership in the test period. Hence, we calculate the (a) Precision and (b) F1 Score of the recommendation as $\frac{|M_g \cap A_g|}{|M_g|}$ and $\frac{2|M_g \cap A_g|}{|M_g| + |A_g|}$ respectively. In Fig. 8, we plot the precision and F1 score (averaged over all groups) for proposed algorithms against the competing algorithms. Evidently, **GN-**Q_M outperforms the competing algorithms in terms of precision whereas **Louvain-**Q_M performs the best in terms of F1 score, demonstrating the elegance of our proposed algorithms for the Meetup dataset.

VIII. Conclusion

The major contribution of this paper is to develop Q_M, a novel modularity index for evaluating the quality of communities in multilayer networks. It is experimentally shown that this index overcomes the limitations of the state of the art multilayer modularity definitions [10], [11] and behaves as per expectation in the various topological scenarios. We have demonstrated the utility of Q_M by developing multilayer community detection algorithms **GN-**Q_M and **Louvain-**Q_M, substituting vanilla modularity by Q_M in classical community detection techniques. We have proved the convergence of both the proposed algorithms along with the complexity analysis. In order to examine the modularity Q_M and evaluate the proposed algorithms in a controlled environment, we have developed a methodology to generate multilayer synthetic networks with pre-planted communities. Our algorithms perform observably better than state-of-the-art community detection techniques for wide spectrum of network parameters. Especially, unlike the competing algorithms, their performance remain almost invariant with respect to the fraction of cross layer vs single layer communities present in the network. Finally, communities discovered by our algorithms exhibit practical applications for recommending locations in Yelp as well as recommending groups in Meetup, reflecting the effectiveness of our approach on empirical dataset.

IX. Acknowledgements

This work has been partially supported by the SAP Labs India Doctoral Fellowship program, DST - CNRS funded Indo - French collaborative project titled "Evolving Communities and Information Spreading" and the French National Research Agency contract CODDDE ANR-13-CORD-0017-01.

References

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

[2] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin, "Catastrophic cascade of failures in interdependent networks," *Nature*, vol. 464, no. 7291, pp. 1025–1028, 2010.

[3] Y.-R. Lin, J. Sun, P. Castro, R. Konuru, H. Sundaram, and A. Kelliher, "Metafac: community discovery via relational hypergraph factorization," in *Proceedings of SIGKDD'09*. ACM, 2009, pp. 527–536.

[4] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela, "Community structure in time-dependent, multiscale, and multiplex networks," *science*, vol. 328, no. 5980, pp. 876–878, 2010.

[5] Z. Kuncheva and G. Montana, "Community detection in multiplex networks using locally adaptive random walks," in *Proceedings of ASONAM'15*. ACM, 2015.

[6] M. Berlingerio, F. Pinelli, and F. Calabrese, "Abacus: frequent pattern mining-based community discovery in multidimensional networks," *DMKD*, vol. 27, no. 3, pp. 294–320, 2013.

[7] Y. Sun, J. Tang, J. Han, C. Chen, and M. Gupta, "Co-evolution of multi-typed objects in dynamic star networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 12, pp. 2942–2955, 2014.

[8] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov, "Clustering on multilayer graphs via subspace analysis on grassmann manifolds," *IEEE Transactions on Signal Processing*, vol. 62, no. 4, pp. 905–918, 2014.

[9] W. Cheng, X. Zhang, Z. Guo, Y. Wu, P. F. Sullivan, and W. Wang, "Flexible and robust co-regularized multi-domain graph clustering," in *Proceedings of KDD '13*. NY, USA: ACM, 2013, pp. 320–328.

[10] X. Liu, W. Liu, T. Murata, and K. Wakita, "A framework for community detection in heterogeneous multi-relational networks," *Advances in Complex Systems*, vol. 17, no. 06, p. 1450018, 2014.

[11] J. Song, S. Tang, X. Liu, Y. Gao, H. Yang, and P. Lu, "A modularity-based method reveals mixed modules from chemical-gene heterogeneous network," *PLoS ONE*, vol. 10, no. 4, pp. 1–16, 04 2015.

[12] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical review E*, vol. 78, no. 4, p. 046110, 2008.

[13] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2005, no. 09, p. P09008, 2005.

[14] M. E. Newman, "Modularity and community structure in networks," *Proceedings of the national academy of sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[15] M. J. Barber, "Modularity and community detection in bipartite networks," *Physical Review E*, vol. 76, no. 6, p. 066102, 2007.

[16] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, 2004.

[17] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.

[18] M. E. J. Newman, "Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality," *Phys. Rev. E*, vol. 64, Jun 2001.

[19] S. Gupta, S. Pathak, and B. Mitra, "Complementary usage of tips and reviews for location recommendation in yelp," in *PAKDD*. Springer, 2015, pp. 720–731.

[20] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: an open architecture for collaborative filtering of netnews," in *Proceedings of CSCW'94*. ACM, 1994, pp. 175–186.

[21] S. Pramanik, M. Gundapuneni, S. Pathak, and B. Mitra, "Predicting group success in meetup." in *ICWSM*, 2016, pp. 663–666.

[22] S. Purushotham and C.-C. J. Kuo, "Personalized group recommender systems for location- and event-based social networks," *ACM Trans. Spatial Algorithms Syst.*, vol. 2, no. 4, pp. 16:1–16:29, Nov. 2016.