Comparison of Communities Detection Algorithms for Multiplex

Chuan Wen, Loe, Henrik Jeldtoft Jensen

Department of Mathematics and Centre for Complexity Science
Imperial College London, London, SW7 2AZ, UK

Abstract

Multiplex is a set of graphs on the same vertex set, i.e. $\{G(V, E_1), \ldots, G(V, E_m)\}$. It is a type of generalized graph to model the multiple relationships in a system with parallel edges between vertices. An important application in Network Science is to capture community structures in multiplex as a way to modularize the system. This paper is a literature review and comparative analysis on the existing communities detection algorithms for multiplex.

Keywords: Multiplex, Communities Detection

1. Introduction

Complexity Science studies the collective behaviour of a system of interacting agents, and a graph (network) is often an apt representation of such systems. Traditionally the agents are expressed as the vertices, and an edge between a vertex pair implies that there are interactions between them.

However the modern outlook in Network Science is to generalize the edges to encapsulate the multiple type of relationships between agents. For example in a social network, people are acquainted through work, school, family, etc. This is to preserve the richness of the data and to reveal deeper perspectives of the system. This is known as a *multiplex*.

Multiplex is a natural transition from graphs and many disciplines independently studied this mathematical model for various applications like communities detection. A community refers to a set of vertices that behaves

Email addresses: `c.loe11@imperial.ac.uk` (Chuan Wen, Loe), `h.jensen@imperial.ac.uk` (Henrik Jeldtoft Jensen)

differently from the rest of the system. This modularizes a complex system into simpler representations to form an overview of the information flow.

The first half of this paper is a literature review of the different communities detection algorithms and some theoretical bounds from graph cutting. Next we propose a suite of benchmark multiplexes and similarity metrics to determine the similarity of various communities detection algorithms. Finally we present the empirical results for this paper.

2. Preliminaries

Definition 2.1. ***(Multiplex)*** *A multiplex $\mathcal{G}$ is a finite set of m graphs on the same vertex set V, where every graph $G^i = (V, E_i)$ has a distinct edge set $E_i \subseteq V \times V$, i.e. $\mathcal{G} = \{G^1(V, E_1), \ldots, G^m(V, E_m)\}$.*

There are many synonymous names for multiplex and occasionally they simultaneously used in the same paper. This assists readers to visualize the system through the different descriptions. E.g. a *"multi-layer network"* describes the multiplex as layers of graphs, i.e. layer i implies G^i. The following synonymous names for multiplex will be avoided for the rest of the paper: **Multigraph** [1, 2], **MultiDimensional Network** [3, 4, 5, 6], **Multi-Relational Network** [7, 8, 9], **MultiLayer Network** [10, 11, 12], **PolySocial Networks** [13], **Multi-Modal Network** [14, 15, 16], **Heterogeneous Networks** [17] and **Multiple Networks** [18].

A more complete list of these names can be found in the literature reviews by Boccaletti *et al.* [19] and Kivelä *et al.* [20]. Note that we will use the i^{th} *layer*, *relationship* or *dimension* to refer to the graph $G^i \in \mathcal{G}$. This is to help us to express certain ideas in a more concrete manner.

For example when layers of graphs are stacked on top of each other, there will be vertex pairs with edges "overlapping" each other (Def. 2.2). The distribution to the number of the overlapping edges is an important characteristic of a multiplex, where it is used to classify different multiplex ensembles [21, 22, 23, 24, 25].

Definition 2.2. ***(Overlapping Edges)*** *Let two edges from two different graphs in $\mathcal{G}$ be $e(u, v) \in E_i$ and $e'(u', v') \in E_j$, where $i \neq j$. The edges e and e' overlap if and only if $e = e'$, i.e. $u = u'$ and $v = v'$.*

If there arise ambiguity to the context, we will distinguish the "community" between a multiplex and a monoplex (a graph in a multiplex) as **multiplex-community** and **monoplex-community** respectively. This will avoid confusion when we review the different multiplex communities detection algorithms.

Many of these multiplex-algorithms divide the multiplex problem into independent communities detection problems on the monoplexes. The solutions for these monoplexes, known as **auxiliary-partitions**, provide the supplementary information for the multiplex-algorithm to aggregate. The principal solution from the aggregation forms the **multiplex-partition**, which defines the communities in the multiplex.

3. Definitions of a Multiplex-Community

A *community* vaguely describes a set of interacting agents that collectively behaves differently from its neighboring agents. However there is no universally accepted formal definition, since the concept of a community depends on the problem domain [26, 27, 28]. We categorize this diversity by the communities' *Local Definitions*, *Global Definitions* and *Vertex Similarity.*

3.1. Local Definition

From the assumption that a community has weak interactions with their neighboring vertices, the evaluation of a community can be isolated from the rest of the network. Thus it is sufficient to establish a community from the perspective of the members in the community.

Consider each graphs in a multiplex as an independent mode of communication between the members, e.g. email, telephone, postal, etc. A high quality community should resume high information flow amongst its members when one of the communication modes fails (1 less graph). Hence Berlingerio et al. proposed the *redundancy* of the communities [3] as a measure to the quality of a multiplex-community.

Definition 3.1. ***(Redundancy)*** *Let $W \subseteq V$ be the set of vertices in a multiplex-community and $P \subseteq W \times W$ be the set of vertex pairs in W that are adjacent in ≥ 1 relationship. The set of redundant vertex pairs are $P' \subseteq P$ where vertex pairs in W that are adjacent in ≥ 2 relationships. The redundancy of W is determined by:*

$$\frac{1}{|\mathcal{G}| \times |P|} \sum_{G^i \in \mathcal{G}} \sum_{\{u,v\} \in P'} \delta(u, v, E_i), \tag{1}$$

where $\delta(u, v, E_i) = 1$ (zero otherwise) if $\{u, v\} \in E_i$.

Eq. 1 counts the number of edges in the multiplex-community where their corresponding vertex pairs are adjacent in two or more graphs. The sum is normalized by the theoretical maximum number of edges between all adjacent vertex pairs, i.e. $(|\mathcal{G}| \times |P|)$. The quality of a multiplex-community is determined by how identical the subgraphs (induced by the vertices of the multiplex-community) are across the graphs in the multiplex.

Thus the *redundancy* does not depends on the number of edges in the multiplex-community, i.e. not a necessary condition to its quality. This can lead to an unusual idea that a community can be low in density. For instance a cycle of overlapping edges form a “community” of equal quality as a complete clique of overlapping edges.

3.2. Global Definition

The global measure of a partition considers the quality of the communities *and* their interactions among the communities. For example the *modularity* function by Newman and Girvan measures how different a monoplex-communities are from a random graph (Def. 3.2) [29].

Definition 3.2. ***(Modularity)*** *Let A_{ij} be the adjacency matrix of a graph with $|E|$ edges and k_i is the degree of vertex i. $\delta(v_i, v_j) = 1$ if v_i and v_j are in the same community, otherwise $\delta(v_i, v_j) = 0$. The* Modularity function *measures how far the communities differs from a random graph:*

$$Q = \frac{1}{2|E|} \sum_{ij} \Big(A_{ij} - \frac{k_i k_j}{2|E|} \Big) \delta(v_i, v_j). \tag{2}$$

Given a fixed partition on the vertex set, the modularity on each of the m graphs in the multiplex differs. Thus a good multiplex-communities suggests that all the monoplex-communites in the graphs have high modularity.

To quantify this concept, Tang et. al claims that if there exists latent communities in the multiplex, a subset of the graphs in the multiplex, $\mathcal{G}' \subset \mathcal{G}$ has sufficient information to find these communities [6]. If the hypothesis is true, then the communities detected from $\mathcal{G}'$ should reflect high modularity on the rest of the graphs in the multiplex, i.e. $\mathcal{G} \setminus \mathcal{G}'$.

In the language of machine learning, pick a random graph $G \in \mathcal{G}$ as the test data and let $\mathcal{G}' = \mathcal{G} \setminus G$ be the training data. The multiplex-partition P yielded from a communities detection algorithm on $\mathcal{G}'$ is evaluated with the modularity function on the test data G. P is a good multiplex-partition if the modularity of partition P on the graph G is maximized. This extends the modularity metric for multiplex.

3.3. Vertex Similarity

Two vertices belongs to a Vertex Similarity community if they are similar by some measure. For example the Edge Clustering Coefficient [30] of a vertex pair in a graph measures the (normalized) number of common neighbors between them. A high Edge Clustering Coefficient implies that there are many common neighbors between the vertex pair, thus suggesting that the two vertices should belong in the same community. In the extension for multiplex, Brodka et. al introduced Cross-Layer Edge Clustering Coefficient (CLECC) [10].

Definition 3.3. ***(Cross-Layer Edge Clustering Coefficient)*** *Given a parameter α, the MIN-Neighbors of vertex v, $N(v,\alpha)$ are the set of vertices that are adjacent to v in at least α graphs. The Cross-Layer Edge Clustering Coefficient of two vertices $u, v \in V$ measures the ratio of their common neighbors to all their neighbors.*

$$CLECC(u,v,\alpha) = \frac{|N(u,\alpha) \cap N(v,\alpha)|}{|N(u,\alpha) \cup N(v,\alpha) \setminus \{u,v\}|}. \tag{3}$$

A pair of vertices in a multiplex of social networks with low CLECC suggests that the individuals do not share a common clique of friends through at least α social networks. Thus it is unlikely that they form a community.

3.4. Densely Connected Community Cores

Communities detection is the process to partition the vertices such that every vertex belongs in at least a community. However there are cases where one just desires some substructures in a multiplex with certain properties. For example a *Dense Connected Community Core* is the set of vertices such that they are in the same community for all the auxiliary-partitions [31].

4. Theoretical Bounds

The Max-Cut problem finds a partition of a graph such that the number of edges induced across the clusters are maximized. Thus the same partition over the complement graph minimizes the number of edges between the clusters. This is known as the Balanced-Min-Cut problem and it is closely related to our communities detection problem, where the number of edges induced between the communities are minimized.

Therefore to extend our understanding for multiplex, we begin with a known result for the Max-Cut problem. It allows us to prove a corollary for the Balanced-Min-Cut problem on multiplex.

4.1. Maximum Cut Problem on Multiplex

Theorem 4.1. *Consider graph $G^1, \dots, G^m$ on the same vertex set V. There exists a k-partition of V into $k \geq 2$ communities $C_1, \dots, C_k$ such that for all $i = 1, \dots, m$ and sufficiently large $|V|$ [32]:*

$$\#\ \textit{edges cut in } G^i \geq \frac{(k-1)|E_i|}{k} - \sqrt{2m|E_i|/k}. \tag{4}$$

Eq. 4 can be improved for cases where the maximum degree is bounded [32] or when $k = m = 2$ [33]. Since the solution for Max-Cut on graphs is NP-complete, the extension to simultaneously Max-Cut all the graphs in a multiplex is naturally NP-complete too. Thus this also implies that the following balanced minimum bisection problem is NP-complete too [34].

4.2. Balanced Minimum Cut Problem on Multiplex

Corollary 4.1. *Consider graph $G^1, \dots, G^m$ on the same vertex set V. There exists a k-partition of V into $k \geq 2$ equal-sized communities $C_1, \dots, C_k$ (i.e. $|C_i| \approx |C_j|$) such that for all $i = 1, \dots, m$ and sufficiently large $|V|$:*

$$\#\ \textit{edges cut in } G^i \leq \binom{n}{2} - \frac{(k-1)|\bar{E}|}{k} - \sqrt{2m|\bar{E}|/k}, \tag{5}$$

where $|\bar{E}| = \binom{n}{2} - |E_i|$.

Proof. Let $\bar{G}^i$ be the complement graph of G^i, and its edge set is denoted by $\bar{E}_i$. Since the maximum number of edges in a graph is $\binom{n}{2}$, hence $|\bar{E}_i| = \binom{n}{2} - |E_i|$. Apply (Max-Cut) Theorem 4.1 on the set of complement graphs $\bar{G}_i$ and substitutes $|\bar{E}|_i$ into the result, the expression in the corollary follows. The proof in Theorem 4.1 ensures that the communities are equal in size. □

A partition that fulfills Eq. 5 is not necessary a good community defined in Section 3, vice versa. However the edges induced between partition classes are often perceived as bottlenecks when information flows through the network/multiplex. They are similar to the bridges between cities and communities. Therefore Communities Detection Algorithms tend to minimize the number of edges between different communities.

Unfortunately none of the algorithms in section 5 guarantees communities of equal size, thus there is no reasonable way to measures the quality of the algorithms with Eq. 4.1. However in the cases it can be applied, a solution that is greater than the bound implies that the algorithm performs worse than randomization. This is due to the proof in Theorem 4.1.

5. Communities Detection Algorithms for Multiplex

The general strategy for existing communities detection algorithm for multiplex is to extract features from the multiplex and reduce the problem to a familiar representation. In solving the reduced representation, the multiplex-communities are then deduced from the auxiliary solutions of the reduced problems.

Thus many multiplex algorithms rely on existing monoplex-communities detection algorithms to get the auxiliary-partitions for the interim steps. The choice of algorithms is often independent of their extension for multiplex, and hence any communities detection algorithm in theory can be chosen to solve the interim steps. In this paper our experiments used Louvain Algorithm to generate the auxiliary-partitions.

5.1. Projection

The naive method is to projected the multiplex into a weighted graph. I.e. let A^i be the adjacency matrix of $G^i \in \mathcal{G}$. The adjacency matrix of the weighted projection of $\mathcal{G}$ is given by $\bar{A} = \frac{1}{m}\sum_{i=1}^{m} A^i$. We will call this the "Projection-Average" of multiplex.

It was been independently proposed as a baseline for more sophisticated multiplex algorithms as the performance is often "sub-par" [3, 4, 35, 7, 5]. In our experiments we will compare this with the unweighted variant, that is the "Projection-Binary" of a multiplex, i.e. $G(V, E_1 \cup \ldots \cup E_m)$.

An alternative weight assignment between vertex pair is to consider the connectivity of their neighbors, where a high ratio of common neighbors implies stronger ties [3]. This is based on the idea that members of the same community tend to interact over the same subset of relations, which was independently proposed by Brodka et. al in Def. 3.3 [10]. This alternative will be known as "Projection-Neighbors".

5.2. Consensus Clustering

The previous strategy aggregates the graphs first, and then it performs the communities detection algorithm over the resultant graph. It is a poor strategy as it neglects the rich information of the dimensions [6]. Therefore the Consensus Clustering strategy is to first apply the communities detection algorithm on the graphs separately as auxiliary partitions, and then the principal clustering (multiplex communities) is derived by aggregating these auxiliary partitions in a meaningful manner.

The key concept behind consensus clustering is to measure the frequency with which two vertices are found in the same community among the auxiliary partitions. Vertices that are frequently in the same monoplex-community are more likely to be in the same multiplex-community. Therefore the communities detection algorithm on the individual graphs on the multiplex determines the structural properties of multiplex-communities, whereas Consensus Clustering determines the relational properties of the multiplex-communities.

5.2.1. Frequent Closed Itemsets Mining

Data-mining is to find a set of items that occurs frequently together in a series of transactions. For example items like milk, cereal and fruits are frequently bought together in supermarkets based on a series of sales transactions. These sets are known as *itemsets*. Berlingerio et. al translates the Consensus Clustering of the auxiliary-partitions as a data-mining problem to discover multiplex-communities [36].

The vertices in the multiplex defines the $|V|$ *transactions* for the data-mining, and the items are tuples (c, d) where the respective vertex belongs in monoplex-community c in dimension d. For example suppose vertex v_i belongs to monoplex-communities c_1, c_5 and c_2 in dimensions d_1, d_2 and d_3 respectively. The i^{th} transaction is the set of items $\{(c_1, d_1), (c_5, d_2), (c_2, d_3)\}$. Therefore when we use data-mining methods like *Frequent Closed Itemsets Mining*, we are able to identify the frequent (relative to a predefined threshold) itemsets as multiplex-communities.

For example each vertex is a customer's transaction in a supermarket, and a community is a target market that the supermarket wants to discover. It is only meaningful if the target market is sufficiently large, and thus we need to defined the minimum community size (e.g. 10). In this case a customer's transaction is his auxiliary-communities membership. Therefore Frequent Closed Itemsets Mining will extract a multiplex-community on at least e.g. 10 vertices with which each customer's transaction is a subset of the target market's itemsets.

5.2.2. Cluster-based Similarity Partitioning Algorithm

Cluster-based Similarity Partitioning Algorithm averages the number of instances vertex pairs are in the same auxiliary-communities. For example in a multiplex with 5 dimensions, if there are 3 instances where vertices v_i and v_j are in the same auxiliary-community, then the similarity value of vertex pair (v_i, v_j) is 3/5.

Once the similarity is measured for all the vertex pairs, the principal cluster is determined with k-means clustering — vertices with the closest similarity at each iteration are grouped together. Therefore vertex pairs that are frequently in the same auxiliary-communities will have high similarity value, and hence more likely to be clustered together in the same principal-community. This is known as Partition Integration by Tang et. al [6].

5.2.3. Generalized Canonical Correlations

Each of the auxiliary-partitions maps the vertices as points in a l-dimensional (this dimension is independent to the dimensions of a multiplex) Euclidean space. The points are positioned in a way that the shorter the shortest path between two vertices are, the closer they are in the Euclidean space. One of such mapping can be achieved by concatenating the top eigenvectors of the adjacency matrix. Thus given d graphs in a multiplex, there are d structural feature matrices S^i of size $l \times n$ where the column in each matrix is the position of a vertex in the l-dimensional Euclidean space.

Tang et. al wants to aggregate the structural feature matrices to a principal structural feature matrix $\bar{S}$ such that the principal partition can be determined from $\bar{S}$ [6]. The "average" $\bar{S} = \frac{1}{d}\sum_{i=1}^{d} S^{(i)}$ however does not result in sensible principal structural feature matrix since the matrix elements between $S^{(i)}$ and $S^{(j)}$ are independent.

A solution to fix this problem is to transform the $S^{(i)}$ such that they are in the same space and their "average" is sensible. That is the same vertex in the d different Euclidean spaces are aligned in the same point in a Euclidean space. Specifically we need a set of linear transformations w_i such that they maximize the pairwise correlations of the $S^{(i)}$, and Generalized Canonical Correlations Analysis is one of such standard statistical tools [37]. This allows us to "average" the structures in a more sensible way:

$$\bar{S} = \frac{1}{d}\sum_{i=1}^{d} S^{(i)} w_i. \tag{6}$$

Finally the principal partition is determined via k-mean clustering of the principal feature matrix $\bar{S}$.

5.3. Bridge Detection

A bridge in a graph refers to an edge with high information flow, like the busy roads between two cities, where the absence of these roads separates the cities into isolated communities. One way to do this is to project the multiplex $\mathcal{G}$ to a weighted network and determine the bridges from the projection. Alternatively one can remove them by the definition of a multiplex-bridge to get the desired partitions.

In social networks, strong edge ties are desirable within the communities. Hence to identify weak ties between vertex pairs, Brodka et. al proposed CLECC (Eq. 3) as a measure. At each iteration, all connected vertex pairs are recomputed and the pair with the lowest CLECC score will be disconnected in all the graphs. The algorithm halts when the desired number of communities (components) are yielded greedily [10]. This is the same strategy presented by Girvan and Newman, where the bridges of a graph were identified by their betweenness centrality score [38].

5.4. Tensor Decomposition

Algebraic Graph Theory is a branch of Graph Theory where algebraic methods like linear algebra are used to solve problems on graph. Hence the natural representation for a multiplex is a 3^{rd}-order tensor (as a multidimensional array) instead of a matrix (2^{nd}-order tensor). The set of m graphs in a multiplex is a set of m $n \times n$ adjacency matrices, which can be represented as a $m \times n \times n$ multidimensional array (tensor) [8]. This allows us to leverage on the available tensor arithmetics like tensor decomposition.

Tensor decompositions are analogues to the singular value decomposition and 'Lower Upper' decomposition in matrices, where they express the tensor into simpler components. For example a PARAFAC tensor decomposition [39] is the rank-k approximation of a tensor $\mathcal{T}$ as a sum of rank-one tensors (vectors $\bar{u}^{(i)}$, $\bar{v}^{(i)}$ and $\bar{w}^{(i)}$), i.e.:

$$\mathcal{T} \approx \sum_{i=1}^{k} \bar{u}^{(i)} \circ \bar{v}^{(i)} \circ \bar{w}^{(i)}. \tag{7}$$

where $\bar{a} \circ \bar{b}$ denotes the vector outer product. The components in the i^{th} factor, $\bar{u}^{(i)}$, $\bar{v}^{(i)}$ and $\bar{w}^{(i)}$, suggest that there are strong ties (possibly a cluster/community) between the elements in $\bar{u}^{(i)}$ and $\bar{v}^{(i)}$ via the dimension in the top component in $\bar{w}^{(i)}$.

For instance suppose the j^{th} element in $\bar{w}^{(i)}$ is the most largest element. This suggests that in the i^{th} community, the top 10 (or any predefined threshold) elements in $\bar{u}^{(i)}$ are in the same cluster as the top 10 elements in $\bar{v}^{(i)}$ via the j^{th} dimension/relationship [40, 41, 42, 43].

6. Benchmark Multiplex

An *Erdős-Rényi Graph* is a graph where vertex pairs are connected with a fixed probability [44]. The random nature of this construction usually does not have meaningful communities structures in them. Hence it is not an good benchmark graph for Communities Detection Algorithms.

A benchmark graph should be similar to the Girvan and Newman Model where some random edges are induced between a set of dense subgraphs (as communities) to form a single connected component/graph. The set of dense subgraphs acts as "ground-truth" communities of the graph for a Communities Detection Algorithm to discover. The goal of this section is to design similar benchmarks for multiplex.

The main challenge is that there is not yet a universally accepted definition of a good multiplex community. Hence there is no methodology for us to construct a benchmark such that it does not favor certain algorithms. Therefore the objective of the following benchmark graphs is **to study the correlations between these multiplex-communities detection algorithms**. This allows us to use a collection of highly uncorrelated algorithms to study different perspectives of a multiplex-community.

6.1. Unstructured Synthetic Random Multiplex

The simplest construction of a random multiplex is to generate a set of independent graphs on the same vertex set. However many algorithms are based on the observations of real world multiplexes and hence will not yield interesting results on such random construction. We name such random multiplexes as *Unstructured Synthetic Random Multiplex* (USRM), and they are analogous to Erdős-Rényi Graphs where Communities Detection Algorithms should not find any meaningful communities in them. Table 1 lists all six combinations of Erdős-Rényi, Watts Strogatz [45] and Barabási-Albert graphs [46] as benchmark USRMs.

For higher dimensional USRMs, we will only consider the combinations of Watts Strogatz and Barabási-Albert graphs as their projections exhibit real-world characteristics like high clustering coefficient and power-law like degree distribution [47]. Furthermore in the experiments (later) we show that only USRMs with Watts Strogatz graphs yield meaningful communities. Hence it is reasonable to include at least one Watts Strogatz graph for multiplexes with > 2 relationships. Thus let USRM-Rd_i refer to a multiplex on d relationships with i Watts Strogatz graphs and $(d - i)$ Barabási-Albert graphs.

Name	Graph 1	Graph 2
USRM1	Erdős-Rényi	Erdős-Rényi
USRM2	Erdős-Rényi	Watts Strogatz
USRM3	Erdős-Rényi	Barabási-Albert
USRM4	Watts Strogatz	Watts Strogatz
USRM5	Watts Strogatz	Barabási-Albert
USRM6	Barabási-Albert	Barabási-Albert

Table 1: Different combinations of USRM benchmark

6.2. Structured Synthetic Random Multiplex

This construction is similar to the Girvan and Newman Model where independently generated communities are connected in a way such that the "ground-truth" communities remains. However we saw above that there are different perspectives of multiplex communities and we want to encapsulate these ideas into a single multiplex benchmark.

Structured Synthetic Random Multiplex (SSRM) is a construction where different definitions of high quality multiplex-communities exists in distinct multiplex-partitions. However at the same time each partition is "less-than-ideal" quality by the other definitions of multiplex-communities.

We begin with a collection of high quality multiplex-communities as defined in section 3. Next we modify these multiplex-communities such that it remains good in one of the three multiplex-communities definitions **and** poor quality by the others. Lastly combine these multiplex-communities into a single multiplex as our SSRM benchmark (Fig. 1).

Fig. 1 shows $\{[c_1, c_2], [c_3, c_4]\}$ as a partition with 2 communities where clusters 1 and 2 form a community and cluster 3 and 4 form the second community. This partition has high redundancy, low modularity and low CLECC multiplex-communities. Whereas partition $\{[c_1, c_3], [c_2, c_4]\}$ has high modularity communities, but low in the remaining metrics. Finally $\{[c_2, c_3], [c_1, c_4]\}$ is the partition where communities have high CLECC vertex pairs. This construction expresses the multi-perspectives of multiplex communities.

6.2.1. High Modularity, Low Redundancy & CLECC Multiplex-Communities

To create a high modularity multiplex-community, we begin with c_1 and c_3 (cluster 1 and 3) as cliques in both dimensions. For these clusters to be low in redundancy, we have to remove some edges in both clusters such that there are very few overlapping edges in the clusters while maintaining high

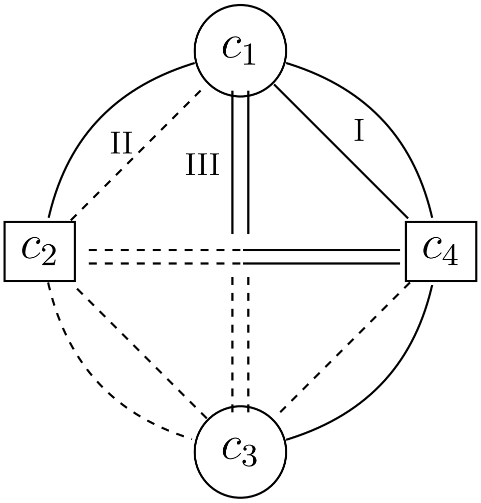


Figure 1: To aid visualization, the edges in this two dimensional SSRM are drawn with solid and dashed lines. Let clusters c_1 and c_3 be dense subgraphs where there are more solid edges than dashed edges. Similarly c_2 and c_4 are dense subgraphs with more dashed edges than solid edges. **I:** The solid edges between c_1 & c_4 implies that there are only solid edges between them. This applies the same to the dashed edges between c_2 & c_3. **II:** All the edges between c_1 & c_2 (or c_3 & c_4) overlap. **III:** None of the edges between c_1 & c_3 (or c_2 & c_4) overlaps. We denote $\{[c_1, c_2], [c_3, c_4]\}$ as a partition with 2 communities where clusters 1 and 2 form a community and cluster 3 and 4 form the second community. This partition has high redundancy, low modularity and low CLECC multiplex-communities.

modularity. This is done by making the first dimension graph in both clusters to be the complement graph in the second dimension.

Next, add edges between c_1 and c_3 such that the resultant cluster is a connected component. We denote $[c_1, c_3]$ as the component that connects c_1 and c_3. To maintain a low redundancy, the new edges cannot overlap.

Finally tweak the clusters such that the CLECC score is low between a significant number of the vertex pairs in the combined component of c_1 and c_3. Specifically we want the vertex pairs connected by the new edges in the previous step to have low CLECC scores. This is possible if c_1 has more edges in the first dimension whereas c_3 has more edges in the second dimensions. In doing so the neighbors of the vertex in c_1 will be significantly different from the neighbors of vertex in c_3, thus a low CLECC score.

The same construction applies to c_2 and c_4 (cluster 2 and 4), where they are similar to c_3 and c_1 respectively.

6.2.2. High CLECC, Low Modularity & Redundancy Multiplex-Communities

Since all the clusters do not have overlapping edges, the redundancy remains low for the multiplex. Therefore the first step is to increase the CLECC of c_1 and c_4, and apply the same construction for c_2 and c_3.

Since c_1 and c_4 are similar from the previous subsection, the neighbors of any vertex in each cluster will be similar too. Therefore by adding new edges between c_1 and c_4 will increase the CLECC score. However these new edges should *only* be drawn in the first dimension, since it is the dominant dimension in $[c_1, c_4]$. This simultaneously reduces the modularity of $[c_1, c_4]$ in the second dimension, since the clusters are not connected and the graph in the second dimension is sparse. This gives a low modularity for multiplex communities while maintaining the high CLECC score.

The construction is similar for c_2 and c_3, expect that only edges in the second dimension connects the clusters together.

6.2.3. High Redundancy, Low Modularity & CLECC Multiplex-Communities

Given $[c_1, c_3]$ have low CLECC score, the same score should apply for $[c_1, c_2]$ since c_2 and c_3 are similar. The main goal is to connect c_1 and c_2 such that the redundancy of $[c_1, c_2]$ is high. Redundancy is measured by Eq. 1, where it counts the number of edges that overlaps. Since there is no overlapping edges at this point of the construction, the simplest way to increase the redundancy is to add new overlapping edges between clusters to form the components $[c_1, c_2]$ and $[c_3, c_4]$.

Although $[c_1, c_2]$ and $[c_3, c_4]$ have relatively high redundancy as compare to other partition in the multiplex, it can still have lower redundancy than a random community in USRM1. To nudge the redundancy higher, it is necessary to add new edges such that there are overlaps in the four clusters. However this might increase the modularity of $[c_1, c_2]$ which we want to avoid. Therefore this final step has to be done incrementally.

6.2.4. Evaluation of the different ground truth partitions

There are three different “ground-truth” partitions, i.e. $\{[c_1, c_2], [c_3, c_4]\}$, $\{[c_2, c_3], [c_1, c_4]\}$ and $\{[c_1, c_3], [c_2, c_4]\}$, where they each represents a different “ideal-partition”. However simultaneously they are “less-than-ideal” from the perspective of the other metrics. Although some of these metrics are correlated (from our experiments) in general, the ground-truth partitions of SSRM show that these metrics are each able to capture essential aspects of the community structure.

	Redundancy	*CLECC	**Modularity
$[c_1, c_2]$	**0.0492**	0.1142	-0.0287
$[c_3, c_4]$	**0.0537**	0.1087	-0.0332
$[c_2, c_3]$	0	**0.1541**	0.007
$[c_1, c_4]$	0	**0.1642**	0.012
$[c_1, c_3]$	0	0.1113	**0.0317**
$[c_2, c_4]$	0	0.1083	**0.0245**
Random	0.0217	0.1056	0.0033

Table 2: The rows are paired up to infer communities in the same partition. E.g. the first two rows are communities of the partition $\{[c_1, c_2], [c_3, c_4]\}$. The redundancy and CLECC score of community $[c_1, c_2]$ are 0.0492 and 0.1142 respectively. ***CLECC:** The CLECC score is the average CLECC score between all vertex pairs in the community. ****Modularity:** The two values in the partition refers to the modularity of the two graphs in the multiplex. E.g. partition $\{[c_1, c_2], [c_3, c_4]\}$ has modularity -0.0287 and -0.0332 for the first and second dimensions of SSRM respectively. A partition is "less-than-ideal" if its measurement is closer to a random partition (last row) than the maximum (bold).

For example Table 2 shows that the partition $\{[c_1, c_2], [c_3, c_4]\}$ has communities with the maximum redundancy. However it has multi-modularity and CLECC scores similar to a random partition, suggesting that it is "less-than-ideal" relative to other metrics.

6.3. Real World Multiplex

6.3.1. Youtube Social Network

Youtube is a video sharing website that allows interactions between the video creators and their viewers. Tang et al. collected 15,088 active users to form a multiplex with 5 relationships where two users are connected if 1) they are in the contact list of each other; 2) their contact list overlaps; 3) they subscribe to the same user's channel; 4) they have subscription from a common user; 5) and they share common favorite videos. [6].

6.3.2. Transportation Multiplex (Multimodal Network)

Cardillo et al. constructed an air traffic multiplex from the data of European Air Transportation (EAT) Network with 450 airports as vertices [48]. An edge is drawn between two vertices if there is a direct flight between them. Each of the 37 distinct airlines in the EAT Network forms a relationship between airports.

7. Comparing Partitions

Normalized Mutual Information [49] and Omega Index [50] will be used to compare the partitions. Although NMI is a popular choice in the literature, only Omega Index allows us to measure solutions where the communities overlaps and accepts vertices with no community membership. Such cases are common in algorithms like Frequent Closed Itemsets Mining and PARAFAC.

7.1. Normalized Mutual Information (NMI)

Let $H(\mathcal{A}) = -\sum_k P(a_k) \log P(a_k)$ be the Shannon entropy where $P(a_k)$ is the probability of a random vertex is in the k^{th} community of partition $\mathcal{A}$. Mutual Information $I(\mathcal{A};\mathcal{B})$ measures the information of the communities-membership of all vertex-pairs in A given the communities-membership in B, vice versa. Roughly speaking given $\mathcal{B}$, how well can we guess that a vertex pair is in the same community in $\mathcal{A}$? Formally this is defined as:

$$I(\mathcal{A};\mathcal{B}) = \sum_j \sum_k P(a_k \cap b_j) \log \frac{P(a_k \cap b_j)}{P(a_k)P(b_j)}, \tag{8}$$

where $P(a_k \cap b_j)$ is the probability that a random vertex is both in k^{th} and j^{th} communities. Basically the larger the intersection of the k^{th} and j^{th} communities of $\mathcal{A}$ and $\mathcal{B}$ respectively is, the higher the Mutual Information. Finally we normalize the Mutual Information score between $[0,1]$ so that NMI=1 implying identical partition:

$$\text{NMI}(\mathcal{A},\mathcal{B}) = \frac{I(\mathcal{A};\mathcal{B})}{[H(\mathcal{A}) + H(\mathcal{B})]/2.} \tag{9}$$

7.2. Omega Index

The unadjusted Omega Index averages the number of vertex pairs that are in the same number of communities. Such vertex pairs are known to be in *agreement*. Consider the case with two partitions $\mathcal{A}$ and $\mathcal{B}$, and the number of communities in them are $|\mathcal{A}|$ and $|\mathcal{B}|$ respectively. The function $t_j(\mathcal{A})$ returns the set of vertex pairs that appears exactly in $j \geq 0$ overlapping communities in $\mathcal{A}$. Thus the unadjusted Omega Index:

$$\omega_u(\mathcal{A};\mathcal{B}) = \frac{1}{\binom{n}{2}} \sum_{j=0}^{max(|\mathcal{A}|,|\mathcal{B}|)} |t_j(\mathcal{A}) \cap t_j(\mathcal{B})|. \tag{10}$$

To account for vertex pairs that are allocated into the same communities by chance, we have to subtract it from expected omega index of a null model:

$$\omega_e(\mathcal{A};\mathcal{B}) = \frac{1}{\binom{n}{2}^2} \sum_{j=0}^{max(|\mathcal{A}|,|\mathcal{B}|)} |t_j(\mathcal{A})| \cdot |t_j(\mathcal{B})|. \tag{11}$$

Finally normalize the Omega Index:

$$\omega(\mathcal{A};\mathcal{B}) = \frac{\omega_u(\mathcal{A};\mathcal{B}) - \omega_e(\mathcal{A};\mathcal{B})}{1 - \omega_e(\mathcal{A};\mathcal{B})}. \tag{12}$$

Identical partitions have a score of 1 and negative Omega Index simply means there are less agreement than pure stochastic coincidence would expect (i.e. not similar). Omega Index can also be used to measure non-overlapping partitions. In such case the metric is then reduced to the Adjusted Rand Index, which is also a popular alternative to NMI [51].

7.3. Notations For Empirical Results

To simplify the plots exhibiting the results from our experiments, we will use some shorthands to denote the algorithms and partitions. For communities detection algorithms, we use $\mathcal{A}$ and $\mathcal{P}$ to denote “Algorithm” and “SSRM Partition” respectively (Table 3).

Notation	Description
$\mathcal{A}_1$	Projection-Binary
$\mathcal{A}_2$	Projection-Average
$\mathcal{A}_3$	Projection-Neighbors
$\mathcal{A}_4$	Cluster-based Similarity Partition Algorithm
$\mathcal{A}_5$	Generalized Canonical Correlations
$\mathcal{A}_6$	CLECC Bridge Detection
$\mathcal{A}_7$	Frequent Closed Itemsets Mining
$\mathcal{A}_8$	PARAFAC, Tensor Decomposition
$\mathcal{P}_1$	$\{[c_1, c_2], [c_3, c_4]\}$ of SSRM
$\mathcal{P}_2$	$\{[c_2, c_3], [c_1, c_4]\}$ of SSRM
$\mathcal{P}_3$	$\{[c_1, c_3], [c_2, c_4]\}$ of SSRM

Table 3: Shorthands for the different algorithms and “ground-truth” communities.

8. Comparison of Multiplex-Communities Detection Algorithms

8.1. Benchmark Parameters

To thoroughly compare the algorithms, the size of the vertex set for the synthetic networks is $|V| = 128$. This is to allow us to run 100 trials on each benchmark within manageable time as algorithms like Cluster-based Similarity Partition can be computationally inefficient ($\sim O(|V|^2)$).

Moreover it is important that the number of edges are equal for all graphs so that none of them dominates the interactions of the multiplex. To ensure that the Erdős-Rényi graphs are connected with high probability, the probability that a vertex pair is connected has to be $> \ln |V|/|V|$. Therefore the number of edges in every graph is $\binom{128}{2} 2 \ln 128/128 \approx 616$.

8.2. Algorithm Parameters

Some of the multiplex-communities detection algorithms require additional parameters which are independent to the network. For example Frequent Closed Itemsets Mining is parameterized by the minimum size of a community. In the case for the synthetic multiplexes on 128 vertices, the minimum size of a community is 10. This is derived by fine-tuning the parameter such that there are ≥ 2 communities in the solution. Similarly for the real-world multiplexes, the size of the community is ≥ 50.

The main parameter for CLECC Bridge Detection is the α in Def. 3.3, which defines a vertex's set of neighbors where they are adjacent in at least α graphs. Despite experimenting with different α, there is no consistent value such that it will always yield meaningful communities for all 100 trials on any given synthetic benchmark. Thus in the experiments, we simply set $\alpha = \lceil |\mathcal{G}|/2 \rceil$ for the CLECC score in Eq. 3.

PARAFAC is parameterized by the rank of the approximation and the threshold to define the top x components in the rank-one tensors. This is usually done by manually fine-tuning [40, 41, 42, 43], however it is infeasible for our numerous random trials. Thus x is chosen for every rank-one tensor such that the difference between the x^{th} and $x + 1^{th}$ element is greater than the average difference among the elements in the rank-one tensor.

Lastly the final step for Cluster-based Similarity Partition and Generalized Canonical Correlations is k-mean clustering. Since these algorithms maximize the modularity of every graph in the multiplex, the best values of k is chosen such that it maximizes the multi-modularity (section 3.2).

8.3. Unstructured Synthetic Random Multiplex

Figure 2 shows the Omega Index of all pairwise multiplex communities detection algorithms for the USRM benchmarks. The last 13 boxplots on the right are the pairwise comparisons with overlapping-communities algorithms, i.e. $\mathcal{A}_7$ and $\mathcal{A}_8$ (Frequent Closed Itemsets Mining and PARAFAC).

The first observation is that $\mathcal{A}_8$ (PARAFAC) is not similar to any of the algorithms. One of the reasons is that it is hard to choose the right parameters (without manual fine-tuning) for PARAFAC such that it is consistently similar to any of the other algorithms. Furthermore there is no strong reason that such parameter exists.

Overlapping-communities algorithm $\mathcal{A}_7$ (Frequent Closed Itemsets Mining) is similar to a class of non-overlapping algorithms, i.e. $\mathcal{A}_1$ to $\mathcal{A}_4$. Specifically Frequent Closed Itemsets Mining is similar to the class of Projection algorithms ($\mathcal{A}_1$ to $\mathcal{A}_3$). In fact the Omega Index for all pairwise comparisons of $\mathcal{F} = \{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4, \mathcal{A}_7\}$ is generally greater than the other algorithm pairs. This is supported by their NMI scores in Fig. 3.

However the family of algorithms $\mathcal{F}$ has low pairwise Omega Index and NMI scores for USRM 1, 3 and 6. There is no fundamental reasons to why the class of projection algorithms ($\mathcal{A}_1$ to $\mathcal{A}_3$) should produce non similar communities, therefore we can deduce that USRM 1, 3 and 6 are not good multiplexes for benchmarks.

The reason is that USRM1 is the combination of two Erdős-Rényi graphs, thus there is no community structure. Whereas USRM 3 and 6 are the combinations of Barabási-Albert graph with Erdős-Rényi and Barabási-Albert respectively. Although Barabási-Albert has structural properties, it has low Clustering Coefficient (similar to Erdős-Rényi), which measures the tendency that vertices cluster together. Therefore the vertices in USRM 3 and 6 do not form communities.

To increase the clustering coefficient of a multiplex, we can introduce a Watts Strogatz graph to the system. We can approximate the clustering coefficient of the projection of such multiplexes [47] and deduce the tendency that communities exists. This allows us to observe interesting relationships between the algorithms from benchmark multiplexes like USRM 2, 4 and 5.

For example USRM 5 in Fig. 3 shows that the similarity range of $\mathcal{F}$ with $\mathcal{A}_6$ (CLECC Bridge Detection) is wide. This is more apparent when we study their relationship with SSRM benchmark.

However in higher dimension, the observations are different where Fig. 4 shows the Omega Index of various parameters of USRM-Rd_i. Firstly the

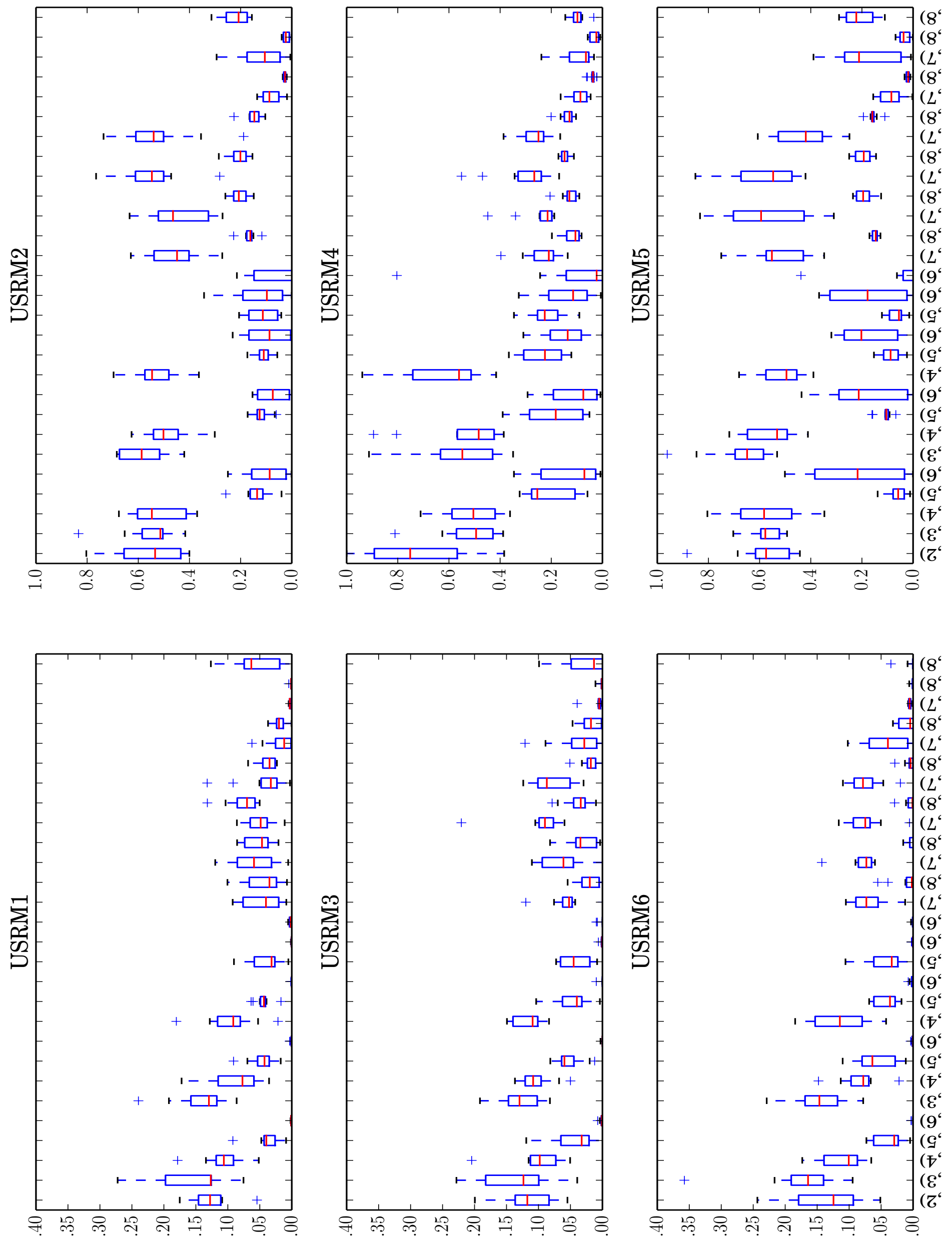


Figure 2: The Omega Index of all pairwise multiplex-communities detection algorithms for the different benchmark USRM. The tuple (i, j) on the x-axis refers to the pairwise comparison of $\mathcal{A}_i$ and $\mathcal{A}_j$. The tuples are arranged such that the comparisons with overlapping-communities algorithms (13 tuples) are placed on the right. Note that the scale of the figures on the left is different from the scale on the right.

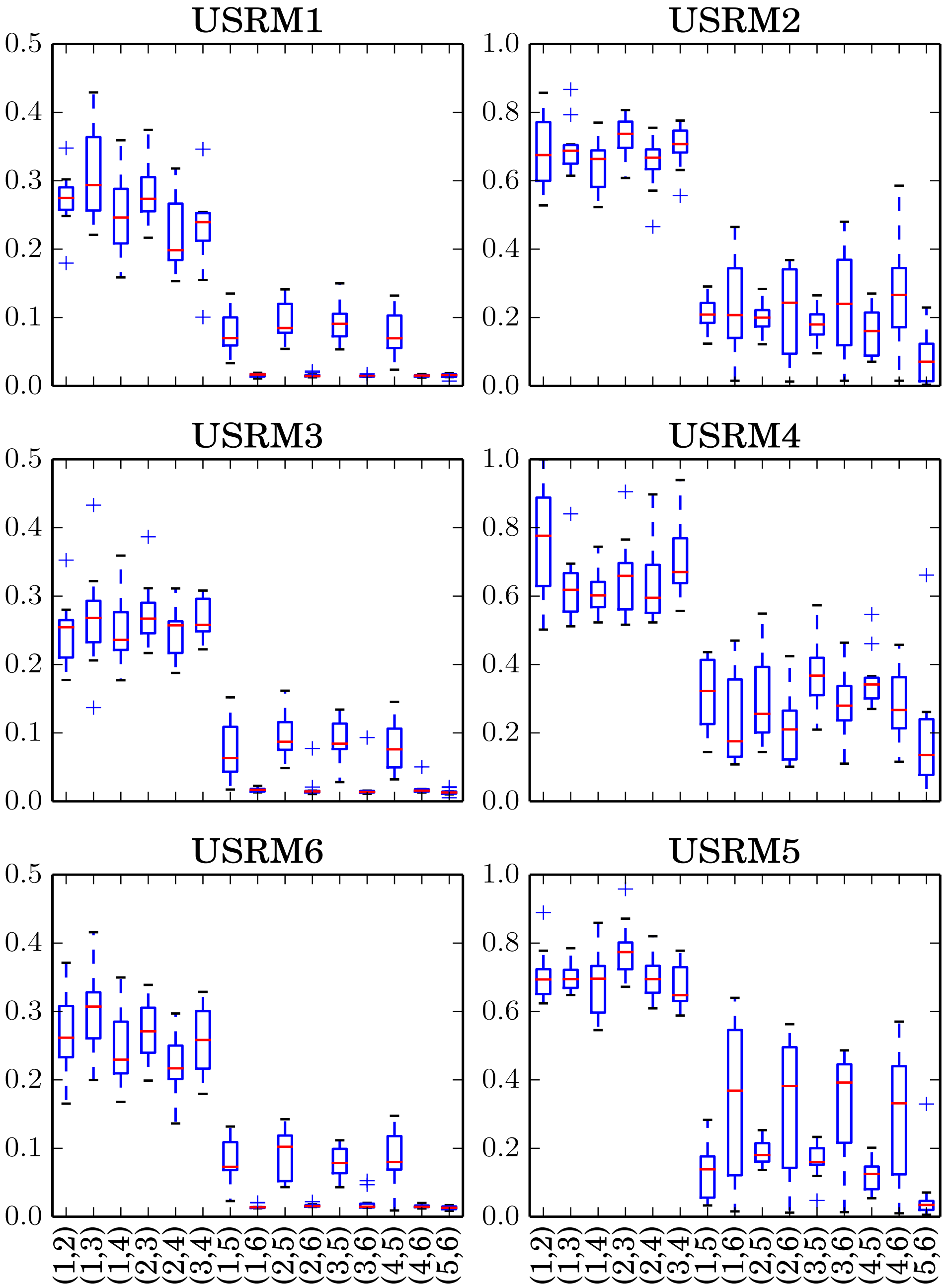


Figure 3: The NMI scores of all pairwise non-overlapping multiplex-communities detection algorithms for USRM benchmarks. The tuples are arranged such that pairwise comparisons of $\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4\}$ are grouped to the left of the boxplots. The "interesting" figures USRM 2, 4 and 5 are placed to the right for comparison.

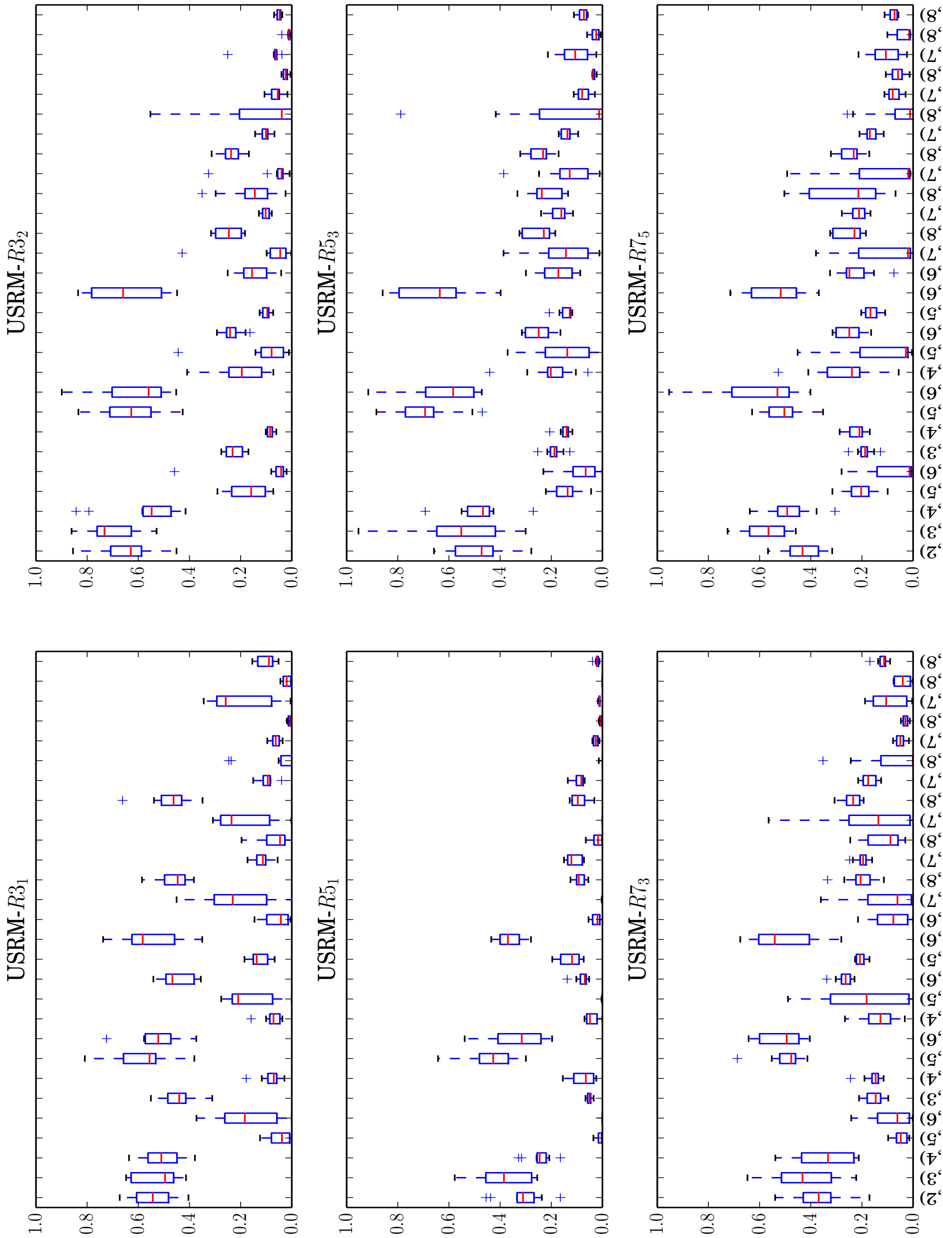


Figure 4: The Omega Index of all pairwise multiplex-communities detection algorithms on USRM-R3_1, USRM-R3_2, USRM-R5_1, USRM-R5_3, USRM-R7_3 and USRM-R7_5. The behavior of the algorithms is drastically different from the 2-dimensional cases.

family of algorithms $\mathcal{F} = \{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4, \mathcal{A}_7\}$ is no longer pairwise similar. Although Projection-Binary ($\mathcal{A}_1$) is somewhat similar to Projection-Average, Projection-Neighbors and Cluster-based Similarity Partition ($\mathcal{A}_2$,$\mathcal{A}_3$,$\mathcal{A}_4$), the rest of the algorithms are not pairwise similar.

The assignment of the weight on the edges for Projection-Average and Projection-Neighbors are clearly different since the former concerns the connectivity between vertex pairs whereas the latter concerns the connectivity between the neighbors of vertex pairs. This difference is more apparent for higher dimensions. However this disparity does not appears in real-world data (section 8.5).

It is particularly interesting that although Projection-Average ($\mathcal{A}_2$) and Cluster-based Similarity Partition ($\mathcal{A}_4$) are not similar, they are both relatively similar to CLECC Bridge Detection ($\mathcal{A}_6$). Moreover only at higher dimensions Projection-Average is similar to Generalized Canonical Correlations ($\mathcal{A}_6$). There is no strong arguments to this statistical observations besides that these algorithms follows the general strategy to prefer vertex pairs that are similar locally.

8.4. Structured Synthetic Random Multiplex

In the previous section, USRM benchmark suggests that $\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4\}$ yields similar communities. However Fig. 5 shows that the SSRM communities from $\mathcal{A}_4$ (Cluster-based Similarity Partition Algorithm) are distinct from the class of projection algorithms ($\mathcal{A}_1$ to $\mathcal{A}_3$). Therefore we can conclude that Cluster-based Similarity Partition Algorithm provides an alternative perspective for multiplex-communities, and it is not reducible to the class of projection algorithms.

Recall from the last observation in the previous section that the similarity range of $\mathcal{A}_6$ (CLECC Bridge Detection) with projection algorithms is wide. Fig. 5 shows that there are cases where they are very different (close to NMI = 0), and there are cases where the NMI > 0.8. In this way our experiments highlights the weakness of the CLECC Bridge Detection algorithm.

CLECC occasionally yields separate components prematurely, where one of the components is a small cluster of vertices or even a single vertex as a community. Therefore it appears that CLECC yields significantly different communities since the rest of the algorithms tend to produce balanced-size communities. Hence if we exclude such cases, CLECC Bridge Detection is similar to projection algorithms for SSRM.

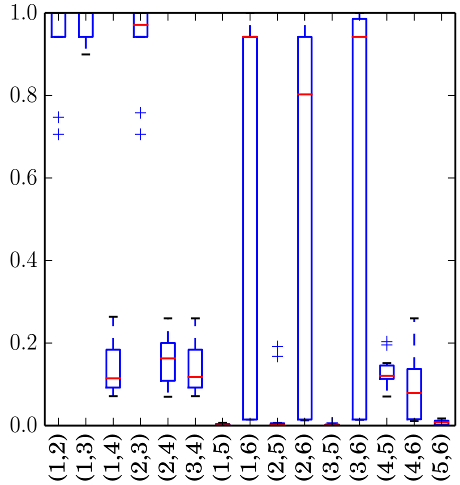


Figure 5: The NMI scores of all pairwise non-overlapping multiplex-communities detection algorithms for SSRM benchmarks. The tuples are arranged in the same way as Fig. 3.

Finally we will compare the algorithms with the "ground-truth" partitions $\mathcal{P}_1$, $\mathcal{P}_2$ and $\mathcal{P}_3$. Table 4 shows that none of the algorithms were able to capture $\mathcal{P}_1$, which is the partition with high redundancy. $\mathcal{A}_3$ (Projection-Neighbors) was proposed to extract high redundancy communities [3].

In contrast, $\mathcal{A}_6$ (CLECC Bridge Detection) was able to find the high CLECC partition $\mathcal{P}_2$. However it does not have any advantage over any of the projection algorithms ($\mathcal{A}_1$ to $\mathcal{A}_3$). This further supports that CLECC Bridge Detection is similar to the class of projection algorithms.

The results by Tang et. al shows that $\mathcal{A}_5$ (Generalized Canonical Correlations) tends to be better than $\mathcal{A}_4$ (Cluster-based Similarity Partition Algorithm) to capture high-modularity communities like $\mathcal{P}_3$ [6]. This was also observed in this experiment.

8.5. Real World Multiplex

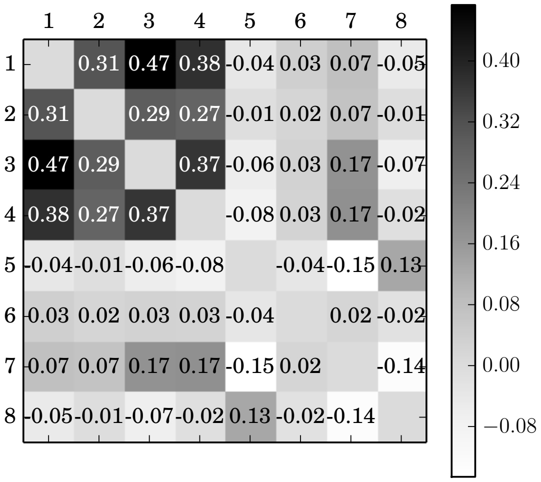
The results for the European Air Transportation Network is similar to the Youtube Social Network, hence Fig. 6 is sufficient for this discussion. The general observation is similar to USRM 2, 4 and 5 in Fig. 2, where the set of projection algorithms, Cluster-based Similarity Partition and Frequent Closed Itemsets Mining ($\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4, \mathcal{A}_7\}$) are relatively similar with pairwise NMI score of ≈ 0.55.

	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$
$\mathcal{A}_1$	0	**0.983**	0
$\mathcal{A}_2$	0.002	**0.94**	0.017
$\mathcal{A}_3$	0	**0.978**	0
$\mathcal{A}_4$	**0.019**	0.14	0.083
$\mathcal{A}_5$	0.004	0.002	**0.158**
$\mathcal{A}_6$	0.006	**0.964**	0.006

Table 4: The NMI scores between the algorithms and the different ground-truth partitions. The entries in bold represents the algorithms that are "close" to the ground-truth partition.

In addition, Fig. 6 highlights that overlapping-communities detection algorithm $\mathcal{A}_8$ (PARAFAC) is relatively more similar to $\mathcal{A}_5$ (Generalized Canonical Correlations) than the other non-overlapping communities detection algorithms. This observation was less apparent in Fig. 2.

Unfortunately $\mathcal{A}_6$ (CLECC Bridge Detection) tends to halt prematurely despite different parameter choices. Hence we did not managed to get any insight for CLECC Bridge Detection in this experiment.



	1	2	3	4	5	6	7	8
1		0.31	0.47	0.38	-0.04	0.03	0.07	-0.05
2	0.31		0.29	0.27	-0.01	0.02	0.07	-0.01
3	0.47	0.29		0.37	-0.06	0.03	0.17	-0.07
4	0.38	0.27	0.37		-0.08	0.03	0.17	-0.02
5	-0.04	-0.01	-0.06	-0.08		-0.04	-0.15	0.13
6	0.03	0.02	0.03	0.03	-0.04		0.02	-0.02
7	0.07	0.07	0.17	0.17	-0.15	0.02		-0.14
8	-0.05	-0.01	-0.07	-0.02	0.13	-0.02	-0.14	

Figure 6: The Omega Index heatmap of all pairwise algorithms for the European Air Transportation Network. This result is similar to the Youtube Social Network.

8.6. Summary

The parameters in CLECC Bridge Detection, Frequent Closed Itemsets Mining and PARAFAC ($\mathcal{A}_6$, $\mathcal{A}_7$ and $\mathcal{A}_8$) require manual fine-tuning to yield meaningful communities. Hence it is not practical to exhaustively test for all configurations for these algorithms. However the outcome of the analysis doesn't change in any essential way for the different choice of parameters.

From USRM benchmarks and real world multiplexes, algorithms in the set $\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4, \mathcal{A}_7\}$ tends to generate relatively similar partitions, i.e. the projection algorithms, Cluster-based Similarity Partition and Frequent Closed Itemsets Mining. However SSRM demonstrates that Cluster-based Similarity Partition is able to capture high redundancy communities and yielded differently from the class of projection algorithms.

Our experiments with USRM and SSRM benchmarks support that $\mathcal{A}_6$ (CLECC Bridge Detection) is similar to the class of projection algorithms $\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3\}$ when the algorithm does not infer small clusters of vertices as communities. Therefore CLECC Bridge Detection is not very stable without careful analysis.

Finally we have to emphasize that the Omega/NMI Index are generally low (< 0.5) for most pairwise comparisons. This implies that the algorithms perceive less-than-ideal multiplex-communities differently. Moreover our results show that there are some vertex clusters within communities that are more likely to be captured by certain algorithms than the others.

This paper emphasizes that there is no notion that one algorithm supersedes another in accuracy as the definition of a multiplex-community varies according to the applications. Therefore there is no meaningful advantages/disadvantages for any of the algorithms as they were designed for different purposes. Some algorithms like Generalized Canonical Correlations prioritize on structural features (e.g. multi-modularity) while others like CLECC Bridge Detection prioritize on relational features (e.g. high density of overlapping edges).

Therefore there is no quantitative way to measure the pros and cons of the algorithms, although by the anecdotal experiences during the experiments, certain algorithms requires more attention to fine-tune their parameters and hence not easily applied in practice. This is particularly true for algorithms that requires > 1 parameters like CLECC Bridge Detection and PARAFAC.

This paper left out the explanation to why in 2-dimensional USRMs the algorithms behave different from higher dimensional USRMs. Currently the literature do not have the sufficient understanding to the stability of the dynamics (e.g. centrality, information propagation, etc) in general when new layers are added into the multiplex. Therefore it is hard for us to make a meaningful statement without further development from multiplex research (read section 9).

The diversity of the algorithms, multiplex parameters, and the different concepts of multiplex-communities illustrate the complexity and multifaceted nature of multiplex research. There is no consistent observation that is true for all the different benchmarks which is the highlight of this paper. There are many degrees of freedom to construct a multiplex benchmark or to tweak the algorithm parameters such that we get a different perspective and conclusion. Thus it is important for this paper to consolidate these algorithms and be consciously aware of the complexity around the problem. Due to the low pairwise similarity of many algorithms, the wrong choice of algorithm (or concept) can easily deviates one from the intended direction!

9. Discussions, Conclusion and Future Work

The literature review and SSRM benchmark highlight the additional complexity to define a multiplex-community. It is a balance between the structural topology of the communities and their relationships. Thus SSRM is a useful benchmark to bring out the fundamental differences between these algorithms. Therefore further research is done to improve SSRM such that the different definitions of communities are more apparent.

The main contribution of this paper is to consolidate and compare all the existing multiplex-communities detection algorithms. The long list of synonymous names for multiplex causes many researchers to be unaware of related efforts for multiplex, and hence has a tendency to make this research some what diffuse and fragmented [19, 20]. Therefore we hope that this paper brings awareness for further developments on multiplex-communities detection algorithm such that researchers can build on to the existing ideas.

During the peer review and the revisions of this paper, more related research (preprints) surfaced into the literature. Some are relatively new research and some are from disciplines that we have not explored during our research. Regrettably it is hard to include them without major rewriting, re-simulations and the time to review these preprints (as they are not peer reviewed yet). However for completeness the following is an outline of these research.

- Barzinpour *et al.* proposed a spectral approach to communities detection. It is similar to Generalized Canonical Correlations (section 5.2.3) where the multiplex is mapped to a Euclidean Space (using the top few eigenvectors) and apply k-mean clustering. In addition from the communities one can define the closeness centrality of multiplex [52].

- Multiplex communities detection can be presented as a heterogeneous data clustering in computer science, specifically database management. Thus using their specialized tools like Relational Bayesian Networks, one can use the result to interpret the communities in multiplex [53].

- Given two auxiliary-partitions of two networks in a multiplex, Hao *et al.* proposed a metric (impact-strength-index) to measure how much influence the monoplex-communities in one of the auxiliary-partitions onto the monoplex-communities of the other auxiliary-partition[54].

- Zhu and Li proposed another type of projection algorithm. The first step is to quantify the importance of every monoplex by measuring how correlated one monoplex is to the rest of the multiplex. In the second step, every monoplex yield a similarity matrix (with another proposed metric) between all pairwise nodes. The projected network is a function of the results from the earlier steps. [55]

- *MutuRank* by Wu *et al.* is also based on the strategy of projection. It uses both the probability distributions that a vertex chooses its neighbors and that the same vertex chooses its dimension to form a distribution on the frequency of the relationships. This relation distribution is then used to project the multiplex in a linear way [56].

- Infomap is a popular monoplex-communities detection algorithm based on the compressibility of a random walk. Domenico *et al.* proposed a multiplex extension by factoring the probability of swapping dimensions for every node's transition probability [57].

- Bródka and Grecki proposed a benchmark multiplex based on a well known network benchmark — LFR Benchmark. Unfortunately only the source code (with little documentation) is given, hence we are unable to further describe the construction [58].

Lastly we thank the anonymous reviewers for their time and constructive comments. Although we began this paper with the goal to consolidate the disparate literature, it appears that the criticisms and the questions from the reviewers also open the Pandora's Box for future work. With hindsight it is really peculiar that none of the algorithms or research (including the first revision of this paper) attempts to understand the change of the communities when layers are added/removed from the system. Hence we believe that the stability of the dynamics (e.g. communities and centrality) of multiplex is paramount for future work.

[1] G. Cormode, S. Muthukrishnan, Space efficient mining of multigraph streams, in: Principles of Database Systems, 2005.

[2] M. Gjoka, C. T. Butts, M. Kurant, A. Markopoulou, Multigraph sampling of online social networks, IEEE J. Sel. Areas Commun. on Measurement of Internet Topologies (2011) –.

[3] M. Berlingerio, M. Coscia, F. Giannotti, Finding and characterizing communities in multidimensional networks., in: ASONAM, 2011.

[4] P. Kazienko, K. Musial, E. Kukla, T. Kajdanowicz, P. Bródka, Multidimensional social network: model and analysis, in: Computational Collective Intelligence. Technologies and Applications, 2011.

[5] G. Rossetti, M. Berlingerio, F. Giannotti, Scalable link prediction on multidimensional networks., in: ICDM Workshops, IEEE, 2011.

[6] L. Tang, X. Wang, H. Liu, Uncovering groups via heterogeneous interaction analysis, in: ICDM '09, 2009, pp. 503–512.

[7] D. A. Davis, R. Lichtenwalter, N. V. Chawla, Multi-relational link prediction in heterogeneous information networks, in: ASONAM, 2011.

[8] M. A. Rodriguez, J. Shinavier, Exposing multi-relational networks to single-relational network analysis algorithms, J. of Informetrics 4 (1).

[9] J. Skvoretz, F. Agneessens, Reciprocity, multiplexity, and exchange: Measures, Quality & Quantity 41 (3) (2007) 341–357.

[10] P. Bródka, T. Filipowski, P. Kazienko, An introduction to community detection in multi-layered social network, CoRR abs/1209.6050.

[11] C. Li, J. Luo, J. Huang, J. Fan, Multi-layer network for influence propagation over microblog, in: Intelligence & Security Informatics, 2012.

[12] E. Cozzo, A. Arenas, Y. Moreno, Stability of Boolean Multilevel Networks, Physical Review E 86 (3). arXiv:1205.3111.

[13] S. A. Applin, M. Fischer, Everybody is talking to each other without talking to each other: Polysocial reality and asynchronous adaptation, The Annual Meeting of the American Anthropological Association.

[14] D. Ambrosino, A. Sciomachen, Hub locations in urban multimodal networks, European Transport Trasporti Europei (51).

[15] L. J. LeBlanc, Transit system network design, Transportation Research Part B: Methodological 22 (5) (1988) 383–390.

[16] A. Nagurney, A multiclass, multicriteria traffic network equilibrium model, Mathematical and Computer Modelling 32 (34).

[17] Y. Dong, J. Tang, S. Wu, J. Tian, N. V. Chawla, J. Rao, H. Cao, Link prediction and recommendation across heterogeneous social networks, in: ICDM, IEEE, 2012, pp. 181–190.

[18] M. Magnani, L. Rossi, Formation of multiple networks, in: Social Computing, Behavioral-Cultural Modeling and Prediction, Springer, 2013.

[19] S. Boccaletti, G. Bianconi, R. Criado, C. del Genio, J. Gómez-Gardeñes, M. Romance, I. Sendiña-Nadal, Z. Wang, M. Zanin, The structure and dynamics of multilayer networks, Physics Reports.

[20] M. Kivel, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, M. A. Porter, Multilayer networks, Journal of Complex Networks-doi:10.1093/comnet/cnu016.

[21] D. Cellai, E. López, J. Zhou, J. P. Gleeson, G. Bianconi, Percolation in multiplex networks with overlap, arXiv preprint arXiv:1307.6359.

[22] K.-M. Lee, J. Y. Kim, W. kuk Cho, K.-I. Goh, I.-M. Kim, Correlated multiplexity and connectivity of multiplex random networks, New Journal of Physics 14 (3) (2012) 033027.

[23] A. L. Chen, F. J. Zhang, H. Li, The degree distribution of the random multigraphs, Acta Mathematica Sinica, English Series 28 (5).

[24] J. M. O. Ranola, S. Ahn, M. E. Sehl, D. J. Smith, K. Lange, A poisson model for random multigraphs., Bioinformatics 26 (16).

[25] G. Bianconi, Statistical mechanics of multiplex networks: Entropy and overlap, Phys. Rev. E 87 (2013) 062806.

[26] S. Fortunato, Community detection in graphs, Physics Reports 486 (3) (2010) 75–174.

[27] M. A. Porter, J.-P. Onnela, P. J. Mucha, Communities in networks, Notices of the AMS 56 (9) (2009) 1082–1097.

[28] J. Xie, S. Kelley, B. K. Szymanski, Overlapping community detection in networks: The state-of-the-art and comparative study, ACM Computing Surveys (CSUR) 45 (4) (2013) 43.

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

[30] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, D. Parisi, Defining and identifying communities in networks, Proceedings of the National Academy of Sciences 101 (9) (2004) 2658.

[31] Z. M. Yin, S. S. Khaing, Multi-layered graph clustering in finding the community cores, International Journal of Advanced Research in Computer Engineering & Technology 2.

[32] D. Khn, D. Osthus, Maximizing several cuts simultaneously., Combinatorics, Probability & Computing 16 (2) (2007) 277–283.

[33] V. Patel, Cutting two graphs simultaneously., Journal of Graph Theory 57 (1) (2008) 19–32.

[34] M. R. Garey, D. S. Johnson, Computers and Intractability; A Guide to the Theory of NP-Completeness, W. H. Freeman & Co., 1990.

[35] N. Nefedov, Multiple-membership communities detection in mobile networks., in: R. Akerkar (Ed.), WIMS, ACM, 2011, p. 64.

[36] M. Berlingerio, F. Pinelli, F. Calabrese, Abacus: frequent pattern mining-based community discovery in multidimensional networks, Data Mining and Knowledge Discovery 27 (3) (2013) 294–320.

[37] J. R. Kettenring, Canonical analysis of several sets of variables, Biometrika 58 (3) (1971) pp. 433–451.

[38] M. Girvan, M. E. Newman, Community structure in social and biological networks, Proceedings of the National Academy of Sciences.

[39] R. A. Harshman, Foundations of the PARAFAC procedure: Models and conditions for an" explanatory" multi-modal factor analysis, UCLA Working Papers in Phonetics 16 (1) (1970) 84.

[40] D. M. Dunlavy, T. G. Kolda, W. P. Kegelmeyer, Multilinear algebra for analyzing data with multiple linkages, Graph Algorithms in the Language of Linear Algebra, Fundamentals of Algorithms, SIAM, Philadelphia (2011) 85–114.

[41] M. Leginus, P. Dolog, V. emaitis, Improving tensor based recommenders with clustering, in: User Modeling, Adaptation, and Personalization, Vol. 7379 of Lecture Notes in Computer Science, Springer, 2012.

[42] A. Mirzal, M. Furukawa, Node-Context Network Clustering using PARAFAC Tensor Decomposition (May 2010). arXiv:1005.0268.

[43] E. E. Papalexakis, L. Akoglu, D. Ienco, Do more views of a graph help? community detection and clustering in multi-graphs, in: 16th International Conference on Information Fusion, 2013.

[44] P. Erdös, A. Rnyi, On the evolution of random graphs, in: Publications of the Mathematical Institute of the Hungarian Academy of Sciences, 1960, pp. 17–61.

[45] D. J. Watts, S. H. Strogatz, Collective dynamics of small-worldnetworks, nature 393 (6684) (1998) 440–442.

[46] R. Albert, A.-L. Barabási, Statistical mechanics of complex networks, Reviews of modern physics 74 (1) (2002) 47.

[47] C. W. Loe, H. J. Jensen, Edge union of networks on the same vertex set, Journal of Physics A: Mathematical and Theoretical 46 (24).

[48] A. Cardillo, J. Gómez-Gardeñes, M. Zanin, M. Romance, D. Papo, F. del Pozo, S. Boccaletti, Emergence of network features from multiplexity, Scientific reports 3.

[49] C. D. Manning, P. Raghavan, H. Schütze, Introduction to information retrieval, Vol. 1, Cambridge university press Cambridge, 2008.

[50] L. M. Collins, C. W. Dent, Omega: A general formulation of the rand index of cluster recovery suitable for non-disjoint solutions, Multivariate Behavioral Research 23 (2) (1988) 231–242.

[51] L. Hubert, P. Arabie, Comparing partitions, Journal of classification 2 (1) (1985) 193–218.

[52] S. A. S. G. J. N. Farnaz Barzinpour, B. Hoda Ali-Ahmadi, Clustering networks heterogeneous data in defining a comprehensive closeness centrality index, Mathematical Problems in Engineering 2014 (202350).

[53] J. Jiang, M. Jaeger, Community detection for multiplex social networks based on relational bayesian networks, in: Foundations of Intelligent Systems, Vol. 8502 of LNCS, 2014, pp. 30–39.

[54] J. Hao, S. Cai, Q. He, Z. Liu, The interaction between multiplex community networks, Chaos: An Interdisciplinary Journal of Nonlinear Science 2011 21 (1).

[55] G. Zhu, K. Li, A unified model for community detection of multiplex networks, in: Web Information Systems Engineering WISE 2014, Vol. 8786 of LNCS, 2014, pp. 31–46.

[56] Z. Wu, W. Yin, J. Cao, G. Xu, A. Cuzzocrea, Community detection in multi-relational social networks, in: Web Information Systems Engineering WISE 2013, Vol. 8181 of LNCS, 2013, pp. 43–56.

[57] A. A. M. R. Manlio De Domenico, Andrea Lancichinetti, Identifying modular flows on multilayer networks reveals highly overlapping organization in social systems (2014). arXiv:1408.2925.

[58] G. T. Bródka P, mlfr benchamark: Testing community detection algorithms in multilayered, multiplex and multiple social networks.
URL `https://www.ii.pwr.edu.pl/ brodka/mlfr.php`