Community Detection in Multiplex Networks using Locally Adaptive Random Walks

Zhana Kuncheva [1] Giovanni Montana [2]

[1]Department of Mathematics
Imperial College London

[2]Department of Biomedical Engineering
King's College London

July 25, 2015

Multiplex Networks

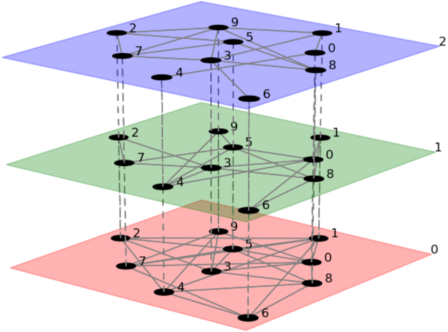


Figure: [Kivel, 2012]

Multiplex Networks

Definition: Multiplex Network

An L-layered multiplex network is a multi-layer undirected graph $\mathcal{M} = (V; A_k)_{k=1}^{L}$, where V is a set of nodes and A_k is the $N \times N$ adjacency matrix representing the set of edges in layer L_k for $k = 1, 2, ..., L$.

Multiplex Networks

Definition: Multiplex Network

An L-layered multiplex network is a multi-layer undirected graph $\mathcal{M} = (V; A_k)_{k=1}^{L}$, where V is a set of nodes and A_k is the $N \times N$ adjacency matrix representing the set of edges in layer L_k for $k = 1, 2, ..., L$.

- Node v_i^k - node $v_i \in V$, $i = 1, 2, ..., N$, in layer L_k.

Multiplex Networks

Definition: Multiplex Network

An L-layered multiplex network is a multi-layer undirected graph $\mathcal{M} = (V; A_k)_{k=1}^{L}$, where V is a set of nodes and A_k is the $N \times N$ adjacency matrix representing the set of edges in layer L_k for $k = 1, 2, ..., L$.

- Node v_i^k - node $v_i \in V$, $i = 1, 2, ..., N$, in layer L_k.
- The connection between nodes v_i and v_j in L_k is given by $A_{ij;k} = A_{ji;k}$. Nodes v_i and v_j in L_k are neighbors if $A_{ij;k} = A_{ji;k} = 1$, otherwise $A_{ij;k} = 0$. Furthermore, $\forall k$, $A_{ij;k} = 0$ for $i = j$.

Multiplex Networks

Definition: Multiplex Network

An L-layered multiplex network is a multi-layer undirected graph $\mathcal{M} = (V; A_k)_{k=1}^{L}$, where V is a set of nodes and A_k is the $N \times N$ adjacency matrix representing the set of edges in layer L_k for $k = 1, 2, ..., L$.

- Node v_i^k - node $v_i \in V$, $i = 1, 2, ..., N$, in layer L_k.
- The connection between nodes v_i and v_j in L_k is given by $A_{ij;k} = A_{ji;k}$. Nodes v_i and v_j in L_k are neighbors if $A_{ij;k} = A_{ji;k} = 1$, otherwise $A_{ij;k} = 0$. Furthermore, $\forall k$, $A_{ij;k} = 0$ for $i = j$.
- Each pair of corresponding nodes in different layers, v_i^k and v_i^l, has an inter-layer connection denoted by $\omega_{i;kl} \in \mathbb{R}$.

Multiplex Community Detection: Problem Formulation

Shared Communities

A shared community is a set of nodes for which several (but not necessarily all) layers provide topological evidence that these nodes form the same community that is shared across these layers.

Multiplex Community Detection: Problem Formulation

Shared Communities

A shared community is a set of nodes for which several (but not necessarily all) layers provide topological evidence that these nodes form the same community that is shared across these layers.

Non-Shared Communities

A non-shared community is a set of nodes which have a densely connected structural pattern specific to one layer.

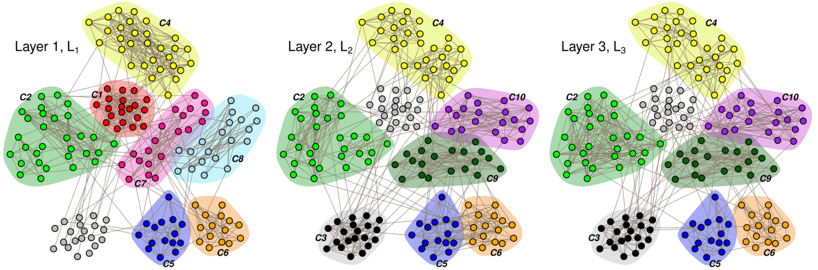
Multiplex Community Detection: Problem Formulation

Shared Communities

A shared community is a set of nodes for which several (but not necessarily all) layers provide topological evidence that these nodes form the same community that is shared across these layers.

Non-Shared Communities

A non-shared community is a set of nodes which have a densely connected structural pattern specific to one layer.



Multiplex Community Detection: Literature Review

- Layer aggregation procedures;

Multiplex Community Detection: Literature Review

- Layer aggregation procedures;
- Cluster ensemble procedures;

Multiplex Community Detection: Literature Review

- Layer aggregation procedures;
- Cluster ensemble procedures;
- Tensor decompositions: a multiplex can be represented as a third order tensor;

Multiplex Community Detection: Literature Review

- Layer aggregation procedures;
- Cluster ensemble procedures;
- Tensor decompositions: a multiplex can be represented as a third order tensor;
- Extensions of community detection algorithms from one to multiple layers:

Multiplex Community Detection: Literature Review

- Layer aggregation procedures;
- Cluster ensemble procedures;
- Tensor decompositions: a multiplex can be represented as a third order tensor;
- Extensions of community detection algorithms from one to multiple layers:

1. Principal Modularity Maximization [Tang et al., 2009];

Multiplex Community Detection: Literature Review

- Layer aggregation procedures;
- Cluster ensemble procedures;
- Tensor decompositions: a multiplex can be represented as a third order tensor;
- Extensions of community detection algorithms from one to multiple layers:

1. Principal Modularity Maximization [Tang et al., 2009];
2. Multislice Modularity Maximization [Mucha et al., 2010];

Multiplex Community Detection: Literature Review

- Layer aggregation procedures;
- Cluster ensemble procedures;
- Tensor decompositions: a multiplex can be represented as a third order tensor;
- Extensions of community detection algorithms from one to multiple layers:

1. Principal Modularity Maximization [Tang et al., 2009];
2. Multislice Modularity Maximization [Mucha et al., 2010];
3. Multiplex Infomap [De Domenico et al., 2015];

Multiplex Community Detection: Literature Review

- Layer aggregation procedures;
- Cluster ensemble procedures;
- Tensor decompositions: a multiplex can be represented as a third order tensor;
- Extensions of community detection algorithms from one to multiple layers:

1. Principal Modularity Maximization [Tang et al., 2009];
2. Multislice Modularity Maximization [Mucha et al., 2010];
3. Multiplex Infomap [De Domenico et al., 2015];
4. Seed-centric algorithm extension [Hmimida and Kanawati, 2015].

Multiplex Community Detection: Single Layer Case

- Random walks are used to unfold the community structure on a network.

Multiplex Community Detection: Single Layer Case

- Random walks are used to unfold the community structure on a network.
- A random walker is expected to get "trapped" for longer times in denser regions defining the communities.

Multiplex Community Detection: Single Layer Case

- Random walks are used to unfold the community structure on a network.
- A random walker is expected to get "trapped" for longer times in denser regions defining the communities.

Walktrap algorithm [Pons and Latapy, 2006]

- Jump probability: $P_{ij} = \frac{A_{ij}}{d_i}$, $d_i = \sum_{j=1}^{N} A_{ij}$;

Multiplex Community Detection: Single Layer Case

- Random walks are used to unfold the community structure on a network.
- A random walker is expected to get "trapped" for longer times in denser regions defining the communities.

Walktrap algorithm [Pons and Latapy, 2006]

- Jump probability: $P_{ij} = \frac{A_{ij}}{d_i}$, $d_i = \sum_{j=1}^{N} A_{ij}$;
- Short random walks of length t, P^t, capture local topology of a network;

Multiplex Community Detection: Single Layer Case

- Random walks are used to unfold the community structure on a network.
- A random walker is expected to get “trapped” for longer times in denser regions defining the communities.

Walktrap algorithm [Pons and Latapy, 2006]

- Jump probability: $P_{ij} = \frac{A_{ij}}{d_i}$, $d_i = \sum_{j=1}^{N} A_{ij}$;
- Short random walks of length t, P^t, capture local topology of a network;
- Define node dissimilarity measure to capture similarity between nodes;

Multiplex Community Detection: Single Layer Case

- Random walks are used to unfold the community structure on a network.
- A random walker is expected to get "trapped" for longer times in denser regions defining the communities.

Walktrap algorithm [Pons and Latapy, 2006]

- Jump probability: $P_{ij} = \frac{A_{ij}}{d_i}$, $d_i = \sum_{j=1}^{N} A_{ij}$;
- Short random walks of length t, P^t, capture local topology of a network;
- Define node dissimilarity measure to capture similarity between nodes;
- Merge nodes using hierarchical clustering;

Multiplex Community Detection: Single Layer Case

- Random walks are used to unfold the community structure on a network.
- A random walker is expected to get "trapped" for longer times in denser regions defining the communities.

Walktrap algorithm [Pons and Latapy, 2006]

- Jump probability: $P_{ij} = \frac{A_{ij}}{d_i}$, $d_i = \sum_{j=1}^{N} A_{ij}$;
- Short random walks of length t, P^t, capture local topology of a network;
- Define node dissimilarity measure to capture similarity between nodes;
- Merge nodes using hierarchical clustering;
- Select best partition by maximizing the modularity function Q [Girvan and Newman, 2002].

Multiplex Community Detection: Locally Adaptive Random Transitions (LART) Algorithm

- Facilitate the exploration of shared and non-shared communities.

Multiplex Community Detection: Locally Adaptive Random Transitions (LART) Algorithm

- Facilitate the exploration of shared and non-shared communities.
- LART is based on a multiplex random walk [Domenico and Sole-Ribalta, 2014].

Multiplex Community Detection: Locally Adaptive Random Transitions (LART) Algorithm

- Facilitate the exploration of shared and non-shared communities.
- LART is based on a multiplex random walk [Domenico and Sole-Ribalta, 2014].
- Contribution: we adapt the transition probabilities of the random walk to depend on the local topological similarity between any pair of layers, at any given node.

Multiplex Community Detection: Locally Adaptive Random Transitions (LART) Algorithm

- Facilitate the exploration of shared and non-shared communities.
- LART is based on a multiplex random walk [Domenico and Sole-Ribalta, 2014].
- Contribution: we adapt the transition probabilities of the random walk to depend on the local topological similarity between any pair of layers, at any given node.
- Result: the random walker spends longer times moving between nodes in communities which are shared across layers.

Multiplex Community Detection: Locally Adaptive Random Transitions (LART) Algorithm

- Facilitate the exploration of shared and non-shared communities.
- LART is based on a multiplex random walk [Domenico and Sole-Ribalta, 2014].
- Contribution: we adapt the transition probabilities of the random walk to depend on the local topological similarity between any pair of layers, at any given node.
- Result: the random walker spends longer times moving between nodes in communities which are shared across layers.
- Using properties of the random walk: introduce a dissimilarity measure between nodes and use it in a hierarchical clustering procedure to detect shared and non-shared communities.

LART: Inter-layer weights

Definition: Inter-layer weights

$$\omega_{i;kl} := |N_{i,k} \cap N_{i,l}|$$

where $N_{i,k} := \{v_j^k : A_{ij;k} = 1\}$ is the set of edges for v_i^k.

LART: Inter-layer weights

Definition: Inter-layer weights

$$\omega_{i;kl} := |N_{i,k} \cap N_{i,l}|$$

where $N_{i,k} := \{v_j^k : A_{ij;k} = 1\}$ is the set of edges for v_i^k.

Definition: Supra-adjacency matrix

$$\mathcal{A}^* := \left(\begin{array}{c|c|c|c} A_1 & W_{12} & \dots & W_{1L} \\ \hline W_{21} & A_2 & & \\ \hline \dots & & \dots & \\ \hline W_{L1} & & & A_L \end{array}\right).$$

LART: Inter-layer weights

Definition: Inter-layer weights

$$\omega_{i;kl} := |N_{i,k} \cap N_{i,l}|$$

where $N_{i,k} := \{v_j^k : A_{ij;k} = 1\}$ is the set of edges for v_i^k.

Definition: Supra-adjacency matrix

$$\mathcal{A}^* := \left(\begin{array}{c|c|c|c} A_1 & W_{12} & \dots & W_{1L} \\ \hline W_{21} & A_2 & & \\ \hline \dots & & \dots & \\ \hline W_{L1} & & & A_L \end{array}\right).$$

- We require $\mathcal{A}^*$ to be "well-behaved", i.e. connected and non-bipartite.

LART: Inter-layer weights

Definition: Inter-layer weights

$$\omega_{i;kl} := |N_{i,k} \cap N_{i,l}|$$

where $N_{i,k} := \{v_j^k : A_{ij;k} = 1\}$ is the set of edges for v_i^k.

Definition: Supra-adjacency matrix

$$\mathcal{A}^* := \left(\begin{array}{c|c|c|c} A_1 & W_{12} & \dots & W_{1L} \\ \hline W_{21} & A_2 & & \\ \hline \dots & & \dots & \\ \hline W_{L1} & & & A_L \end{array}\right).$$

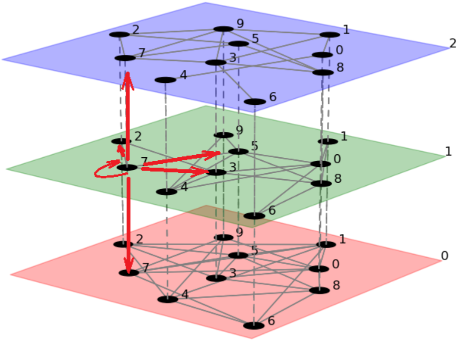
- We require $\mathcal{A}^*$ to be "well-behaved", i.e. connected and non-bipartite.
- Use $\mathcal{A}$ obtained from $\mathcal{A}^*$ by replacing the entry A_j with $A_j + \varepsilon I$ and W_{ij} with $W_{ij} + \varepsilon I$; here I is the $N \times N$ identity matrix and $0 < \varepsilon \leq 1$.

LART: Transition Probabilities

- The structure of $\mathcal{M}$ allows four possible moves that a random walker can make when in node v_i^k.

LART: Transition Probabilities

- The structure of $\mathcal{M}$ allows four possible moves that a random walker can make when in node v_i^k.



[Kivel, 2012]

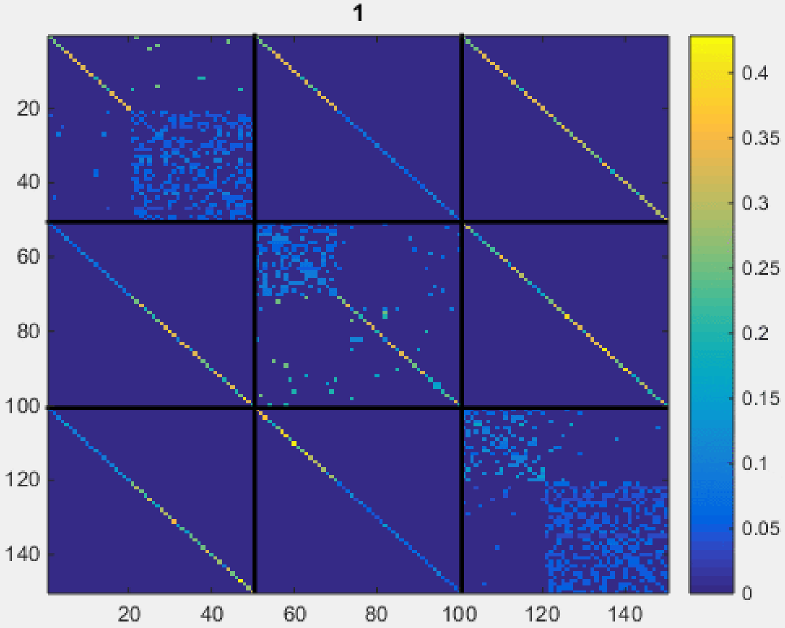
LART: Transition Probabilities

- The structure of $\mathcal{M}$ allows four possible moves that a random walker can make when in node v_i^k.
- The corresponding transition probabilities associated to these four possible moves are defined as

$$\mathcal{P}_{(i,k)(i,k)} := \frac{\varepsilon}{\kappa_{i,k}} \qquad \mathcal{P}_{(i,k)(j,k)} := \frac{A_{(i,k)(j,k)}}{\kappa_{i,k}}$$

$$\mathcal{P}_{(i,k)(i,l)} := \frac{\omega_{i;kl} + \varepsilon}{\kappa_{i,k}} \qquad \mathcal{P}_{(i,k)(j,l)} := 0$$

where $\kappa_{i,k}$ is the multiplex degree of node v_i^k in $\mathcal{A}$ defined as $\kappa_{i,k} := \sum_{j,l} \mathcal{A}_{(i,k)(j,l)}$.


1
20
40
60
80
100
120
140
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0

LART: Node Dissimilarity Measure

Node Dissimilarity Measure - Same Layer

When v_i^k and v_j^k are in the same layer, their dissimilarity is defined as:

$$S(t)_{(i,k)(j,k)} := \sqrt{\sum_{h=1}^{N}\sum_{m=1}^{L} \frac{\left(\mathcal{P}^t_{(i,k)(h,m)} - \mathcal{P}^t_{(j,k)(h,m)}\right)^2}{\kappa_{(h,m)}}}.$$

Node Dissimilarity Measure - Different Layers

When v_i^k and v_j^l are in two different layers, L_k and L_l, we define the dissimilarity as:

$$S(t)_{(i,k)(j,l)} := \sqrt{s_1 + s_2 + s_3}$$

where

$$s_1 := \sum_{h=1}^{N} \left(\frac{\mathcal{P}^t_{(i,k)(h,k)}}{\sqrt{\kappa_{(h,k)}}} - \frac{\mathcal{P}^t_{(j,l)(h,l)}}{\sqrt{\kappa_{(h,l)}}} \right)^2$$

$$s_2 := \sum_{h=1}^{N} \left(\frac{\mathcal{P}^t_{(i,k)(h,l)}}{\sqrt{\kappa_{(h,l)}}} - \frac{\mathcal{P}^t_{(j,l)(h,k)}}{\sqrt{\kappa_{(h,k)}}} \right)^2$$

$$s_3 := \sum_{h=1}^{N} \sum_{\substack{m=1;\\ m \neq k,l}}^{L} \frac{\left(\mathcal{P}^t_{(i,k)(h,m)} - \mathcal{P}^t_{(j,l)(h,m)}\right)^2}{\kappa_{(h,m)}}.$$

LART: Hierarchical Clustering

- Advantage to using an agglomerative clustering to merge nodes in communities: we ensure that the obtained communities are connected.

LART: Hierarchical Clustering

- Advantage to using an agglomerative clustering to merge nodes in communities: we ensure that the obtained communities are connected.
- We use the multiplex modularity Q_M proposed in [Mucha et al., 2010] as a criterion to select the best partition:

$$Q_M(\gamma) = \frac{1}{2\mu} \sum_{C \in \pi} \left[\sum_{(i,k),(i,l) \in C} \omega_{i;kl} + \sum_{(i,k),(i,l) \in C} [A_{ij;k} - \gamma_k \frac{d_{i,k} d_{j,k}}{2\mu_k}] \right]$$

where $2\mu = \sum_{i,j,k} A_{i,j;k}$, $d_{i,k} = \sum_j A_{ij;k}$, π is the partition into communities C, and γ_k is the resolution parameter for layer L_k.

Simulations: Comparative performance

- Compare the performance of LART to other algorithms:

Simulations: Comparative performance

- Compare the performance of LART to other algorithms:
  - Multiplex Modularity Maximization (MM).

Simulations: Comparative performance

- Compare the performance of LART to other algorithms:
  - Multiplex Modularity Maximization (MM).
  - Principal Modularity Maximization (PMM).

Simulations: Comparative performance

- Compare the performance of LART to other algorithms:
  - Multiplex Modularity Maximization (MM).
  - Principal Modularity Maximization (PMM).
  - Multiplex Infomap (IM).

Simulations: Comparative performance

- Compare the performance of LART to other algorithms:
  - Multiplex Modularity Maximization (MM).
  - Principal Modularity Maximization (PMM).
  - Multiplex Infomap (IM).
  - Apply WalkTrap algorithm on each layer. Merge communities based on similarity measures - S_T (for topological overlap) and S_M (normalized mutual information).

Simulations: Comparative performance

- Compare the performance of LART to other algorithms:
  - Multiplex Modularity Maximization (MM).
  - Principal Modularity Maximization (PMM).
  - Multiplex Infomap (IM).
  - Apply WalkTrap algorithm on each layer. Merge communities based on similarity measures - S_T (for topological overlap) and S_M (normalized mutual information).
- Five different simulation scenarios:

Simulations: Comparative performance

- Compare the performance of LART to other algorithms:
  - Multiplex Modularity Maximization (MM).
  - Principal Modularity Maximization (PMM).
  - Multiplex Infomap (IM).
  - Apply WalkTrap algorithm on each layer. Merge communities based on similarity measures - S_T (for topological overlap) and S_M (normalized mutual information).
- Five different simulation scenarios:
  - Test robustness to noise (S4) and uncovering hidden structures (S1) - shared across all three ($L = 3$) layers;

Simulations: Comparative performance

- Compare the performance of LART to other algorithms:
  - Multiplex Modularity Maximization (MM).
  - Principal Modularity Maximization (PMM).
  - Multiplex Infomap (IM).
  - Apply WalkTrap algorithm on each layer. Merge communities based on similarity measures - S_T (for topological overlap) and S_M (normalized mutual information).
- Five different simulation scenarios:
  - Test robustness to noise (S4) and uncovering hidden structures (S1) - shared across all three ($L = 3$) layers;
  - Test ability to detect and distinguish between shared and non-shared communities (S2 and S3) across three ($L = 3$) layers;

Simulations: Comparative performance

- Compare the performance of LART to other algorithms:
  - Multiplex Modularity Maximization (MM).
  - Principal Modularity Maximization (PMM).
  - Multiplex Infomap (IM).
  - Apply WalkTrap algorithm on each layer. Merge communities based on similarity measures - S_T (for topological overlap) and S_M (normalized mutual information).
- Five different simulation scenarios:
  - Test robustness to noise (S4) and uncovering hidden structures (S1) - shared across all three ($L = 3$) layers;
  - Test ability to detect and distinguish between shared and non-shared communities (S2 and S3) across three ($L = 3$) layers;
  - Mixture of different community structures (S5) across four ($L = 4$) layers;

Simulations: Comparative performance

- Compare the performance of LART to other algorithms:
  - Multiplex Modularity Maximization (MM).
  - Principal Modularity Maximization (PMM).
  - Multiplex Infomap (IM).
  - Apply WalkTrap algorithm on each layer. Merge communities based on similarity measures - S_T (for topological overlap) and S_M (normalized mutual information).
- Five different simulation scenarios:
  - Test robustness to noise (S4) and uncovering hidden structures (S1) - shared across all three ($L = 3$) layers;
  - Test ability to detect and distinguish between shared and non-shared communities (S2 and S3) across three ($L = 3$) layers;
  - Mixture of different community structures (S5) across four ($L = 4$) layers;
- For each scenario: 150 synthetic multiplexes, community sizes vary between $[10, 100]$ nodes, within-community edge probability $0.25 \leq p \leq 0.40$.

Simulations: Comparative performance

Table: Performance of various algorithms in five simulated scenarios (NMI similarity)

	S1	S2	S3	S4	S5
LART	0.99 ± 0.02	0.89 ± 0.07	0.97 ± 0.03	0.98 ± 0.04	0.96 ± 0.06
MM	0.98 ± 0.04	0.81 ± 0.07	0.83 ± 0.04	0.97 ± 0.04	0.92 ± 0.09
IM	0.43 ± 0.07	0.64 ± 0.14	0.81 ± 0.11	0.60 ± 0.10	0.53 ± 0.09
PMM	0.95 ± 0.15	0.52 ± 0.16	0.68 ± 0.02	0.97 ± 0.07	0.84 ± 0.21
ST	0.69 ± 0.07	0.76 ± 0.13	0.83 ± 0.05	0.72 ± 0.04	0.71 ± 0.11
SM	0.68 ± 0.07	0.78 ± 0.12	0.84 ± 0.06	0.71 ± 0.05	0.72 ± 0.09

- For LART and MM: report best result over resolution parameter $\gamma = 0.25, 0.75, 1, 1.25, 1.50, 1.75, 2, 2.25, 2.5, 2.75, 3$.
- Consider $t = 3L$.
- $\varepsilon = 1$.

Simulations: Comparative performance

Table: Performance of competing algorithms in five simulated scenarios for different inter-layer weights (NMI similarity)

	S1	S2	S3	S4	S5
LART	0.99 ± 0.02	0.89 ± 0.07	0.97 ± 0.03	0.98 ± 0.04	0.96 ± 0.06
LART(ω=1)	0.96 ± 0.10	0.79 ± 0.12	0.97 ± 0.04	0.77 ± 0.05	0.90 ± 0.13
LART(ω=0.5)	0.84 ± 0.13	0.85 ± 0.12	0.93 ± 0.07	0.73 ± 0.02	0.87 ± 0.10
LART(ω=0.1)	0.69 ± 0.07	0.88 ± 0.08	0.81 ± 0.04	0.72 ± 0.04	0.73 ± 0.10
MM	0.98 ± 0.04	0.81 ± 0.07	0.83 ± 0.04	0.97 ± 0.04	0.92 ± 0.09
MM(ω=1)	1.00 ± 0.00	0.62 ± 0.13	0.67 ± 0.02	0.98 ± 0.03	0.88 ± 0.18
MM(ω=0.5)	0.84 ± 0.12	0.61 ± 0.14	0.82 ± 0.01	0.80 ± 0.04	0.79 ± 0.16
MM(ω=0.1)	0.73 ± 0.06	0.62 ± 0.13	0.82 ± 0.01	0.78 ± 0.05	0.72 ± 0.14
IM	0.43 ± 0.07	0.64 ± 0.14	0.81 ± 0.11	0.60 ± 0.10	0.53 ± 0.09
IM(ω=1)	0.43 ± 0.07	0.64 ± 0.14	0.81 ± 0.11	0.60 ± 0.10	0.53 ± 0.09
IM(ω=0.5)	0.89 ± 0.13	0.89 ± 0.05	0.80 ± 0.11	0.94 ± 0.07	0.81 ± 0.10
IM(ω=0.1)	0.89 ± 0.13	0.89 ± 0.05	0.80 ± 0.11	0.94 ± 0.07	0.81 ± 0.10
IM(tele)	0.89 ± 0.13	0.89 ± 0.05	0.80 ± 0.10	0.94 ± 0.07	0.81 ± 0.10

Simulations: Tuning parameters

- LART performance is robust to parameter value γ - $\gamma \in [0.75, 1.75]$ provide similar results.

Simulations: Tuning parameters

- LART performance is robust to parameter value γ - $\gamma \in [0.75, 1.75]$ provide similar results.
- MM performance depends heavily on parameter value γ.

Simulations: Tuning parameters

- LART performance is robust to parameter value γ - $\gamma \in [0.75, 1.75]$ provide similar results.
- MM performance depends heavily on parameter value γ.
- LART performance is robust to parameter value t - $t \in [6, 15]$ provide similar results.

Simulations: Tuning parameters

- LART performance is robust to parameter value γ - $\gamma \in [0.75, 1.75]$ provide similar results.
- MM performance depends heavily on parameter value γ.
- LART performance is robust to parameter value t - $t \in [6, 15]$ provide similar results.
- Adding white noise slightly decreases results but performance is barely affected by up to 10% of white noise edges.

Simulations: Tuning parameters

- LART performance is robust to parameter value γ - $\gamma \in [0.75, 1.75]$ provide similar results.
- MM performance depends heavily on parameter value γ.
- LART performance is robust to parameter value t - $t \in [6, 15]$ provide similar results.
- Adding white noise slightly decreases results but performance is barely affected by up to 10% of white noise edges.
- These results are valid for different number of layers $L = 2, 3, 4, 5$.

Conclusions

- LART performs well for detecting shared and non-shared community structures.

Conclusions

- LART performs well for detecting shared and non-shared community structures.
- LART performs comparatively well to competing algorithms for detecting communities shared across all layers.

Conclusions

- LART performs well for detecting shared and non-shared community structures.
- LART performs comparatively well to competing algorithms for detecting communities shared across all layers.
- LART is stable for different γ and t values.

Conclusions

- LART performs well for detecting shared and non-shared community structures.
- LART performs comparatively well to competing algorithms for detecting communities shared across all layers.
- LART is stable for different γ and t values.
- The introduced inter-layer weights and corresponding locally adaptive probabilities prove to be beneficial for shared and non-shared community detection.

De Domenico, M., Lancichinetti, A., Arenas, A., and Rosvall, M. (2015).
Identifying Modular Flows on Multilayer Networks Reveals Highly Overlapping Organization in Interconnected Systems.
Phys. Rev. X, 5(1):011027.

Domenico, M. D. and Sole-Ribalta, A. (2014).
Navigability of interconnected networks under random failures.
PNAS, 111(23):8351–8356.

Girvan, M. and Newman, M. E. J. (2002).
Community structure in social and biological networks.
Proc. Natl. Acad. Sci. U. S. A., 99(12):7821–6.

Hmimida, M. and Kanawati, R. (2015).
Community Detection in Multiplex Networks: A Seed-Centric Approach.
Networks Heterog. Media, 10(1):71–85.

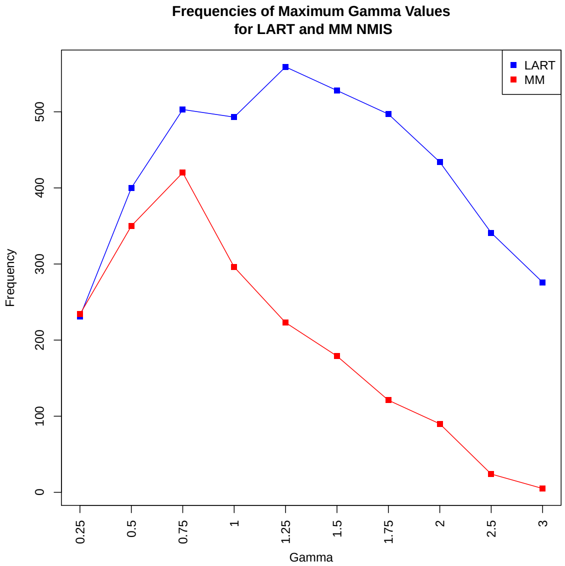
Kivel, M. (2012).
Multilayer network library.

Mucha, P. J., Richardson, T., Macon, K., Porter, M. A., and Onnela, J.-P. (2010).
Community Structure in Time-Dependent, Multiscale, and Multiplex Networks.
Science (80-.)., 328.

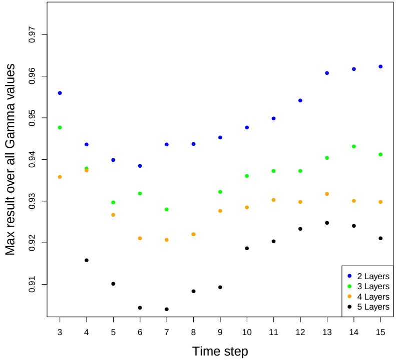
Pons, P. and Latapy, M. (2006).
Computing communities in large networks using random walks.
J. Graph Algorithms Appl., 10.2:1910218.

Tang, L., Wang, X., and Liu, H. (2009).
Uncovering Groups via Heterogeneous Interaction Analysis.
In *2009 Ninth IEEE Int. Conf. Data Min.*, pages 503–512. IEEE.

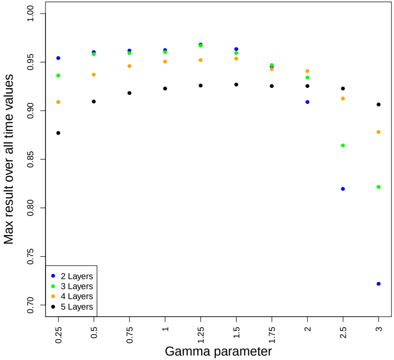
Simulations: Comparative performance



Simulations: Tuning Parameters

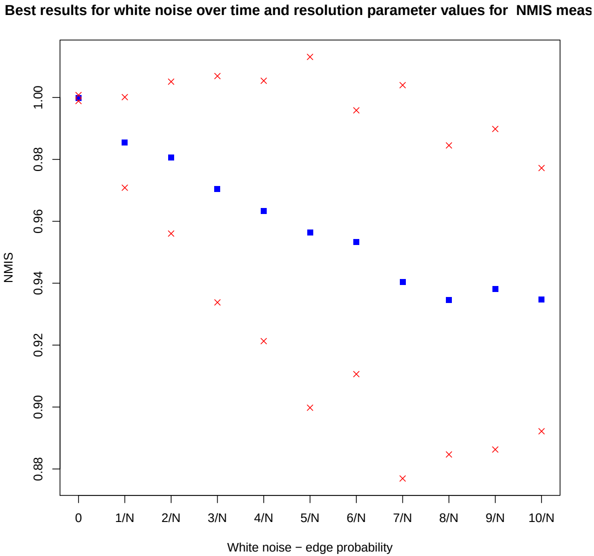


: Results for varying time steps t

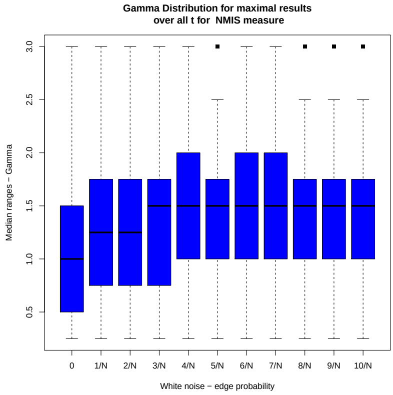


: Results for different γ parameters

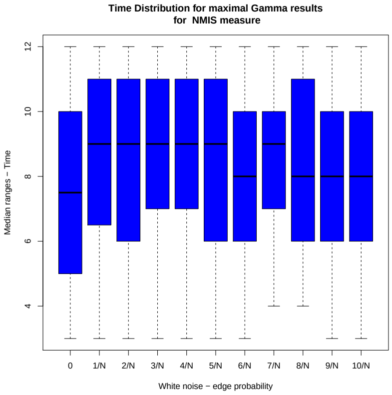
Simulations: White Noise



Simulations: White Noise



: Median ranges for Gamma values that produce best result at each white noise level.



: Median ranges for time values (over best Gamma result) that produce best result at each white noise level.