TBSI 清华-伯克利深圳学院 Tsinghua-Berkeley Shenzhen Institute

BiNeTClus: Bipartite Network Community Detection Based on Transactional Clustering

MOHAMED BOUGUESSA and KHALED NOURI

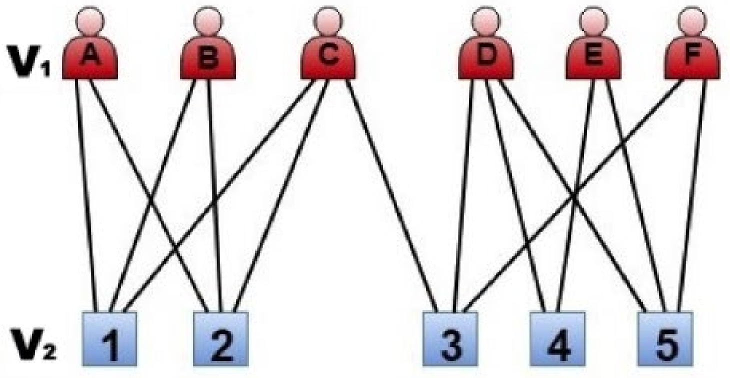
Presenter: Wanda Li
18 March, 2021

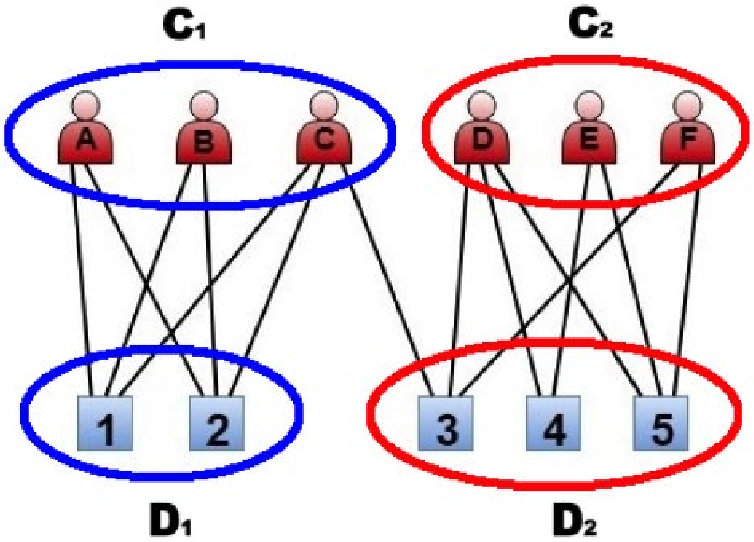
Content

Part 1
Background

1.1 Problem

- Community detection in bipartite networks
  - What is a community?
    - a group of nodes densely connected to each other and loosely linked with the nodes of the other groups
      - projecting a bipartite graph to homogeneous graphs, or simply ignore node attributes
    - a set of nodes of the same type that share a lot of connections to nodes of the second type





1.2 Related Work

- Transform a bipartite graph to a simple graph, then apply a standard community detection algorithm
  - a link between two V_1 nodes is created if they connect to the same node of the other type
- No transform but find communities of both types of nodes
  - BRIM(Bipartite Recursively Induced Modules) and its derivatives: Adaptive BRIM, LP-BRIM = LPA(Label Propagation Algorithm) + BRIM, LPAb, LPAb+ = LPAb + MSG(Multi-Step Greedy agglomerative)
- Maximize a probability function by moving nodes between communities
  - BiSBM, BiLouvain

1.3 Limitations of Existing Work

1. **Loss of relevant topological information** due to the transformation of the bipartite network to standard plain graphs.

2. **Difficulty** in **detecting communities** in the presence of many non-discriminating nodes with atypical connections that hide the community structures.

3. Manually specifying several **input parameters**, including the number of communities to be identified.

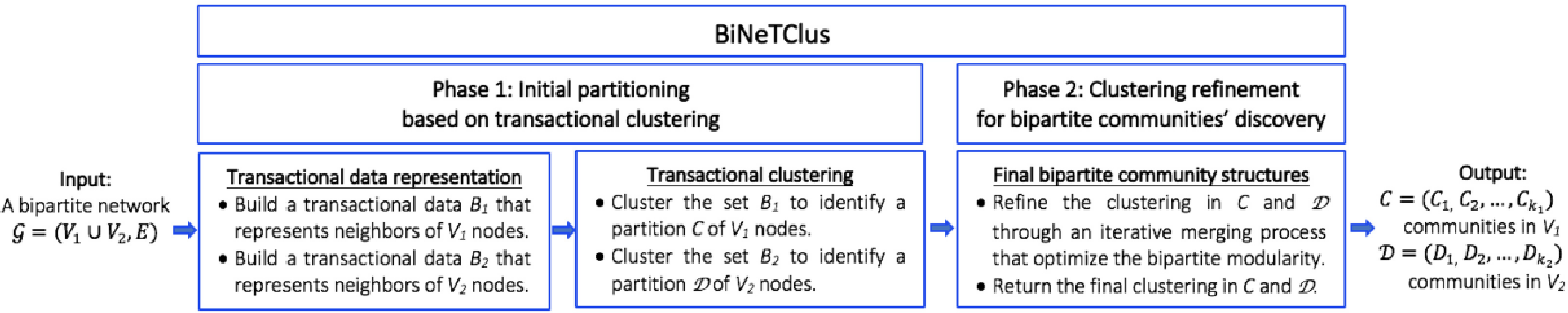
1.4 Method Comparison

Table 1. BiNeTClus vs. Mainstream Bipartite Community Detection Approaches

Approach	Handel non-discriminating nodes with atypical connections?	Projection-based?	Parameter-laden?
BiNeTClus	Yes	No	No
Alzahrani and Horadam [5]	No	Yes	No
Melamed [6]	No	Yes	Yes
Barber [10]	No	No	No
Liu and Murata [11]	No	No	No
Liu and Murata [12]	No	No	No
Pesantez and Kalyanaraman [14]	No	No	Yes
Barber and Clark [15]	No	No	No
Larremore et al. [16]	No	No	Yes

Part 2
Method

2 Flowchart



Bipartite network: $\mathcal{G} = (V_1 \cup V_2, E)$

$$V_1 = \{u_1, \dots, u_p\} \; V_2 = \{v_1, \dots, v_q\}$$

A partition of V_1 into k_1 communities $\mathcal{C} = \{C_1, C_2, \dots, C_{k_1}\}$

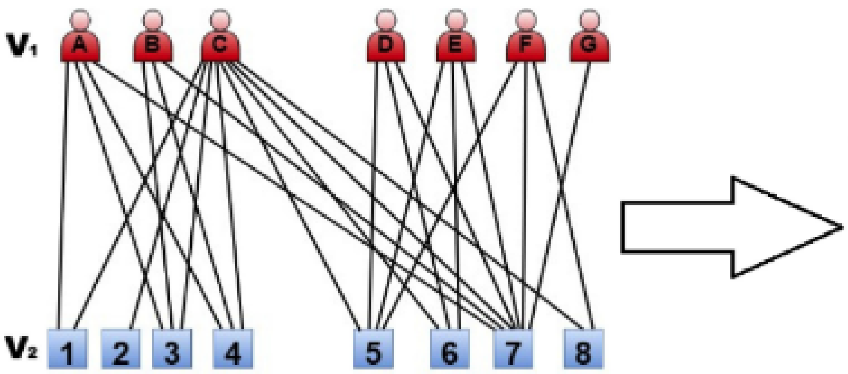
$$\mathcal{D} = \{D_1, D_2, \dots, D_{k_2}\}$$

2.1 Initial Partitioning

(1) Transactional Data Representation

- Main idea: A bipartite network can be represented as a type of transactional data without loss of information.
- Divide the resulting transactional dataset to B_1 and B_2
- Define each transaction T_{u_x} in $S_{V_1} = \{T_{u_1}, T_{u_2}, \ldots, T_{u_p}\}$ by the set $I_{V_2} = \{v_1, v_2, \ldots, v_q\}$ reflecting u_x's neighbors in V_2 ; and the same for nodes in V_2.
- Cluster transactions in B_1 and B_2 separately to make sure identified clusters contain nodes of same type.

2.1 Initial Partitioning



The transactional data set B_1

Transactions	Items
T_A	{1, 3, 4, 7}
T_B	{3, 4, 7}
T_C	{1, 2, 3, 4, 5, 6, 7, 8}
T_D	{5, 6, 7}
T_E	{5, 6, 7}
T_F	{5, 7, 8}
T_G	{7}

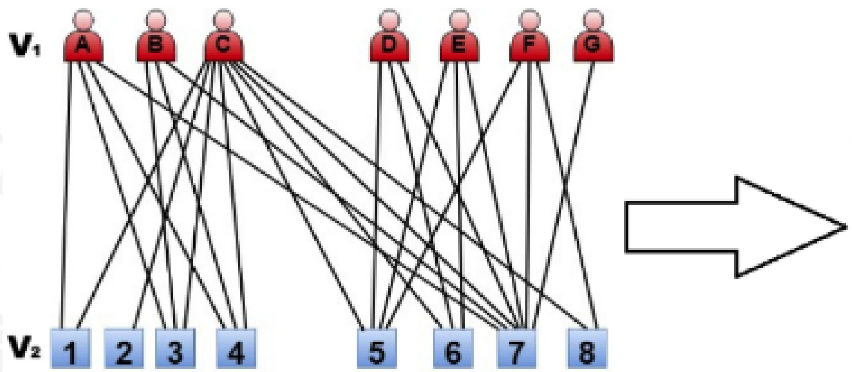
Transactions in B_1 represent neighbors of V_1 nodes.

The transactional data set B_2

Transactions	Items
T_1	{A, C}
T_2	{C}
T_3	{A, B, C}
T_4	{A, B, C}
T_5	{C, D, E, F}
T_6	{C, D, E}
T_7	{A, B, C, D, E, F, G}
T_8	{C, F}

Transactions in B_2 represent neighbors of V_2 nodes.

2.1 Initial Partitioning



The transactional data set B_1

Transactions	Items
T_A	{1, 3, 4, 7}
T_B	{3, 4, 7}
T_C	{1, 2, 3, 4, 5, 6, 7, 8}
T_D	{5, 6, 7}
T_E	{5, 6, 7}
T_F	{5, 7, 8}
T_G	{7}

Transactions in B_1 represent neighbors of V_1 nodes.

The transactional data set B_2

Transactions	Items
T_1	{A, C}
T_2	{C}
T_3	{A, B, C}
T_4	{A, B, C}
T_5	{C, D, E, F}
T_6	{C, D, E}
T_7	{A, B, C, D, E, F, G}
T_8	{C, F}

Transactions in B_2 represent neighbors of V_2 nodes.

Item C and 7: very high freq == less important

Item G and 2: very low freq == less important

2.1 Initial Partitioning

(2) Transactional Clustering

- Main objective: divide transaction $B_\bullet$ based on distribution of items into clusters C_T .
- **Objective function**: $\mathcal{O}(C_T) = \sum_{j=1}^{k} \left[\frac{r_j}{r} . \mathcal{F}(C_{T_j}) \right]$
  - r: # transactions in the transactional data.
  - r_j: # transactions in C_{T_j} .
  - $\mathcal{F}(C_{T_j}) = \frac{1}{r_j} \sum_{\eta \in C_{T_j}} \mathcal{N}(\eta, C_{T_j}) \times \mathcal{W}(\eta, C_{T_j}) \times \mathcal{W}(\eta, B_\bullet)$: quality of cluster C_{T_j}
  - η: an item
  - $\mathcal{N}(\eta, C_{T_j})$: #$\eta$ in C_{T_j}

2.1 Initial Partitioning

(2) Transactional Clustering

- *Local importance*: tradeoff between compactness and separation.

$$\mathcal{W}(\eta, C_{T_j}) = \mathcal{P}(\eta|C_{T_j}) \times (1 - \Lambda(\eta, C_{T_j}))$$

$$\mathcal{P}(\eta|C_{T_j}) = \frac{number\ of\ transactions\ in\ C_{T_j}\ that\ contain\ the\ item\ \eta}{size\ of\ C_{T_j}}$$

$$\Lambda(\eta, C_{T_j}) = \frac{number\ of\ transactions\ located\ outside\ C_{T_j}\ that\ contain\ the\ item\ \eta}{the\ total\ number\ of\ transaction\ in\ B_{\bullet}\ that\ contain\ the\ item\ \eta}$$

- *Global importance*: Measure whether η is rare or omnipresent.

$$\mathcal{W}(\eta, B_{\bullet}) = \mathcal{N}(\eta, B_{\bullet}) \times \Phi(\eta, B_{\bullet})$$

$$\Phi(\eta, B_{\bullet}) = \log[\mathcal{N}(\eta, B_{\bullet}) \times (1 - \mathcal{P}(\eta|B_{\bullet})) + 1]$$

$$\mathcal{P}(\eta|B_{\bullet}) = \frac{number\ of\ transactions\ in\ B_{\bullet}\ that\ contain\ the\ item\ \eta}{size\ of\ B_{\bullet}}$$

2.1 Algorithm 1

ALGORITHM 1: Transactional clustering

Data: A transactional dataset $B_\bullet$

Result: $C_T = \{C_{T_1}, C_{T_2}, \ldots, C_{T_k}\}$: a partitioning of $B_\bullet$ into k clusters

```
begin
    Assign each transaction T_s (s = 1, ..., r) in B• to an existing or new cluster that maximizes O(C_T);
    repeat
        Reassign each T_s to an existing or new cluster to maximize O(C_T);
    until no transaction is reassigned;
    return C_T = {C_T1, C_T2, ..., C_Tk};
end
```

* Applied independently onto the two transactional sets

2.2 Clustering Refinement

TBSI 清华-伯克利深圳学院 Tsinghua-Berkeley Shenzhen Institute

Clustering Refinement for Bipartite Communities' Discovery

- Main objective: optimize the bipartite modularity (Murata+) on the partition.

- Modularity *Murata*+:

$$Q_M^+ = \sum_{C}(e_{lm} - a_l a_m) + \sum_{\mathcal{D}}(e_{ml} - a_l a_m)$$

  - Find corresponding community from the other side by:

$$C_l = \arg\max_m (e_{ml} - a_l a_m) \quad D_m = \arg\max_l (e_{lm} - a_l a_m)$$

  - e_{lm}: the fraction of all links that connect nodes in C_l to nodes in D_m
  - a_l, a_m: the fraction of links within C_l and D_m

- Advantage: reduce #input nodes; take structural properties into consideration. → higher quality of community detection

2.2 Algorithm 2

Clustering Refinement for Bipartite Communities' Discovery

ALGORITHM 2: BiNeTClus

Data: $\mathcal{G} = (V_1 \cup V_2, E)$: a bipartite network

Result: $\mathcal{C} = \{C_1, C_2, \ldots, C_{k_1}\}$: communities in V_1

$\mathcal{D} = \{D_1, D_2, \ldots, D_{k2}\}$: communities in V_2

begin

`// Phase 1: Initial partitioning based on transactional clustering`

Represent $\mathcal{G} = (V_1 \cup V_2, E)$ as type of two transactional data: B_1 and B_2; `// Each transaction in` B_1 `consists of all neighboring nodes of type` V_2 `of each node in` V_1`. Similarly, each transaction in` B_2 `consists of all neighboring nodes of type` V_1 `of each node in` V_2

`// Next, using the transactional clustering process described by Algorithm 1, cluster, separately,` B_1 `and` B_2 `to identify an initial partitioning of` $\mathcal{G}$

Apply Algorithm 1 to cluster the set B_1;

Store the identified clusters of type V_1 in $\mathcal{C}$;

Apply Algorithm 1 to cluster the set B_2;

Store the identified clusters of type V_2 in $\mathcal{D}$;

`// Phase 2: Clustering refinement for bipartite communities' discovery`

Define N as a list of size $|V_1 + V_2|$ containing the initial partitioning of Algorithm 1 where each element in this list indicates the membership of a node to a cluster;

Define C_N as a list containing the index of each cluster in N;

Based on the initial clustering, compute Q_M^+ for $\mathcal{G}$ using (9); ⟸ $Q_M^+ = \sum_{\mathcal{C}} (e_{lm} - a_l a_m) + \sum_{\mathcal{D}} (e_{ml} - a_l a_m)$

$Q_M^{+first} \leftarrow Q_M^+$;

2.2 Algorithm 2

Clustering Refinement for Bipartite Communities' Discovery

```
    repeat
        for each cluster R in C_N do
            Identify the list of candidate clusters that can potentially be merged with the cluster R;
            Store the identified clusters in candidate_cluster;
            for each candidate cluster C_R in candidate_cluster do
                Compute Q_M^{+new} using (9) by considering R and C_R in the same cluster;
                // Evaluate Modularity gain
                if Q_M^{+new} > Q_M^{+first} then
                    Q_M^{+first} ← Q_M^{+new};
                    cluster_fus ← C_R; // cluster_fus here is the selected candidate cluster that
                        maximizes the modularity
                end
            end
            if Q_M^{+first} > Q_M^{+} then
                Q_M^{+} ← Q_M^{+first}
                Merge R and cluster_fus;
                Update C_N according to the merged clusters;
                Update C or D according to the type of nodes within the merged clusters;
            end
        end
    until it is no longer possible to increase the modularity Q_M^{+};
    Return C, D;
end
```

2.3 Complexity Analysis

TBSI 清华-伯克利深圳学院 Tsinghua-Berkeley Shenzhen Institute

- For phase 1: time complexity depends on #iterations
- Experimental results: #iterations does not grow more than linearly with:
  - k_1: #clusters in B_1 (clusters of type V_1),
  - p: size of the transactional data B_1 (#nodes of type V_1)
  - q: #items in B_1 (#nodes of type V_2 corresponding to neighbors of V_1 nodes).
- So clustering nodes of type V_1 cost $O(k_1 pq)$; overall cost is $O((k_1 + k_2)pq)$

2.3 Complexity Analysis

- For phase 2: time complexity depends on #communities
- For (k_1+k_2) communities, there are $k_1(k_1-1) + k_2(k_2-1)$ possible combinations
- In practice:
  - #merges significantly decreases
  - $k_1, k_2 << p, q$
- *… claim the effective of the heuristic*

Part 3
Experiments

3.1 Compared Algorithms

- LPAb+
- LP-BRIM
- Adaptive BRIM

Uncover communities contain both types of nodes
→ divide detected communities

- BiSBM
  - #communities should be set

Non-deterministic
→ run 3 times

- BiLouvain

Require manual parameters
→ Set parameters following original paper

3.2 Evaluation Criteria

清华-伯克利深圳学院
Tsinghua-Berkeley Shenzhen Institute

- Internal
  - Normalized Mutual Information(NMI)

$$NMI(P_1, P_2) = \frac{-2\sum_{i=1}^{k_{P_1}}\sum_{j=1}^{k_{P_2}} N_{ij}\log(\frac{N_{ij}n}{N_i N_j})}{\sum_{i=1}^{k_{P_1}} N_i\log(\frac{N_i}{n}) + \sum_{j=1}^{k_{P_2}} N_j\log(\frac{N_j}{n})}$$

- N: confusion matrix with N_{ij} indicating #nodes in the *ith* cluster of the partition P_1 and the *jth* cluster of the partition P_2.
- N_i: #nodes in the *ith* cluster of the partition P_1
- k_{P_1}: #communities in P_1
- n: #nodes

3.2 Evaluation Criteria

- External
  - Coverage

$$Coverage(\mathcal{G}) = \frac{\sum_{i=1}^{k} e_{G_i}}{E}$$

    - measures the internal density within the subgraph G_i .
    - G_i: subgraph enclosing community C_l and its co-cluster mate D_m
    - e_{G_i}: #links in G_i
    - E: #all links

  - Bipartite Modularity Density (BMD)

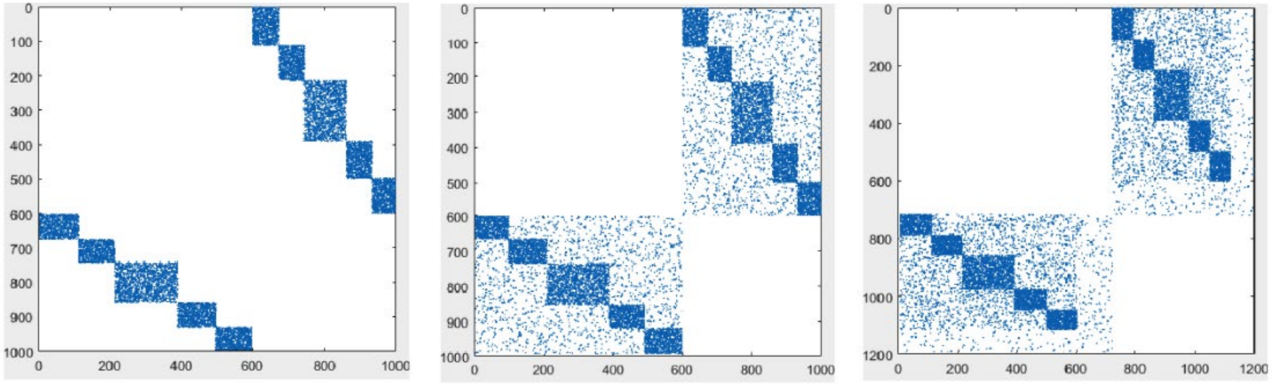
$$BMD(\mathcal{G}) = \sum_{i=1}^{k} BMD(G_i)$$

$$BMD(G_i) = D_{in}(G_i) - D_{out}(G_i)$$

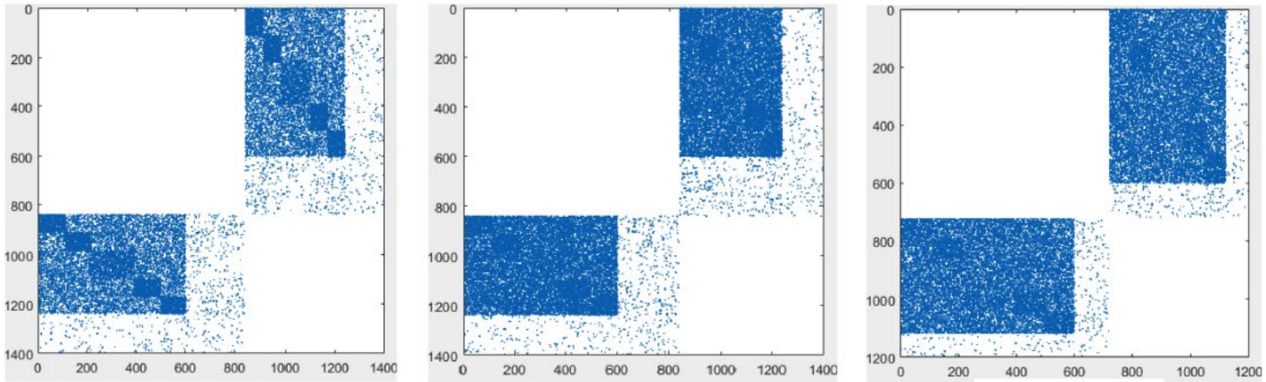
    - considers within- and between-subgraph density.

3.3 Synthetic Network Results

- ***mpx****:* average proportion of links between a node (of one type) and nodes (of the second type) located outside its co-cluster.
- $\boldsymbol{nd_{nd}}$: percentage of non-discriminating (i.e., sparsely connected) nodes.



(a) $mpx = 0$, $nd_{nd} = 0\%$ (b) $mpx = 0.2$, $nd_{nd} = 0\%$ (c) $mpx = 0.2$, $nd_{nd} = 20\%$



(d) $mpx = 0.4$, $nd_{nd} = 40\%$ (e) $mpx = 0.5$, $nd_{nd} = 40\%$ (f) $mpx = 0.6$, $nd_{nd} = 20\%$

3.3 Synthetic Network Results

TBSI 清华-伯克利深圳学院 Tsinghua-Berkeley Shenzhen Institute

Table 2. Results on Networks with $mxp = 0$ and Different Percentages of Sparsely Connected Node (n_{nd})

Algorithms	$n_{nd} = 0\%$	$n_{nd} = 10\%$	$n_{nd} = 20\%$	$n_{nd} = 30\%$	$n_{nd} = 40\%$
BiNeTClus	1	1	1	1	1
BiLouvain	1	1	1	1	0.98
Adaptive BRIM	1	1	0.98	0.96	0.96
LP-BRIM	1	1	0.98	0.98	0.96
BiSBM	1	0.79	0.72	0.66	0.61
LPAb+	1	1	0.98	1	0.95

Table 3. Results on Networks with $mxp = 0.2$ and Different Percentages of Sparsely Connected Node (n_{nd})

Algorithms	$n_{nd} = 0\%$	$n_{nd} = 10\%$	$n_{nd} = 20\%$	$n_{nd} = 30\%$	$n_{nd} = 40\%$
BiNeTClus	1	1	1	1	1
Bilouvain	1	1	1	1	1
Adaptive BRIM	0.97	0.92	0.87	0.85	0.71
LP-BRIM	1	0.98	0.97	0.94	0.79
BiSBM	1	0.79	0.71	0.66	0.54
LPAb+	1	1	1	1	0.8

3.3 Synthetic Network Results

Table 4. Results on Networks with $mxp = 0.4$ and Different Percentages of Sparsely Connected Node (n_{nd})

Algorithms	n_{nd} = 0%	n_{nd} = 10%	n_{nd} = 20%	n_{nd} = 30%	n_{nd} = 40%
BiNeTClus	0.98	**0.97**	**0.97**	**0.81**	**0.64**
BiLouvain	0.98	**0.97**	0.85	0.63	0.56
Adaptive BRIM	0.81	0.86	0.74	0.71	0.58
LP-BRIM	0.77	0.96	0.92	0.78	0.59
BiSBM	**1**	0.76	0.70	0.63	0.58
LPAb+	0.91	**0.97**	0.96	0.65	0.60

3.4 Real Network Results

- Five real-world bipartite networks:
  - Corporate Leadership: people V.S. companies
  - American Revolution: people V.S. organizations
  - Crime: people V.S. crimes
  - Malaria: genes V.S. gene substrings
  - arXiv: authors V.S. articles

- No ground truth
  - →only considered external criteria

Bipartite network	$\mid V_1 \mid$	$\mid V_2 \mid$	$\mid E \mid$
Corporate Leadership	20	24	99
Americain Revolution	136	5	160
Crime	829	551	1,476
Malaria	297	806	2,965
arXiv	16,726	22,015	58,595

3.4 Real Network Results

Table 6. Performance Results Evaluated with the Coverage

Algorithms	Corporate Leadership	Americain Revolution	Crime	Malaria	arXiv
BiNeTClus	**0.77**	**0.85**	0.90	0.68	**0.84**
BiLouvain	0.64	**0.85**	0.80	0.64	0.81
Adaptive BRIM	0.62	**0.85**	0.80	0.66	—
LP-BRIM	0.63	**0.85**	0.82	0.66	—
LPAb+	0.62	**0.85**	**0.96**	**0.75**	—

Table 7. Performance Results Evaluated with the Bipartite Modularity Density

Algorithms	Corporate Leadership	Americain Revolution	Crime	Malaria	arXiv
BiNeTClus	**1.5**	**0.62**	0.75	0.074	**0.78**
BiLouvain	−0.13	**0.62**	0.44	−0.20	0.66
Adaptive BRIM	−0.27	**0.62**	0.43	−0.01	—
LP-BRIM	−0.20	**0.62**	0.49	0.022	—
LPAb+	−0.27	**0.62**	**0.95**	**0.69**	—

Part 4
Conclusion

4.1 The Algorithm

TBSI 清华-伯克利深圳学院 Tsinghua-Berkeley Shenzhen Institute

- Parameter-free
- Capable of handling network with many atypical (i.e., sparsely or massively) connections

4.2 Take-away

- Improve one metric at a step
- Adopt joint strategy
- Writing style: friendly, logical and well-organized

THANK YOU