Community detection algorithms survey and overlapping communities

Presented by
Sai Ravi Kiran Mallampati
(sairavi5@vt.edu)

Outline

- Various community detection algorithms: Intuition *
- Evaluation of the discussed community detection algorithms *
- Overlapping communities in real-world networks **
- Results of a k-clique community detection algorithm in various networks **

* Meng Wang, Chaokun Wang, Jeffrey Xu Yu, Jun Zhang. Community Detection in Social Networks: An In-depth Bench-marking Study with a Procedure-Oriented Framework. VLDB 2015.
** Gergely Palla, Imre Derenyi, Illes Farkas, Tarmas Vicsek. Uncovering the overlapping community structure of complex networks in nature and society

Communities –Definition

In an undirected graph *G(V, E) (|V|=n; |E|=m)*,

Set of Communities: $Coms = \{V_1', V_2', .., V_{cn}'\}$

where $\bigcup_{i=1}^{cn} V_i' \subseteq V$ and

cn is the total number of communities

Coms should satisfy: $V_i' \bigcap V_j' = \phi$

Outliers - Definition

- Each node need not necessarily be in a community
- Outliers are nodes which cannot be grouped in to any of the communities
- Set of Outliers:

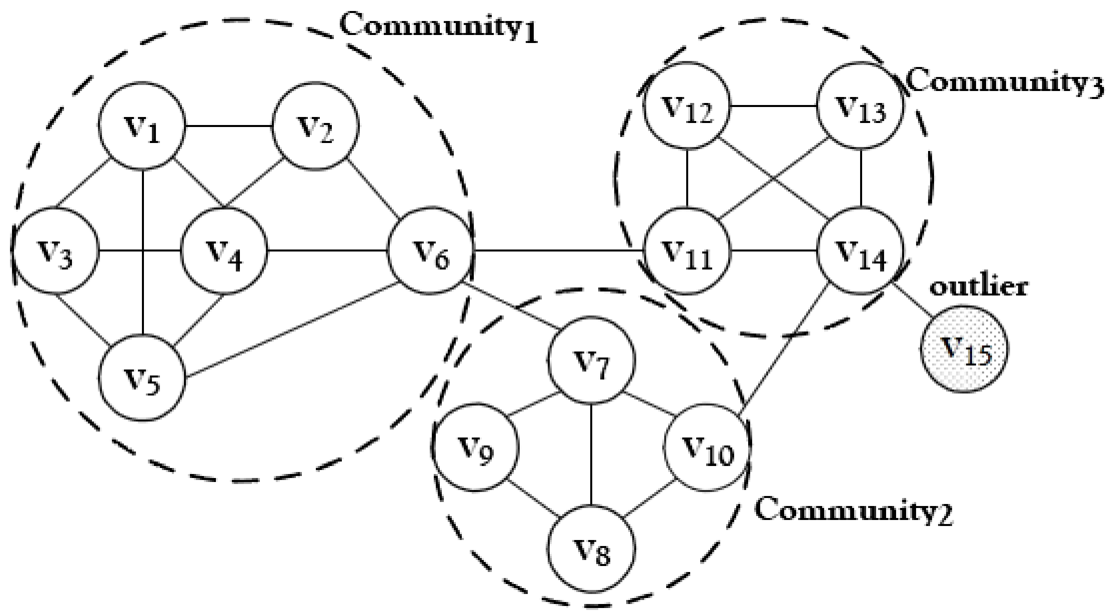
$$Outs = \{v \mid v \in V, \neg \exists V_i' \in Coms \wedge v \in V_i'\} = V - \bigcup_{i=1}^{cn} V_i'$$

- Outliers directly identified or produced by getting nodes from tiny groups whose size is less than the *minimal valid size (mvs)* of communities

$$Coms \bigcup Outs = V$$

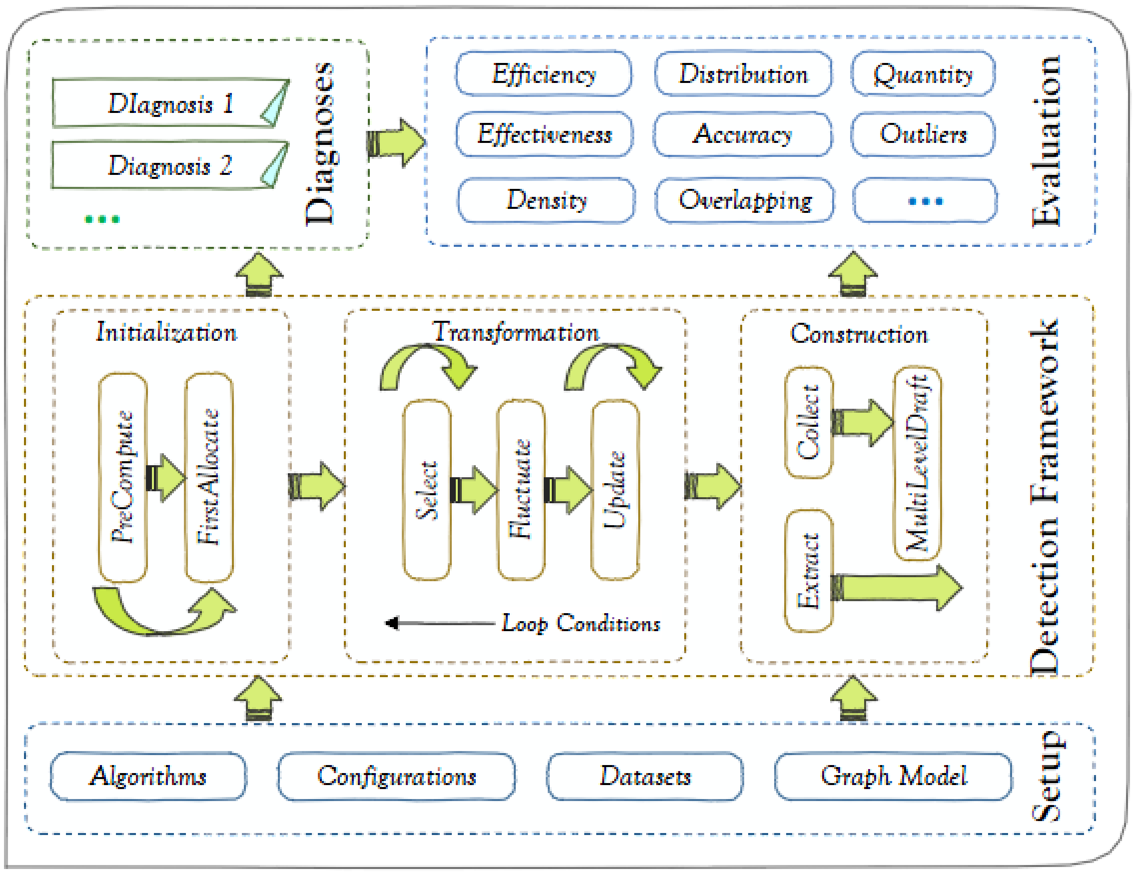
$$Coms \bigcap Outs = \phi$$

Communities and Outliers - Example



mvs = 2

Benchmark for community detection



Detection – Generalized procedure

Input: $G(V,E)$, mvs and T_{max}
Output: $R(Coms, Outs)$

initialize ϕ and Π;
$T \leftarrow 0$, $S_{R_{imp}} \leftarrow \emptyset$, $S_R \leftarrow \emptyset$, $Cad^T \leftarrow \emptyset$;
$\textsc{PreCompute}(G, \phi)$;
$R^T_{imp} \leftarrow \textsc{FirstAllocate}(G, \Pi)$;

INITIALIZATION PHASE

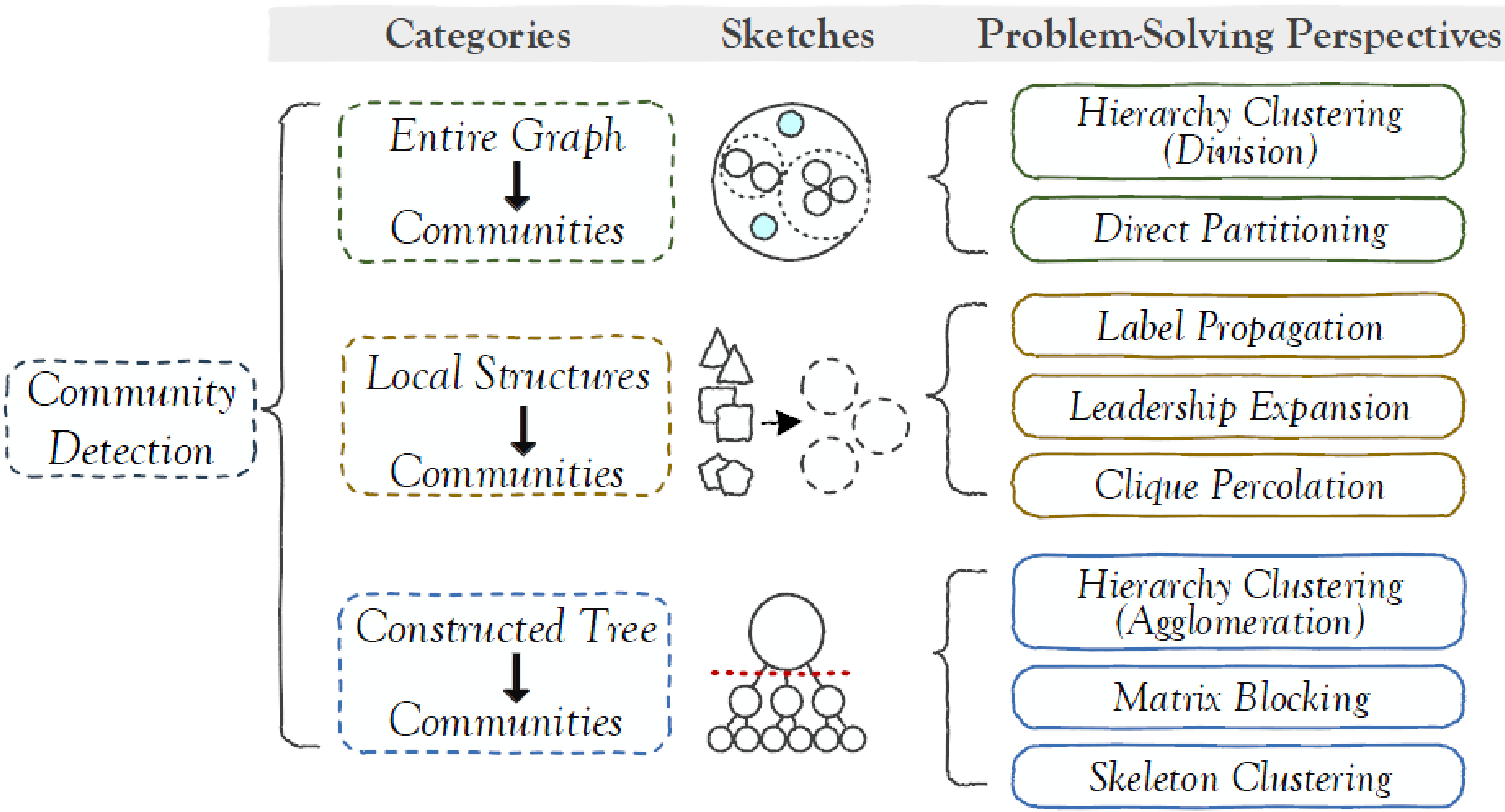
while $T \,!= T_{max} \&\& !\textsc{Stable}(\Pi) \&\& !\textsc{Optimal}(\phi)$ **do**
    $Cad^T \leftarrow \textsc{Select}(G, \Pi)$;
    $R^T_{imp} \leftarrow \textsc{Fluctuate}(Cad^T, \Pi)$;
    $\textsc{Update}(\textsc{Involve}(Cad^T), \phi)$;
    $S_{R_{imp}} \leftarrow S_{R_{imp}} \cup R^T_{imp}$;
    $T{+}{+}$;
end while

TRANSFORMATION PHASE

if $S_{R_{imp}}$ has multiple results **then**
    for each $level \in \Pi$ **do**
        $S_R \leftarrow S_R \cup \textsc{Collect}(S_{R_{imp}})$;
    end for
    $R \leftarrow \textsc{MultiLevelDraft}(S_R, \phi \text{ or } \psi)$;
else if $S_{R_{imp}}$ has no obvious result **then**
    $R \leftarrow \textsc{Extract}(\Pi, \phi \text{ or } \psi)$;
else R is obtained in the iteration;
end if
$R.Outs \leftarrow R.Outs \cup \textsc{Erase}(R.Coms, mvs)$;
$\textsc{Order}(R.Coms)$;
return R;

CONSTRUCTION PHASE

Community detection algorithms - Overview



Community detection - Parameters

- Propinquity measure (Φ): For a subset M of the graph G, propinquity measure gives nearness of the inner-connections
- Revelatory structure (π): Organize the graph elements and get the community structure from the intertwined connections among them

Hierarchy clustering

- Communities formed in a multi-level structure progressively on the bases of the original graph
- Two types:
  – Agglomeration algorithm
  – Division algorithm

Hierarchy clustering - Properties

Π = Hierarchy tree (Different for agglomerative and division algorithms)

Modularity is specific proposed division of a network into communities

Global modularity, $Q = \sum_{i=1}^{cn} \left[\frac{I_i}{m} - \left(\frac{2I_i + O_i}{2m} \right)^2 \right]$

I_i – number of internal relationships within C_i

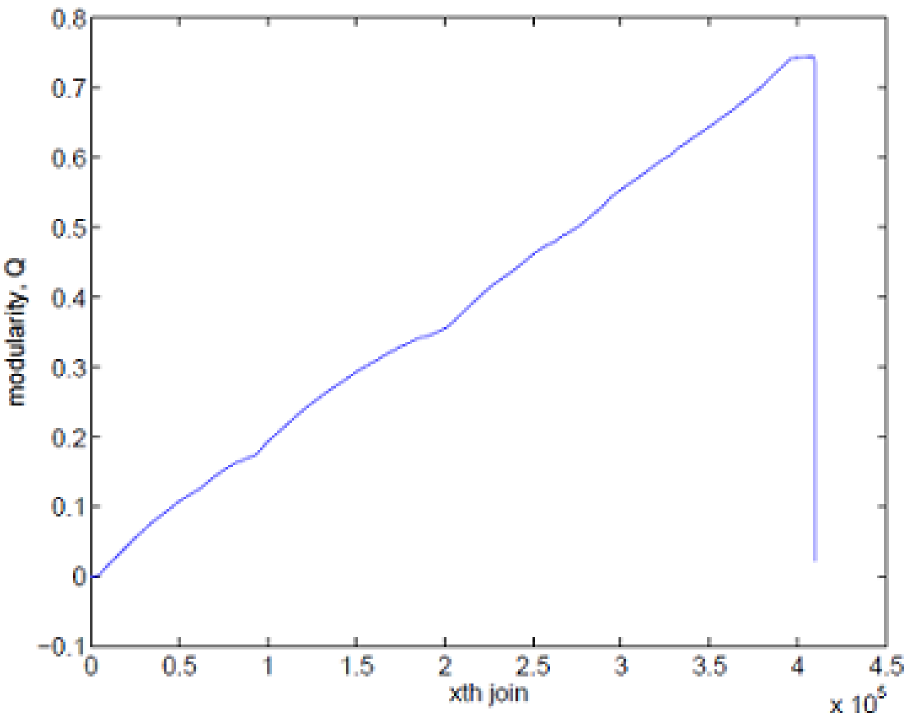
O_i – number of outgoing relationships between nodes in C_i and any node outside

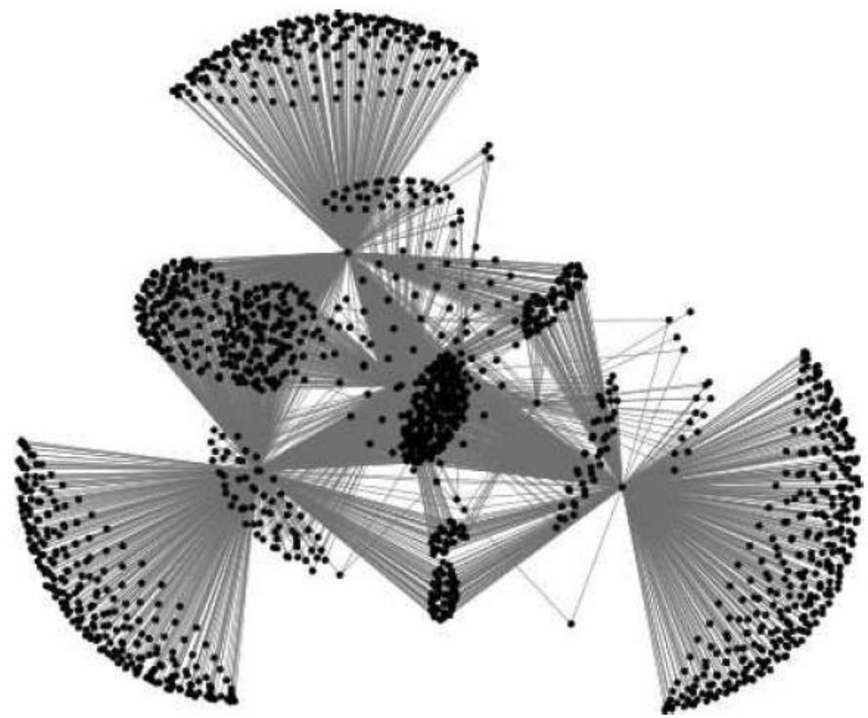
Φ = Q → To be optimized

Hierarchy clustering - Implementation

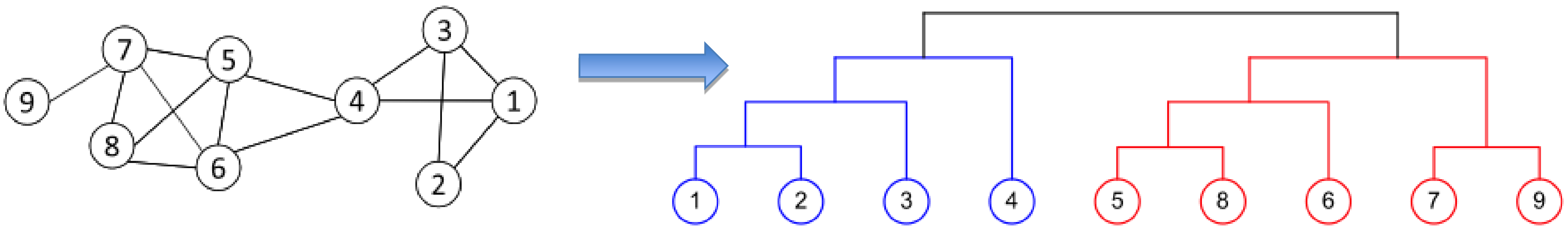
1. Initialize Φ(G)
2. Each node should be taken as a single tree
3. Choose two communities (candidate trees) whose combination may lead to a maximum increase of Φ in the current graph
4. Combine them to form a new tree (community)
5. If $\Delta\Phi < 0$ and Φ of the current graph is optimal, Gather the result at each level in π
6. Else, repeat steps 3 and 4

Hierarchy clustering – Graph visualization





Hierarchy clustering – Example



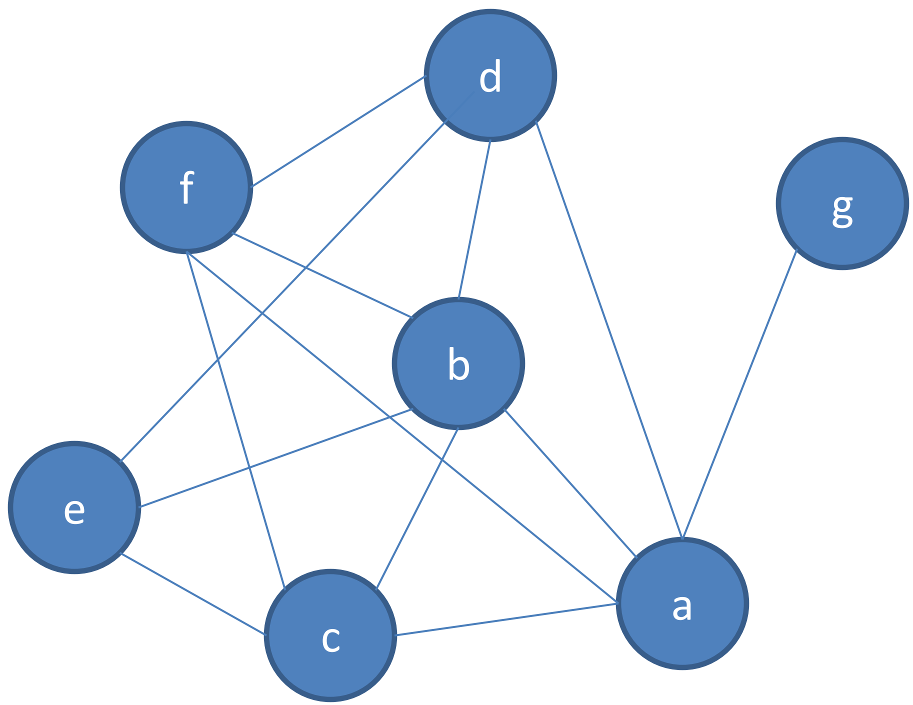
Direct Partitioning

- Partitioning a graph directly gives communities
- Each node $v \in \pi$ has the degree $d(v) \geq k$ and each relationship has at least *k triangles*
- Φ(r) of a relationship is the number of triangles containing r

Direct Partitioning - Implementation

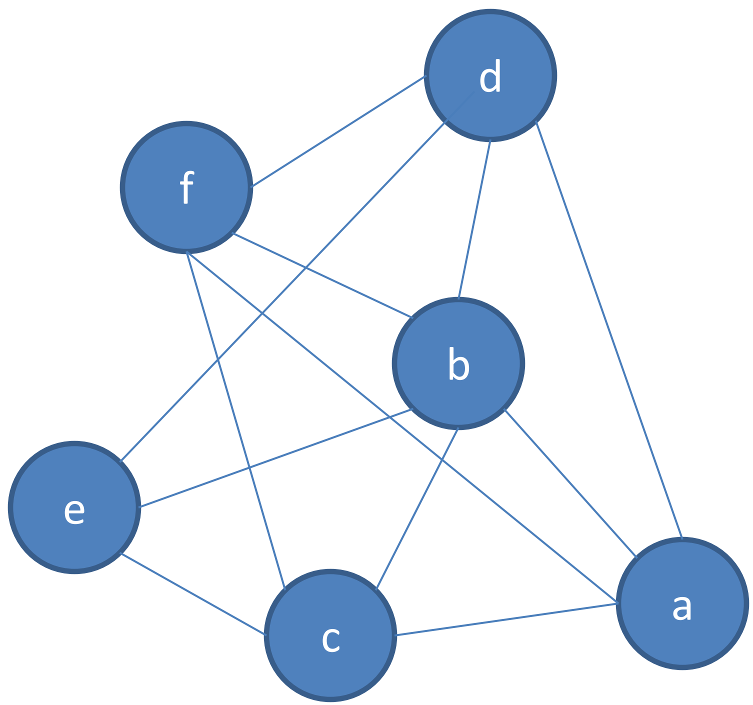
- Mark all nodes with degree < k + 1 as outliers and remove these nodes
- Initialize $\Phi(r)$ for each relationship
- Select relationships for whom $\Phi(r) < k$ and remove all these edges from the current graph
- When there is no edge to be marked, we get several connected components
- Each living relationship belongs to *k* triangles so that the mutual friends between a pair of connected nodes is maximum

Direct Partitioning - Example



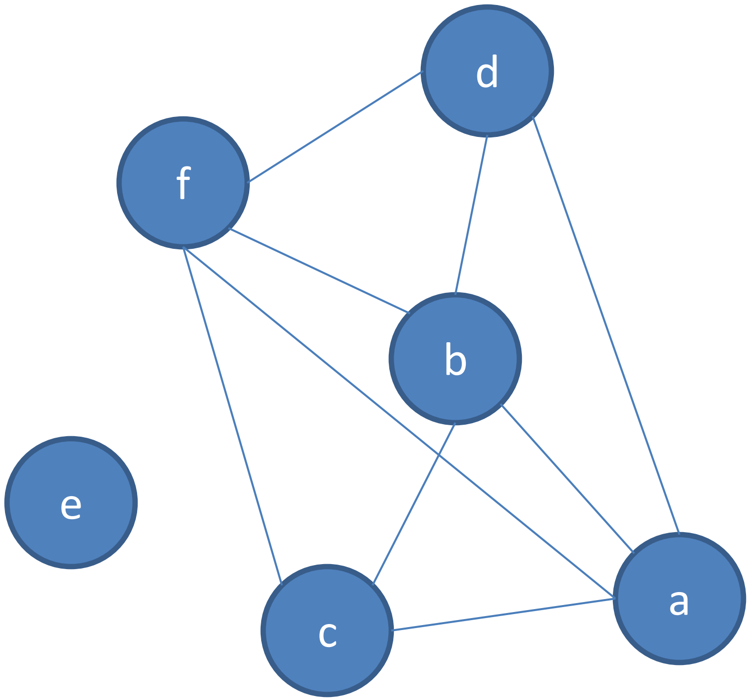
k=2

Direct Partitioning - Example



Remove node ‘g’ from the graph since it has degree less than k+1
Add it to outliers
Remove all edges which have number of triangles < k
Remove edges <e, b>, <e, c>, <e, d>

Direct Partitioning - Example

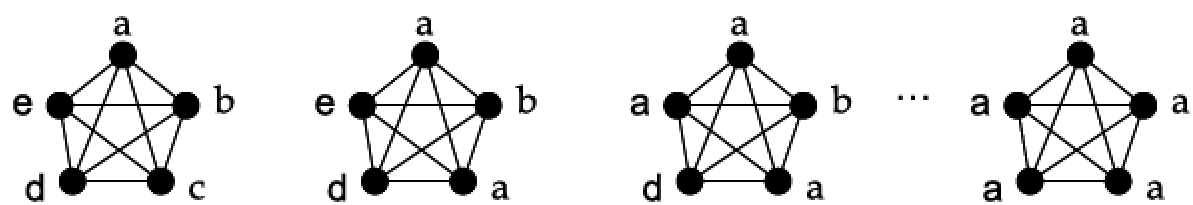


Add node ‘e’ to outliers
Add the remaining nodes to a community

Label Propagation

- Community detection by different labels spreading among neighbors
- Label distribution with different labels for different communities
- Implementation:
  1. Initially, nodes are assigned unique labels
  2. Each node changes its label to the most frequent one of its neighbors' labels
  3. Repeat the previous step for T_{max} times

Label Propagation - Example



Leadership expansion

$$\phi(v_i) = d(i)$$

- Find the degree for each node in the graph
- Elect the *ld* nodes with most degree in the graph such that neighborhoods of any two leaders have an intersection less than λ_{max}

Leadership expansion Contd.

- Each node explores its neighbors within *l* hops and counts the common neighbors with each leader
- If common neighbors with any leader > λ_{min} the node joins that leader's community
- New leaders are re-elected by re-computing $\Phi(v_i)$ for the graph at each step

Clique Percolation

- All cliques of size k are found out
- Two cliques are adjacent if they share k-1 nodes → Clique graph
- Each connected component in the clique graph forms a community

Matrix Blocking - Properties

- Community detection by finding dense subgraphs
- Π – Hierarchy tree

$$\phi(v_i, v_j) = \frac{< A(:,i), A(: j) >}{| A(:,i) \| A(:, j) |}$$

where *A* is the adjacency matrix of *G*

Matrix Blocking - Implementation

1. First, compute $\Phi(v_i, v_j)$ of each pair of nodes
2. Sort the triplets (i, j, $\Phi(v_i, v_j)$) in decreasing order to form a queue CQ
3. Pop the triplet with maximum Φ and find trees corresponding to nodes *i* and *j*
4. Combine two trees to form a new branch in Π if there is no common ancestor
5. Repeat 3 and 4 until the tree is built (n-1 times)

Skeleton clustering

- Π – Hierarchy tree

$$\phi(r_{ij}) = \frac{|N(i) \bigcap N(j)|}{\sqrt{N(i)}\sqrt{N(j)}}$$

N is a self-contained neighborhood of a node

Dataset information

Small scale real-world networks with ground truth communities

Name	n	m	cn	Supp.
Strike	24	38	3	/
Football	180	788	11	61 outliers, 4 hubs

Large scale real-world networks

Name	n	m	cc_{avg}	$diam$
CA-HepPh	12,008	118,505	0.6115	13
Email-Enron	36,691	183,830	0.4970	11
BrightKit	58,228	214,078	0.1723	16
Gowalla	196,591	950,327	0.2367	14
DBLP	317,080	1,049,866	0.6324	21
Amazon	334,863	925,872	0.3967	44
YouTube	1,134,890	2,987,624	0.0808	20

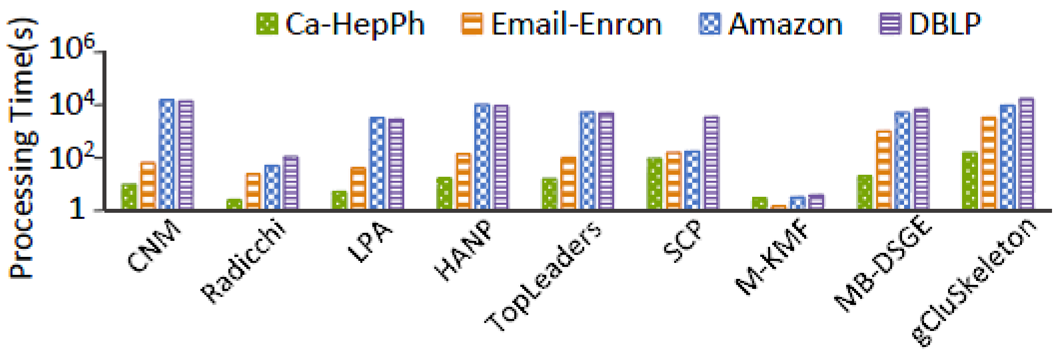
Synthetic networks with ground truth communities

Name	n	m	d	d_{max}	c_{min}	c_{max}
S_10K	10,000	50,302	10	50	20	100
S_100K	100,000	504,371	10	50	50	100
S_200K	200,000	953,230	10	100	60	200
S_500K	500,000	2,938,555	10	250	100	500

Evaluation of the detection algorithms

- Efficiency
- Accuracy
- Effectiveness
- Outliers

Evaluation - Efficiency

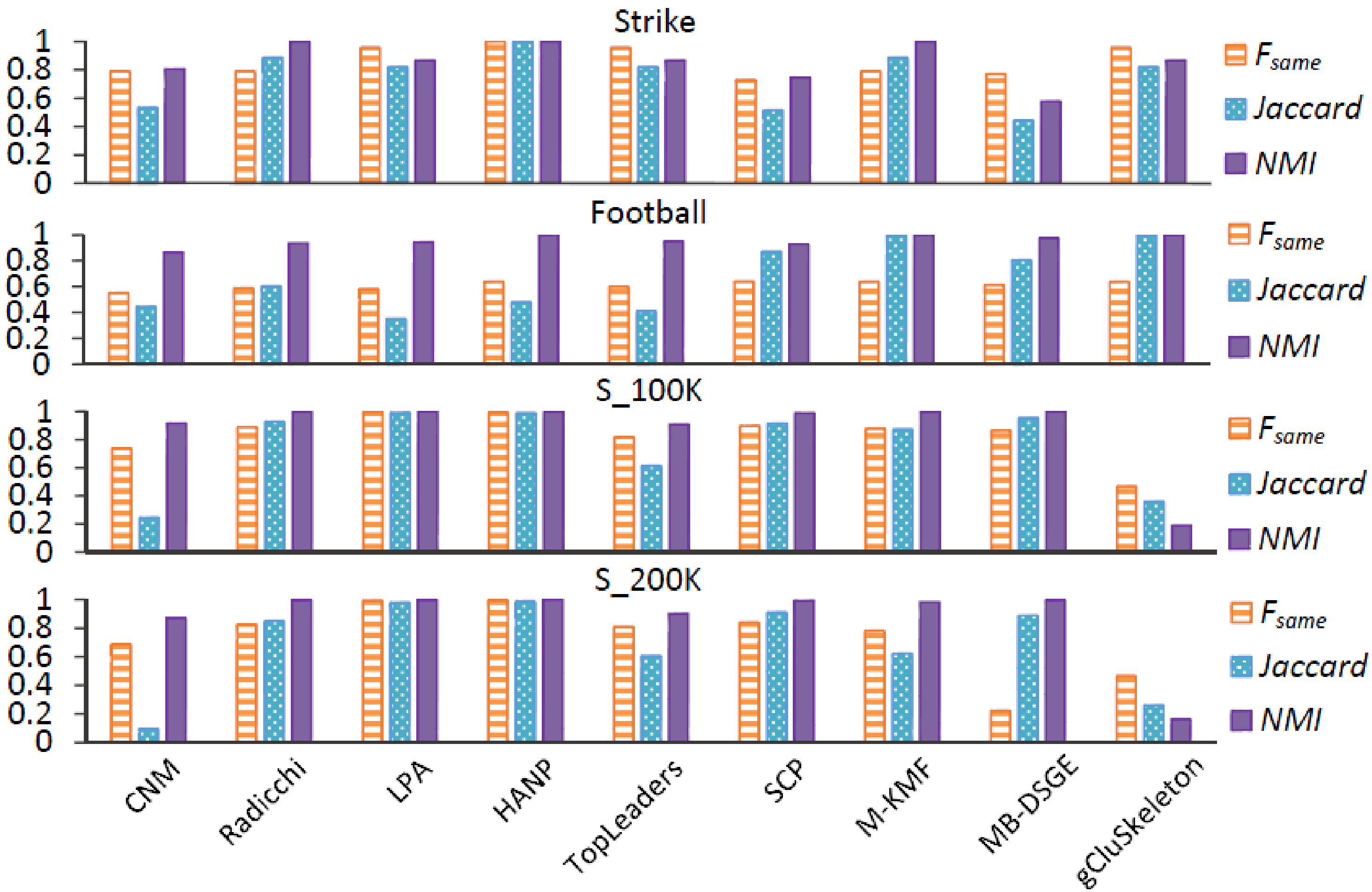


Algorithm	Detection method
CNM	Hierarchy clustering agglomeration
Radichi	Hierarchy clustering division
LPA, HANP	Label propagation
TopLeaders	Leadership expansion
SCP	Clique percolation
M-KMF	Direct Partitioning
MB-DSGE	Matrix Blocking
gCluSkeleton	Skeleton clustering

Evaluation – Accuracy metrics

- Cross Common Fraction (F_{same}) compares each pair of communities in which one comes from the detected result and the other comes from the real one, to find the maximal shared parts
- Jaccard-index compares the number of node pairs in algorithm-produced results and the ground truth communities
- Normalized mutual information (NMI) stands for the agreement of two results

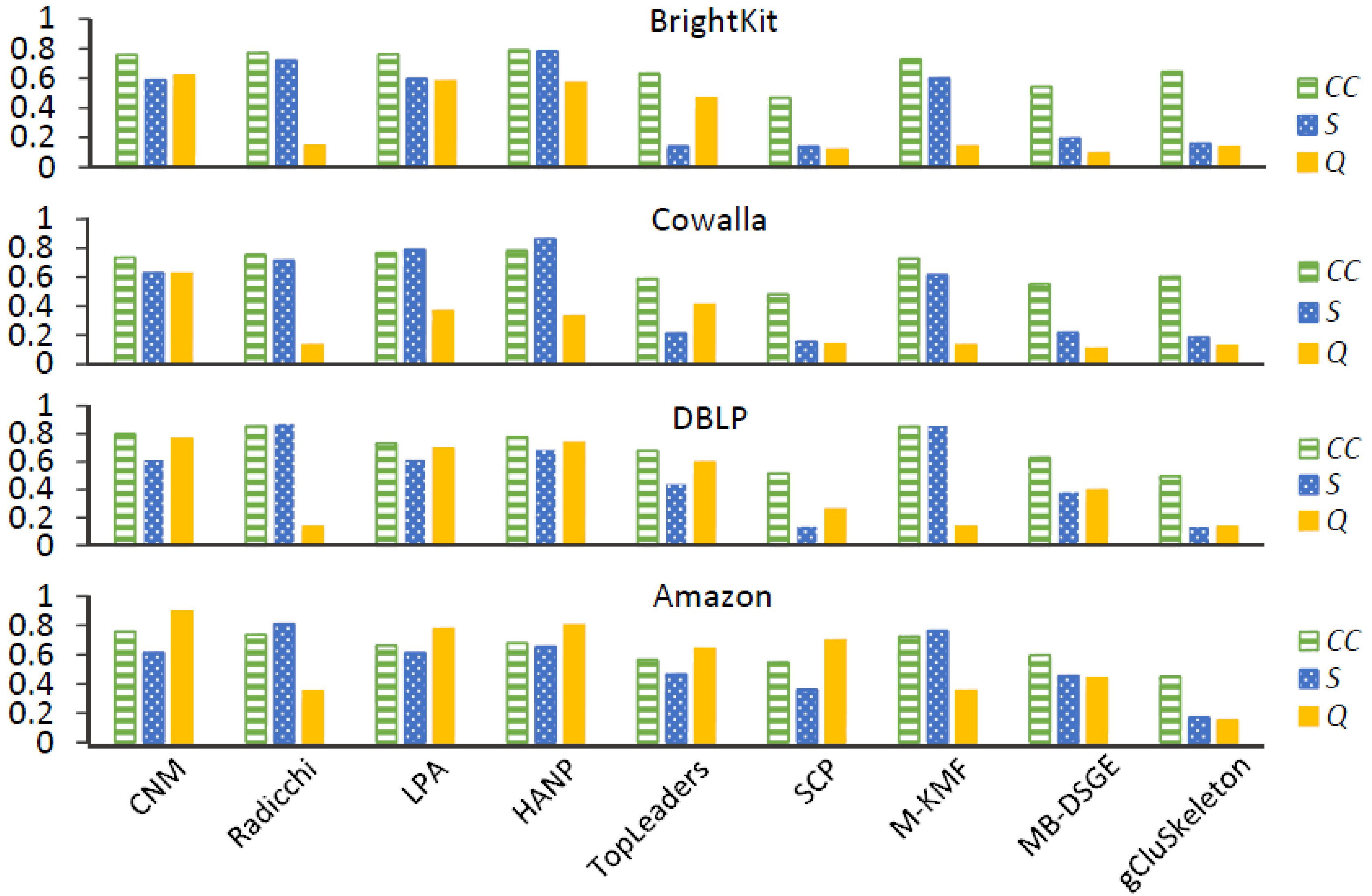
Evaluation - Accuracy



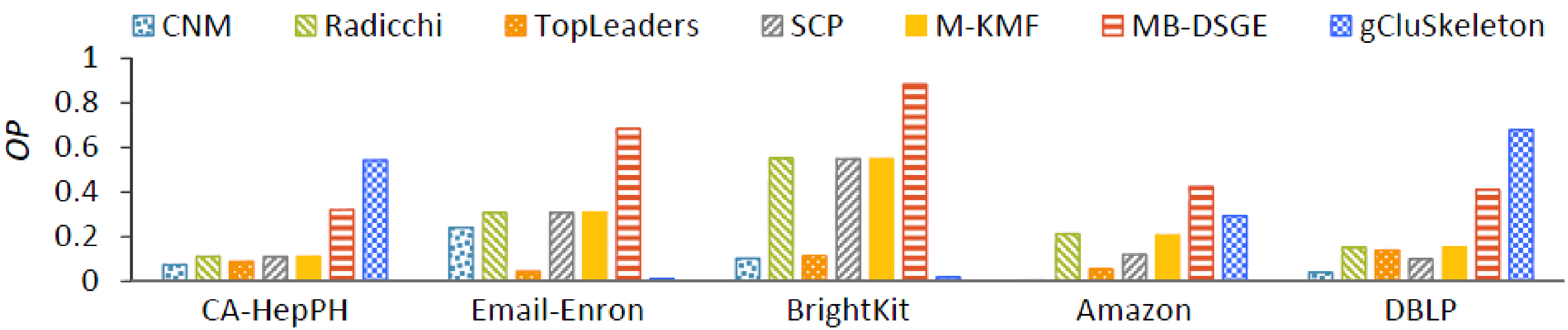
Evaluation – Effectiveness metrics

- Clustering coefficient: Tendency of nodes to cluster together
- Strength: Most members are in the same community with their neighbors
- Modularity: Measures how well the nodes are assigned to different communities

Evaluation - Effectiveness

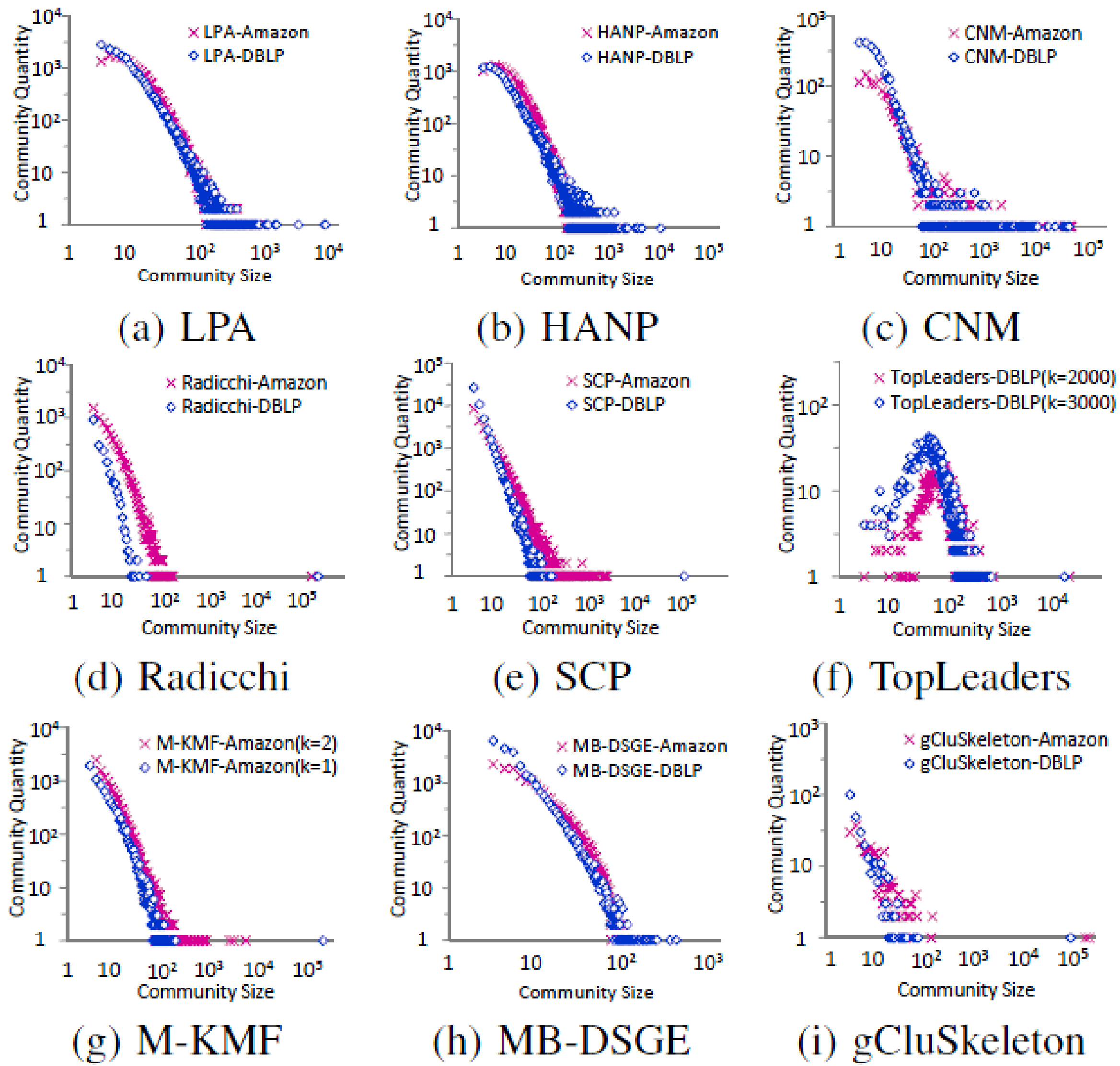


Evaluation – Outliers



OP = |Outs|/|V|

Community distribution

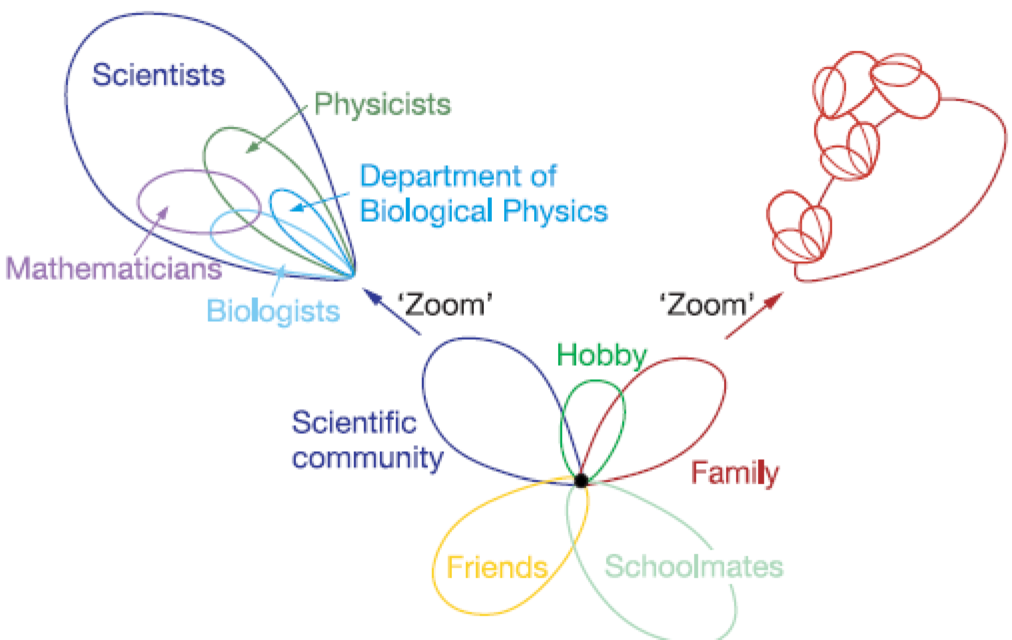


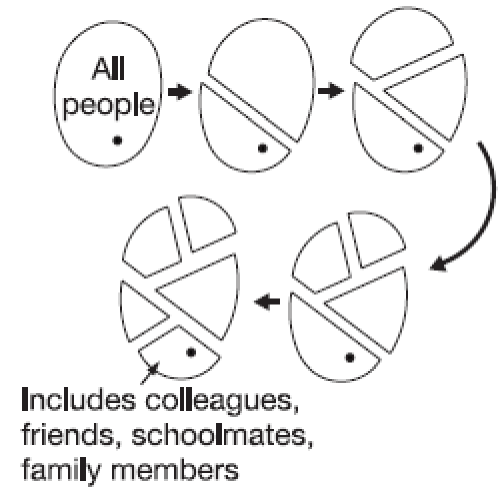
Overlapping communities and detection

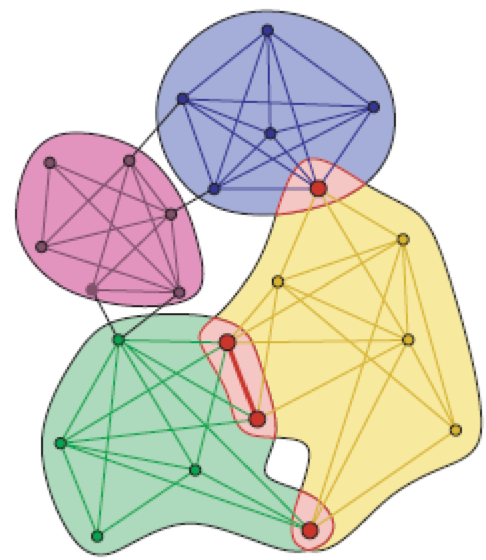
Overlapping communities

- In a graph, a node might belong to multiple communities
- This scenario not handled by Non-overlapping community detection methods
- Real networks have communities which overlap and are nested

Overlapping communities – Contd.







Overlapping communities – Properties

Membership number, m_i is the number of communities that the node i belongs to

Two communities α and β can share $s_{\alpha,\beta}^{ov}$ nodes, the overlap size between these communities

Number of links in community α is its community degree, d_{α}^{com}

Size of a community α is the number of nodes in alpha, s_{α}^{com}

k-clique community finding algorithm

- A member in a community is linked to many other members, but not necessarily to all other nodes in the community
- A community can be regarded as a union of cliques with smaller size → k-cliques where k is the number of nodes in each of these cliques
- Adjacent k-cliques → Two k-cliques are adjacent if they share *k-1* nodes
- k-clique-community → Union of all k-cliques that can be reached from each other through a series of adjacent k-cliques

k-clique community finding Contd.

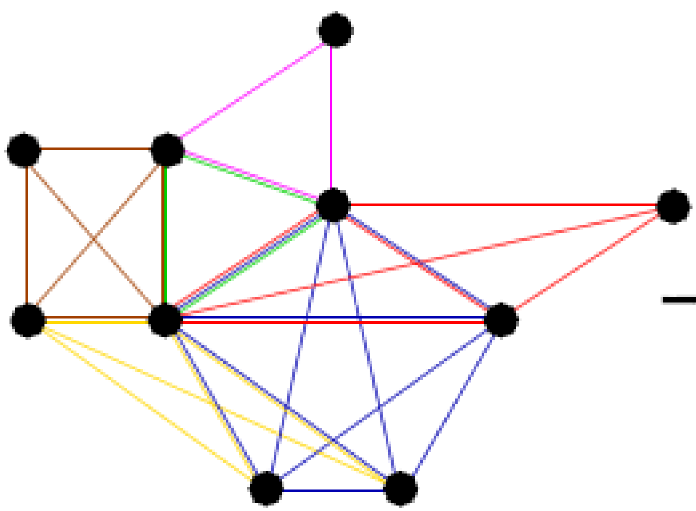
The k-clique communities shrink as k increases

- For k=2, the k-clique-communities simply have to share a node
- For k=3, the k-clique-communities have to share a link with each other

Finding k-clique communities

- We find k-clique communities from cliques in an undirected graph
- We first locate cliques
- We then construct the clique-clique overlap matrix as follows:
  - Row/Column indicates clique
  - Matrix elements → No. of common nodes between two cliques
  - Diagonal entries → Clique size

Finding k-clique communities Contd.



	5	3	2	1	3	1
	3	4	2	1	1	1
	2	2	3	2	1	2
	1	1	2	3	0	1
	3	1	1	0	4	2
	1	1	2	1	2	4

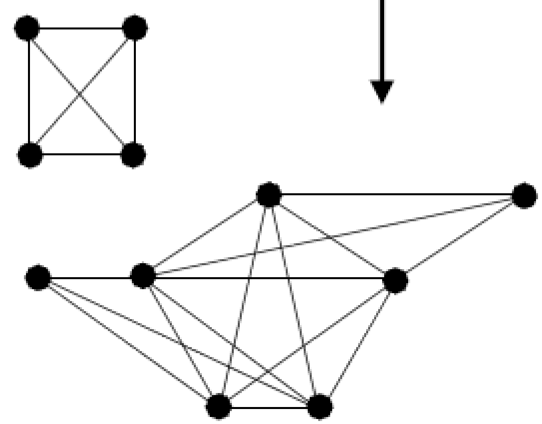
Finding k-clique communities Contd.

- Replace all entries of cliques that have < k-1 elements common with other cliques with 0
- Replace all off diagonal entries in clique-clique overlap matrix that are less than k with 0
- Replace non-diagonal entries in the matrix that are greater than k-1 with 1

Finding k-clique communities Contd.

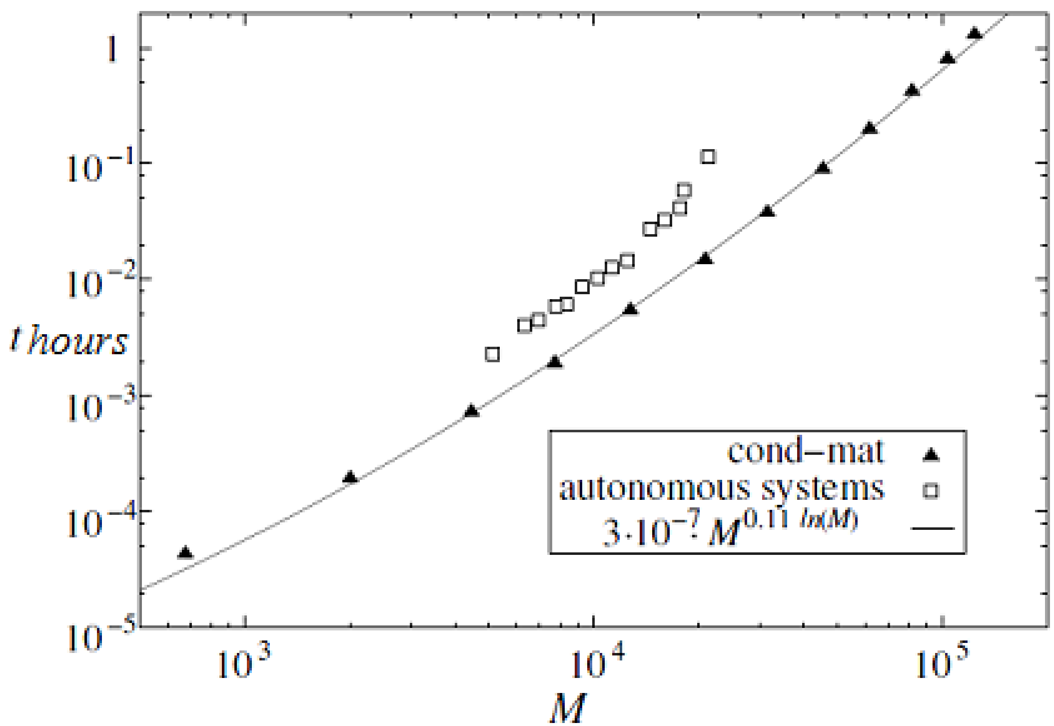
	5	3	2	1	3	1
	3	4	2	1	1	1
	2	2	3	2	1	2
	1	1	2	3	0	1
	3	1	1	0	4	2
	1	1	2	1	2	4

	1	1	0	0	1	0
	1	1	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
	1	0	0	0	1	0
	0	0	0	0	0	1

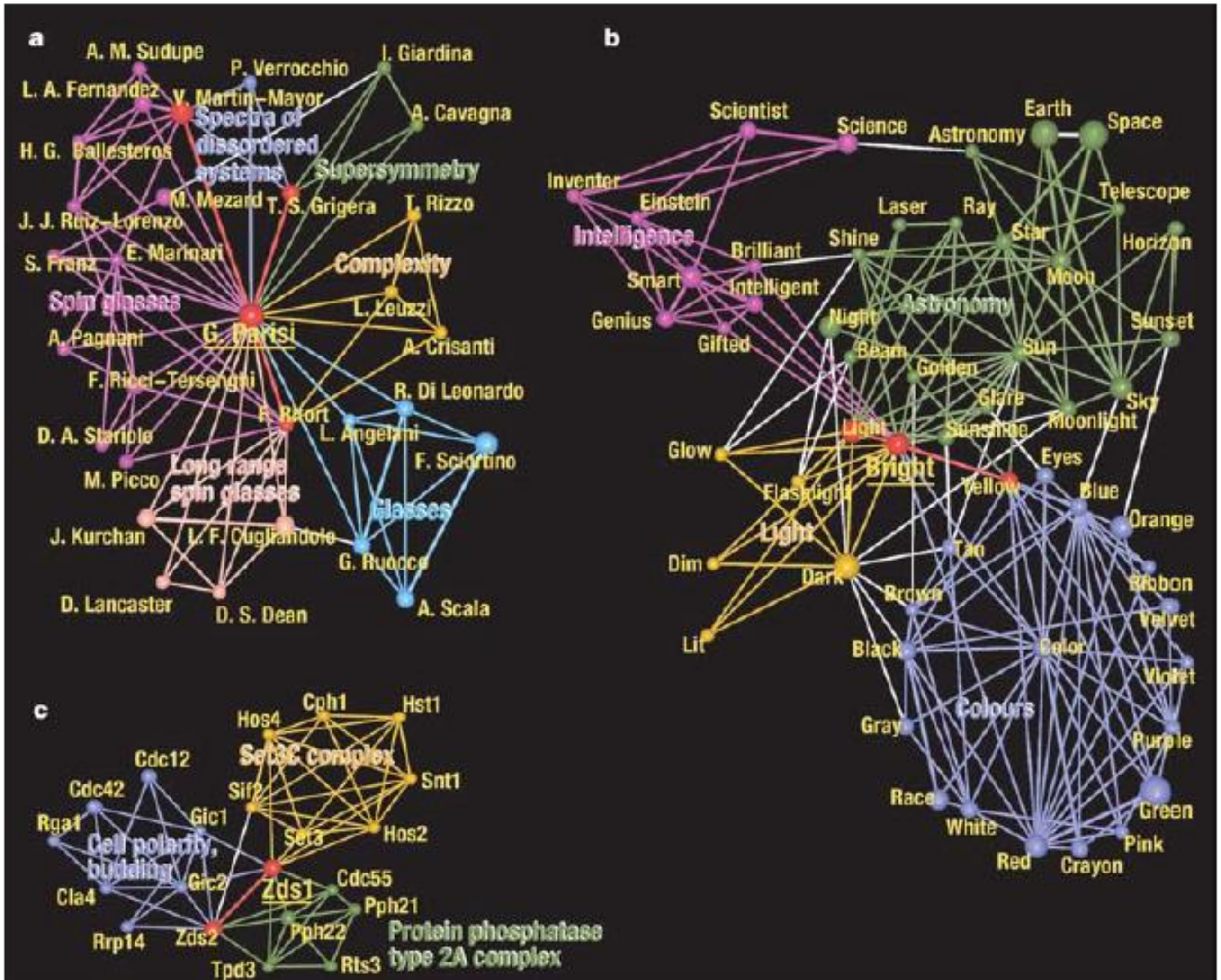


Finding k-clique communities Contd.

Finding full set of k-clique communities is found in exponential time for improved accuracy

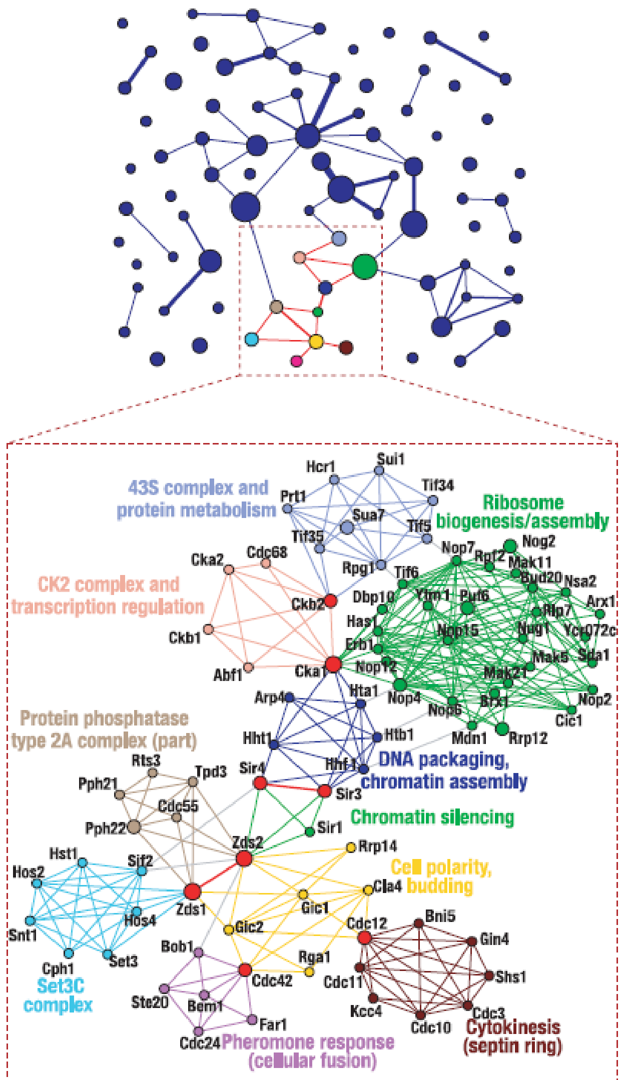


k-clique communities in real networks

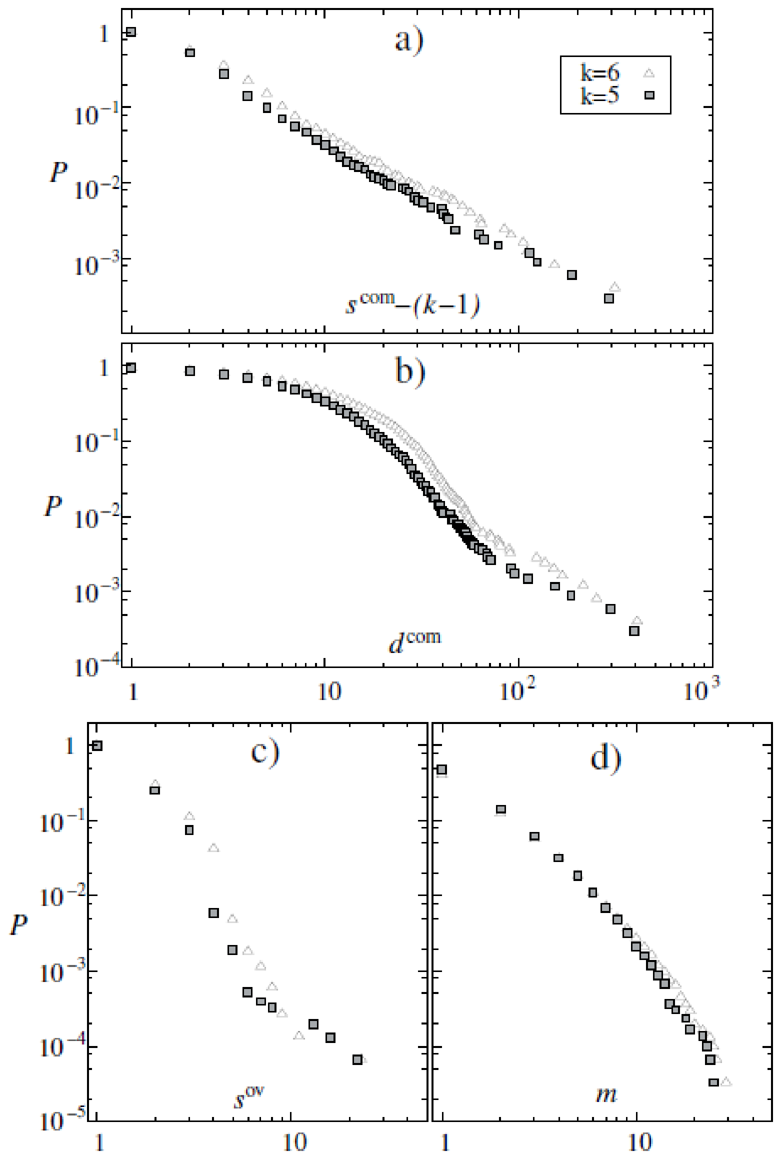


a. Communities in the co-authorship network
b. Communities in word association network
c. Protein-protein interaction

Finding k-clique communities – Protein-Protein interaction



Community statistics at different k values



Co-authorship network of the Los Alamos condensed matter e-print archive
Squares – k=5
Triangles – k=6

Cumulative distributions of the
a. community size
b. k-clique community degree
c. overlap size
d. membership number

Overlapping communities on weighted and directed links

- Arbitrary network → binary network
  - Directionality ignored in edges
  - All edges with weights less than threshold weight w* to be removed
- Prune weaker links and retain stronger ones
- f* - Fraction of links stronger than w*

Overlapping communities on weighted and directed links Contd.

- High k and w* → Less communities
- At a certain critical point, the largest community becomes twice as big as the second largest one

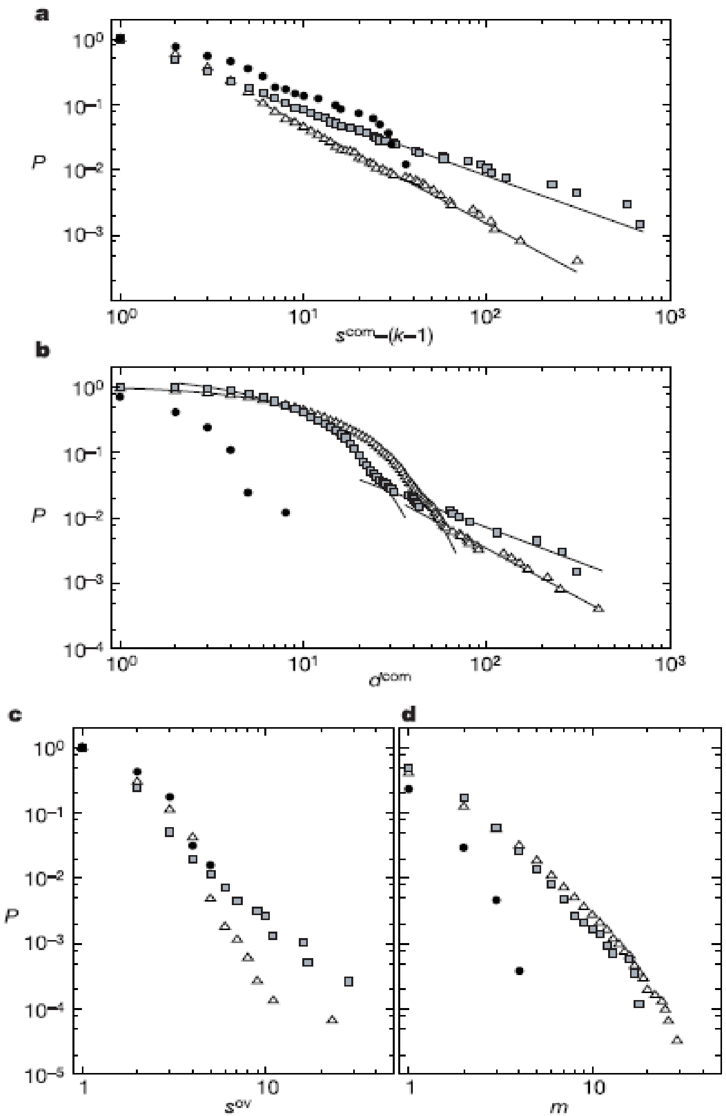
Community statistics on real networks

Statistical properties of the network of communities

Network	N^{com}	$\langle d^{com} \rangle$	$\langle C^{com} \rangle$	$\langle r \rangle$
Co-authorship	2,450	12.10	0.44	0.58
Word association	670	11.33	0.56	0.72
Protein interaction	82	1.54	0.17	0.26

N^{com} is the number of communities, $\langle d^{com} \rangle$ is the average community degree, $\langle C^{com} \rangle$ is the average clustering coefficient of the network of communities, and $\langle r \rangle$ is the average fraction of shared nodes in the communities.

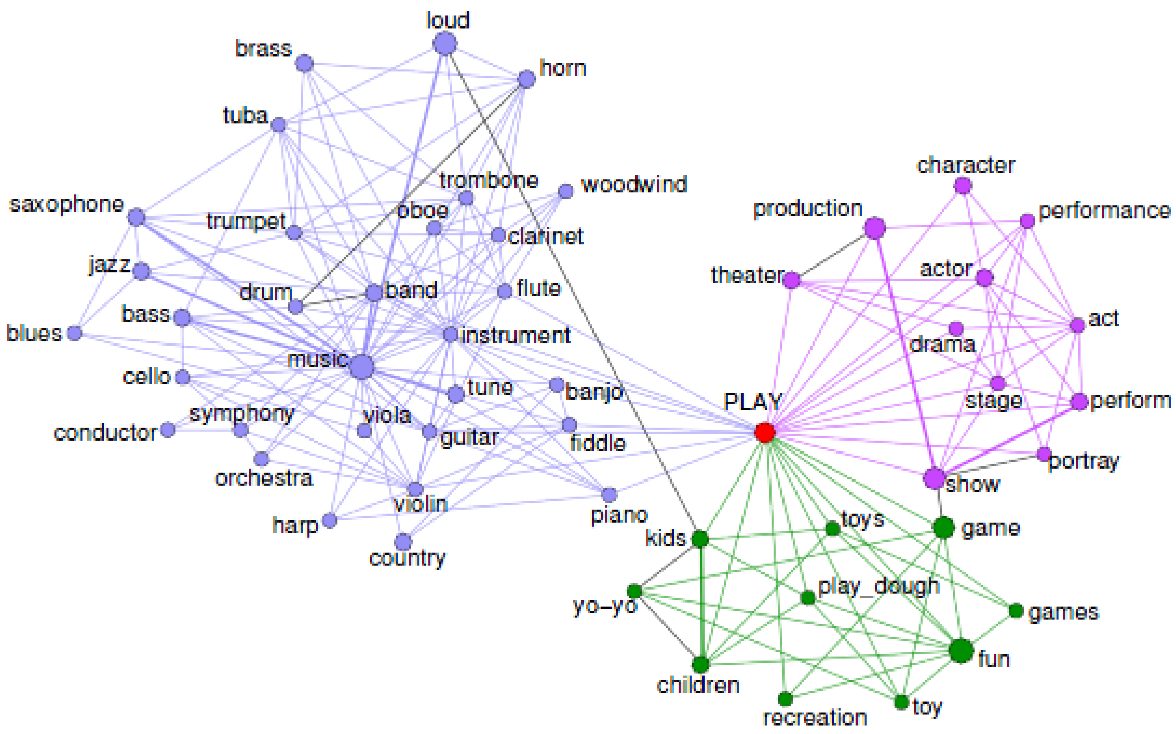
Community statistics on real networks



Co-authorship network of the Los Alamos Condensed Matter archive (triangles, k = 6, f* = 0.93),
The word association network of the South Florida Free Association norms (squares, k = 4, f* = 0.67), and
The protein interaction network DIP database (circles, k = 4).

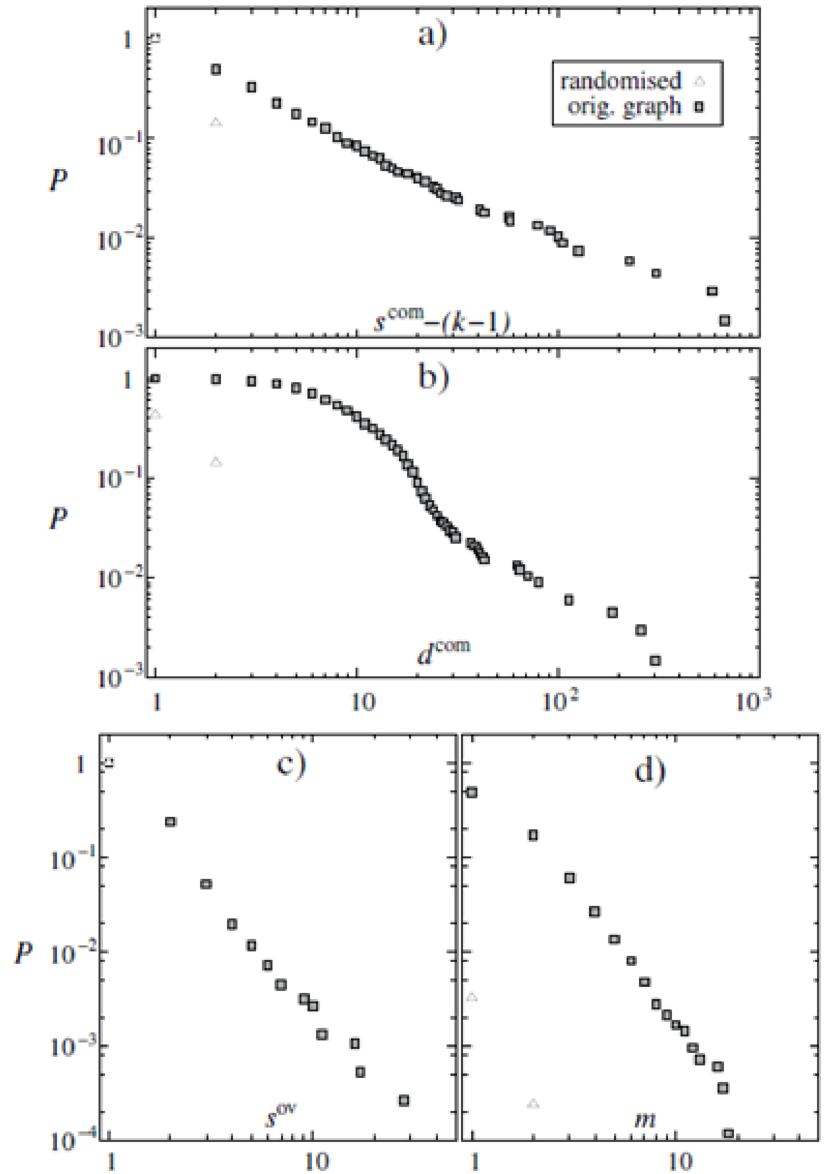
Cumulative distribution of the:
a. community size
b. community degree
c. overlap size
d. membership number

Communities on real networks



K-clique communities of the word 'PLAY' in the South Florida Free Association norm list for k = 4 and w*=0.025

Community statistics on random graphs



Word association network of the South Florida Free Association norm list.
Squares – original graph
Triangles – randomized graph

Cumulative distributions of the
a. community size
b. community degree
c. overlap size
d. membership number

Conclusion

- We discussed the various community detection methods and evaluated them
- Overlapping communities were discussed
- Studied the k-clique community finding and the cumulative distributions for various real networks and random graphs