

Subgraphs and Community Structure of Networks

Saptarshi Ghosh

Department of CSE, IIT Kharagpur

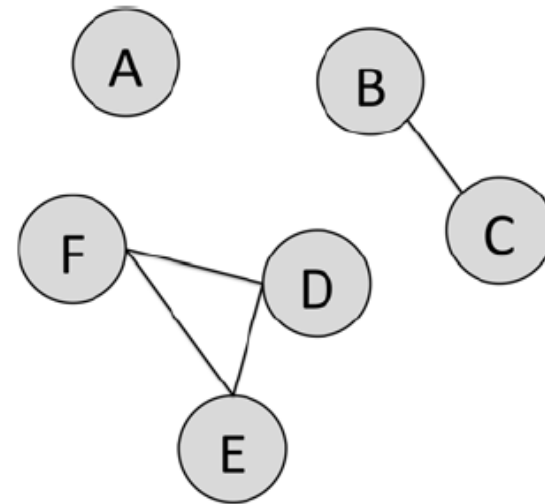
Social Computing course, CS60017

Subgraphs

- A subset of nodes and edges in a network
- Given a (social) network, what are some subgraphs of interest?

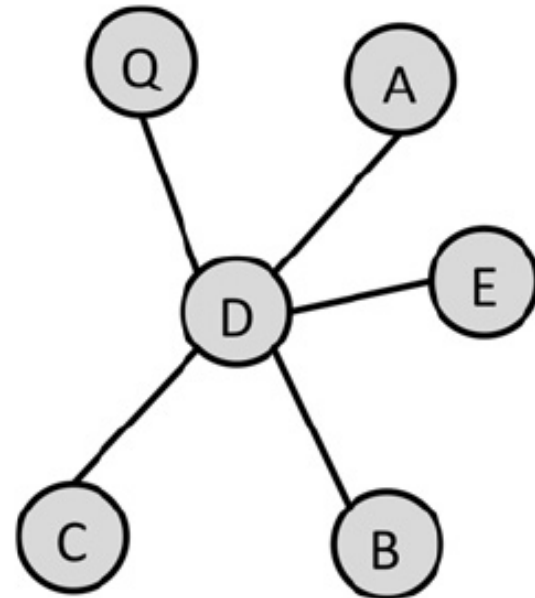
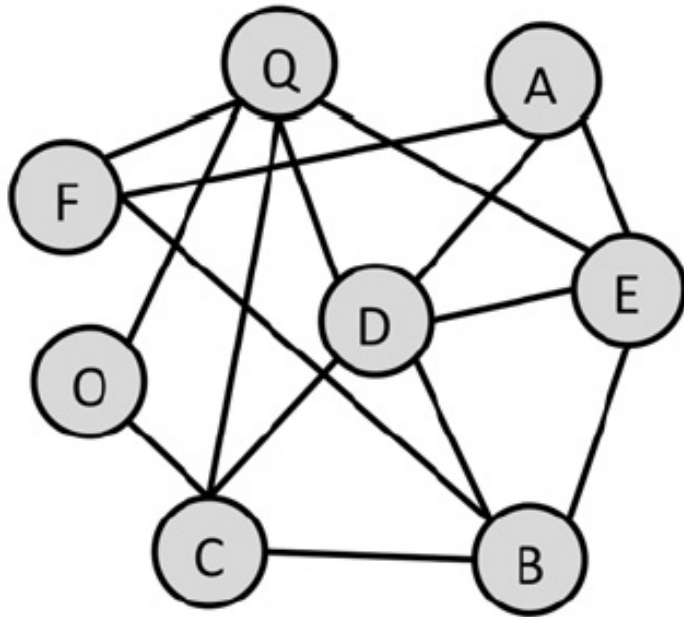
Subgraphs

- A subset of nodes and edges in a network
- Given a (social) network, what are some subgraphs of interest?
 - Singletons: Isolated nodes
 - Connected components
 - Triads or triangles
 - Larger cliques



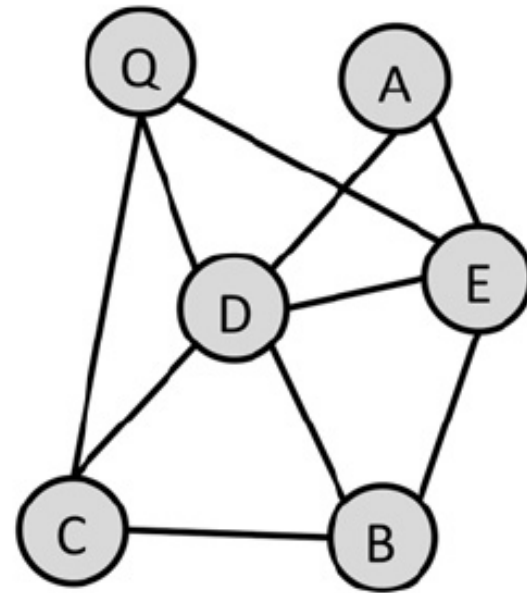
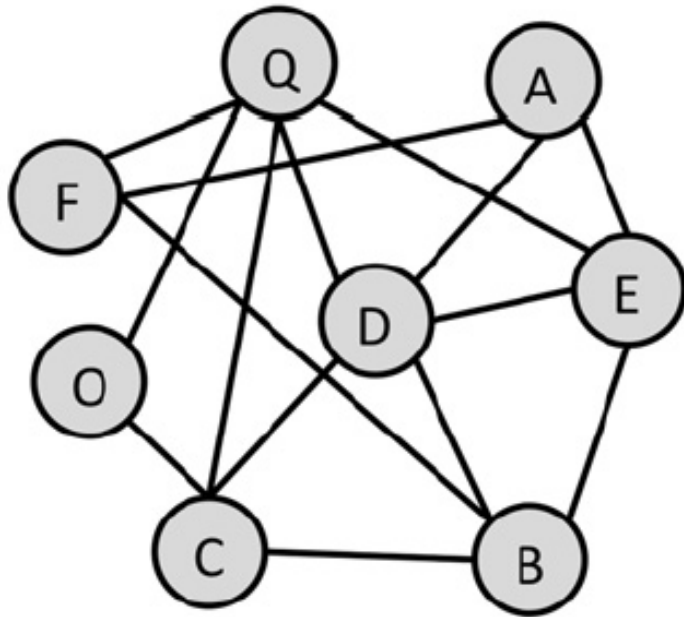
Egocentric networks

- From the perspective of a node (user)
- **1-degree egocentric network**: a node and all its connections to its neighbors



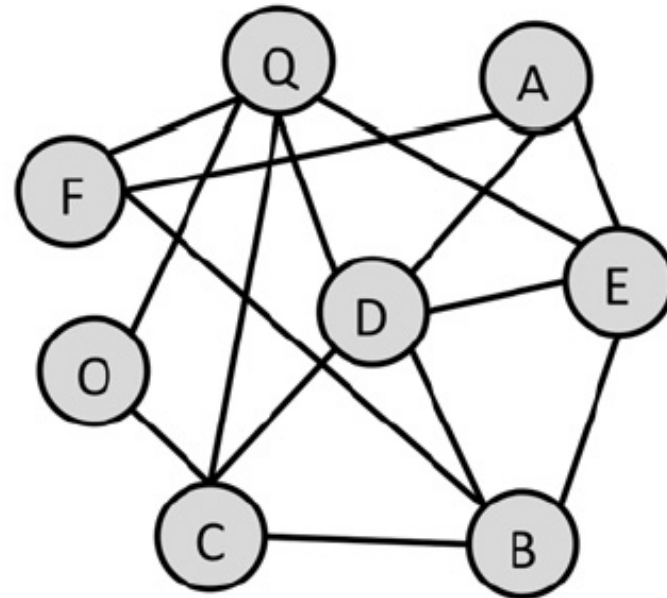
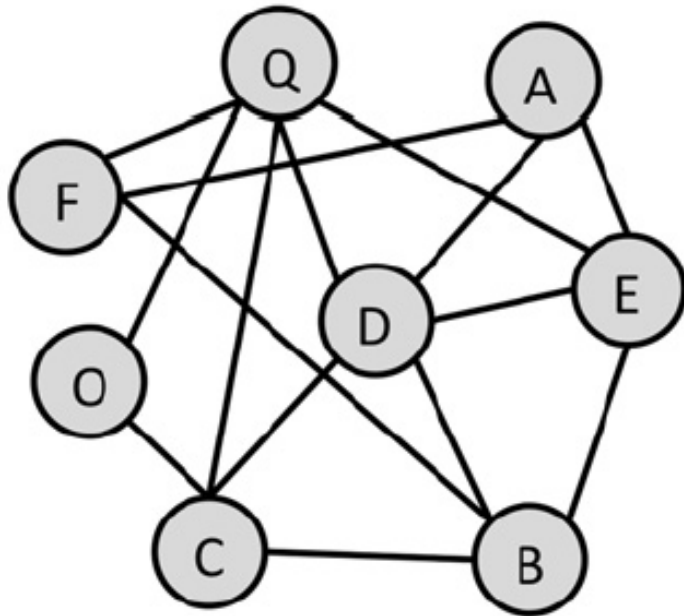
Egocentric networks

- **1.5-degree egocentric network**: a node, all its connections to its neighbors, and the connections among the neighbors



Egocentric networks

- **2-degree egocentric network**: a node, all its neighbors, all neighbors of neighbors, and the connections among all these nodes



Communities

- Community or network cluster
 - Typically a group of nodes having more and / or better interactions among its members, than between its members and the rest of the network
- No unique formal definition

COMMUNITY DETECTION

Community detection algorithms

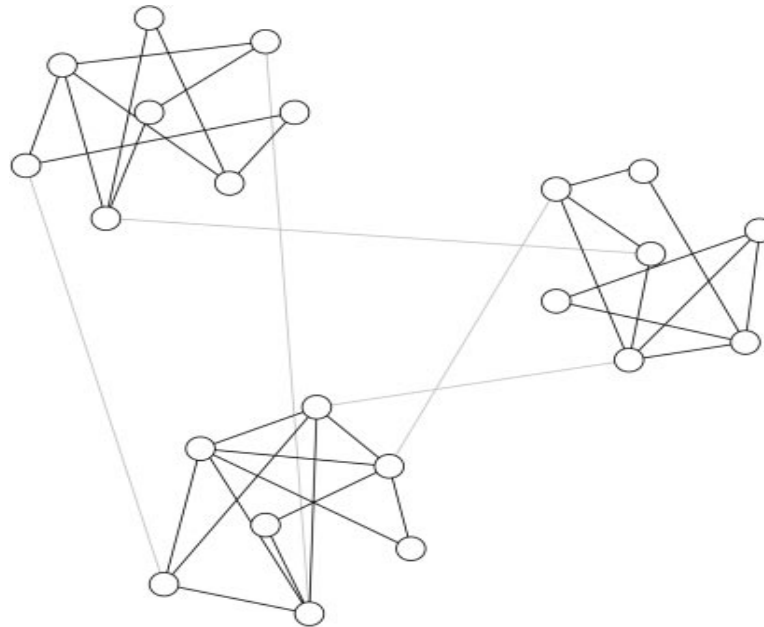
- Lot of applications – identifying similar nodes, close friends, recommendation, ...
- Challenging
 - Communities are not well-defined
 - Number of communities in a network is not known

Two broad types of algorithms

- Detection of disjoint communities
 - Each community is a partition of the network
 - Detection of overlapping communities
 - A node can be members of multiple communities
-

Algorithm by Girvan & Newman

- Community structure in social and biological networks, PNAS, 2002
- Focus on edges that are most “between” communities



Edge betweenness

- Edge betweenness of an edge e : fraction of shortest paths between all pairs of vertices, which run through e
 - Edges between communities are likely to have high betweenness centrality
 - Progressively remove edges having high betweenness centrality, to separate communities from one another
-

Girvan-Newman algorithm

1. Compute betweenness centrality for all edges
 2. Remove the edge with highest betweenness centrality
 3. Re-compute betweenness centrality for all edges affected by the removal
 4. Repeat steps 2 and 3 until no edges remain
- Time complexity
 - Graph of n vertices and m edges: betweenness centrality of all edges can be computed in $O(mn)$ time
 - Hence, worst case time complexity: $O(m^2n)$

How many communities?

- Community structure of a graph is hierarchical, with smaller communities nested within larger ones
 - Represented as a **hierarchical clustering tree:**
dendrogram
 - A “slice” through the tree at any level gives a certain number of communities
 - Which level to slice at?
-

An example dendrogram

Hierarchical clustering algorithms

- **Agglomerative** algorithms (bottom-up)
 - Clusters / communities iteratively merged if their similarity is sufficiently high
 - **Divisive** algorithms (top-down)
 - Clusters / communities iteratively split by removing edges
 - Both can be represented by dendrograms
 - Need some way to decide at what level to slice the dendrogram – **what is a good community structure?**
-

What is a good community structure?

- A few large communities, or many small communities?
 - Often depends on the end application
 - Example: find communities in an OSN for
 - Application 1: personalized recommendation to users
 - Application 2: map user-accounts to data centers located in some places
-

Objective functions for CD

- Community or network cluster
 - Typically a group of nodes having more and / or better interactions among its members, than between its members and the rest of the network
 - Typical CD algorithms
 - Choose an **objective function** that captures the above intuition
 - Optimize the objective function using heuristics or approximation algorithms
-

OBJECTIVE FUNCTIONS FOR COMMUNITY DETECTION

Empirical Comparison of Algorithms for Network
Community Detection, Leskovec et al., WWW 2010

Various objective functions

- Two criteria of interest for measuring **how well a particular set S of nodes represents a community**
 - Number of edges among the nodes within S
 - Number of edges between nodes in S and rest of network
 - Two types of objective functions
 - Single criterion – considers any one of the above criteria
 - Multi criterion – considers both the above criteria
-

Multi-criterion scores

- Consider both the criteria for measuring quality of a set S of nodes
- Lower values of $f(S)$ signify a more community-like set of nodes

Notations

- $G = (V, E)$ is the network.
 - $n = |V|$ = number of nodes
 - $m = |E|$ = number of edges
 - $d(u) = k_u$ = degree of node u
-
- S : set of nodes
 - n_s = number of nodes in S
 - m_s = number of edges **within S** (both nodes in S)
 - c_s = number of edges **on the boundary of S**
-

Expansion

$$f(S) = \frac{c_S}{n_S}$$

- Number of edges per node in S , that points outside the set S

Internal density

$$f(S) = 1 - \frac{m_S}{n_S(n_S - 1)/2}$$

- Internal edge density of the set S
-

Cut Ratio

$$f(S) = \frac{c_S}{n_S(n - n_S)}$$

- Fraction of all possible edges leaving the set S
-

Conductance

$$f(S) = \frac{c_S}{2m_S + c_S}$$

- Fraction of total edge volume that points outside the cluster
 - Edge volume = sum of node-degrees
 - Denominator: total connection from nodes in S to all nodes in graph G
-

Normalized Cut

$$f(S) = \frac{c_S}{2m_S + c_S} + \frac{c_S}{2(m - m_S) + c_S}$$

- Originally proposed in “Normalized cuts and Image Segmentation” by Shi et al, IEEE TPAMI, 2000
 - Some doubts about the denominator of the second term
-

Normalized cut – original definition

- Partition graph $G = (V, E)$ into two partitions A and B

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v).$$

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}, \quad (2)$$

where $assoc(A, V) = \sum_{u \in A, t \in V} w(u, t)$ is the total connection from nodes in A to all nodes in the graph and $assoc(B, V)$ is similarly defined.

- According to this definition, denominator of second term likely to be $2(m - m_s - c_s) + c_s = 2(m - m_s) - c_s$

Maximum Out Degree Fraction (ODF)

$$\max_{u \in S} \frac{|\{(u, v) : v \notin S\}|}{d(u)}$$

- Maximum fraction of edges of a node in S , that points outside the set S
-

Average ODF

$$f(S) = \frac{1}{n_S} \sum_{u \in S} \frac{|\{(u, v) : v \notin S\}|}{d(u)}$$

- Average fraction of edges of nodes in S, that points outside S

Flake ODF

$$f(S) = \frac{|\{u: u \in S, |\{(u, v): v \in S\}| < d(u)/2\}|}{n_S}$$

- Fraction of nodes in S that have fewer edges pointing inside S , than to outside S

Observations by Leskovec et al.

- Internal density and Maximum-ODF are not good measures for community quality
 - Does not show much variation, except for very small communities
 - Cut ratio has high variance
 - communities of similar sizes can have very different numbers of edges pointing outside
 - Both very low variance and very high variance undesirable for objective functions for CD
-

Observations by Leskovec et al.

- Flake-ODF prefers larger communities
- Conductance, expansion, normalized cut, average-ODF all exhibit qualitatively similar behavior and give best scores to similar clusters

Single-criterion scores

- Consider only one of the two criteria for measuring quality of a set S of nodes
 - Two simple single-criterion scores:
 - **Volume**: Sum of degrees of the nodes in S
 - **Edges Cut**: c_S : Number of edges needed to be removed to disconnect nodes in S from the rest of the network
-

Modularity-based measures

- A set of nodes is a good community if the number of edges within the set is significantly more than what can be expected by random chance
 - Modularity $Q = 1/K * (m_s - E(m_s))$
 - Number of edges m_s within set S, minus expected number of edges within the set S
 - K is a constant, used for normalization
-

Modularity ratio

$$\frac{m_S}{E(m_S)}$$

- Alternative measure of how well set S represents a community
 - Ratio of the number of edges among nodes in S , and expected number of such edges
-

Expected number of edges

- Null model: Erdos-Renyi random network having the same node degree sequence as given network
 - Randomized realization of a given network, realized in practice using Configuration Model
 - Cut each edge into two half-edges or stubs
 - Randomly connect each stub to any stub
 - Expected to have no community structure
-

Mathematical definition of Modularity

- For two particular nodes i and j :
 - Number of edges between the nodes: A_{ij}
 - Degrees: k_i, k_j
 - Expected number of links between i and j : $k_i k_j / 2m$
- Do the nodes i and j have more edges than expected by random chance?

$$A_{ij} - k_i k_j / 2m$$

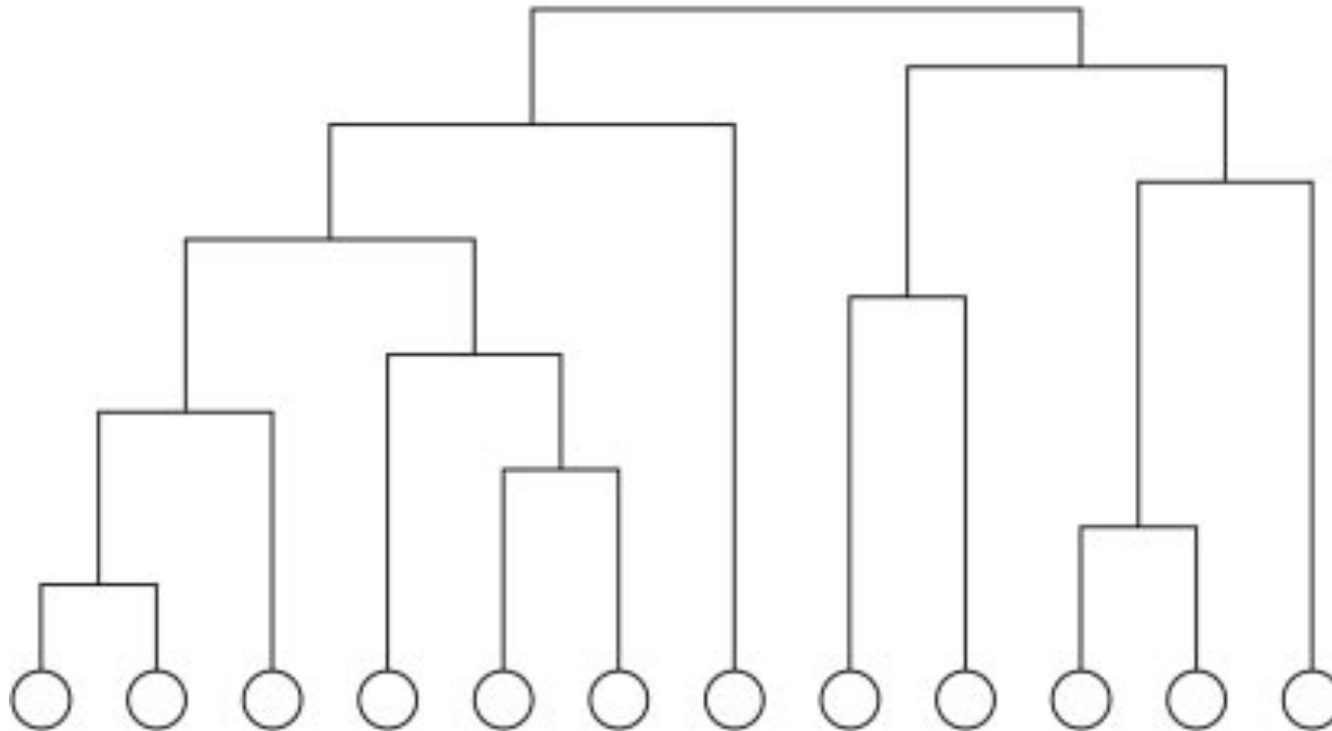
Modularity for a given network

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(C_i, C_j)$$

- The delta function is 1 if both nodes i and j are in the same community ($C_i = C_j$), 0 otherwise
- Consider a network with two communities $c1, c2$
 - Q is the fraction of edges that fall within $c1$ or $c2$, minus the expected number of edges within $c1$ and $c2$ for a random graph with the same node degree distribution as the given network

Using modularity for CD

- Approach 1: use Modularity to decide at which level to slice the dendrogram



Using modularity for CD

- Approach 1: use Modularity to decide at which level to slice the dendrogram
 - Approach 2: Optimize modularity
 - Exhaustive maximization is NP-hard
 - Heuristics and approximations used
-

Greedy algorithm for maximizing Q

- Fast algorithm for detecting community structure in networks, Newman, PRE 69(6), 2004
- Greedy agglomerative hierarchical clustering
 - Start with n clusters, each containing a single node
 - Add edges such that the new partitioning gives the maximum increase (minimum decrease) of modularity wrt the previous partitioning
 - A total of n partitionings found, with number of clusters varying from n to 1
 - Select the partitioning having highest modularity

Most popular Q optimization algorithm

- Louvain algorithm:

- <https://perso.uclouvain.be/vincent.blondel/research/louvain.html>

- Optimization in two steps

- Step 1: look for small communities - optimizing Q locally
 - Step 2: aggregate nodes in the same community and build a **new network whose nodes are the communities**
 - Repeat iteratively until a maximum of modularity is attained and a hierarchy of communities is produced
 - Time: approx $O(n \log n)$
-

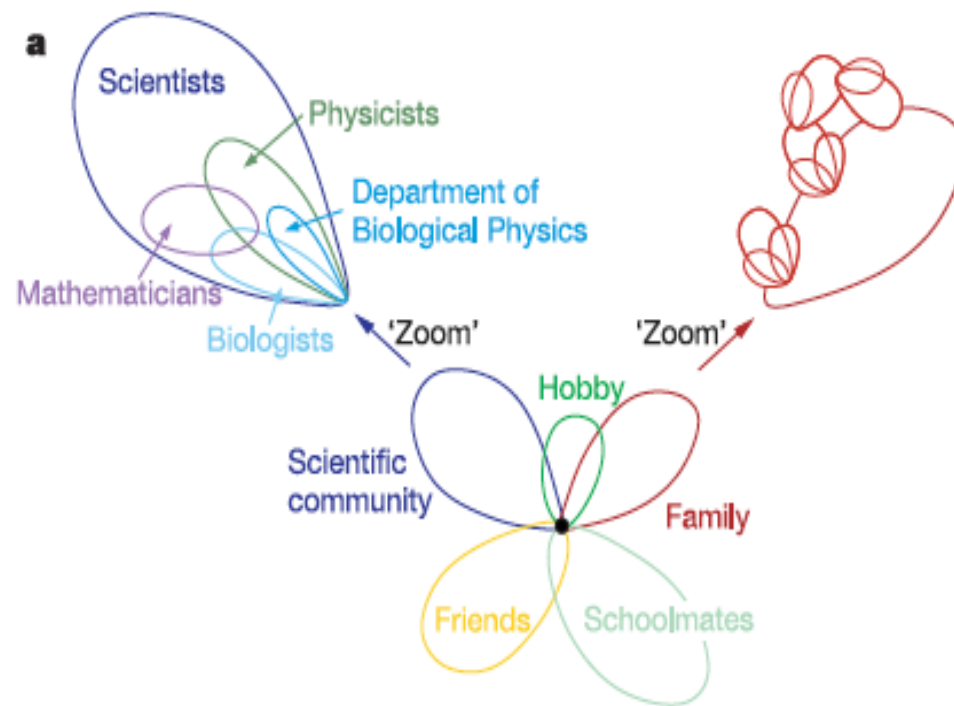
For reading

- Many subsequent works have suggested improvements for maximizing modularity
 - Reducing time complexity
 - Normalizing with number of edges to minimize bias towards larger communities
 - ...
 - Read “Community detection in graphs” by Fortunato, Physics Reports, 2010.
-

OVERLAPPING COMMUNITY DETECTION

Overlapping communities

- Nodes in real networks are often parts of multiple overlapping communities



Two algorithms

- **Clique Percolation Method**

- Uncovering the overlapping community structure of complex networks in nature and society, Palla et al., Nature Letters, vol. 435, 2005

- **Link communities**

- Link communities reveal multiscale complexity in networks, Ahn et al., Nature Letters, vol. 466, 2010
-

Clique Percolation Method

- Concept:
 - Internal edges of communities likely to be part of cliques
 - Inter-community edges unlikely to be part of cliques
- Adjacent k -cliques: two k -cliques are adjacent if they share $k-1$ nodes

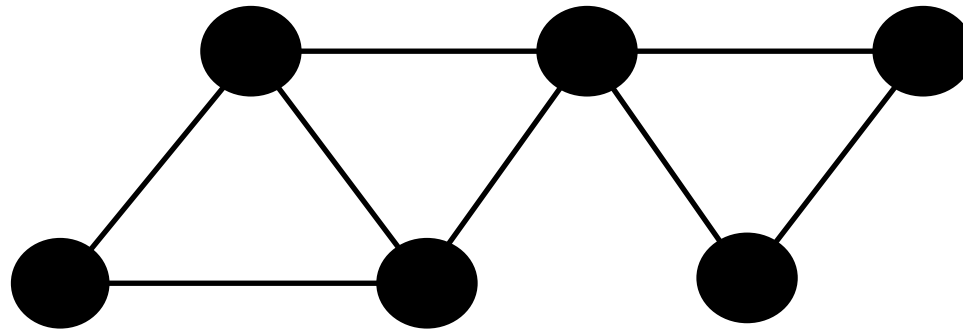
Some material on CPM borrowed from slides by Eugene Lim

k-Clique Communities

■ Adjacent k-cliques

Two k-cliques are adjacent when they share **k-1** nodes

$k = 3$

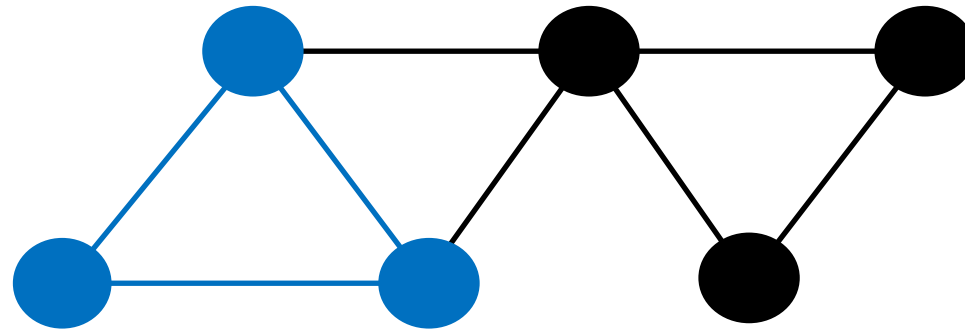


k-Clique Communities

■ Adjacent k-cliques

Two k-cliques are adjacent when they share **k-1** nodes

$k = 3$



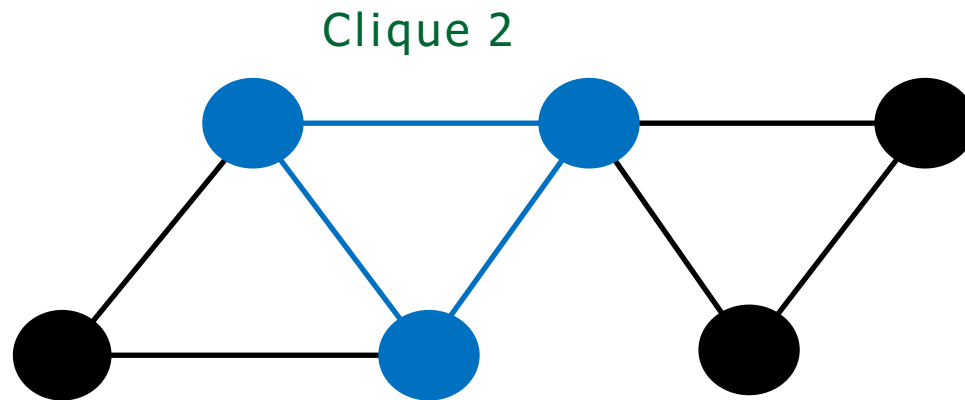
Clique 1

k-Clique Communities

■ Adjacent k-cliques

Two k-cliques are adjacent when they share **k-1** nodes

k = 3

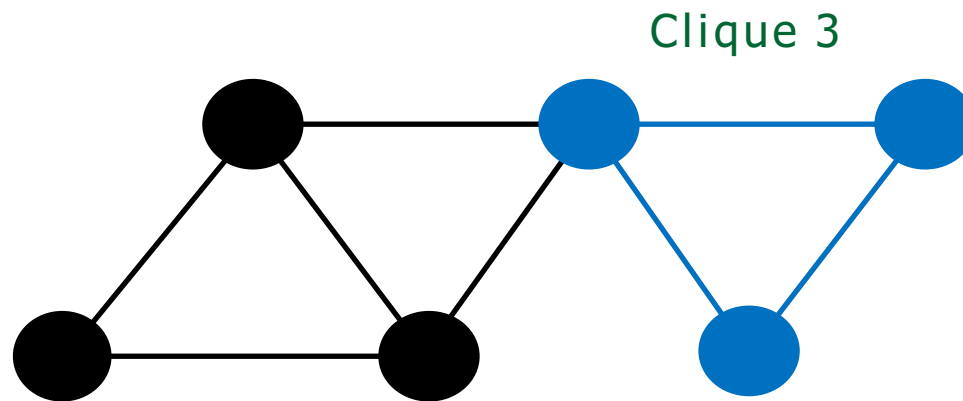


k-Clique Communities

■ Adjacent k-cliques

Two k-cliques are adjacent when they share **k-1** nodes

k = 3

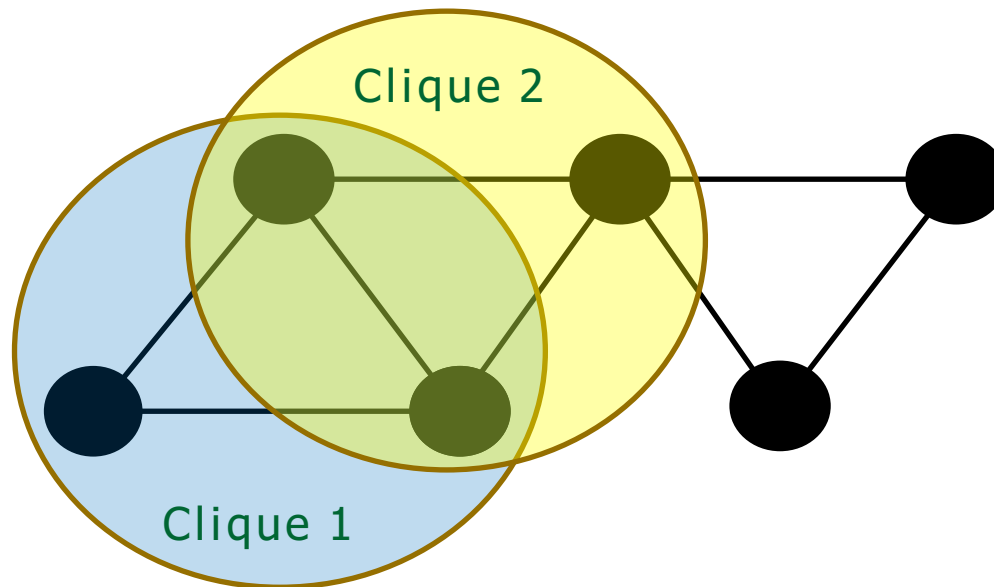


k-Clique Communities

■ Adjacent k-cliques

Two k-cliques are adjacent when they share **k-1** nodes

$k = 3$

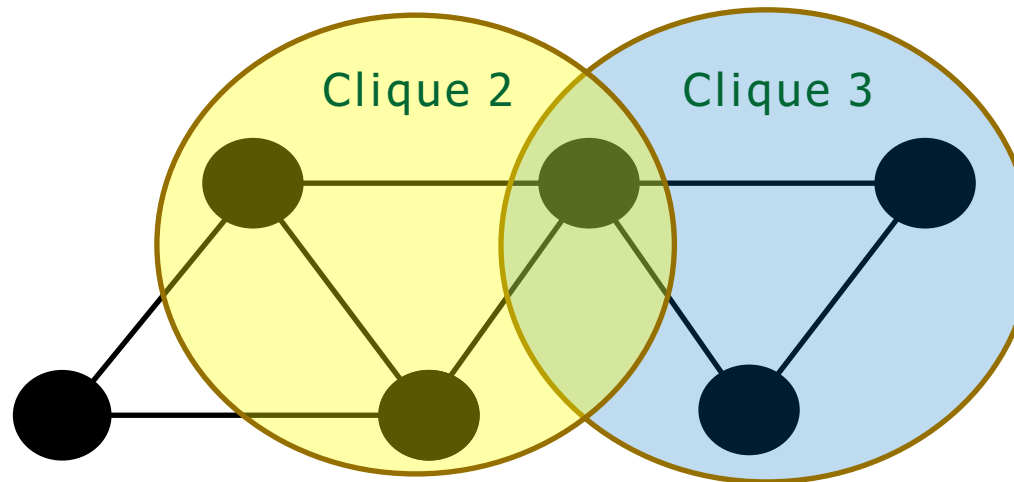


k-Clique Communities

■ Adjacent k-cliques

Two k-cliques are adjacent when they share **k-1** nodes

$k = 3$



k-Clique Communities

- **k-clique community**

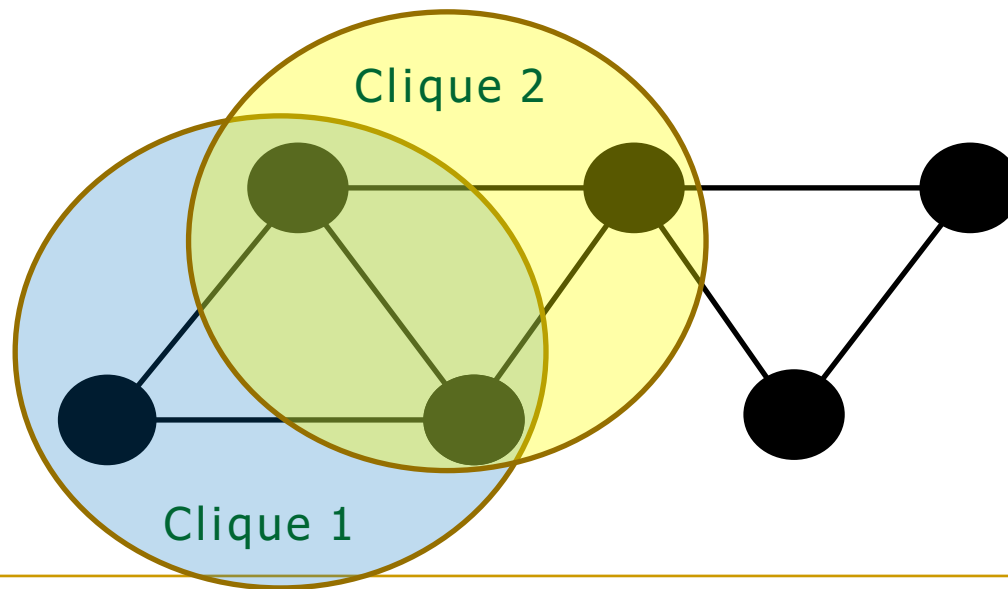
Union of all k-cliques that can be reached from each other
through a series of adjacent k-cliques

k-Clique Communities

■ k-clique community

Union of all k-cliques that can be reached from each other through a series of adjacent k-cliques

$k = 3$

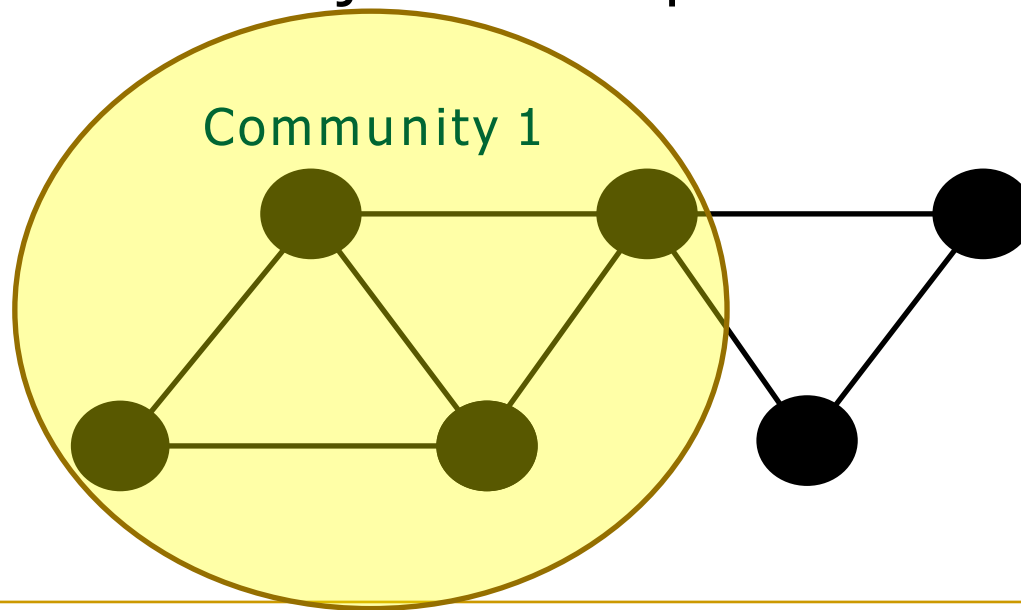


k-Clique Communities

■ k-clique community

Union of all k-cliques that can be reached from each other through a series of adjacent k-cliques

$k = 3$

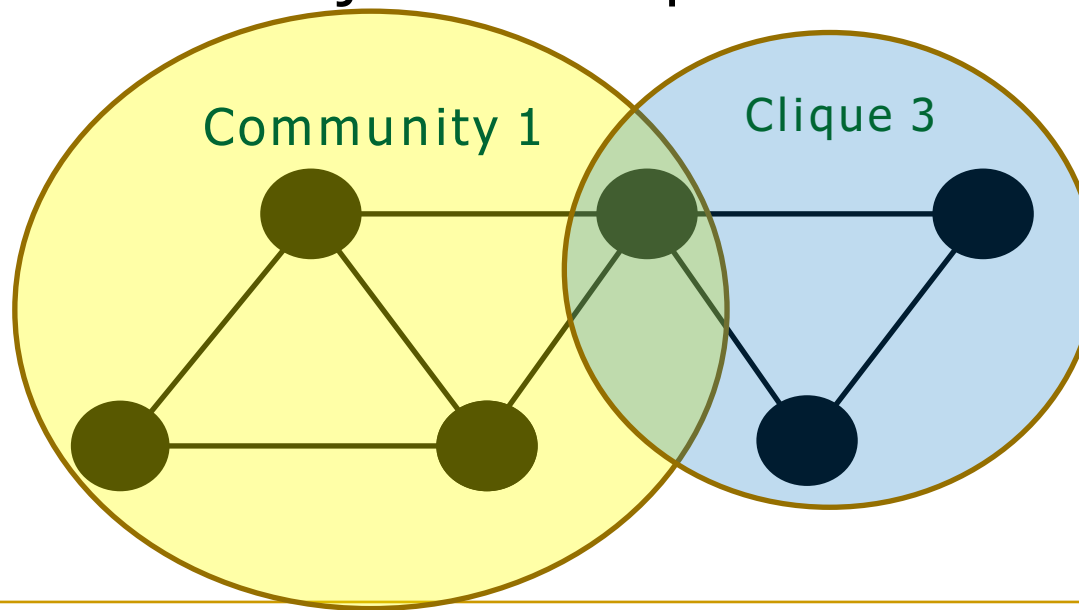


k-Clique Communities

■ k-clique community

Union of all k-cliques that can be reached from each other through a series of adjacent k-cliques

$k = 3$

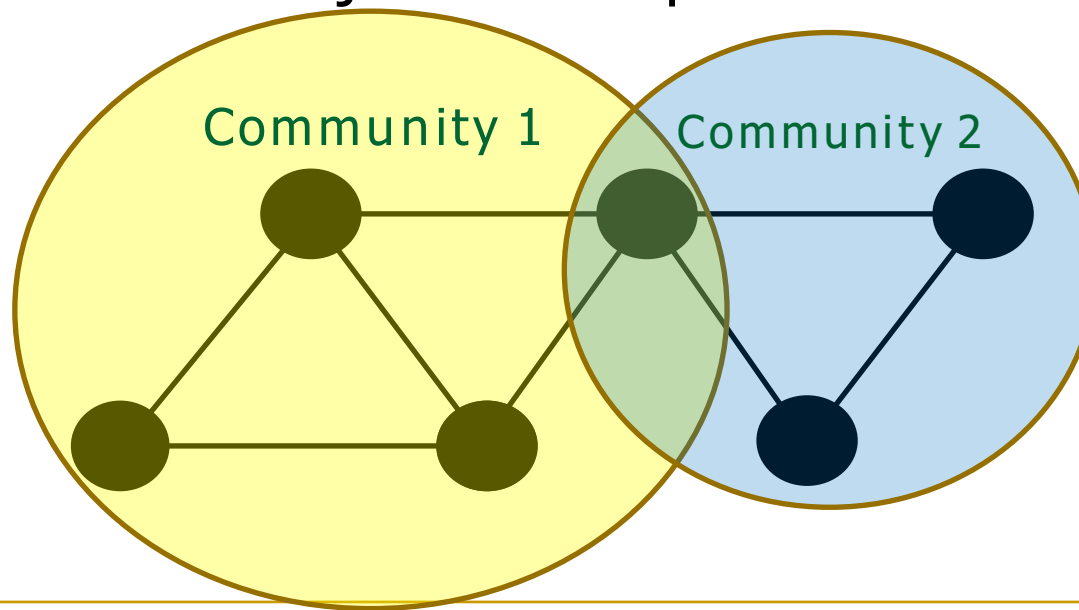


k-Clique Communities

■ k-clique community

Union of all k-cliques that can be reached from each other through a series of adjacent k-cliques

$k = 3$



Algorithm

- Locate maximal cliques
- Convert from cliques to k-clique communities

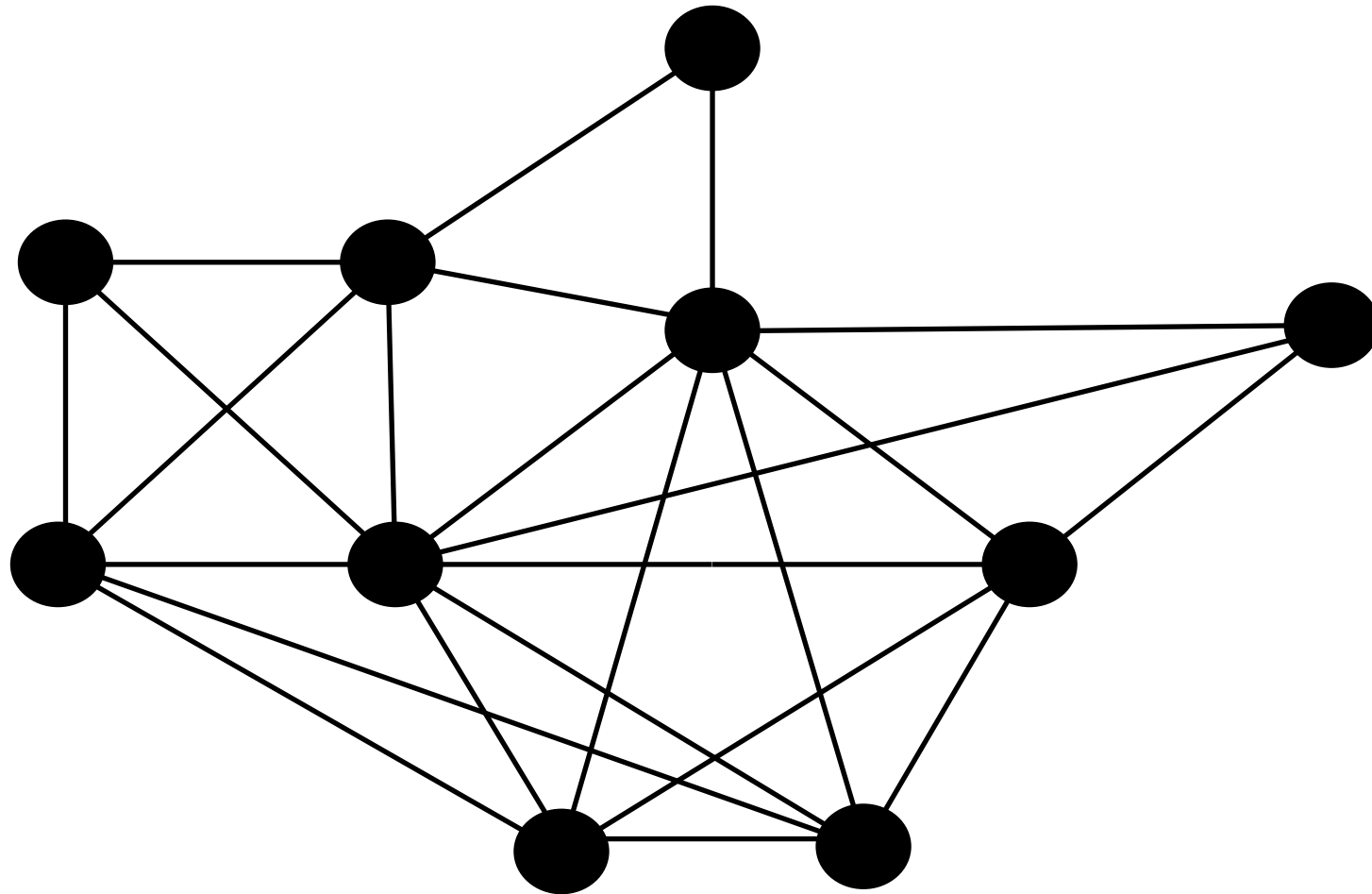
Locate Maximal Cliques

- Largest possible clique size can be determined from degrees of vertices
 - Starting from this size, find all cliques, then reduce size by 1 and repeat
-

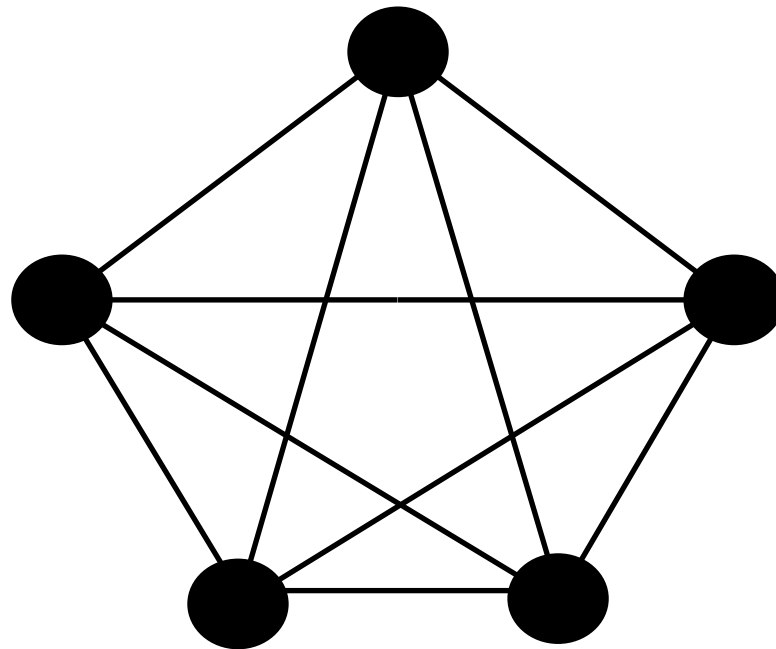
Algorithm

- Locate maximal cliques
- Convert from cliques to k-clique communities

Cliques to k-Clique Communities

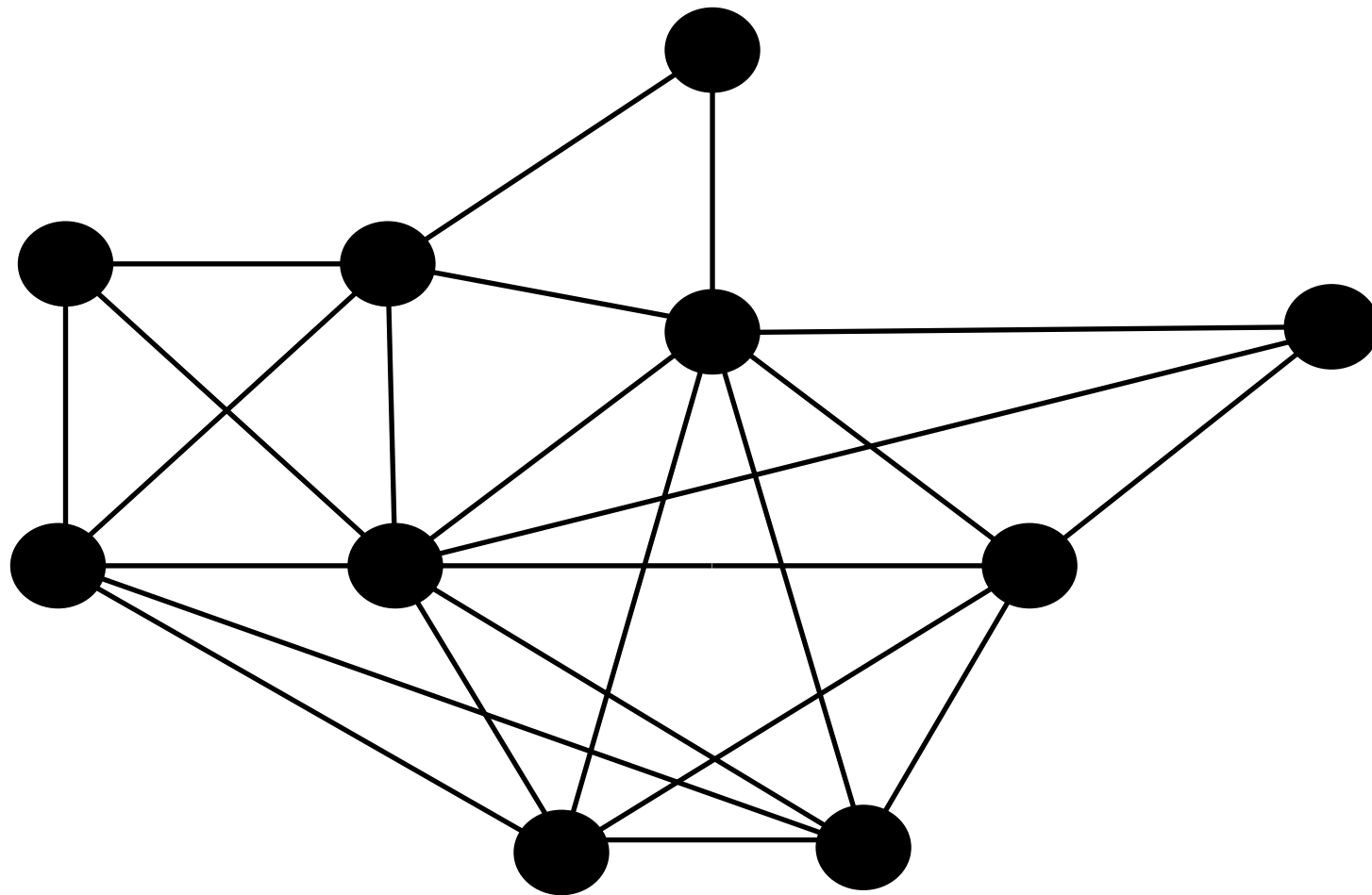


Cliques to k-Clique Communities

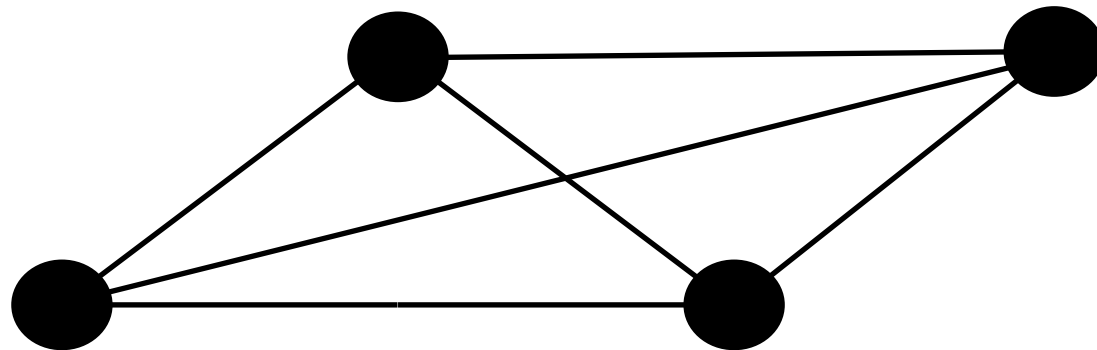


Clique 1: 5-clique

Cliques to k-Clique Communities

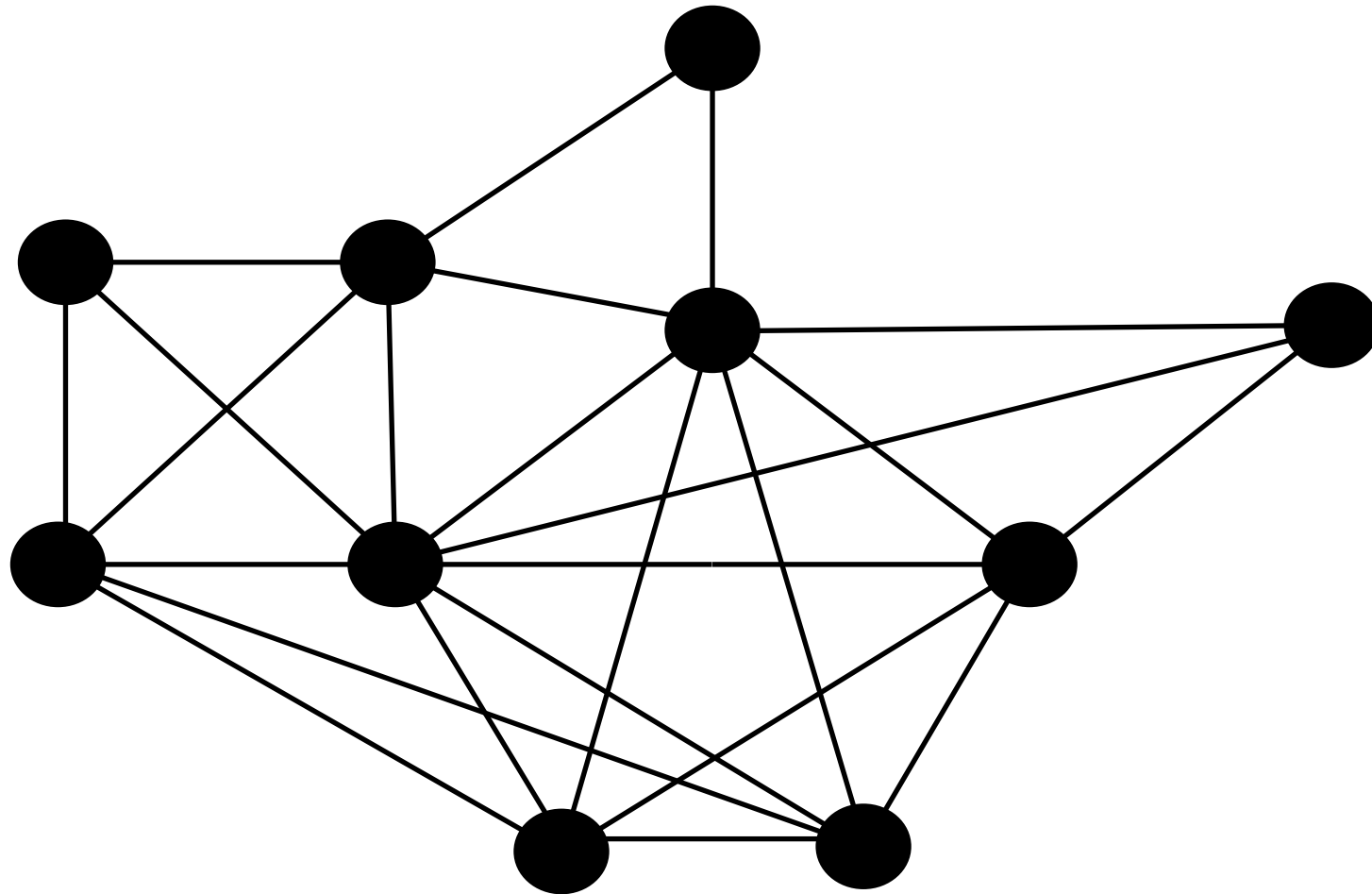


Cliques to k-Clique Communities



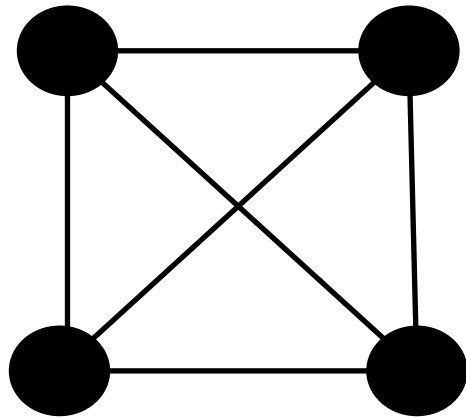
Clique 2: 4-clique

Cliques to k-Clique Communities

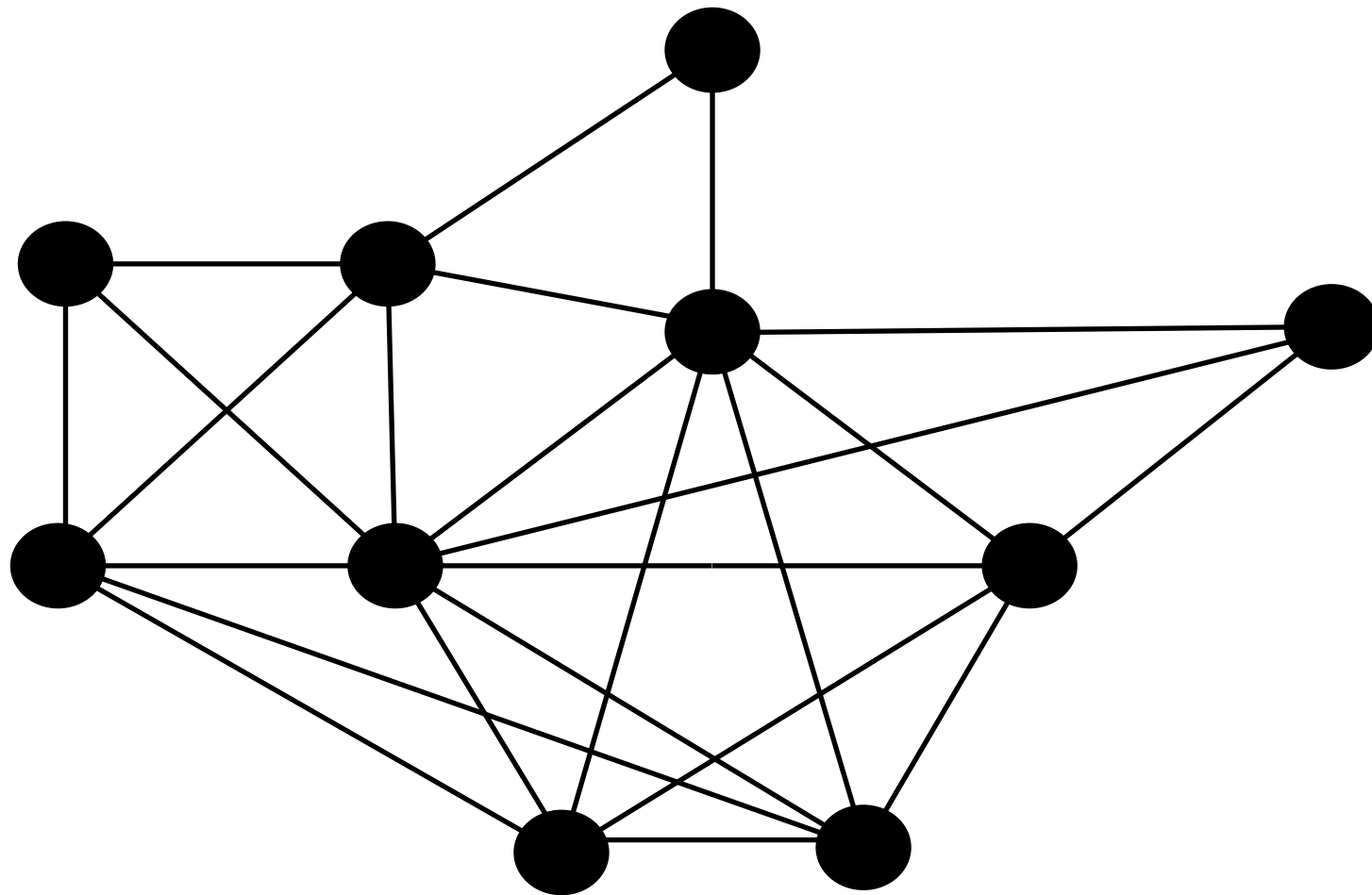


Cliques to k-Clique Communities

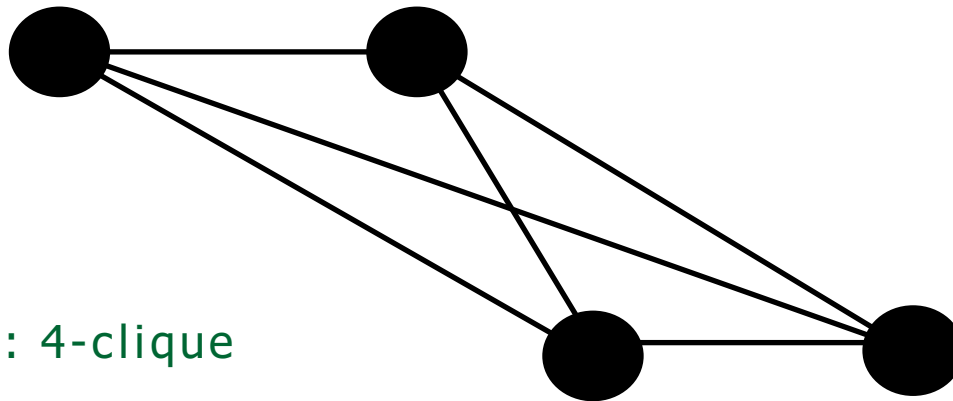
Clique 3: 4-clique



Cliques to k-Clique Communities

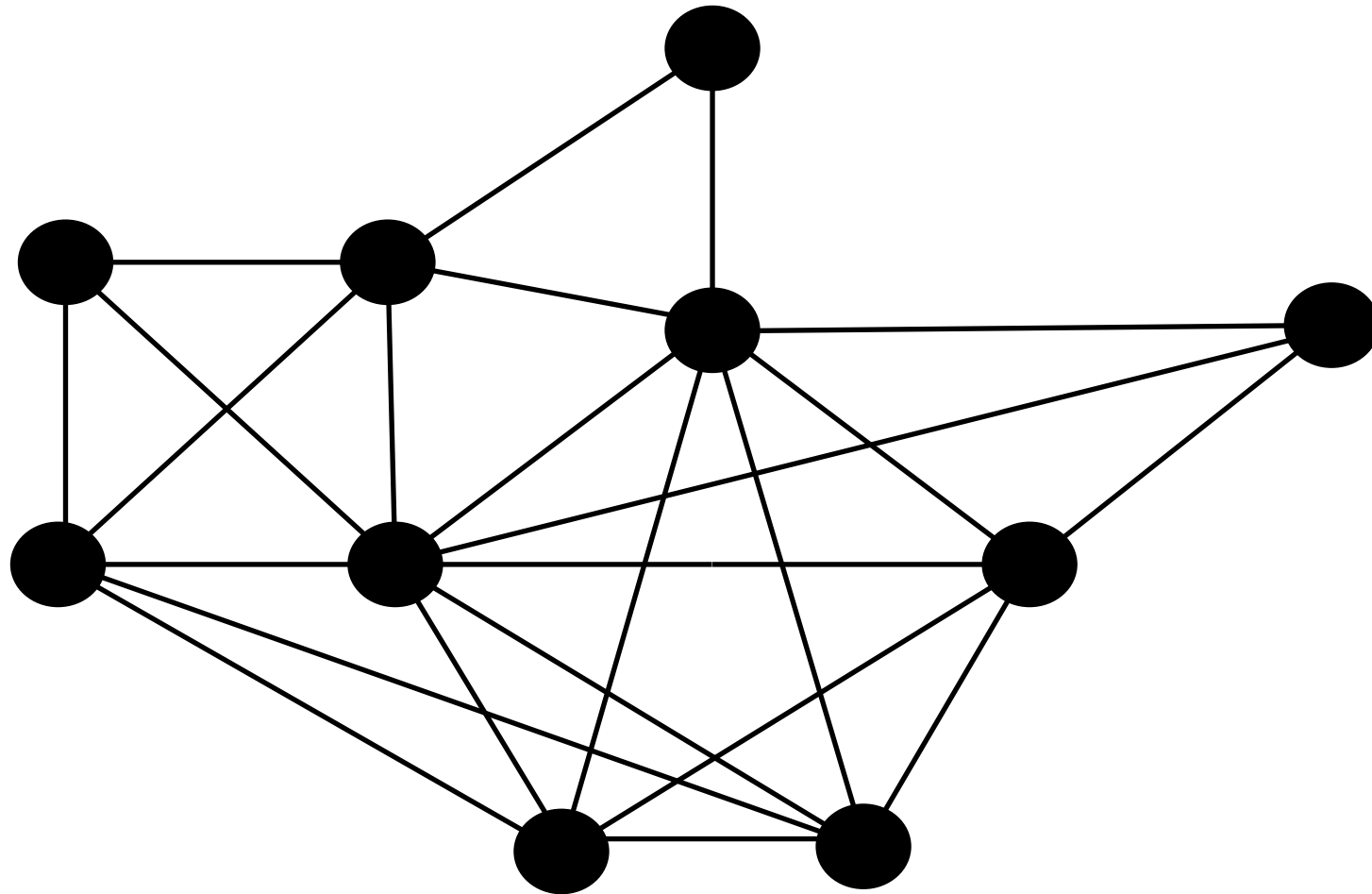


Cliques to k-Clique Communities

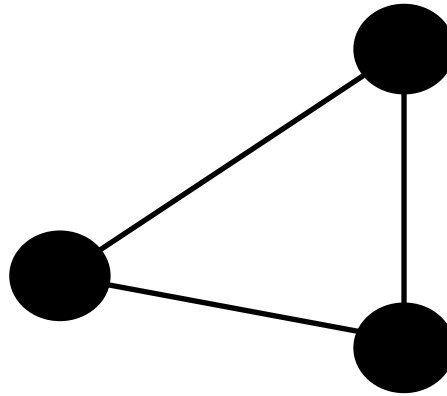


Clique 4: 4-clique

Cliques to k-Clique Communities

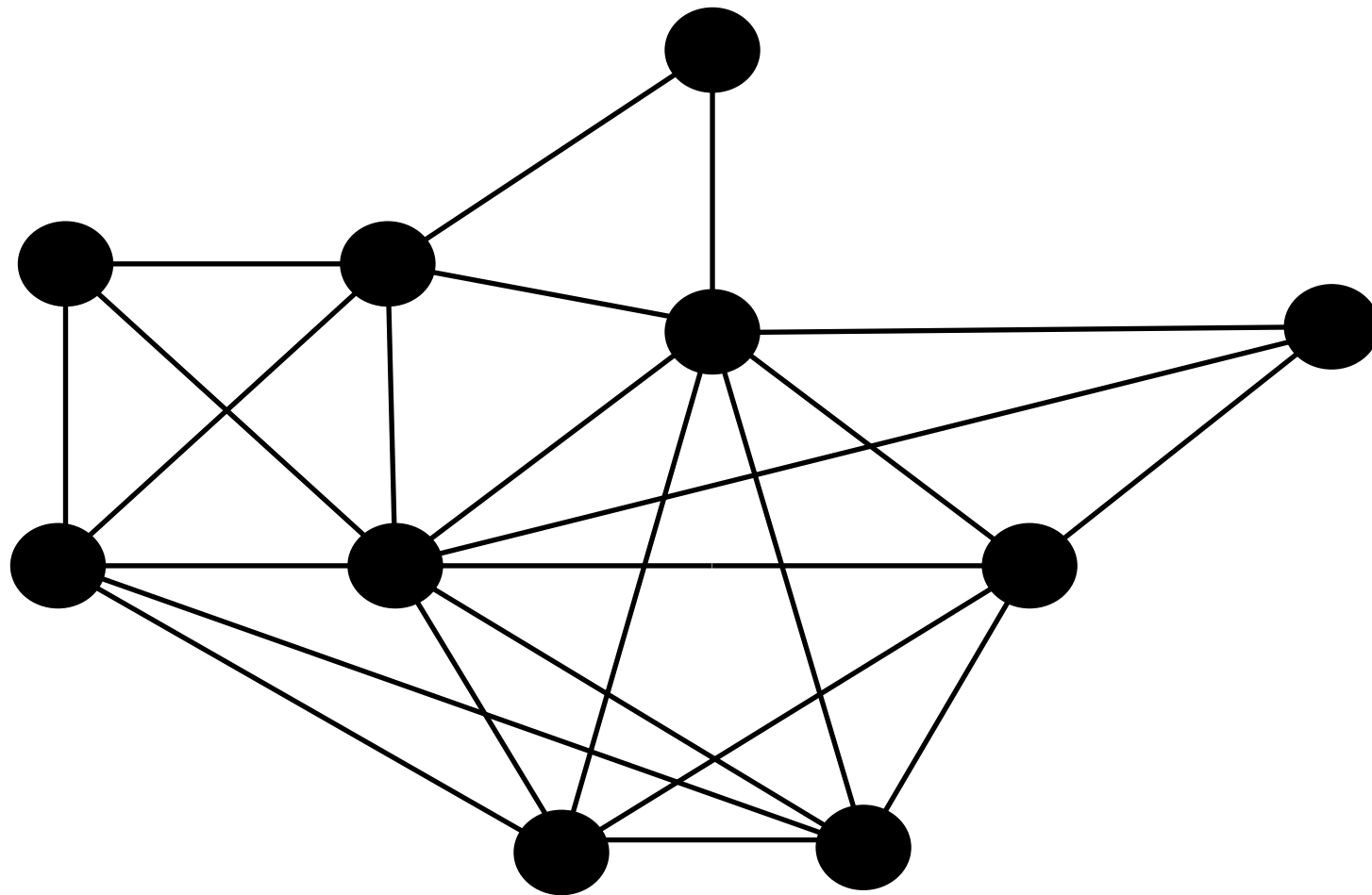


Cliques to k-Clique Communities



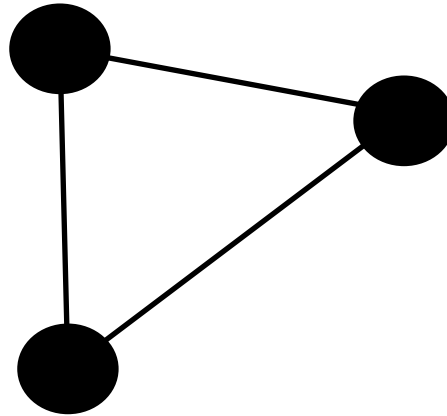
Clique 5: 3-clique

Cliques to k-Clique Communities



Cliques to k-Clique Communities

Clique 6: 3-clique



Cliques to k-Clique Communities

Clique-Clique overlap matrix

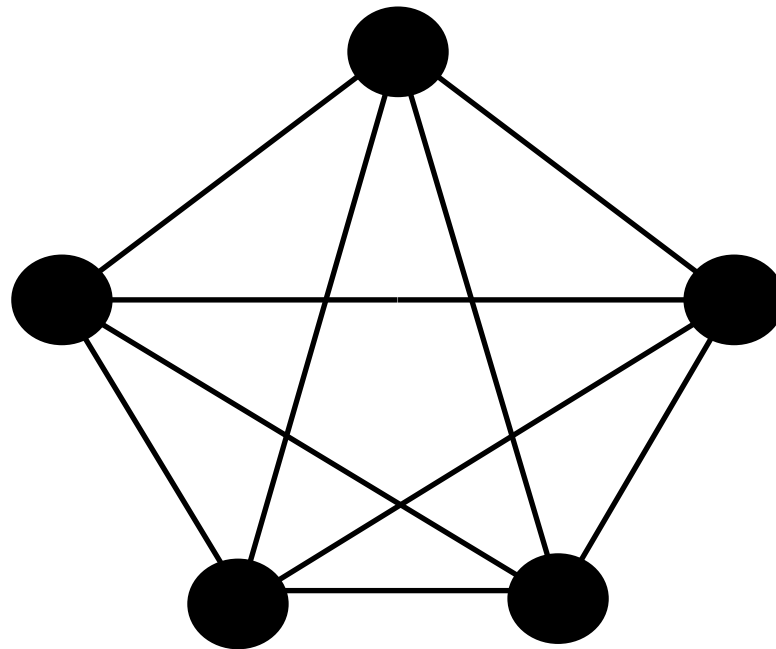
	1	2	3	4	5	6
1	5					
2		4				
3			4			
4				4		
5					3	
6						3

Cliques to k-Clique Communities

Clique-Clique overlap matrix

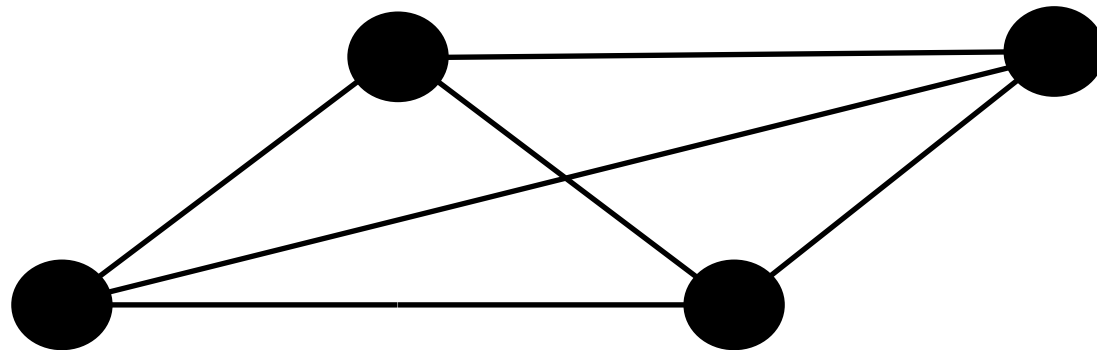
	1	2	3	4	5	6
1	5	3	1	3	1	2
2	3	4	1	1	1	2
3	1	1	4	2	1	2
4	3	1	2	4	0	1
5	1	1	1	0	3	2
6	2	2	2	1	2	3

Cliques to k-Clique Communities



Clique 1: 5-clique

Cliques to k-Clique Communities



Clique 2: 4-clique

Cliques to k-Clique Communities

Clique-Clique overlap matrix

	1	2	3	4	5	6
1	5	3	1	3	1	2
2	3	4	1	1	1	2
3	1	1	4	2	1	2
4	3	1	2	4	0	1
5	1	1	1	0	3	2
6	2	2	2	1	2	3

Cliques to k -Clique Communities

- For a given value of k , k -clique communities:
 - Connected clique components in which neighboring cliques linked to each other by at least $k-1$ common nodes
 - How to find k -clique communities from the clique-clique overlap matrix?
 - Erase every diagonal element smaller than k
 - Erase every off-diagonal element smaller than $k-1$
 - Replace remaining elements by 1
 - Carry out a component analysis of this matrix
-

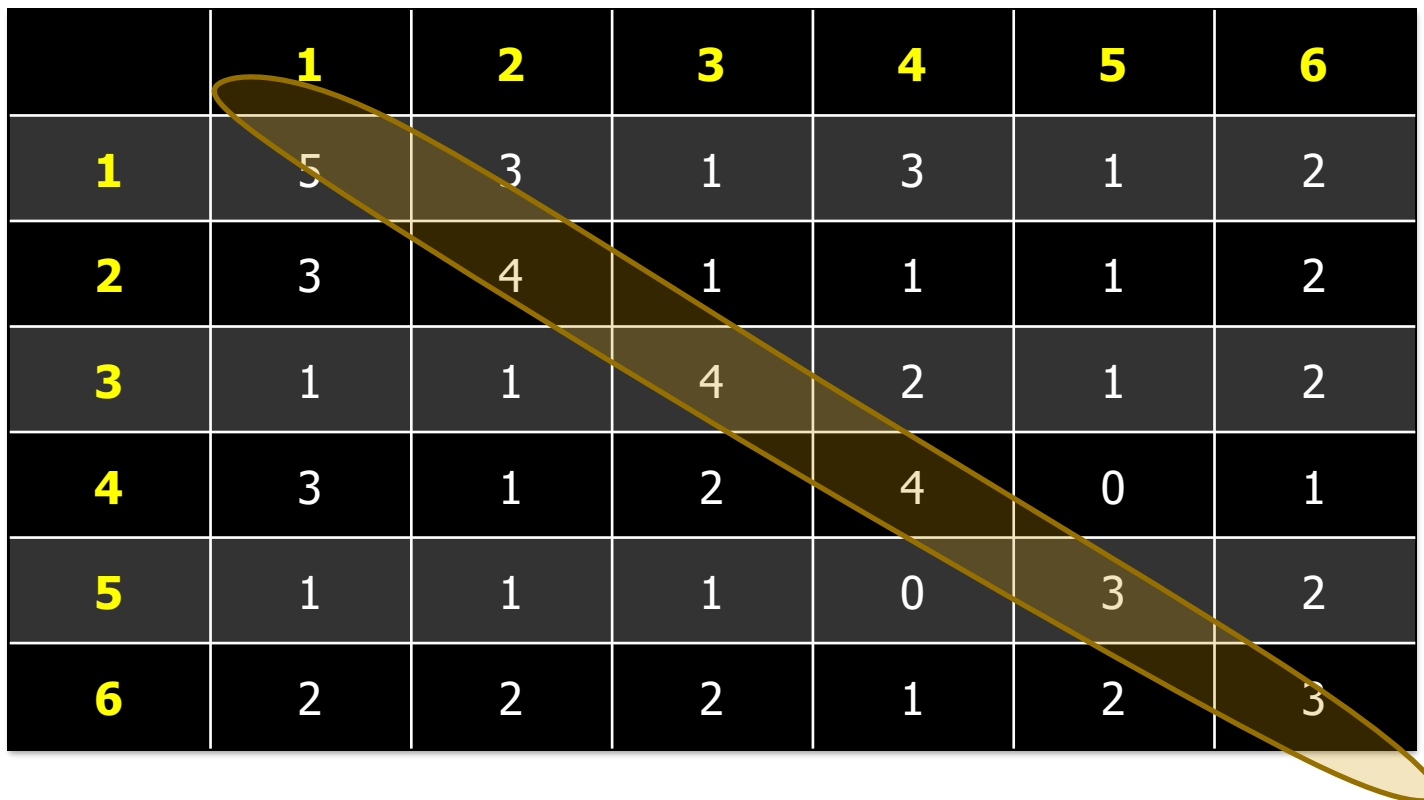
Cliques to k-Clique Communities

k=4

	1	2	3	4	5	6
1	5	3	1	3	1	2
2	3	4	1	1	1	2
3	1	1	4	2	1	2
4	3	1	2	4	0	1
5	1	1	1	0	3	2
6	2	2	2	1	2	3

Cliques to k-Clique Communities

k=4



	1	2	3	4	5	6
1	5	3	1	3	1	2
2	3	4	1	1	1	2
3	1	1	4	2	1	2
4	3	1	2	4	0	1
5	1	1	1	0	3	2
6	2	2	2	1	2	3

Cliques to k-Clique Communities

$k=4$

	1	2	3	4	5	6
1	5	3	1	3	1	2
2	3	4	1	1	1	2
3	1	1	4	2	1	2
4	3	1	2	4	0	1
5	1	1	1	0	0	2
6	2	2	2	1	2	0

Delete if less than k

Cliques to k-Clique Communities

k=4

	1	2	3	4	5	6
1	5	3	1	3	1	2
2	3	4	1	1	1	2
3	1	1	4	2	1	2
4	3	1	2	4	0	1
5	1	1	1	0	0	2
6	2	2	2	1	2	0

Cliques to k-Clique Communities

k=4

	1	2	3	4	5	6
1	5	3	1	3	1	2
2	3	4	1	1	1	2
3	1	1	4	2	1	2
4	3	1	2	4	0	1
5	1	1	1	0	0	2
6	2	2	2	1	2	0

Cliques to k-Clique Communities

$k=4$

	1	2	3	4	5	6
1	5	3	0	3	0	0
2	3	4	0	0	0	0
3	0	0	4	0	0	0
4	3	0	0	4	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Delete if less than $k-1$

Cliques to k-Clique Communities

k=4

	1	2	3	4	5	6
1	5	3	0	3	0	0
2	3	4	0	0	0	0
3	0	0	4	0	0	0
4	3	0	0	4	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Cliques to k-Clique Communities

k=4

	1	2	3	4	5	6
1	1	1	0	1	0	0
2	1	1	0	0	0	0
3	0	0	1	0	0	0
4	1	0	0	1	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Change all non-zeros to 1

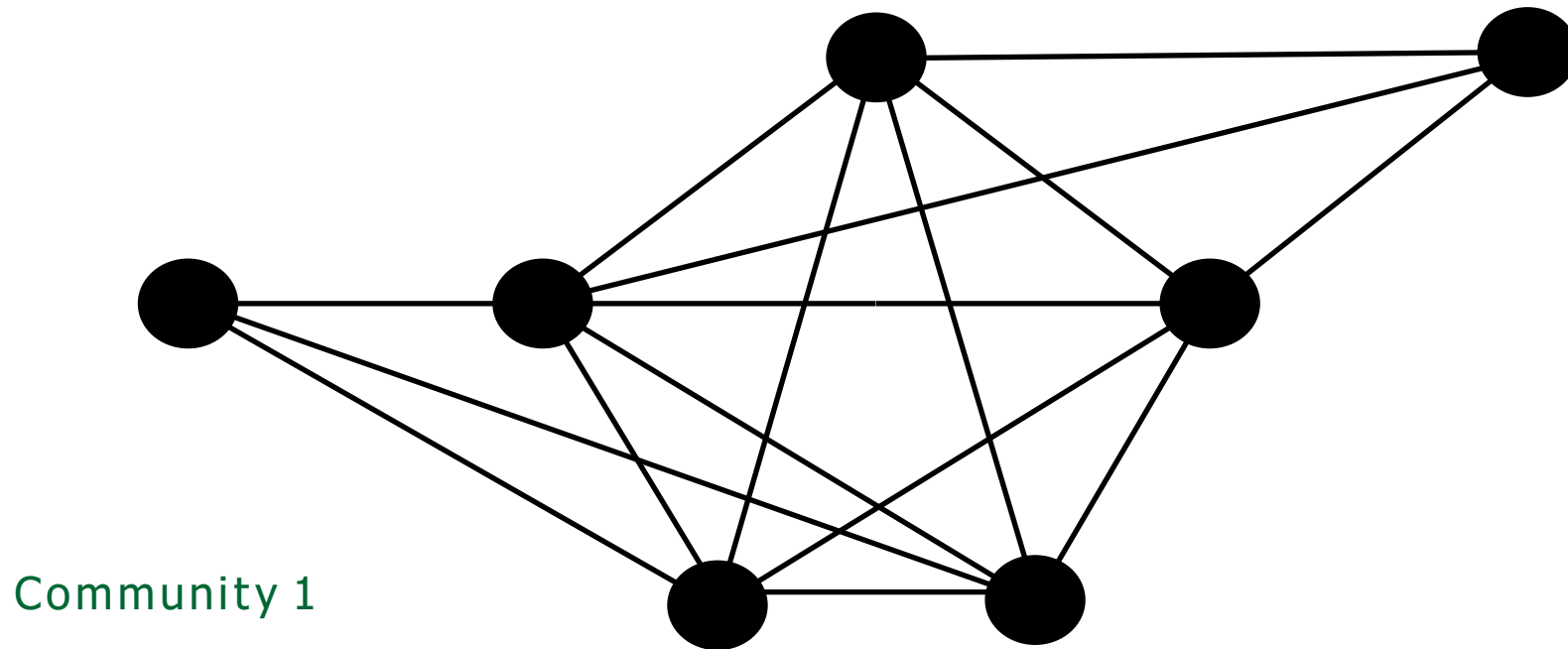
Cliques to k-Clique Communities

k=4

	1	2	3	4	5	6
1	1	1	0	1	0	0
2	1	1	0	0	0	0
3	0	0	1	0	0	0
4	1	0	0	1	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

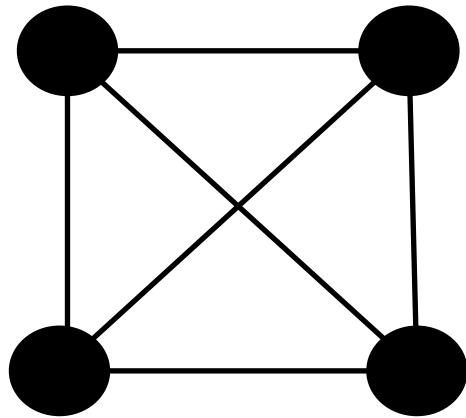
Cliques to k-Clique Communities

$k=4$



Cliques to k -Clique Communities

$k=4$



Community 2

Clique Percolation Method: Analysis

- Believed to be non-polynomial
 - No closed formula can be given
 - However, claimed to be efficient on real systems
 - Limitations
 - Fail to give meaningful covers for graph with few cliques
 - With too many cliques, might give a trivial community structure
-

Link communities

- A node might belong to multiple communities
 - For a person: family, co-workers, friends, ...
 - A link often exists for one dominant reason
 - Two people are in the same family, or are co-workers
 - Link community: a set of closely inter-related links
-

Identifying Link communities

- Hierarchical clustering with a similarity between links to build a dendrogram
 - Each leaf of the dendrogram is a link from the original network
 - Branches of the dendrogram are link communities
 - Slice the dendrogram at a suitable level
 - Each link placed in a single community
 - Each node inherits membership of the communities of all its links
-

For hierarchical clustering

- Two questions to be answered
 - How to measure similarity between items?
 - At which level to slice the dendrogram?
-

Similarity measure between links

- Node i and its neighboring nodes: $n_+(i)$
- Similarity measured only between pairs of links which share a node
- Similarity between e_{ik} and e_{jk} :

$$S(e_{ik}, e_{jk}) = |n_+(i) \cap n_+(j)| / |n_+(i) \cup n_+(j)|.$$

Which level to slice the dendrogram?

■ Measure: Partition density D

- Total number of links in network: M
- $\{P_1, P_2, \dots, P_C\}$: partition of links into C subsets
- P_c has n_c nodes and m_c links

$$D_c = \frac{m_c - (n_c - 1)}{n_c(n_c - 1)/2 - (n_c - 1)}$$

- Partition density is average of D_c weighted by the fraction of links present in P_c

$$D = \frac{2}{M} \sum_c m_c \frac{m_c - (n_c - 1)}{(n_c - 2)(n_c - 1)}$$

How to evaluate a CD algorithm?

- Assume a known community structure $X = \{x_1, x_2, \dots, x_I\}$
 - An algorithm finds a community structure $Y = \{y_1, y_2, \dots, y_J\}$
 - How close is Y to X ?
 - Several existing measures
 - Purity
 - Rand index
 - Normalized Mutual Information (NMI) [has been extended to overlapping communities]
 - Generalized Measures for the Evaluation of Community Detection Methods, by Labatut (<https://arxiv.org/abs/1303.5441>)
-

DIFFERENT TYPES OF GROUPS IN A SOCIAL NETWORK

Different methods to identify groups

- Identifying groups based on network structure – community detection algorithms
 - How about identifying groups based on content, e.g., text or profile attributes?
 - Deep Twitter Diving: Exploring Topical Groups in Microblogs at Scale, Bhattacharya et al., CSCW 2014
-

Identified topical groups in Twitter

Topical Groups = Experts + Seekers

Experts: Users who have expertise on the topic

Seekers: Users who are interested in the topic



@BarackObama
Expert on Politics

@BarackObama
Seeker on Basketball



Identifying topical groups at scale

- Crawled data for first 38 million users in Twitter
- 88 Million lists, 1.5 Billion social links
- Identified 36 thousand topical groups

Diversity: Topics and Group Size

No. of seekers	Number of experts					
	< 100	100 – 500	500 – 1K	1K – 5K	5K – 10K	> 10K
< 1K	(5416) <i>geology, karate, malaria, neurology, tsunami, psychiatry, radiology, pediatrics, dermatology, dentistry</i>	(132) <i>volleyball, philosophers, tarot, perfume, florists, copy-writers, taxi, esperanto</i>				
1K – 5K	(915) <i>biology, chemistry, swimmers, astrophysics, multimedia, semiconductor, renewable-energy, breast-cancer, judaism</i>	(428) <i>painters, astrology, sociology, geography, forensics, anthropology, genealogy, archaeology, gluten, diabetes, neuroscience</i>	(17) <i>architects, insurance, second-life, police, progressives, creativity</i>			
5K – 10K	(166) <i>malware, gnu, robot, chicago-sports, gospel-music, space-exploration, wall-street</i>	(202) <i>horror, agriculture, atheism, attorneys, furniture, art-galleries, ubuntu</i>	(34) <i>psychology, poetry, catholic, hospitals, autism, jazz</i>	(2) <i>coffee, dealers</i>		
10K – 50K	(174) <i>ipod, ipad, virus, Liverpool-FC, choreographers, heavy-metal, backstreet-boys, world-cup,</i>	(312) <i>olympics, physics, theology, earthquake, opera, makeup, Adobe, wrestlers, typography, american-idol</i>	(146) <i>tennis, linux, astronomy, yoga, animation, manga, doctors, realtors, wildlife, rugby, forex, php, java,</i>	(67) <i>law, history, beer, golf, librarians, theatre, military, poker, conservatives, vegan</i>		
50K–100K	(7) <i>bbc-radio, UK-celebs, christian-leaders, superstars</i>	(61) <i>hackers, programmers, bicycle, GOP, fantasy-football, NCAA, wwe, sci-fi</i>	(35) <i>medicine, cyclists, investors, recipes, NHL, xbox, triathlon, Google</i>	(37) <i>hotels, museums, hockey, architecture, charities, weather, space</i>		
> 100K	(3) <i>headlines, brits</i>	(49) <i>pop-culture, gospel, BBC, reality-tv, bollywood</i>	(58) <i>religion, actresses, gadgets, graphic-design, directors, lifestyle, gossip, commentators, youtube</i>	(140) <i>books, government, comedy, environment, baseball, soccer, hollywood, iphone, economics, money</i>	(25) <i>fashion, education, wine, photography, radio, restaurants, science, SEO</i>	(17) <i>music, tech, business, politics, food, sports, celebs, health, media, bloggers, travel, writers</i>

A Small Number of Very Popular Groups

No. of seekers	Number of experts					
	< 100	100 – 500	500 – 1K	1K – 5K	5K – 10K	> 10K
< 1K	(5416) <i>geology, karate, malaria, neurology, tsunami, psychiatry, radiology, pediatrics, dermatology</i>	(132) <i>volleyball, philosophers, tarot, perfume, florists, copy-writers, taxi, esperanto</i>				
1K – 5K	(915) <i>history, astrology, media, renewal, breast-cancer</i>	(37) <i>hotels, museums, hockey, architecture, charities, weather, space</i>				
5K – 10K	(166) <i>robot, gospel-exploration</i>	(140) <i>books, government, comedy, environment, baseball, soccer, hollywood, iphone, economics, money</i>	(25) <i>fashion, education, wine, photography, radio, restaurants, science, SEO</i>	(17) <i>music, tech, business, politics, food, sports, celebs, health, media, bloggers, travel, writers</i>		
10K – 50K	(174) <i>virus, choreography, metal, world-cup</i>					
50K – 100K	(7) <i>bloggers, celebs, leaders, superstars</i>					
		GOP, fantasy-football, NCAA, wwe, sci-fi	xbox, triathlon, Google	architecture, charities, weather, space		
> 100K	(3) <i>headlines, brits</i>	(49) <i>pop-culture, gospel, BBC, reality-tv, bollywood</i>	(58) <i>religion, actresses, gadgets, graphic-design, directors, lifestyle, gossip, commentators, youtube</i>	(140) <i>books, government, comedy, environment, baseball, soccer, hollywood, iphone, economics, money</i>	(25) <i>fashion, education, wine, photography, radio, restaurants, science, SEO</i>	(17) <i>music, tech, business, politics, food, sports, celebs, health, media, bloggers, travel, writers</i>

Thousands of Specialized Niche Groups

No. of seekers	Number of experts					
	< 100	100 – 500	500 – 1K	1K – 5K	5K – 10K	> 10K
< 1K	(5416) geology, karate, malaria, neurology, tsunami, psychiatry, radiology, pediatrics, dermatology, dentistry	(132) volleyball, philosophers, tarot, perfume, florists, copy-writers, taxi, esperanto				
1K – 5K	(915) biology, chemistry, astrophysics, media, semiconductor, renewable-energy, breast-cancer, judaism	(5416) geology, karate, malaria, neurology, tsunami, psychiatry, radiology, pediatrics, dermatology, dentistry	(132) volleyball, philosophers, tarot, perfume, florists, copy-writers, taxi, esperanto			
5K – 10K	(166) malware, robot, chicago, gospel-music, exploration, wall	(915) biology, chemistry, swimmers, astrophysics, multi-media, semiconductor, renewable-energy, breast-cancer, judaism	(428) painters, astrology, sociology, geography, forensics, anthropology, genealogy, archaeology, gluten, diabetes, neuroscience			
10K – 50K	(174) ipod, virus, Liverpool, choreographers, metal, backstreet, world-cup,					
50K – 100K	(7) bbc-radio, celebs, ch, leaders, supersta					
> 100K	(3) headlines, brits	NCAA, wwe, sci-fi (49) pop-culture, gospel, BBC, reality-tv, bollywood	(58) religion, actresses, gadgets, graphic-design, directors, lifestyle, gossip, commentators, youtube	ties, weather, space (140) books, government, comedy, environment, baseball, soccer, hollywood, iphone, economics, money	(25) fashion, education, wine, photography, radio, restaurants, science, SEO	(17) music, tech, business, politics, food, sports, celebs, health, media, bloggers, travel, writers

Breaking the Twitter stereotype

- Twitter stereotype

- Popular news on few topics such as sports, entertainment, politics, technology
- Celebrity gossip, current news, and chatter

- Breaking the stereotype

- Majority of the population discuss few popular topics, but
 - Smaller groups interested in thousands of niche, specialized topics
-

Detecting topical groups

- We followed content-based approach to identify topical groups
 - Could community detection algorithms be used on the social network to detect them?
 - Applied BGLL / Louvain algorithm on the Twitter subscription network to identify communities
-

Detecting topical groups

- Louvain largely unable to detect topical groups, especially the smaller ones (on niche topics)
 - Communities detected by Louvain fare better on structural measures like cut-ratio, conductance
 - Topical groups do not have good structural quality
 - Poor values for standard community quality metrics such as cut-ratio and conductance
-

Why do groups form?

- “Common Identity and Bond Theory”
 - Prentice et. al. “Asymmetries in Attachments to Groups and to Their Members: Distinguishing Between Common-Identity and Common-Bond Groups”, Personality and Social Psychology Bulletin, 1994
 - Identity based groups
 - Bond based groups
-

Common Identity and Bond Theory

Identity Based Groups

Low Reciprocity

Low Personal Interactions

High Topicality of discussions

Examples:

Fans at a football match,
Attendees at a conference

Bond Based Groups

High Reciprocity

High Personal Interactions

Low Topicality of discussions

Examples:

Family, personal friends

Analysis of 50 topical groups

- Low reciprocity among members
 - Few one-to-one interactions
 - Most tweets posted by experts are related to topic
 - → Topical groups are identity-based which are difficult to detect via community detection algorithms
-