Chapter 3.
Community Detection and Evaluation

October 2012
Youn-Hee Han
http://link.koreatech.ac.kr
Community Detection

- Social networks demonstrate a strong community effect
- Actors in a network tend to form closely-knit groups
  - The groups are also called communities, clusters, cohesive subgroups or modules in different contexts
- Generally speaking, individuals interact more frequently with members within group than those outside the group
- Detecting community is a core problem in social network analysis
  - Finding out communities helps for other related social computing tasks

![Diagram of a network with communities](image.png)

**FIG. 1**: A schematic representation of a network with community structure. In this network, there are three communities of densely connected vertices (circles with solid lines), with a much lower density of connections (gray lines) between them.
Prologue

◊ Community Detection

Communities of Political Blogs

Lada Adamic and Natalie Glance, The Political Blogosphere and the 2004 U.S. Election: Divided They Blog, 2005
Prologue

Two types of groups in social media
- **Explicit Groups**: formed by user subscriptions
- **Implicit Groups**: implicitly formed by social interactions

Four approaches to detect communities
- **node-centric**
  - Each node in a group satisfies certain properties
- **group-centric**
  - Consider the connections within a group as a whole.
  - The group has to satisfy certain properties without zooming into node-level
- **network-centric**
  - Partition the whole network into several disjoint sets
- **hierarchy-centric**
  - Construct a hierarchical structure of communities
3.1 Node-Centric Community Detection

- Node-Centric Community Detection
  - It require each node in a group to satisfy certain properties.
  - Method Classification
    - Complete mutuality \(\Rightarrow\) clique (clique percolation method)
    - Rechability (geodesic distance) \(\Rightarrow\) k-clique, k-club

- Clique
  - a maximum complete subgraph in which all nodes are adjacent to each other
  - In the above network, there is a clique of 4 nodes, \{5, 6, 7, 8\}.
  - cliques of larger sizes are of much more interest
  - search for the maximum cliques in a graph is an NP-hard problem
3.1 Node-Centric Community Detection

◆ A brute-force approach to find a maximum clique
  - Suppose we now look at node $v_l$
  - Maintain a queue of cliques
    - Initial Time: the queue has a clique of one single node $\{v_l\}$.
  - Perform the following
    - Pop a clique from the queue, say, a clique $B_k$ of size $k$.
      Let $v_l$ denote the last added node into $B_k$.
    - For each of $v_l$'s neighbor $v_j$ (to remove duplicates, we may look at only those nodes whose index is larger than $v_l$), form a new candidate set $B_{k+1} = B_k \cup \{v_j\}$.
    - Validate whether $B_{k+1}$ is a clique by checking whether $v_j$ is adjacent to all nodes in $B_k$. Add to the queue if $B_{k+1}$ is a clique.

1) $v_4 \rightarrow$ the primitive clique $\{4\}$ is added to the queue
2) pop $B_1 = \{4\} \rightarrow \{4,5\}, \{4,6\}$ are added to the queue
3) pop $B_2 = \{4,5\} \rightarrow 5$ is the last added node $\rightarrow \{4,5,6\}, \{4,5,7\}, \{4,5,8\}$ is a candidate clique $\rightarrow \{4,5,6\}$ is added to the queue
4) ...

- This exhaustive search works for small-scale networks, but it becomes impractical for large-scale networks.
3.1 Node-Centric Community Detection

Pruning Strategy to find a maximum clique

- For a clique of size $k$, each node in the clique should maintain at least degree $k - 1$.
- Hence, those nodes with degree less than $k - 1$ cannot be included in the maximum clique, thus can be pruned.

Pruning procedure

- A sub-network is sampled from the given network.
- A clique in the sub-network can be found in a manner
  - The maximum clique found on the sub-network (say, it contains $k$ nodes) serves as the lower bound for pruning.
  - That is, the maximum clique in the original network should contain at least $k$ members.
- The nodes with degree less than or equal to $k - 1$, in conjunction with their edges can be removed from future consideration.
- This process is repeated until the maximum clique can be identified.
3.1 Node-Centric Community Detection

❖ Pruning Strategy to find a maximum clique
  – As social media networks follow a power law distribution for node degrees
    • the majority of nodes have a low degree
  – So, pruning strategy can reduce the network size significantly.
  – Example
    • we randomly sample a sub-network
      
      ![Network Diagram]

      • A maximal clique in the sub-network is of size 3 ({1,2,3}, {1,3,4}, or {4,5,6}).
      • So, all the nodes of degree less than or equal to 2 can be removed from consideration
        ➢ nodes 9 and 2 can be pruned ➢ nodes 1 and 3 is pruned ➢ node 4 is pruned
      • we obtain a much smaller network of nodes {5, 6, 7, 8} where a clique of size 4 can be identified
3.1 Node-Centric Community Detection

- Clique percolation method (CPM) to find overlapping communities
  - Procedure
  - Given a user specified parameter $k$,
  - 1) Find out all cliques of size $k$ in the given network
  - 2) Construct a “clique graph”
    - Two cliques are adjacent if they share “$k - 1$” nodes
  - 3) Each connected component in the clique graph is a “$k$-clique community”
  - Example with $k = 3$
    - we can identify all the cliques of size 3 as follows:
      \[
      \{1, 2, 3\} \quad \{1, 3, 4\} \quad \{4, 5, 6\} \quad \{5, 6, 7\} \quad \{5, 6, 8\} \quad \{5, 7, 8\} \quad \{6, 7, 8\}
      \]
    - From clique graph, we obtain two 3-clique communities: \{1, 2, 3, 4\} and \{4, 5, 6, 7, 8\}
      - In other words, we obtain two overlapping communities.
      - Note that “node 4” belongs to both communities.

3.1 Node-Centric Community Detection

- **Clique percolation method (CPM)** to find overlapping communities
  - Example with $k=4$

3.1 Node-Centric Community Detection

Using **Reachability** to find a community

- **k-clique**
  - a maximal subgraph in which the largest geodesic distance between any two nodes is no greater than $k$. 
    
    $d(v_i, v_j) \leq k \quad \forall v_i, v_j \in V_s$
  
  - where $V_s$ is the set of nodes in the k-clique.
  
  - Note that the geodesic distance is defined on the original network.
    
    ➢ Thus, the geodesic is not necessarily included in the k-clique.
    
    ➢ So the k-clique may have a diameter greater than $k$.
  
  - For example,
    
    ➢ \{1, 2, 3, 4, 5\} form a 2-clique. But, $d(v_4, v_5)$ within the group is 3
3.1 Node-Centric Community Detection

- Using **Reachability** to find a community
  - *k-club*
    - It restricts the geodesic distance within the group to be no greater than *k*.
    - It is a maximal substructure of diameter *k*.
    - The definition of *k-club* is more strict than that of *k-clique*.

- Other community definition based on the rechability
  - *k-plex*, *k-core*, LS sets, and Lambda sets

- It remains a challenge to detect them in large-scale networks.
3.2 Group-Centric Community Detection

- A group-centric criterion considers edges inside a group as whole
  - It allows some nodes in the group to have low connectivity as long as the group overall satisfies certain requirements.

- Density-based community (a quasi-clique)
  - A subgraph $G_s(V_s, E_s)$ is $\gamma$-dense if $\frac{E_s}{V_s(V_s-1)/2} \geq \gamma$.
  - Quasi-clique becomes a clique when $\gamma = 1$.
  - Quasi-clique does not guarantee reachability for each node in the group.
  - It allows the degree of a node to vary, thus is more suitable for large-scale networks.
3.2 Group-Centric Community Detection

Density-based community (a quasi-clique)

- Pruning strategy to find the maximal $\gamma$-dense quasi-clique in a network
  - Phase I: Local search
    - Sample a sub-network from the given network and search for a maximal quasi-clique in the sub-network.
      - A randomized search strategy can be exploited.
      - A greedy approach to expand a quasi-clique by encompassing those high-degree neighboring nodes until the density drops below $\gamma$.
    - As a result, we know a $\gamma$-dense quasi-clique of size $k$
3.2 Group-Centric Community Detection

Density-based community (a quasi-clique)

- Pruning strategy to find the maximal $\gamma$-dense quasi-clique in a network
  - Phase II: Heuristic pruning
    - prune “peelable” nodes and their incident edges.
      - A node $v$ is peelable if $v$ and its neighbors all have degree less than $k \cdot \gamma$
      - Such a node $v$ is less likely to contribute to a larger quasi-clique by including such a node.
    - We can start from low-degree nodes and recursively remove peelable nodes in the original network.
    - This process is repeated until the network is reduced to a maximal quasi-clique
  - the algorithm does not guarantee to be optimal, but it works reasonably well in most cases
3.3 Network-Centric Community Detection

(Network-Centric Community Detection)
  - consider the global topology of a network.
  - It aims to partition nodes of a network into a number of disjoint sets
    - A group in this case is not defined independently

(Vertex Similarity)
  - Vertex similarity is defined in terms of the similarity of their social circles,
    e.g., the number of friends they share in common

  - Various Definitions and Methods for vertex similarity
    - Structurally equivalence
    - Automorphic equivalence
    - Regular equivalence
    - k-means algorithm with connection feature
    - Jaccard similarity
    - Cosine similarity
3.3 Network-Centric Community Detection

Structurally Equivalence
- Actors $v_i$ and $v_j$ are structurally equivalent, if for any actor $v_k$ that $v_k \neq v_i$ and $v_k \neq v_j$, $e(v_i, v_k) \in E$ iff $e(v_j, v_k) \in E$.
  - actors $v_i$ and $v_j$ are connecting to exactly the same set of actors in a network
  - For example,
    - nodes 1 and 3 are structurally equivalent
    - nodes 5 and 6 are structurally equivalent

- Nodes of the same equivalence class form a community
  - But, it is too restrictive for practical use
  - Other relaxed definitions of equivalence such as “automorphic equivalence” and “regular equivalence” are proposed
    - But no scalable approach exists to find them.
3.3 Network-Centric Community Detection

**Jaccard and Cosine Similarity**

- For two nodes \(v_i\) and \(v_j\) in a network, the similarity between the two are defined as

\[
Jaccard(v_i, v_j) = \frac{|N_i \cap N_j|}{|N_i \cup N_j|} = \frac{\sum_k A_{ik} A_{jk}}{|N_i| + |N_j| - \sum_k A_{ik} A_{jk}}
\]

\[
\text{Cosine}(v_i, v_j) = \frac{A_i \cdot A_j}{||A_i|| \times ||A_j||} = \frac{\sum_k A_{ik} A_{jk}}{\sqrt{\sum_s A_{is}^2 \cdot \sum_t A_{jt}^2}} = \frac{|N_i \cap N_j|}{\sqrt{|N_i| \cdot |N_j|}}
\]

- where \(|*|\) is the cardinality of the set, \(\cdot\) denote the inner product of vectors, \(||*||\) is the norm (or magnitude) of the vector.

- For example, \(N_4 = \{1,3,5,6\}\) and \(N_6 = \{4,5,7,8\}\)

\[
Jaccard(4, 6) = \frac{|\{5\}|}{|\{1,3,4,5,6,7,8\}|} = \frac{1}{7} \quad \text{Cosine}(4, 6) = \frac{|\{5\}|}{\sqrt{4 \cdot 4}} = \frac{1}{4}
\]
3.3 Network-Centric Community Detection

◊ Jaccard and Cosine Similarity

- For example, \(N_7 = \{5,6,8,9\}\) and \(N_9 = \{7\}\)
  - \(N_7 \cap N_9 = \emptyset\)

\[ Jaccard(7,9) = 0 \quad \text{Cosine}(7,9) = 0 \]

- However, two nodes are likely to share some similarity if they are connected

- A modification
  - Include node \(v\) when we compute \(N_v\)
  - \(N_7 = \{5,6,7,8,9\}\) and \(N_9 = \{7,9\}\)

\[
Jaccard(7,9) = \frac{|\{7,9\}|}{|\{5,6,7,8,9\}|} = \frac{2}{5} \quad \text{Cosine}(4,6) = \frac{|\{7,9\}|}{\sqrt{5 \cdot 2}} = \frac{2}{\sqrt{10}}
\]
3.3 Network-Centric Community Detection

💎 Jaccard and Cosine Similarity
- Similarity-based community detection method
  - 1) Set a community threshold $\sigma$
  - 2) the similarity for each pair of nodes in the given graph
  - 3) If the similarity of two nodes is over the threshold $\sigma$, the two nodes form the same community.

\[
\sigma = 0.5
\]

- Compute the similarity for each pair of nodes
  - time-consuming when $n$ is very large
  - [Note] Shingling Algorithm
    - Discovering Large Dense Subgraphs in Massive Graphs, David Gibson Ravi Kumar Andrew Tomkins, IBM Almaden Research Center
    - Finding dense clusters in web graph, Prof. Donald J. Patterson Scribe: Minh Doan, Ching-wei Huang, Siripen Pongpaichet
3.4 Hierarchy-Centric Community Detection

◮ Hierarchy-centric methods
  – build a hierarchical structure of communities based on network topology
  – two types of hierarchical clustering
    • Divisive
    • Agglomerative

◮ Divisive Clustering
  – 1. Put all objects in one cluster
  – 2. Repeat until all clusters are singletons
    • a) choose a cluster to split
      ➢ what criterion?
    • b) replace the chosen cluster with the sub-clusters
      ➢ split into how many?
3.4 Hierarchy-Centric Community Detection

Divisive Clustering

- A Method: Cut the “weakest” tie
  - At each iteration, find out the weakest edge.
    - This kind of edge is most likely to be a tie connecting two communities.
  - Remove the edge.
    - Once a network is decomposed into two connected components, each component is considered a community.
  - Update the strength of links.
  - This iterative process is applied to each community to find sub-communities.
3.4 Hierarchy-Centric Community Detection

Divisive Clustering

  - find the weak ties based on “edge betweenness”
  - Edge betweenness
    - the number of shortest paths between pair of nodes pass along the edge
    - utilized to find the “weakest” tie for hierarchical clustering

\[
C_B(e(v_i, v_j)) = \begin{cases} 
\frac{\sigma_{st}(e(v_i, v_j))}{\sigma_{st}} & \text{if } i < j \\
0 & \text{if } i = j \\
C_B(e(v_j, v_i)) & \text{if } i > j 
\end{cases}
\]

- where
  - \(\sigma_{st}\) is the total number of shortest paths between nodes \(v_s\) and \(v_t\)
  - \(\sigma_{st}(e(v_i, v_j))\) is the number of shortest paths between nodes \(v_s\) and \(v_t\) that pass along the edge \(e(v_i, v_j)\).

- It proposed to progressively remove the edges with the highest betweenness.
3.4 Hierarchy-Centric Community Detection

Divisive Clustering

  
  - Example

$$
\sigma_{st}(e(v_1, v_2)) = \sigma_{st}(e(v_2, v_1))
$$

- Negatives for divisive clustering
  
  - edge betweenness-based scheme requires high computation
  - removal of an edge will lead to the recomputation of betweenness for all edges
3.4 Hierarchy-Centric Community Detection

◊ Agglomerative Clustering
  – begins with base (singleton) communities
  – merges them into larger communities with certain criterion.
    • One example criterion: modularity
      ➢ Let $e_{ij}$ be the fraction of edges in the network that connect nodes in community $i$ to those in community $j$
      ➢ Let $a_i = \sum_j e_{ij}$, then the modularity $Q = \sum_i (e_{ii} - a_i^2)$
      ➢ values approaching $Q = 1$ indicate networks with strong community structure
      ➢ values for real networks typically fall in the range from 0.3 to 0.7
    • Two communities are merged if the merge results in the largest increase of overall modularity
      – The merge continues until no merge can be found to improve the modularity.
3.4 Hierarchy-Centric Community Detection

◊ Agglomerative Clustering

– In the dendrogram, the circles at the bottom represent the individual nodes of the network.
– As we move up the tree, the nodes join together to form larger and larger communities, as indicated by the lines, until we reach the top, where all are joined together in a single community.
– Alternatively, the dendrogram depicts an initially connected network splitting into smaller and smaller communities as we go from top to bottom.
– A cross section of the tree at any level, such the one indicated by a dotted line, will give the communities at that level.
3.4 Hierarchy-Centric Community Detection

 shm Divisive vs. Agglomerative Clustering
 – Zachary's karate club study

Zachary observed 34 members of a karate club over a period of two years. During the course of the study, a disagreement developed between the administrator (34) of the club and the club's instructor (1), which ultimately resulted in the instructor's leaving and starting a new club, taking about a half of the original club's members with him.
3.4 Hierarchy-Centric Community Detection

 Omnibus vs. Agglomerative Clustering

- **Divisive**

- **Agglomerative**