CHAPTER 3

Community Detection and Evaluation

Social networks of various kinds 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 cohesive groups in a social network (i.e., *community detection*) remains a core problem in social network analysis. Finding out these groups also helps for other related social computing tasks. Many approaches have been proposed in the past. These approaches can be separated into four categories: node-centric, group-centric, network-centric, and hierarchy-centric (Tang and Liu, 2010b). We introduce definitions and present representative community detection approaches in each category.

3.1 NODE-CENTRIC COMMUNITY DETECTION

The community detection methods based on node-centric criteria require *each node* in a group to satisfy certain properties. We discuss these methods according to these criteria.

3.1.1 COMPLETE MUTUALITY

An ideal cohesive subgroup is a *clique*. It is a maximum complete subgraph in which all nodes are adjacent to each other. For example, in the network in Figure 1.1, there is a clique of 4 nodes, {5, 6, 7, 8}. Typically, cliques of larger sizes are of much more interest. However, the search for the maximum cliques in a graph is an NP-hard problem.

One brute-force approach is to traverse all nodes in a network. For each node, check whether there is any clique of a specified size that contains the node. Suppose we now look at node v_ℓ. We can maintain a queue of cliques. It is initialized with a clique of one single node $\{v_\ell\}$. Then we perform the following:

- Pop a clique from the queue, say, a clique B_k of size k. Let v_i denote the last added node into B_k.
- For each of v_i's neighbor v_j (to remove duplicates, we may look at only those nodes whose index is larger than v_i), 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.

Take the network in Figure 1.1 as an example. Suppose we start from node $B_1 = \{4\}$. For each of its friends with a larger index, we obtain a clique of size 2. Thus, we have $\{4, 5\}$ and $\{4, 6\}$ added into the queue. Now suppose we pop $B_2 = \{4, 5\}$ from the queue. Its last added element is node 5. We can expand the set following node 5's connections and generate three candidate sets: $\{4, 5, 6\}$, $\{4, 5, 7\}$ and $\{4, 5, 8\}$. Among them, only $\{4, 5, 6\}$ is a clique as node 6 is connected both nodes 4 and 5. Thus, $\{4, 5, 6\}$ is appended to the queue for further expansion for larger cliques.

The exhaustive search above works for small-scale networks, but it becomes impractical for large-scale networks. If the goal is to find out a maximum clique, then a strategy is to effectively prune those nodes and edges that are unlikely to be contained in the 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. We can recursively apply the pruning procedure below to a given network:

- A sub-network is sampled from the given network. A clique in the sub-network can be found in a greedy manner, e.g., expanding a clique by adding an adjacent node with the highest degree.
- 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. Hence, in order to find a clique of size larger than k, the nodes with degree less than or equal to $k - 1$, in conjunction with their connections can be removed from future consideration. As social media networks follow a power law distribution for node degrees, i.e., the majority of nodes have a low degree, this pruning strategy can reduce the network size significantly.

This process is repeated until the original network is shrunk into a reasonable size and the maximum clique can either be identified directly, or have already been identified in one of the sub-networks. A similar pruning strategy is discussed for directed networks as well (Kumar et al., 1999).

Suppose we randomly sample a sub-network from the network in Figure 1.1. It consists of nodes 1 to 6. A maximal clique in the sub-network is of size 3 ($\{1, 2, 3\}$ or $\{1, 3, 4\}$). If there exists a larger clique (i.e., size > 3) in the original network, all the nodes of degree less than or equal to 2 can be removed from consideration. Hence, nodes 9 and 2 can be pruned. Then, the degree of nodes 1 and 3 is reduced to 2, thus they can also be removed. This further leaves node 4 with only 2 connections, which can be removed as well. After this pruning, we obtain a much smaller network of nodes $\{5, 6, 7, 8\}$. And in this pruned network, a clique of size 4 can be identified. It is exactly the maximum clique.

A clique is a very strict definition, and it can rarely be observed in a huge size in real-world social networks. This structure is very unstable as the removal of any edge in it will render it an invalid clique. Practitioners typically use identified cliques as cores or seeds for subsequent expansion for a

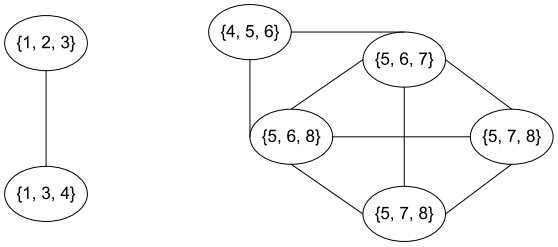


Figure 3.1: A Clique Graph

community. The *clique percolation method* (CPM) is such a scheme to find overlapping communities in networks (Palla et al., 2005). Given a user specified parameter k, it works as follows:

- Find out all cliques of size k in the given network;
- Construct a clique graph. Two cliques are adjacent if they share $k - 1$ nodes;
- Each connected component in the clique graph is a community.

Take the network in Figure 1.1 as an example. For $k = 3$, we can identify all the cliques of size 3 as follows:

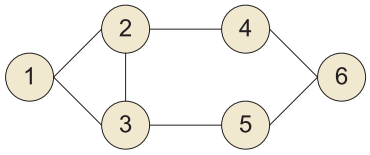
{1, 2, 3}	{1, 3, 4}	{4, 5, 6}	{5, 6, 7}	{5, 6, 8}	{5, 7, 8}	{6, 7, 8}

Then we have the clique graph as in Figure 3.1. Two cliques are connected as long as they share $k - 1$ (2 in our case) nodes. In the clique graph, there are two connected components. The nodes in each component fall into one community. Consequently, we obtain two communities: {1, 2, 3, 4} and {4, 5, 6, 7, 8}. Note that node 4 belongs to both communities. In other words, we obtain two overlapping communities.

The clique percolation method requires the enumeration of all the possible cliques of a fixed size k. This can be computational prohibitive for large-scale social media networks. Other forms of subgraph close to a clique thus are proposed to capture the community structure, which will be discussed next.

3.1.2 REACHABILITY

This type of community considers the reachability among actors. In the extreme case, two nodes can be considered as belonging to one community if there exists a path between the two nodes. Thus each connected component is a community. The components can be efficiently identified in $O(n + m)$ time (Hopcroft and Tarjan, 1973), linear with respect to number of nodes and edges in a network. However, in real-world networks, a giant component tends to form while many others are singletons and minor communities (Kumar et al., 2006). Those minor communities can be identified as connected components. Yet more efforts are required to find communities in the giant component.



cliques: {1, 2, 3}
2-cliques: {1, 2, 3, 4, 5, }, {2, 3, 4, 5, 6}
2-clubs: {1, 2, 3, 4, }, {1, 2, 3, 5}, {2, 3, 4, 5, 6}

Figure 3.2: An example to show the difference of k-clique and k-club (based on (Wasserman and Faust, 1994))

Conceptually, there should be a short path between any two nodes in a group. Some well-studied structures in social sciences are the following:

- *k-clique* is a maximal subgraph in which the largest geodesic distance between any two nodes is no greater than k. That is,

$$d(v_i, v_j) \leq k \quad \forall v_i, v_j \in V_s$$

  where V_s is the set of nodes in the subgraph. Note that the geodesic distance is defined on the original network. Thus, the geodesic is not necessarily included in the group structure. So a k-clique may have a diameter greater than k. For instance, in Figure 3.2, {1, 2, 3, 4, 5} form a 2-clique. But the geodesic distance between nodes 4 and 5 within the group is 3.

- *k-club* 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. A k-club is often a subset of a k-clique. In the example in Figure 3.2, The 2-clique structure {1, 2, 3, 4, 5} contains two 2-clubs, {1, 2, 3, 4} and {1, 2, 3, 5}.

There are other definitions of communities such as k-plex, k-core, LS sets, and Lambda sets (Wasserman and Faust, 1994). They are typically studied in traditional social sciences. Solving the k-club problem often requires involved combinatorial optimization (McClosky and Hicks, 2009). It remains a challenge to generalize them to large-scale networks.

3.2 GROUP-CENTRIC COMMUNITY DETECTION

A group-centric criterion considers connections inside a group as whole. It is acceptable to have some nodes in the group to have low connectivity as long as the group overall satisfies certain requirements. One such example is *density-based groups*. A subgraph $G_s(V_s, E_s)$ is γ-dense (also called a *quasi-clique* (Abello et al., 2002)) if

$$\frac{E_s}{V_s(V_s - 1)/2} \geq \gamma. \tag{3.1}$$

Clearly, the quasi-clique becomes a clique when $\gamma = 1$. Note that this density-based group-centric criterion does not guarantee reachability for each node in the group. It allows the degree of a node to vary[1], thus is more suitable for large-scale networks.

However, it is not a trivial task to search for quasi-cliques. Strategies similar to those of finding cliques can be exploited. In Abello et al. (2002), a greedy search followed by pruning is employed to find the maximal γ-dense quasi-clique in a network. The iterative procedure consists of two steps - local search and heuristic pruning.

- *Local search*: Sample a sub-network from the given network and search for a maximal quasi-clique in the sub-network. A greedy approach is to aggressively expand a quasi-clique by encompassing those high-degree neighboring nodes until the density drops below γ. In practice, a randomized search strategy can also be exploited.

- *Heuristic pruning*: If we know a γ-dense quasi-clique of size k, then a heuristic is to prune those "peelable" nodes and their incident edges. A node v is *peelable* if v and its neighbors all have degree less than $k\gamma$ because it 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 reasonable size so that a maximal quasi-clique can be found directly. Though the solution returned by the algorithm does not guarantee to be optimal, it works reasonably well in most cases (Abello et al., 2002).

3.3 NETWORK-CENTRIC COMMUNITY DETECTION

Network-centric community detection has to consider the global topology of a network. It aims to *partition* nodes of a network into a number of disjoint sets. Typically, network-centric community detection aims to optimize a criterion defined over a network partition rather than over one group. A group in this case is not defined independently.

3.3.1 VERTEX SIMILARITY

Vertex similarity is defined in terms of the similarity of their social circles, e.g., the number of friends two share in common. A key related concept is *structural 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 $e(v_j, v_k) \in E$. In other words, actors v_i and v_j are connecting to exactly the same set of actors in a network. If the interaction is represented as a matrix, then rows (columns) of v_i and v_j are the same except for the diagonal entries. Nodes 1 and 3 in Figure 1.1 are structurally equivalent. So are nodes 5 and 6. A closer examination at its adjacency matrix (Table 1.3) reveals that those structurally equivalent nodes share the same rows (columns). Nodes of the same equivalence class form a community.

[1] It removes the need of being connected to at least k other nodes in the same group.

Since structural equivalence is too restrictive for practical use, other relaxed definitions of equivalence such as *automorphic equivalence* and *regular equivalence* are proposed (Hanneman and Riddle, 2005), but no scalable approach exists to find automorphic equivalence or regular equivalence. Alternatively, some simplified similarity measures can be used. They consider one's connections as features for actors, and they assume actors sharing similar connections tend to reside within the same community. Once a similarity measure is determined, classical k-means clustering or hierarchical clustering algorithm (Tan et al., 2005) can be applied to find communities in a network.

Commonly used similarity measures include Jaccard similarity (Gibson et al., 2005) and cosine similarity (Hopcroft et al., 2003). For two nodes v_i and v_j in a network, the similarity between the two are defined as

$$Jaccard(v_i, v_j) = \frac{|N_i \cap N_j|}{|N_i \cup N_j|} = \frac{\sum_k A_{ik}A_{jk}}{|N_i| + |N_j| - \sum_k A_{ik}A_{jk}}, \tag{3.2}$$

$$Cosine(v_i, v_j) = \frac{\sum_k A_{ik}A_{jk}}{\sqrt{\sum_s A_{is}^2 \cdot \sum_t A_{jt}^2}} = \frac{|N_i \cap N_j|}{\sqrt{|N_i| \cdot |N_j|}}. \tag{3.3}$$

In the equations, N_i denotes the neighbors of node v_i and $|*|$ the cardinality. Both similarity measures are within the range between 0 and 1.

For example, in the network in Figure 1.1, $N_4 = \{1, 3, 5, 6\}$, and $N_6 = \{4, 5, 7, 8\}$. Thus, the similarity between the two nodes are:

$$Jaccard(4, 6) = \frac{|\{5\}|}{|\{1, 3, 4, 5, 6, 7, 8\}|} = \frac{1}{7}, \tag{3.4}$$

$$Cosine(4, 6) = \frac{|\{5\}|}{\sqrt{4 \cdot 4}} = \frac{1}{4}. \tag{3.5}$$

However, based on this definition, the similarity of two adjacent nodes could be 0. For example, the similarity of nodes 7 and 9 is 0 because $N_7 = \{5, 6, 8, 9\}$, $N_9 = \{7\}$, and $N_7 \cap N_9 = \phi$, even though they are connected. This is reasonable from the perspective of structural equivalence. However, from the correlation aspect, statistically, two nodes are likely to share some similarity if they are connected. A modification is to include node v when we compute N_v. In equivalence, the diagonal entries of the adjacency matrix of a network is set to 1 rather than default 0. In this case, $N_7 = \{5, 6, 7, 8, 9\}$, $N_9 = \{7, 9\}$. It follows that $N_7 \cap N_9 = \{7, 9\}$:

$$Jaccard(7, 9) = \frac{|\{7, 9\}|}{|\{5, 6, 7, 8, 9\}|} = \frac{2}{5},$$

$$Cosine(7, 9) = \frac{|\{7, 9\}|}{\sqrt{2 \cdot 5}} = \frac{2}{\sqrt{10}}.$$

Normal similarity-based methods have to compute the similarity for each pair of nodes, totaling $O(n^2)$. It is time-consuming when n is very large. Thus, Gibson et al. (2005) present an

efficient two-level shingling algorithm for fast computation of web communities. Generally speaking, the *shingling* algorithm maps each vector (the connection of actors) into a constant number of "shingles". If two actors are similar, they share many shingles; otherwise, they share few. After initial shingling, each shingle is associated with a group of actors. In a similar vein, the shingling algorithm can be applied to the first-level shingles as well. So similar shingles end up sharing the same meta-shingles. Then all the actors relating to one meta-shingle form one community. This two-level shingling can be efficiently computed even for large-scale networks. Its time complexity is approximately linear to the number of edges.

3.3.2 LATENT SPACE MODELS

A latent space model maps nodes in a network into a low-dimensional Euclidean space such that the proximity between nodes based on network connectivity are kept in the new space (Hoff et al., 2002; Handcock et al., 2007), then the nodes are clustered in the low-dimensional space using methods like k-means (Tan et al., 2005). One representative approach is *multi-dimensional scaling* (MDS) (Borg and Groenen, 2005). Typically, MDS requires the input of a proximity matrix $P \in \mathbb{R}^{n\times n}$, with each entry P_{ij} denoting the distance between a pair of nodes i and j in the network. Let $S \in \mathbb{R}^{n\times \ell}$ denote the coordinates of nodes in the ℓ-dimensional space such that S is column orthogonal. It can be shown (Borg and Groenen, 2005; Sarkar and Moore, 2005) that

$$SS^T \approx -\frac{1}{2}(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T)(P \circ P)(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T) = \widetilde{P}, \tag{3.6}$$

where I is the identity matrix, $\mathbf{1}$ an n-dimensional column vector with each entry being 1, and $\circ$ the element-wise matrix multiplication. It follows that S can be obtained via minimizing the discrepancy between $\widetilde{P}$ and SS^T as follows:

$$\min \|SS^T - \widetilde{P}\|_F^2. \tag{3.7}$$

Suppose V contains the top ℓ eigenvectors of $\widetilde{P}$ with largest eigenvalues, Λ is a diagonal matrix of top ℓ eigenvalues $\Lambda = diag(\lambda_1, \lambda_2, \cdots, \lambda_\ell)$. The optimal S is $S = V\Lambda^{\frac{1}{2}}$. Note that this multi-dimensional scaling corresponds to an eigenvector problem of matrix $\widetilde{P}$. Thus, the classical k-means algorithm can be applied to S to find community partitions.

Take the network in Figure 1.1 as an example. Given the network, the geodesic distance between each pair of nodes is given in the proximity matrix P as in Eq. (3.8). Hence, we can

compute the corresponding matrix $\widetilde{P}$ following Eq. (3.6).

$$P = \begin{bmatrix} 0 & 1 & 1 & 1 & 2 & 2 & 3 & 3 & 4 \\ 1 & 0 & 1 & 2 & 3 & 3 & 4 & 4 & 5 \\ 1 & 1 & 0 & 1 & 2 & 2 & 3 & 3 & 4 \\ 1 & 2 & 1 & 0 & 1 & 1 & 2 & 2 & 3 \\ 2 & 3 & 2 & 1 & 0 & 1 & 1 & 1 & 2 \\ 2 & 3 & 2 & 1 & 1 & 0 & 1 & 1 & 2 \\ 3 & 4 & 3 & 2 & 1 & 1 & 0 & 1 & 1 \\ 3 & 4 & 3 & 2 & 1 & 1 & 1 & 0 & 2 \\ 4 & 5 & 4 & 3 & 2 & 2 & 1 & 2 & 0 \end{bmatrix}, \tag{3.8}$$

$$\widetilde{P} = \begin{bmatrix} 2.46 & 3.96 & 1.96 & 0.85 & -0.65 & -0.65 & -2.21 & -2.04 & -3.65 \\ 3.96 & 6.46 & 3.96 & 1.35 & -1.15 & -1.15 & -3.71 & -3.54 & -6.15 \\ 1.96 & 3.96 & 2.46 & 0.85 & -0.65 & -0.65 & -2.21 & -2.04 & -3.65 \\ 0.85 & 1.35 & 0.85 & 0.23 & -0.27 & -0.27 & -0.82 & -0.65 & -1.27 \\ -0.65 & -1.15 & -0.65 & -0.27 & 0.23 & -0.27 & 0.68 & 0.85 & 1.23 \\ -0.65 & -1.15 & -0.65 & -0.27 & -0.27 & 0.23 & 0.68 & 0.85 & 1.23 \\ -2.21 & -3.71 & -2.21 & -0.82 & 0.68 & 0.68 & 2.12 & 1.79 & 3.68 \\ -2.04 & -3.54 & -2.04 & -0.65 & 0.85 & 0.85 & 1.79 & 2.46 & 2.35 \\ -3.65 & -6.15 & -3.65 & -1.27 & 1.23 & 1.23 & 3.68 & 2.35 & 6.23 \end{bmatrix}.$$

Suppose we want to map the original network into a 2-dimensional space; we obtain V, Λ, and S as follows:

$$V = \begin{bmatrix} -0.33 & 0.05 \\ -0.55 & 0.14 \\ -0.33 & 0.05 \\ -0.11 & -0.01 \\ 0.10 & -0.06 \\ 0.10 & -0.06 \\ 0.32 & 0.11 \\ 0.28 & -0.79 \\ 0.52 & 0.58 \end{bmatrix}, \quad \Lambda = \begin{bmatrix} 21.56 & 0 \\ 0 & 1.46 \end{bmatrix}, \quad S = V\Lambda^{1/2} = \begin{bmatrix} -1.51 & 0.06 \\ -2.56 & 0.17 \\ -1.51 & 0.06 \\ -0.53 & -0.01 \\ 0.47 & -0.08 \\ 0.47 & -0.08 \\ 1.47 & 0.14 \\ 1.29 & -0.95 \\ 2.42 & 0.70 \end{bmatrix}.$$

The network can be visualized in the 2-dimensional space in Figure 3.3. Because nodes 1 and 3 are structurally equivalent, they are mapped into the same position in the latent space. So are nodes 5 and 6. k-means can be applied to S in order to obtain disjoint partitions of the network. At the end, we obtain two clusters {1, 2, 3, 4}, {5, 6, 7, 8, 9}, which can be represented as a partition

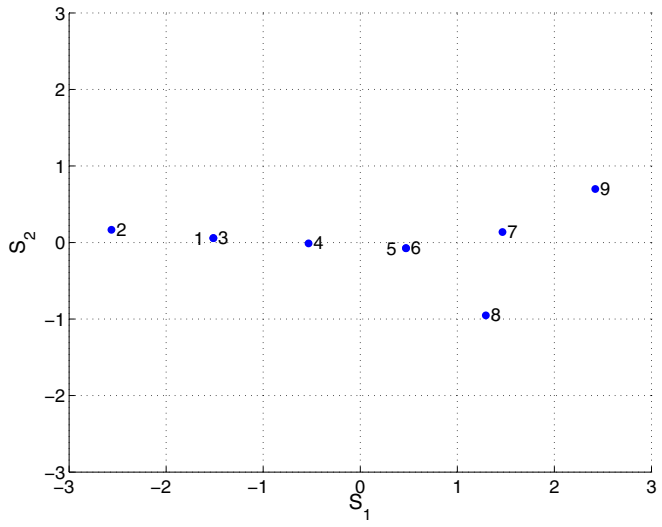


Figure 3.3: Network in the Latent Space

matrix below:

$$H = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}.$$

3.3.3 BLOCK MODEL APPROXIMATION

Block models approximate a given network by a block structure. The basic idea can be visualized in Tables 3.1 and 3.2. The adjacency matrix of the network in Figure 1.1 is shown in Table 3.1. We highlight those entries that indicate an edge between two nodes. The adjacency matrix can be approximated by a block structure as shown in Table 3.2. Each block represents one community. Therefore, we approximate a given adjacency matrix A as follows:

$$A \approx S\Sigma S^T, \tag{3.9}$$

where $S \in \{0, 1\}^{n \times k}$ is the block indicator matrix with $S_{ij} = 1$ if node i belongs to the j-th block, Σ a $k \times k$ matrix indicating the block (group) interaction density, and k the number of blocks. A

Table 3.1: Adjacency Matrix

-	1	1	1	0	0	0	0	0
1	-	1	0	0	0	0	0	0
1	1	-	1	0	0	0	0	0
1	0	1	-	1	1	0	0	0
0	0	0	1	-	1	1	1	0
0	0	0	1	1	-	1	1	0
0	0	0	0	1	1	-	1	1
0	0	0	0	1	1	1	-	0
0	0	0	0	0	0	1	0	-

Table 3.2: Ideal Block Structure

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

natural objective is to minimize the following:

$$\min \|A - S\Sigma S^T\|_F^2. \tag{3.10}$$

The discreteness of S makes the problem NP-hard. We can relax S to be continuous but satisfy certain orthogonal constraints, i.e., $S^T S = I_k$, then the optimal S corresponds to the top k eigenvectors of A with maximum eigenvalues. Similar to the latent space model, k-means clustering can be applied to S to recover the community partition H.

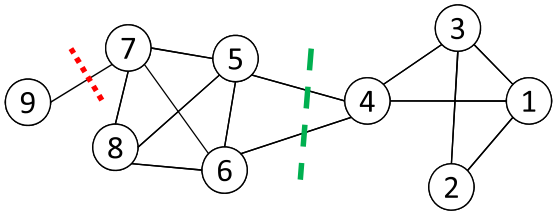


Figure 3.4: Two Different Cuts of the Toy Network in Figure 1.1

For the network in Figure 1.1, the top two eigenvectors of the adjacency matrix are

$$S = \begin{bmatrix} 0.20 & -0.52 \\ 0.11 & -0.43 \\ 0.20 & -0.52 \\ 0.38 & -0.30 \\ 0.47 & 0.15 \\ 0.47 & 0.15 \\ 0.41 & 0.28 \\ 0.38 & 0.24 \\ 0.12 & 0.11 \end{bmatrix}, \Sigma = \begin{bmatrix} 3.5 & 0 \\ 0 & 2.4 \end{bmatrix}.$$

As indicated by the sign of the second column of S, nodes $\{1, 2, 3, 4\}$ form a community, and $\{5, 6, 7, 8, 9\}$ is another community, which can be obtained by a k-means clustering applied to S.

3.3.4 SPECTRAL CLUSTERING

Spectral clustering (Luxburg, 2007) is derived from the problem of graph partition. Graph partition aims to find out a partition such that the cut (the total number of edges between two disjoint sets of nodes) is minimized. For instance, the green cut (thick dashed line) between two sets of nodes $\{1, 2, 3, 4\}$ and $\{5, 6, 7, 8, 9\}$ in Figure 3.4 is 2 as there are two edges $e(4, 5)$ and $e(4, 6)$. Intuitively, if two communities are well separated, the cut between them should be small. Hence, a community detection problem can be reduced to finding the minimum cut in a network. This *minimum cut* problem can be solved efficiently. It, however, often returns imbalanced communities, with one being trivial or a singleton, i.e., a community consisting of only one node. In the network in Figure 3.4, for example, the minimum cut is 1, between $\{9\}$ and $\{1, 2, 3, 4, 5, 6, 7, 8\}$.

Therefore, the objective function is modified so that the group sizes of communities are considered. Two commonly used variants are *ratio cut* and *normalized cut*. Let $\pi = (C_1, C_2, \cdots, C_k)$ be a graph partition such that $C_i \cap C_j = \phi$ and $\cup_{i=1}^{k} C_i = V$. The ratio cut and the normalized cut

are defined as:

$$\text{Ratio Cut}(\pi) = \frac{1}{k}\sum_{i=1}^{k}\frac{cut(C_i, \bar{C}_i)}{|C_i|}, \tag{3.11}$$

$$\text{Normalized Cut}(\pi) = \frac{1}{k}\sum_{i=1}^{k}\frac{cut(C_i, \bar{C}_i)}{vol(C_i)}. \tag{3.12}$$

where $\bar{C}_i$ is the complement of C_i, and $vol(C_i) = \sum_{v\in C_i} d_v$. Both objectives attempt to minimize the number of edges between communities, yet avoid the bias of trivial-size communities like singletons.

Suppose we partition the network in Figure 1.1 into two communities, with $C_1 = \{9\}$ and $C_2 = \{1, 2, 3, 4, 5, 6, 7, 8\}$. Let this partition be denoted as π_1. It follows that $cut(C_1, \bar{C}_1) = 1$, $|C_1| = 1$, $|C_2| = 8$, $vol(C_1) = 1$, and $vol(C_2) = 27$. Consequently,

$$\begin{aligned}\text{Ratio Cut}(\pi_1) &= \frac{1}{2}\left(\frac{1}{1}+\frac{1}{8}\right) = 9/16 = 0.56,\\ \text{Normalized Cut}(\pi_1) &= \frac{1}{2}\left(\frac{1}{1}+\frac{1}{27}\right) = 14/27 = 0.52.\end{aligned}$$

Now for another more balanced partition π_2 with $C_1 = \{1, 2, 3, 4\}$, and $C_2 = \{5, 6, 7, 8, 9\}$, we have

$$\begin{aligned}\text{Ratio Cut}(\pi_2) &= \frac{1}{2}\left(\frac{2}{4}+\frac{2}{5}\right) = 9/20 = 0.45 < \text{Ratio Cut}(\pi_1),\\ \text{Normalized Cut}(\pi_2) &= \frac{1}{2}\left(\frac{2}{12}+\frac{2}{16}\right) = 7/48 = 0.15 < \text{Normalized Cut}(\pi_1).\end{aligned}$$

Though the cut of partition π_1 is smaller, partition π_2 is preferable based on the ratio cut or the normalized cut.

Nevertheless, finding the minimum ratio cut or normalized cut is NP-hard. An approximation is to use spectral clustering. Both ratio cut and normalized cut can be formulated as a min-trace problem like below

$$\min_{S\in\{0,1\}^{n\times k}} Tr(S^T\widetilde{L}S), \tag{3.13}$$

with the (normalized) graph Laplacian $\widetilde{L}$ defined as follows:

$$\widetilde{L} = \begin{cases} D - A & \text{(Graph Laplacian for Ratio Cut)}\\ I - D^{-1/2}AD^{-1/2} & \text{(Normalized Graph Laplacian for Normalized Cut)}\end{cases} \tag{3.14}$$

with $D = diag(d_1, d_2, \cdots, d_n)$. Akin to block model approximation, we solve the following spectral clustering problem based on a relaxation to S (Luxburg, 2007).

$$\min_{S} Tr(S^T\widetilde{L}S) \quad s.t.\ S^TS = I_k \tag{3.15}$$

Then, S corresponds to the top eigenvectors of $\widetilde{L}$ with the smallest eigenvalues. For the network in Figure 1.1,

$$D = diag(3, 2, 3, 4, 4, 4, 4, 3, 1);$$

and the graph Laplacian is

$$\widetilde{L} = D - A = \begin{bmatrix} 3 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 4 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 4 & -1 & -1 & -1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 4 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 4 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & -1 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix},$$

with its two smallest eigenvectors being

$$S = \begin{bmatrix} 0.33 & -0.38 \\ 0.33 & -0.48 \\ 0.33 & -0.38 \\ 0.33 & -0.12 \\ 0.33 & 0.16 \\ 0.33 & 0.16 \\ 0.33 & 0.30 \\ 0.33 & 0.24 \\ 0.33 & 0.51 \end{bmatrix}.$$

Typically, the first eigenvector does not contain any community information. For the example above, all the nodes are assigned with the same value, meaning that all reside in the same community. Thus, the first eigenvector is often discarded. In order to find out k communities, $k - 1$ smallest eigenvectors (except the first one) are used to feed into k-means for clustering. In our example, the second column of S, as indicated by the sign, tells us that the network can be divided into two groups $\{1, 2, 3, 4\}$ and $\{5, 6, 7, 8, 9\}$.

3.3.5 MODULARITY MAXIMIZATION

Modularity (Newman, 2006a) is proposed specifically to measure the strength of a community partition for real-world networks by taking into account the degree distribution of nodes. Given a network of n nodes and m edges, the *expected number of edges* between nodes v_i and v_j is $d_i d_j / 2m$, where d_i and d_j are the degrees of node v_i and v_j, respectively. Considering one edge from node v_i connecting to all nodes in the network randomly, it lands at node v_j with probability $d_j / 2m$. As there

are d_i such edges, the expected number of connections between the two are $d_i d_j/2m$. For example, the network in Figure 1.1 has 9 nodes and 14 edges. The expected number of edges between nodes 1 and 2 is $3 \times 2/(2 \times 14) = 3/14$.

So $A_{ij} - d_i d_j/2m$ measures how far the true network interaction between nodes i and j (A_{ij}) deviates from the expected random connections. Given a group of nodes C, the strength of community effect is defined as

$$\sum_{i \in C, j \in C} A_{ij} - d_i d_j/2m.$$

If a network is partitioned into k groups, the overall community effect can be summed up as follows:

$$\sum_{\ell=1}^{k} \sum_{i \in C_\ell, j \in C_\ell} \left(A_{ij} - d_i d_j/2m\right).$$

Modularity is defined as

$$Q = \frac{1}{2m} \sum_{\ell=1}^{k} \sum_{i \in C_\ell, j \in C_\ell} \left(A_{ij} - d_i d_j/2m\right). \tag{3.16}$$

where the coefficient $1/2m$ is introduced to normalize the value between -1 and 1. Modularity calibrates the quality of community partitions thus can be used as an objective measure to maximize.

We can define a modularity matrix B as $B_{ij} = A_{ij} - d_i d_j/2m$. Equivalently,

$$B = A - \mathbf{d}\mathbf{d}^T/2m. \tag{3.17}$$

where $\mathbf{d} \in R^{n \times 1}$ is a vector of each node's degree. Let $S \in \{0, 1\}^{n \times k}$ be a community indicator matrix with $S_{i\ell} = 1$ if node i belongs to community C_ℓ, and s_ℓ the ℓ-th column of S. Modularity can be reformulated as

$$Q = \frac{1}{2m} \sum_{\ell=1}^{k} \mathbf{s}_\ell B \mathbf{s}_\ell = \frac{1}{2m} Tr(S^T B S). \tag{3.18}$$

With a spectral relaxation to allow S to be continuous, the optimal S can be computed as the top k eigenvectors of the modularity matrix B (Newman, 2006b) with the maximum eigenvalues.

For example, the modularity matrix of the toy network is

$$B = \begin{bmatrix}
-0.32 & 0.79 & 0.68 & 0.57 & -0.43 & -0.43 & -0.43 & -0.32 & -0.11 \\
0.79 & -0.14 & 0.79 & -0.29 & -0.29 & -0.29 & -0.29 & -0.21 & -0.07 \\
0.68 & 0.79 & -0.32 & 0.57 & -0.43 & -0.43 & -0.43 & -0.32 & -0.11 \\
0.57 & -0.29 & 0.57 & -0.57 & 0.43 & 0.43 & -0.57 & -0.43 & -0.14 \\
-0.43 & -0.29 & -0.43 & 0.43 & -0.57 & 0.43 & 0.43 & 0.57 & -0.14 \\
-0.43 & -0.29 & -0.43 & 0.43 & 0.43 & -0.57 & 0.43 & 0.57 & -0.14 \\
-0.43 & -0.29 & -0.43 & -0.57 & 0.43 & 0.43 & -0.57 & 0.57 & 0.86 \\
-0.32 & -0.21 & -0.32 & -0.43 & 0.57 & 0.57 & 0.57 & -0.32 & -0.11 \\
-0.11 & -0.07 & -0.11 & -0.14 & -0.14 & -0.14 & 0.86 & -0.11 & -0.04
\end{bmatrix}$$

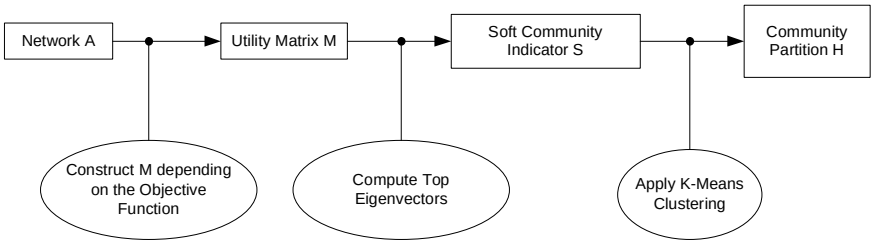


Figure 3.5: A Unified Process for Some Representative Community Detection Methods

Its top two maximum eigenvectors are

$$S = \begin{bmatrix} 0.44 & -0.00 \\ 0.38 & 0.23 \\ 0.44 & -0.00 \\ 0.17 & -0.48 \\ -0.29 & -0.32 \\ -0.29 & -0.32 \\ -0.38 & 0.34 \\ -0.34 & -0.08 \\ -0.14 & 0.63 \end{bmatrix}.$$

Note how the partition information is expressed by the first column of S.

3.3.6 A UNIFIED PROCESS

The above four representative community detection methods - latent space models, block model approximation, spectral clustering, and modularity maximization - can be unified in a process as in Figure 3.5. The process is composed of four components with three intermediate steps. Given a network, a utility matrix is constructed. Depending on the objective function, we can construct different utility matrices:

$$\text{Utility Matrix } M = \begin{cases} \widetilde{P} \text{ in Eq. (3.6)} & \text{(latent space models)} \\ A \text{ in Eq. (3.9)} & \text{(block model approximation)} \\ \widetilde{L} \text{ in Eq. (3.14)} & \text{(spectral clustering)} \\ B \text{ in Eq. (3.17)} & \text{(modularity maximization)} \end{cases} \tag{3.19}$$

After obtaining the utility matrix, we obtain the *soft community indicator* S that consists of the top eigenvectors with the largest (or smallest subject to formulation) eigenvalues. The selected eigenvectors capture the prominent interaction patterns, representing approximate community partitions. This step can also be considered as a de-noising process since we only keep those top eigenvectors

that are indicative of community structures. To recover the discrete partition H, a k-means clustering algorithm is applied. Note that all the aforementioned community detection methods differ subtly by constructing different utility matrices.

The community detection methods presented above, except the latent space model, are normally applicable to medium-size networks (say, 100,000 nodes). The latent space model requires an input of a proximity matrix of the geodesic distance of any pair of nodes, which costs $O(n^3)$ to compute the pairwise geodesic distances. Moreover, the utility matrix of the latent space model is neither sparse nor structured, incurring $O(n^3)$ time to compute its eigenvectors. This high computational cost hinders its application to real-world large-scale networks. By contrast, block models, spectral clustering and modularity maximization are typically much faster[2].

3.4 HIERARCHY-CENTRIC COMMUNITY DETECTION

Another line of community detection research is to build a hierarchical structure of communities based on network topology. This facilitates the examination of communities at different granularity. There are mainly two types of hierarchical clustering: divisive, and agglomerative.

3.4.1 DIVISIVE HIERARCHICAL CLUSTERING

Divisive clustering first partitions the nodes into several disjoint sets. Then each set is further divided into smaller ones until each set contains only a small number of (say, only one) actors. The key here is how to split a network into several parts. Some partition methods such as block models, spectral clustering, and latent space models can be applied recursively to divide a community into smaller sets.

One particular divisive clustering algorithm receiving much attention is to recursively remove the "weakest" tie in a network until the network is separated into two or more components. The general principle is as follows:

- At each iteration, find out the edge with least strength. This kind of edge is most likely to be a tie connecting two communities.
- Remove the edge and then update the strength of links.
- Once a network is decomposed into two connected components, each component is considered a community. The iterative process above can be applied to each community to find subcommunities.

Newman and Girvan (2004) proposes to find the weak ties based on *edge betweenness*. Edge betweenness is highly related to the betweenness centrality discussed in Section 2.1. Edge betweenness is defined to be the number of shortest paths that pass along one edge (Brandes, 2001). If

[2]The utility matrix of modularity maximization is dense, but it is a sparse matrix plus a low rank update as in Eq. (3.17). This structure can be exploited for fast eigenvector computation (Newman, 2006b; Tang et al., 2009).

Table 3.3: Edge Betweenness

	1	2	3	4	5	6	7	8	9
1	0	4	1	9	0	0	0	0	0
2	4	0	4	0	0	0	0	0	0
3	1	4	0	9	0	0	0	0	0
4	9	0	9	0	10	10	0	0	0
5	0	0	0	10	0	1	6	3	0
6	0	0	0	10	1	0	6	3	0
7	0	0	0	0	6	6	0	2	8
8	0	0	0	0	3	3	2	0	0
9	0	0	0	0	0	0	8	0	0

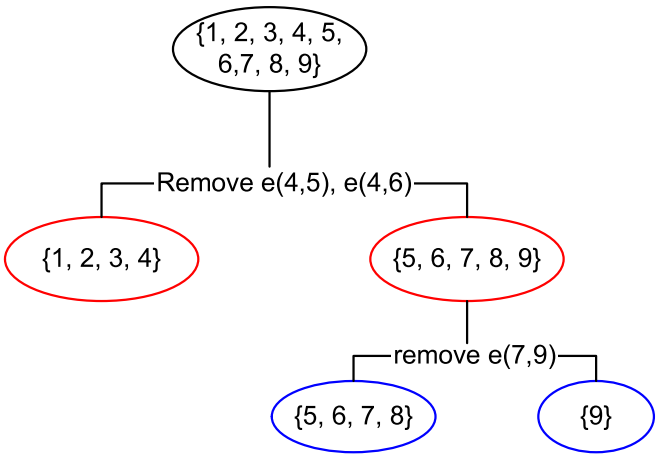


Figure 3.6: The Process of the Newman-Girvan Algorithm Applied to the Toy Network

two communities are joined by only a few cross-group edges, then all paths through the network from nodes in one community to the other community have to pass along one of these edges. Edge betweenness is a measure to count how many shortest paths between pair of nodes pass along the edge, and this number is expected to be large for those between-group edges. The Newman-Girvan algorithm suggests progressively removing edges with the highest betweenness. It will gradually disconnect the network, naturally leading to a hierarchical structure.

The edge betweenness of the network in Figure 1.1 is given in Table 3.3. For instance, the betweenness of $e(1, 2)$ is 4. Since all shortest paths from node 2 to any node in $\{4, 5, 6, 7, 8, 9\}$ has either to pass $e(1, 2)$ or $e(1, 3)$, leading to a weight of $6 \times 1/2 = 3$ for $e(1, 2)$. Meanwhile, $e(1, 2)$ is the shortest path between nodes 1 and 2. Hence, the betweenness of $e(1, 2)$ is $3 + 1 = 4$. An algorithm to compute the edge betweenness is included in Appendix B.

As seen in the table, both edges $e(4, 5)$ and $e(4, 6)$ have highest edge betweenness. Suppose we randomly remove one (say $e(4, 5)$). Then in the resultant network, the edge with the highest betweenness is $e(4, 6)$ (with betweenness being 20). After removing the edge $e(4, 6)$, the network is decomposed into two communities. At this point, $e(7, 9)$ becomes the edge with the highest betweenness. Its removal results in two new communities $\{5, 6, 7, 8\}$ and $\{9\}$. Then a similar procedure can be applied to each community to further divide them into smaller ones. The overall process of the first few steps is shown in Figure 3.6.

However, this divisive hierarchical clustering based on edge betweenness presses hard for computation. As we have discussed in Section 2.1, betweenness for nodes or edges takes $O(nm)$ time (Brandes, 2001). Moreover, each removal of an edge will lead to the recomputation of betweenness for all edges within the same connected component. Its high computational cost hinders its application to large-scale networks.

3.4.2 AGGLOMERATIVE HIERARCHICAL CLUSTERING

Agglomerative clustering begins with base communities and merges them successively into larger communities following certain criterion. One such criterion is modularity (Clauset et al., 2004). Two communities are merged if doing so results in the largest increase of overall modularity. We can start from treating each node as a separate base community and merge communities. The merge continues until no merge can be found to improve the modularity. Figure 3.7 shows the resultant dendrogram based on agglomerative hierarchical clustering applied to the network in Figure 1.1. Nodes 7 and 9 are merged first, and then 1 and 2, and so on. Finally, we obtain two communities at the top $\{1, 2, 3, 4\}$ and $\{5, 6, 7, 8, 9\}$.

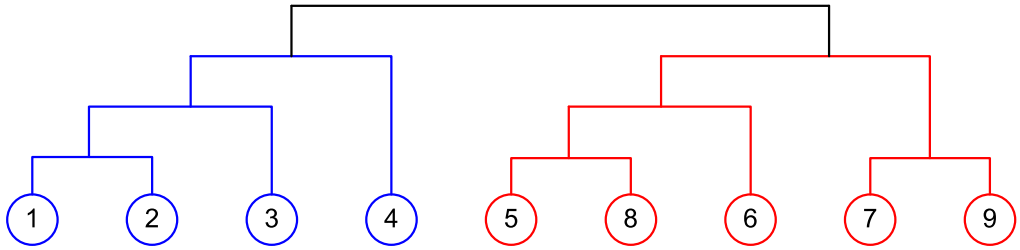


Figure 3.7: Dendrogram according to Agglomerative Clustering based on Modularity

It is noticed that this algorithm incurs many imbalanced merges (i.e., a large community merges with a tiny community, such as the merge of node 4 with $\{1, 2, 3\}$), resulting in a high computational cost (Wakita and Tsurumi, 2007). Hence, the merge criterion is modified accordingly by considering the size of communities. In the new scheme, communities of comparable sizes are joined first, leading to a more balanced hierarchical structure of communities and to improved efficiency. A state-of-the-art for such hierarchical clustering is the Louvain method (Blondel et al., 2008). It starts from each node as a base community and determines which of its neighbor should be

merged to based on the modularity criterion. After the first scan, some base communities are merged into one. Then, each community's degree and connection information is aggregated and treated as a single node for further merge. This multi-level approach is extremely efficient to handle large-scale networks.

So far, we have discussed some general ideas and different community detection approaches. We can hardly conclude which one is the best. It highly depends on the task at hand and available network data. Other things being equal, we need to resort to evaluation for comparing different community detection methods.

3.5 COMMUNITY EVALUATION

Part of the reason that there are so many assorted definitions and methods, is that there is no clear ground truth information about a community structure in a real world network. Therefore, different community detection methods are developed from various applications of specific needs. We now depict strategies commonly adopted to evaluate identified communities in order to facilitate the comparison of different community detection methods. Depending on available network information, one can take different strategies for comparison:

- Groups with self-consistent definitions. Some groups like cliques, k-cliques, k-clubs, k-plexes and k-cores can be examined immediately once a community is identified. We can simply check whether the extracted communities satisfy the definition.

- Networks with ground truth. That is, the community membership for each actor is known. This is an ideal case. This scenario hardly presents itself in real-world large-scale networks. It usually occurs for evaluation on synthetic networks generated based on predefined community structures (e.g., (Tang et al., 2008)), or some well-studied tiny networks like Zachary's karate club with 34 members (Newman, 2006b). To compare the ground truth with identified community structures, visualization can be intuitive and straightforward (Newman, 2006b). If the number of communities is small (say 2 or 3 communities), it is easy to determine a one-to-one mapping between the identified communities and the ground truth. So conventional classification measures (Tan et al., 2005) such as accuracy, F1-measure can be also used. In Figure 3.8, e.g., one can say one node 2 is wrongly assigned.

  However, when there are many communities, it may not be clear what a correct mapping of communities from the ground truth to a clustering result. In Figure 3.8, the ground truth has two communities whereas the clustering result consists of three. Both communities {1, 3} and {2} map to the community {1, 2, 3} in the ground truth. Hence, some measure that can average all the possible mappings should be considered. Normalized mutual information (NMI) is a commonly used one (Strehl and Ghosh, 2003).

  Before we introduce NMI, we briefly review some information-theoretic concepts. In information theory, the information contained in a distribution is called *entropy*, which is defined

Figure 3.8: Comparing Ground Truth with Clustering Result. Each number denotes a node, and each circle or block denotes a community.

below:

$$H(X) = -\sum_{x \in X} p(x) \log p(x). \tag{3.20}$$

Mutual information calibrates the shared information between two distributions:

$$I(X; Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log \left(\frac{p(x, y)}{p_1(x) p_2(y)} \right). \tag{3.21}$$

Since $I(X; Y) \leq H(X)$ and $I(X; Y) \leq H(Y)$, a *normalized mutual information* (NMI) between two variables X and Y is

$$NMI(X; Y) = \frac{I(X; Y)}{\sqrt{H(X) H(Y)}}. \tag{3.22}$$

We can consider a partition as a probability distribution of one node falling into one community. Let π^a, π^b denote two different partitions of communities. $n_{h,\ell}$, n_h^a, n_ℓ^b are, respectively, the number of actors simultaneously belonging to the h-th community of π^a and ℓ-th community of π^b, the number of actors in the h-th community of partition π^a, and the number of actors in the ℓ-th community of partition π^b. Thus,

$$\begin{aligned} H(\pi^a) &= \sum_{h}^{k^{(a)}} \frac{n_h^a}{n} \log \frac{n_h^a}{n}, \\ H(\pi^b) &= \sum_{\ell}^{k^{(b)}} \frac{n_\ell^b}{n} \log \frac{n_\ell^b}{n}, \\ I(\pi^a; \pi^b) &= \sum_{h} \sum_{\ell} \frac{n_{h,\ell}}{n} \log \left(\frac{\frac{n_{h,\ell}}{n}}{\frac{n_h^a}{n} \frac{n_\ell^b}{n}} \right). \end{aligned}$$

$n = 6$
$k^{(a)} = 2$
$k^{(b)} = 3$

	n_h^a
h=1	3
h=2	3

	n_l^b
l=1	2
l=2	1
l=3	3

$n_{h,l}$	l=1	l=2	l=3
h=1	2	1	0
h=2	0	0	3

Figure 3.9: Computation of NMI to compare two clusterings in Figure 3.8.

In the formula, $\frac{n_{h,\ell}}{n}$, in essence, estimates the probability of the mapping from community h in $\pi^{(a)}$ to community ℓ in $\pi^{(b)}$. Consequently,

$$NMI(\pi^a; \pi^b) = \frac{\sum_{h=1}^{k^{(a)}} \sum_{\ell=1}^{k^{(b)}} n_{h,\ell} \log \left(\frac{n \cdot n_{h,l}}{n_h^{(a)} \cdot n_\ell^{(b)}} \right)}{\sqrt{\left(\sum_{h=1}^{k^{(a)}} n_h^{(a)} \log \frac{n_h^a}{n} \right) \left(\sum_{\ell=1}^{k^{(b)}} n_\ell^{(b)} \log \frac{n_\ell^b}{n} \right)}}. \tag{3.23}$$

NMI is a measure between 0 and 1. It equals 1 when π^a and π^b are the same.

As for the example in Figure 3.8, the partitions can be rewritten in another form as follows:

$$\begin{aligned} \pi^a &= [1, 1, 1, 2, 2, 2] \quad \text{(ground truth)} \\ \pi^b &= [1, 2, 1, 3, 3, 3] \quad \text{(clustering result)} \end{aligned}$$

The network has 6 nodes, with each assigned to one community. Here, the numbers are the community ids of each clustering. The corresponding quantity of each term in Eq. (3.23) is listed in Figure 3.9. The resultant NMI following Eq. (3.23) is 0.83.

Another way is to consider all the possible pairs of nodes and check whether they reside in the same community. It is considered an error if two nodes of the same community are assigned to different communities, or two nodes of different communities are assigned to the same community. Let $C(v_i)$ denote the community of node v_i. We can construct a contingency table below: a, b, c and d are frequencies of each case. For instance, a is the frequency that

		Ground Truth	
		$C(v_i) = C(v_j)$	$C(v_i) \neq C(v_j)$
Clustering Result	$C(v_i) = C(v_j)$	a	b
	$C(v_i) \neq C(v_j)$	c	d

two nodes are assigned into the same community in the ground truth as well in the clustering result. It is noticed that the total sum of frequencies is the number of all possible pairs of

nodes in a network, i.e., $a + b + c + d = n(n-1)/2$. Based on the frequencies, the accuracy of clustering can be computed as

$$accuracy = \frac{a+d}{a+b+c+d} = \frac{a+d}{n(n-1)/2}.$$

Take Figure 3.8 as an example. We have $a = 4$. Specifically, {1, 3},{4, 5}, {4, 6}, {5, 6} are assigned into the same community in the ground truth and clustering result. Any pair between {1, 2, 3} and {4, 5, 6} are being assigned to different communities, thus $d = 9$. Consequently, the accuracy of the clustering result is $(4 + 9)/(6 \times 5/2) = 13/15$.

- Networks with semantics. Some networks come with semantic or attribute information of nodes and connections. In this case, the identified communities can be verified by human subjects to check whether it is consistent with the semantics, for instance, whether the community identified in the Web is coherent to a shared topic (Flake et al., 2000; Clauset et al., 2004), and whether the clustering of coauthorship network captures the research interests of individuals. This evaluation approach is applicable when the community is reasonably small. Otherwise, selecting the top-ranking actors as representatives of a community is a commonly used approach. Since this approach is qualitative, it can hardly be applied to all communities in a large network, but it is quite helpful for understanding and interpretation of community patterns. For example, tag clouds of representative nodes in two communities in blogosphere are shown in Figure 3.10 (Tang, 2010). Though these two communities are extracted based on network topology, they both capture certain semantic meanings. The first community is about *animals*, and the other one is about *health*.

Figure 3.10: Tag Clouds of Extracted Communities based on (Tang, 2010)

- Networks without ground truth or semantic information. This is the most common situation, yet it requires objective evaluation most. Normally, one resorts to some quantitative measure for network validation. That is, the quality measure Q is a function of a partition π and a network A. We can use a similar procedure as cross validation in classification for validation. It extracts communities from a (training) network and then compares them with those of the same network (e.g., constructed from a different date) or another related network based on a different type of interaction.

  In order to quantify the quality of extracted community structure, a common measure being used is modularity (Newman, 2006a). Once we have a network partition, we can compute its modularity with respect to one network. The method with higher modularity wins. Another comparable approach is to use the identified community as a base for link prediction, i.e., two actors are connected if they belong to the same community. Then, the predicted network is compared with the true network, and the deviation is used to calibrate the community structure. Since social media networks demonstrate strong community effect, a better community structure should predict the connections between actors more accurately. This is basically checking how far the true network deviates from a block model based on the identified communities.

Community detection is still an active and evolving field. We present some widely used community detection and evaluation strategies. In (Fortunato, 2010), one can find a comprehensive survey. Social media, however, often presents more than just a single friendship network. It might involve heterogeneous types of entities and interactions. In the next chapter, we will discuss further how to integrate different kinds of interaction information together to find robust communities in social media.

Bibliography

J. Abello, M. G. C. Resende, and S. Sudarsky. Massive quasi-clique detection. In *LATIN*, pages 598–612, 2002. DOI: 10.1007/3-540-45995-2_51 34, 35

N. Agarwal, H. Liu, L. Tang, and P. S. Yu. Identifying the influential bloggers in a community. In *WSDM '08: Proceedings of the international conference on Web search and web data mining*, pages 207–218, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-927-9. DOI: 10.1145/1341531.1341559 8

A. Anagnostopoulos, R. Kumar, and M. Mahdian. Influence and correlation in social networks. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 7–15, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-193-4. DOI: 10.1145/1401890.1401897 27, 28, 29

R. Andersen and K. J. Lang. Communities from seed sets. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 223–232, New York, NY, USA, 2006. ACM. ISBN 1-59593-323-9. DOI: 10.1145/1135777.1135814 9

S. Aral, L. Muchnik, and A. Sundararajan. Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. *Proceedings of the National Academy of Sciences*, 106(51):21544–21549, 2009. URL `http://www.pnas.org/content/106/51/21544.full`. DOI: 10.1073/pnas.0908800106 29

S. Asur, S. Parthasarathy, and D. Ucar. An event-based framework for characterizing the evolutionary behavior of interaction graphs. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 913–921, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-609-7. DOI: 10.1145/1281192.1281290 9, 76, 78

L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks: membership, growth, and evolution. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 44–54, New York, NY, USA, 2006. ACM. ISBN 1-59593-339-5. DOI: 10.1145/1150402.1150412 9, 76

L. Backstrom, C. Dwork, and J. Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 181–190, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-654-7. DOI: 10.1145/1242572.1242598 11

A.-L. Barabási and R. Albert. Emergence of Scaling in Random Networks. *Science*, 286(5439): 509–512, 1999. URL http://www.sciencemag.org/cgi/content/abstract/286/5439/509. DOI: 10.1126/science.286.5439.509 7, 8

L. Becchetti, P. Boldi, C. Castillo, and A. Gionis. Efficient semi-streaming algorithms for local triangle counting in massive graphs. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 16–24, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-193-4. DOI: 10.1145/1401890.1401898 8

V. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008:P10008, 2008. DOI: 10.1088/1742-5468/2008/10/P10008 48

I. Borg and P. Groenen. *Modern Multidimensional Scaling: theory and applications*. Springer, 2005. 37

U. Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25(2): 163–177, 2001. DOI: 10.1080/0022250X.2001.9990249 16, 18, 46, 48, 97

J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *UAI '98: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, 1998. 10

D. Chakrabarti and C. Faloutsos. Graph mining: Laws, generators, and algorithms. *ACM Comput. Surv.*, 38(1):2, 2006. ISSN 0360-0300. 7, 8

D. Chakrabarti, R. Kumar, and A. Tomkins. Evolutionary clustering. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 554–560, New York, NY, USA, 2006. ACM. ISBN 1-59593-339-5. DOI: 10.1145/1150402.1150467 79

W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 199–208, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-495-9. DOI: 10.1145/1557019.1557047 26

W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *KDD '10: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1029–1038, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0055-1. DOI: 10.1145/1835804.1835934 26

Y. Chi, X. Song, D. Zhou, K. Hino, and B. L. Tseng. Evolutionary spectral clustering by incorporating temporal smoothness. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 153–162, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-609-7. DOI: 10.1145/1281192.1281212 79, 80

N. A. Christakis and J. H. Fowler. The spread of obesity in a large social network over 32 years. *The New England Journal of Medicine*, 357(4):370–379, 2007. DOI: 10.1056/NEJMsa066082 29

A. Clauset, M. Newman, and C. Moore. Finding community structure in very large networks. *Arxiv preprint cond-mat/0408187*, 2004. DOI: 10.1103/PhysRevE.70.066111 48, 52

D. Crandall, D. Cosley, D. Huttenlocher, J. Kleinberg, and S. Suri. Feedback effects between similarity and social influence in online communities. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 160–168, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-193-4. DOI: 10.1145/1401890.1401914 27, 29

P. Desikan and J. Srivastava. I/o efficient computation of first order markov measures for large and evolving graphs. In *Proceedings of the Tenth Workshop on Web Mining and Web Usage Analysis (WebKDD)*, 2008. 8

I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–274, New York, NY, USA, 2001. ACM. ISBN 1-58113-391-X. DOI: 10.1145/502512.502550 69

J. Diesner, T. L. Frantz, and K. M. Carley. Communication networks from the enron email corpus "it's always about the people. enron is no different". *Comput. Math. Organ. Theory*, 11(3):201–228, 2005. ISSN 1381-298X. DOI: 10.1007/s10588-005-5377-0 6

P. Domingos and M. Richardson. Mining the network value of customers. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 57–66, New York, NY, USA, 2001. ACM. ISBN 1-58113-391-X. DOI: 10.1145/502512.502525 24

Y. Dourisboure, F. Geraci, and M. Pellegrini. Extraction and classification of dense communities in the web. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 461–470, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-654-7. DOI: 10.1145/1242572.1242635 9

D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010. 19, 27

K. El-Arini, G. Veda, D. Shahaf, and C. Guestrin. Turning down the noise in the blogosphere. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 289–298, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-495-9. DOI: 10.1145/1557019.1557056 24

G. W. Flake, S. Lawrence, and C. L. Giles. Efficient identification of web communities. In *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–160, New York, NY, USA, 2000. ACM. ISBN 1-58113-233-6. DOI: 10.1145/347090.347121 9, 52

R. W. Floyd. Algorithm 97: Shortest path. *Commun. ACM*, 5(6):345, 1962. ISSN 0001-0782. DOI: 10.1145/367766.368168 18

S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75 – 174, 2010. ISSN 0370-1573. URL http://www.sciencedirect.com/science/article/B6TVP-4XPYXF1-1/2/99061fac6435db4343b2374d26e64ac1. DOI: 10.1016/j.physrep.2009.11.002 53

D. Gibson, R. Kumar, and A. Tomkins. Discovering large dense subgraphs in massive graphs. In *VLDB '05: Proceedings of the 31st international conference on Very large data bases*, pages 721–732. VLDB Endowment, 2005. ISBN 1-59593-154-6. 9, 36

E. Gilbert and K. Karahalios. Predicting tie strength with social media. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 211–220, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-246-7. DOI: 10.1145/1518701.1518736 18, 20

G. H. Golub and C. F. Van Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996. ISBN 0-8018-5414-8. 18, 71

L. Goodman. Snowball sampling. *The Annals of Mathematical Statistics*, 32(1):148–170, 1961. DOI: 10.1214/aoms/1177705148 94

A. Goyal, F. Bonchi, and L. V. Lakshmanan. Learning influence probabilities in social networks. In *WSDM '10: Proceedings of the third ACM international conference on Web search and data mining*, pages 241–250, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-889-6. DOI: 10.1145/1718487.1718518 21

M. Granovetter. The Strength of Weak Ties. *American Journal of Sociology*, 78(6):1360, 1973. DOI: 10.1086/225469 18

M. Granovetter. Threshold models of collective behavior. *American journal of sociology*, 83(6):1420, 1978. DOI: 10.1086/226707 22

D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 491–501, New York, NY, USA, 2004. ACM. ISBN 1-58113-844-X. DOI: 10.1145/988672.988739 23

M. S. Handcock, A. E. Raftery, and J. M. Tantrum. Model-based clustering for social networks. *Journal of The Royal Statistical Society Series A*, 127(2):301–354, 2007. URL http://ideas.repec.org/a/bla/jorssa/v170y2007i2p301-354.html. 37

R. Hanneman and M. Riddle. *Introduction to Social Network Methods*. http://faculty.ucr.edu/ hanneman/, 2005. 36

M. Hechter. *Principles of Group Solidarity*. University of California Press, 1988. 8

S. Hill, F. Provost, and C. Volinsky. Network-based marketing: Identifying likely adopters via consumer networks. *Statistical Science*, 21(2):256–276, 2006. DOI: 10.1214/088342306000000222 29

P. D. Hoff, A. E. Raftery, and M. S. Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, 2002. DOI: 10.1198/016214502388618906 37

J. Hopcroft and R. Tarjan. Algorithm 447: efficient algorithms for graph manipulation. *Commun. ACM*, 16(6):372–378, 1973. ISSN 0001-0782. DOI: 10.1145/362248.362272 33

J. Hopcroft, O. Khan, B. Kulis, and B. Selman. Natural communities in large linked networks. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 541–546, New York, NY, USA, 2003. ACM. ISBN 1-58113-737-0. DOI: 10.1145/956750.956816 36, 78

J. Hopcroft, O. Khan, B. Kulis, and B. Selman. Tracking evolving communities in large linked networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101 (Suppl 1):5249–5253, 2004. URL `http://www.pnas.org/content/101/suppl.1/5249.abstract`. DOI: 10.1073/pnas.0307750100 78, 79

B. A. Huberman, D. M. Romero, and F. Wu. Social networks that matter: Twitter under the microscope. *First Monday*, 14(1), 2009. 20

A. Java, A. Joshi, and T. Finin. Detecting Commmunities via Simultaneous Clustering of Graphs and Folksonomies. In *Proceedings of the Tenth Workshop on Web Mining and Web Usage Analysis (WebKDD)*. ACM, August 2008. (Held in conjunction with The 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2008)). 9

D. B. Johnson. Efficient algorithms for shortest paths in sparse networks. *J. ACM*, 24(1):1–13, 1977. ISSN 0004-5411. DOI: 10.1145/321992.321993 18

D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM New York, NY, USA, 2003. DOI: 10.1145/956750.956769 8, 21, 23, 24, 25, 26

M.-S. Kim and J. Han. A particle-and-density based evolutionary clustering method for dynamic networks. *Proc. VLDB Endow.*, 2(1):622–633, 2009. ISSN 2150-8097. 82

P. Kolari, T. Finin, and A. Joshi. SVMs for the blogosphere: Blog identification and splog detection. In *AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs*, 2006a. 11

P. Kolari, A. Java, and T. Finin. Characterizing the splogosphere. In *Proceedings of the 3rd Annual Workshop on Weblogging Ecosystem: Aggregation, Analysis and Dynamics, 15th World Wid Web Conference*. Citeseer, 2006b. 11

G. Kossinets and D. J. Watts. Empirical analysis of an evolving social network. *Science*, 311(5757): 88–90, 2006. URL http://www.sciencemag.org/cgi/content/abstract/311/5757/88. DOI: 10.1126/science.1116869 76

G. Kossinets, J. Kleinberg, and D. Watts. The structure of information pathways in a social communication network. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 435–443, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-193-4. DOI: 10.1145/1401890.1401945 21

R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the web for emerging cyber-communities. *Comput. Netw.*, 31(11-16):1481–1493, 1999. ISSN 1389-1286. DOI: 10.1016/S1389-1286(99)00040-7 32

R. Kumar, J. Novak, P. Raghavan, and A. Tomkins. On the bursty evolution of blogspace. *World Wide Web*, 8(2):159–178, 2005. ISSN 1386-145X. DOI: 10.1007/s11280-004-4872-4 75, 76, 78

R. Kumar, J. Novak, and A. Tomkins. Structure and evolution of online social networks. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 611–617, New York, NY, USA, 2006. ACM. ISBN 1-59593-339-5. DOI: 10.1145/1150402.1150476 33, 76

T. La Fond and J. Neville. Randomization tests for distinguishing social influence and homophily effects. In *WWW '10: Proceedings of the 19th international conference on World wide web*, pages 601–610, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-799-8. DOI: 10.1145/1772690.1772752 27, 29

J. Leskovec and C. Faloutsos. Sampling from large graphs. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 631–636, New York, NY, USA, 2006. ACM. ISBN 1-59593-339-5. DOI: 10.1145/1150402.1150479 8

J. Leskovec and E. Horvitz. Planetary-scale views on a large instant-messaging network. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 915–924, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-085-2. DOI: 10.1145/1367497.1367620 5, 6

J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Trans. Knowl. Discov. Data*, 1(1):2, 2007a. ISSN 1556-4681. DOI: 10.1145/1217299.1217301 76

J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 420–429, New York, NY, USA, 2007b. ACM. ISBN 978-1-59593-609-7. DOI: 10.1145/1281192.1281239 8, 24, 26

J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins. Microscopic evolution of social networks. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 462–470, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-193-4. DOI: 10.1145/1401890.1401948 76

D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *J. Am. Soc. Inf. Sci. Technol.*, 58(7):1019–1031, 2007. ISSN 1532-2882. DOI: 10.1002/asi.20591 10, 11

Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng. Analyzing communities and their evolutions in dynamic social networks. *ACM Trans. Knowl. Discov. Data*, 3(2):1–31, 2009. ISSN 1556-4681. DOI: 10.1145/1514888.1514891 82

B. Long, Z. M. Zhang, X. Wú, and P. S. Yu. Spectral clustering for multi-type relational data. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 585–592, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2. DOI: 10.1145/1143844.1143918 72

B. Long, P. S. Yu, and Z. M. Zhang. A general model for multiple view unsupervised learning. In *SDM '08: Proceedings of SIAM International Conference on Data Mining*, pages 822–833, 2008. 64

U. v. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007. ISSN 0960-3174. DOI: 10.1007/s11222-007-9033-z 41, 42

S. A. Macskassy and F. Provost. Classification in networked data: A toolkit and a univariate case study. *J. Mach. Learn. Res.*, 8:935–983, 2007. ISSN 1533-7928. 85, 86

S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1:24–45, 2004. ISSN 1545-5963. DOI: 10.1109/TCBB.2004.2 69

B. McClosky and I. V. Hicks. Detecting cohesive groups. http://www.caam.rice.edu/ ivhicks/CokplexAlgorithmPaper.pdf, 2009. 34

M. McPherson, L. Smith-Lovin, and J. M. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27:415–444, 2001. DOI: 10.1146/annurev.soc.27.1.415 27

F. Menczer. Web crawling. In B. Liu, editor, *Web Data Mining*, chapter 8, pages 273–322. Springer, 2006. 94

A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 29–42, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-908-1. DOI: 10.1145/1298306.1298311 6

T. M. Mitchell. Mining Our Reality. *Science*, 326(5960):1644–1645, 2009. URL http://www.sciencemag.org. DOI: 10.1126/science.1174459 11

A. A. Nanavati, S. Gurumurthy, G. Das, D. Chakraborty, K. Dasgupta, S. Mukherjea, and A. Joshi. On the structural properties of massive telecom call graphs: findings and implications. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 435–444, New York, NY, USA, 2006. ACM. ISBN 1-59593-433-2. DOI: 10.1145/1183614.1183678 6

G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of approximations for maximizing submodular set functions-I. *Mathematical Programming*, 14(1):265–294, 1978. DOI: 10.1007/BF01588971 26

M. Newman. Modularity and community structure in networks. *PNAS*, 103(23):8577–8582, 2006a. DOI: 10.1073/pnas.0601602103 43, 53

M. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 74(3), 2006b. URL http://dx.doi.org/10.1103/PhysRevE.74.036104. DOI: 10.1103/PhysRevE.74.036104 44, 46, 49

M. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004. URL http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0308217. DOI: 10.1103/PhysRevE.69.026113 9, 46, 97

M. Newman, A.-L. Barabasi, and D. J. Watts, editors. *The Structure and Dynamics of Networks*. 2006. 6

J. Onnela, J. Saramäki, J. Hyvönen, G. Szabó, D. Lazer, K. Kaski, J. Kertész, and A. Barabási. Structure and tie strengths in mobile communication networks. *PNAS*, 104(18):7332–7336, 2007. DOI: 10.1073/pnas.0610245104 19

L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. URL http://ilpubs.stanford.edu:8090/422/. Previous number = SIDL-WP-1999-0120. 8, 16

G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814–818, 2005. DOI: 10.1038/nature03607 33, 78

G. Palla, A.-L. Barabasi, and T. Vicsek. Quantifying social group evolution. *Nature*, 446(7136): 664–667, April 2007. DOI: 10.1038/nature05670 9, 76, 77, 78

J. Pearl. *Causality: models, reasoning, and inference*. Cambridge University Press, 2000. 28

M. Ramezani, J. Sandvig, R. Bhaumik, R. Burke, and B. Mobasher. Exploring the impact of profile injection attacks in social tagging systems. In *Proceedings of Workshop on Web Mining and Web Usage Analysis*, 2008. 11

M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 61–70, New York, NY, USA, 2002. ACM. ISBN 1-58113-567-X. DOI: 10.1145/775047.775057 8, 24

K. Saito, M. Kimura, K. Ohara, and H. Motoda. Behavioral analyses of information diffusion models by observed data of social network. In *SBP*, pages 149–158, 2010. DOI: 10.1007/978-3-642-12079-4_20 21

P. Sarkar and A. W. Moore. Dynamic social network analysis using latent space models. *SIGKDD Explor. Newsl.*, 7(2):31–40, 2005. ISSN 1931-0145. DOI: 10.1145/1117454.1117459 37, 82

T. C. Schelling. Dynamic models of segregation. *Journal of Mathematical Sociology*, 1:143–186, 1971. DOI: 10.1080/0022250X.1971.9989794 22

P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93, 2008. 85

C. Shirky. *Here Comes Everybody: The Power of Organizing without Organizations*. The Penguin Press, 2008. 1

P. Singla and M. Richardson. Yes, there is a correlation: - from social networks to personal behavior on the web. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 655–664, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-085-2. DOI: 10.1145/1367497.1367586 26

A. Strehl and J. Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3:583–617, 2003. ISSN 1533-7928. DOI: 10.1162/153244303321897735 49, 65

J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu. Graphscope: parameter-free mining of large time-evolving graphs. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 687–696, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-609-7. DOI: 10.1145/1281192.1281266 82, 83

Y. Sun, Y. Yu, and J. Han. Ranking-based clustering of heterogeneous information networks with star network schema. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 797–806, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-495-9. DOI: 10.1145/1557019.1557107 56

P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison Wesley, 2005. 36, 37, 49, 85, 101

L. Tang. *Learning with Large-Scale Social Media Networks*. PhD thesis, Arizona State University, 2010. URL `http://www.public.asu.edu/~ltang9/thesis.pdf`. 52, 58

L. Tang and H. Liu. Scalable learning of collective behavior based on sparse social dimensions. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*, pages 1107–1116, New York, NY, USA, 2009a. ACM. ISBN 978-1-60558-512-3. DOI: 10.1145/1645953.1646094 91

L. Tang and H. Liu. Relational learning via latent social dimensions. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 817–826, New York, NY, USA, 2009b. ACM. ISBN 978-1-60558-495-9. DOI: 10.1145/1557019.1557109 57, 88

L. Tang and H. Liu. Toward predicting collective behavior via social dimension extraction. *IEEE Intelligent Systems*, 25:19–25, 2010a. ISSN 1541-1672. DOI: 10.1109/MIS.2010.36 89, 90

L. Tang and H. Liu. Graph mining applications to social network analysis. In C. Aggarwal and H. Wang, editors, *Managing and Mining Graph Data*, chapter 16, pages 487–513. Springer, 2010b. DOI: 10.1007/978-1-4419-6045-0_16 31

L. Tang, H. Liu, J. Zhang, and Z. Nazeri. Community evolution in dynamic multi-mode networks. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 677–685, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-193-4. DOI: 10.1145/1401890.1401972 9, 49, 72

L. Tang, X. Wang, and H. Liu. Uncovering groups via heterogeneous interaction analysis. In *ICDM '09: Proceedings of IEEE International Conference on Data Mining*, pages 503–512, 2009. 9, 46, 55, 64, 67

L. Tang, X. Wang, H. Liu, and L. Wang. A multi-resolution approach to learning with overlapping communities. In *Proceedings of Workshop on Social Media Analytics*, 2010. 92

L. Tang, H. Liu, and J. Zhang. Identifying evolving groups in dynamic multi-mode networks. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, forthcoming. 9, 56, 73

C. Tantipathananandh, T. Berger-Wolf, and D. Kempe. A framework for community identification in dynamic social networks. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 717–726, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-609-7. DOI: 10.1145/1281192.1281269 82

M. Thelwall. Bloggers under the london attacks:top information sources and topics. In *WWW:3rd annual workshop on webloging ecosystem: aggreation, analysis and dynamics*, 2006. 2

J. Travers and S. Milgram. An experimental study of the small world problem. *Sociometry*, 32(4): 425–443, 1969. DOI: 10.2307/2786545 5

K. Wakita and T. Tsurumi. Finding community structure in mega-scale social networks: [extended abstract]. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 1275–1276, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-654-7. DOI: 10.1145/1242572.1242805 48

S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994. 3, 5, 8, 13, 34, 68

D. J. Watts and P. S. Dodds. Influentials, networks, and public opinion formation. *Journal of Consumer Research*, 34(4):441–458, 2007. DOI: 10.1086/518527 26

D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998. DOI: 10.1038/30918 7, 8

R. Xiang, J. Neville, and M. Rogati. Modeling relationship strength in online social networks. In *WWW '10: Proceedings of the 19th international conference on World wide web*, pages 981–990, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-799-8. DOI: 10.1145/1772690.1772790 20

T. Yang, Y. Chi, S. Zhu, Y. Gao, and R. Jin. A bayesian approach toward finding communities and their evolutions in dynamic social networks. In *SDM '09: Proceedings of SIAM International Conference on Data Mining*, 2009. 82

H.-J. Zeng, Z. Chen, and W.-Y. Ma. A unified framework for clustering heterogeneous web objects. In *WISE '02: Proceedings of the 3rd International Conference on Web Information Systems Engineering*, pages 161–172, Washington, DC, USA, 2002. IEEE Computer Society. ISBN 0-7695-1766-8. DOI: 10.1109/WISE.2002.1181653 9

X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML '03: Proceedings of the Twentieth International Conference on Machine Learning*, pages 912–919, 2003. 87