Chapter 3

Social Network Analysis

3.1 Social Network

The relationships between social entities can be depicted in a social network. This network is represented as an undirected graph with a set of actors, such as individuals or organizations, as nodes and ties between them as edges. An example of this concept can be seen in a friendship network on Facebook where students from the same department of a college may form a group due to their shared location and academic background. In contrast, students with different backgrounds, education, or geography have a lower likelihood of forming networks. The homogeneity of the network can reflect the similarity between nodes in terms of parameters like location or education.

A *graph* is represented by $G = (V, E)$, where $V = \{v_1, v_2, ..., v_n\}$ is the set of n nodes and E is the set of edges. The nodes represent the network entities and edges represent the relationships between the entities. An edge $e_{ij} \in E$ between two nodes v_i and v_j is represented by a pair of the nodes (v_i, v_j). A graph G is called a *weighted graph* if a weight is assigned to each edge.

Example 3.1. An illustration of a small social network is shown in Figure 3.1. The nodes, labeled as a to j, are the entities, and the edges signify the relationships, such as friendship, between them. For instance, node b has a friendship relationship with nodes a, c, d, and e.

Clusters in a network are groups of nodes or vertices within the network that exhibit a higher degree of connectivity to each other compared to nodes outside of the group. Edges and vertices are not distributed uniformly, but rather in locally dense groups. Common interests or goals, friendship, or other similarities between actors are common reasons for the implicit or explicit formation of groups.

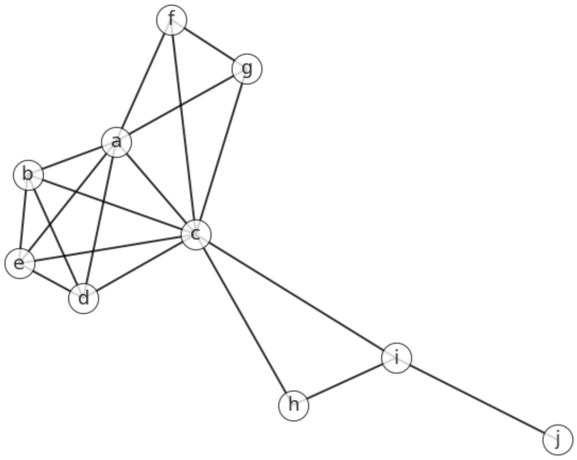


Fig. 3.1 A small social network.

3.2 Varieties of Social Networks

There are numerous instances of social networks beyond just networks of friends. Let's list some other examples of networks that display a local quality of relationships.

Telephone Networks

The example in Figure 3.2 illustrates a small telephone network. The nodes in the network represent individual phone numbers and if a call was made between two nodes during a specific time period, such as last month or "ever," an edge is formed between them. The edges can be weighted based on the number of calls made between the phones throughout the specified time period.

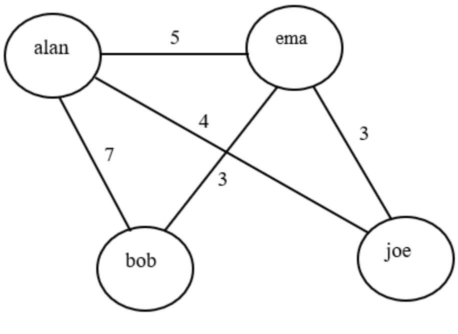


Fig. 3.2 A small telephone network.

In a telephone network, communities will arise from groups of individuals who communicate frequently, such as friends, members of a club, or people who work for the same company.

Email Networks

Individuals are represented by the nodes, which are email addresses. An edge indicates that at least one email was sent between the two addresses in at least one direction. Alternatively, if there were emails in both directions, we could only place an edge. We avoid seeing spammers as "friends" with all of their victims in this way. Another option is to categorize edges as strong or weak. Communication in both directions is represented by strong edges, and communication in one direction is represented by weak edges. The communities identified in email networks are formed by the same kind of groupings that we see in telephone networks. Individuals who text other people on their cell phones form a similar network.

Collaboration Networks

Individuals who have published research articles are depicted as nodes. Edges exist between two individuals who have co-authored one or more papers. The strength of these edges can be determined by the number of joint publications. Communities can be formed in this network based on the authors who focus on a particular subject matter.

Another way to look at the same data is as a graph with nodes that are publications. If two papers share at least one author, they are related by an edge. We now create communities, which are collections of writings on a single subject.

There are various additional types of data that can be used to create two networks in the same way. We may look at the people who edit Wikipedia pages as well as the articles that they edit, for example. If two editors have edited the same article, they are related. The communities are groups of editors who share a common interest in a particular topic. The creation of a network of articles that are connected if they were modified by the same person can be used to find communities of articles on similar topics.

This concept can be applied to the data used in collaborative filtering, where two networks are established, one for customers and one for items. The communities in these networks are determined by customers who buy similar items and by products that are purchased by the same customers.

Other Examples of Social Graphs

Many other processes, particularly those demonstrating locality, result in graphs that resemble social graphs. Information networks (documents, web graphs, patents), infrastructure networks (roads, airlines, water pipelines, power grids), biological networks (genes, proteins, food-webs of animals eating each other), and other sorts, such as product co-purchasing networks, are only a few examples (e.g., Groupon).

3.3 Social-Network Graph Drawing

A drawing of a graph is a pictorial representation of the vertices and edges of a graph. It is important to note that this visual representation should not be confused with the graph itself, as different layouts can represent the same graph. In abstract, what truly matters is identifying which pairs of nodes are connected by edges.

There are many different graph layout strategies available that can be used to visualise a graph. We discuss the force-directed placement strategy proposed by Fruchterman and Reingold [1991] in this dissertation.

Normally, graphs are depicted with their vertices as points in a plane and their edges as line segments or curves connecting those points. Fruchterman-Reingold concentrates on the most general class of graphs: general, undirected graphs, drawn with straight edges. In their paper, they introduced an algorithm that attempts to produce aesthetically–pleasing, two–dimensional pictures of graphs by doing simplified simulations of physical systems.

They are concerned with drawing general undirected graphs according to some generally–accepted aesthetic criteria:

- Evenly distribute the vertices in the frame.
- Minimize edge crossings.
- Make edge lengths uniform.
- Reflect inherent symmetry.
- Conform to the frame.

They have only two principles for graph drawing:

- Vertices that are neighbors should be drawn near each other.
- Vertices should not be drawn too close to each other.

How close vertices should be placed depends on how many there are and how much space is available. For more details please read Fruchterman and Reingold [1991].

3.4 Social-Network Graphs Clustering

A *cluster* is a set of nodes, and *clustering* is the process of grouping a set of nodes into subsets, where each subset represents a cluster. Consider the nodes: $S = x_1, x_2, ..., x_n$. Clustering divides this set into k subsets $(C_1, C_2, ..., C_k)$, each representing a cluster of similar nodes. That is, $S = C_1 \cup C_2 \cup ... \cup C_k$ and clustering $C = \{C_1, C_2, ..., C_k\}$.

A measure of similarity or dissimilarity is required to organize similar nodes into groups. The degree to which two nodes are alike is measured numerically by the similarity between them. As a result, dissimilarity is a numerical measure of how distinct the two nodes are. The higher the similarity, the lower the dissimilarity, and the lower the similarity, the higher the dissimilarity. When discussing dissimilarity, the term "distance" is commonly employed. The discussion of the various clustering techniques and their applications is relevant because the objective of clustering aligns with our aim of identifying groups of actors who are similar or close to each other.

3.4.1 Measures for Social-Network Graphs

The first step in applying typical clustering techniques to a social-network graph is defining a distance measure. Depending on what the weights on the graph's edges indicated, as in a telephone network (see Figure 3.2), these weights may be used to label a distance measure. However, when the edges are unweighted, as in a "friends" graph (see Figure 3.1), we are limited in our ability to establish one suitable distance.

Distance Measure

In social network clustering, distance measures are used to quantify the similarity or dissimilarity between nodes (individuals) within the network. These measures help identify clusters or communities of nodes that exhibit similar patterns of connections or interactions.

A distance $d(x,y)$ between two nodes x and y fulfils the following properties:

- *Non-negativity:* $d(x,y) \geq 0$ for all x and y
- *Identity of indiscernibles:* $d(x,y) = 0$ only if $x = y$
- *Symmetry:* $d(x,y) = d(y,x)$ for all x and y
- *Triangle inequality:* $d(x,z) \leq dist(x,y) + d(y,z)$ for all x, y and z

Distance measures that adhere to all of the specified properties are referred to as metrics. These properties can be beneficial for certain applications, for instance, if the triangle inequality property holds, clustering can be performed more efficiently. Here are some common distance measures used in social network clustering:

Euclidean distance

This measure calculates the straight-line distance between two nodes in a multi-dimensional space. In social networks, each dimension may represent a different attribute or feature of the nodes. Euclidean distance can be used to assess the dissimilarity between nodes based on their attribute values. The Euclidean distance between two nodes in n-dimensional space is defined by the following formula:

$d(P,Q) = \sqrt{\sum_{i=1}^{n} (P_i - Q_i)^2}$
$where\ P = \{p_1, p_2, ..., p_n\}$ and $Q = \{q_1, q_2, ..., q_n\}$.

Manhattan distance

The Manhattan distance between two nodes $P = \{p_1, p_2, ..., p_n\}$ and $Q = \{q_1, q_2, ..., q_n\}$ in n-dimensional space is the sum of the distances in each dimension.
$d(P,Q) = \sum_{i=1}^{n} |P_i - Q_i|$

Distance functions must satisfy amongst other the triangle inequality property. But sometimes the triangle inequality property is violated. The reason is that when there are three nodes connected by two edges, if there is an edge between nodes A and B and another edge between nodes B and C but no edge between nodes A and C, then the distance between nodes A and C is greater than the sum of the distances between nodes A and B and nodes B and C.

Similarity Measure

The similarity measure is a way of measuring how nodes or objects are related or being close to each other. Typically, the similarity values are expressed within the range of $[0,1]$. A similarity value closer to 1 indicates a stronger similarity between the objects, whereas a value closer to 0 indicates less similarity. The degree to which objects are alike is determined by a measure that assesses their similarities. As a result, calculating the similarity between all pairs of objects results in a quadratic similarity matrix.

Similarities, also have some well known properties.

1. $s(x,y) = 1$ (or maximum similarity) only if $x = y$

2. $s(x,y) = s(y,x)$ for all x and y (Symmetry)

where $s(x,y)$ is the similarity between nodes, x and y.

If we transform distance values into similarity values within the range of $[0,1]$ using the function $s(x,y) = e^{-d(x,y)}$, we obtain for property TI:

$$d(x,z) \leq d(x,y) + d(y,z) \xLeftrightarrow{\cdot(-1)} -d(x,z) \geq -d(x,y) - d(y,z) \xLeftrightarrow{(exp)}$$
$$e^{-d(x,z)} \geq e^{-d(x,y)-d(y,z)} \Leftrightarrow e^{-d(x,z)} \geq e^{-d(x,y)} \cdot e^{-d(y,z)}$$

which gives, $s(x,z) \geq s(x,y) \cdot s(y,z)$. And thus, $d(x,z) \leq d(x,y) + d(y,z) \Leftrightarrow s(x,z) \geq s(x,y) \cdot s(y,z)$ holds.

Cosine similarity

Cosine similarity is a technique used to determine the similarity between two documents or to rank documents in relation to a set of query words. Given vectors P and Q, the cosine similarity between these vectors is calculated as:

$sim(P,Q) = \frac{P \cdot Q}{|P||Q|}$

where $P \cdot Q = \sum_{i=1}^{n} P_i Q_i$ is the dot product of the vectors P and Q, and $|P|$ is the Euclidean norm of vector $P = \{p_1, p_2, ..., p_n\}$, which is defined as $\sqrt{p_1^2 + p_2^2 + ... + p_n^2}$, or the length of the vector. Similarly, $|Q|$ is the Euclidean norm of vector Q. The cosine similarity measure is calculated as the cosine of the angle between the two vectors. If the cosine value is equal to 0, it indicates that the vectors are orthogonal and have no match. The closer the cosine value is to 1, the lower the angle between the vectors and the better the match. Note that cosine similarity is referred to as a non-metric measure as it does not satisfy all the properties of a metric measure.

Jaccard similarity index

The Jaccard similarity index (also known as the Jaccard similarity coefficient) analyzes members from two sets to determine which are common and which are unique. It's a proportional measure of similarity between two sets of data, ranging from 0% to 100%. The larger the proportion, the closer the two groups are. Although it is simple to use, it is particularly sensitive to tiny sample sizes and can produce incorrect findings, especially when dealing with very small samples or data sets with missing observations.

The Jaccard similarity measures is defined as the cardinality of the intersection of sets divided by the cardinality of the union of the sample sets.

$J(P,Q) = \frac{|P \cap Q|}{|P \cup Q|}$

CQQL

CQQL, which stands for Commuting Quantum Query Language, is a language used for querying the similarity between two objects. It provides the ability to incorporate weighting into queries and formulate logic-based queries that include both Boolean and similarity conditions. The evaluation of a CQQL condition returns a similarity value within the interval of $[0,1]$ for the similarity condition, indicating the closeness of attribute values between two objects. A CQQL expression can be seen as a similarity measure when all atomic conditions are similarity measures between two objects and CQQL combines them by logic junctors. The boolean values $true$ for perfect match and $false$ otherwise are mapped to the score values 1 or 0, respectively, for the boolean condition. A detailed discussion can be found in Schmitt [2008] and Schmitt [2019].

3.4.2 Matrices that describe Graphs

To utilize the concept of matrix algebra in finding good partitions in a graph, it's necessary to have an understanding of three different matrices that represent aspects of the graph. Consider a set of n individuals, denoted as $P = \{p_1, p_2, ..., p_n\}$. The first should be familiar: the ***adjacency matrix*** $Adj = \{a_{ij}\}$ is the $n \times n$ matrix defined as

$$a_{ij} = \begin{cases} 1 & \text{if there is an edge } (p_i, p_j) \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

The adjacency matrix for the social network of Figure 3.1 appears below:

$$Adj = \begin{array}{c|cccccccccc|} & a & b & c & d & e & f & g & h & i & j \\ a & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ b & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ c & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ d & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ e & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ f & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ g & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ h & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ i & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ j & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}$$

The second matrix we need is the ***degree matrix*** for a graph. This example matrix has nonzero entries only on the diagonal. The entry for row and column i is the degree of the ith node.

The degree matrix for the social network of Figure 3.1 is shown below:

$$Deg = \begin{array}{c|cccccccccc|}
 & a & b & c & d & e & f & g & h & i & j \\
\hline
a & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
b & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
c & 0 & 0 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
d & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\
e & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\
f & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 \\
g & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 \\
h & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\
i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\
j & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}$$

The Laplacian matrix, represented as $L = Deg - Adj$, is a matrix that can be derived from a graph's adjacency matrix Adj and degree matrix Deg. The Laplacian matrix is the difference between the degree matrix and the adjacency matrix. The diagonal entries of the Laplacian matrix are the same as those in the degree matrix Deg. For entries off the diagonal, at the i^{th} row and j^{th} column, a value of -1 is present if there is an edge between nodes i and j, and a value of 0 if not. The Laplacian matrix has the property that each row and each column adds up to zero, which is typical of any Laplacian matrix.

The Laplacian matrix for the social network 3.1 is shown below:

$$L = \begin{array}{c|cccccccccc|}
 & a & b & c & d & e & f & g & h & i & j \\
\hline
a & 6 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 \\
b & -1 & 4 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\
c & -1 & -1 & 8 & -1 & -1 & -1 & -1 & -1 & -1 & 0 \\
d & -1 & -1 & -1 & 4 & -1 & 0 & 0 & 0 & 0 & 0 \\
e & -1 & -1 & -1 & -1 & 4 & 0 & 0 & 0 & 0 & 0 \\
f & -1 & 0 & -1 & 0 & 0 & 3 & -1 & 0 & 0 & 0 \\
g & -1 & 0 & -1 & 0 & 0 & -1 & 3 & 0 & 0 & 0 \\
h & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 2 & -1 & 0 \\
i & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 3 & -1 \\
j & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1
\end{array}$$

3.4.3 Clustering Methods

The various clustering techniques discussed in the following can be adapted to graphs as they are based on a general distance or similarity measure. We will briefly touch upon their suitability and direct readers to later chapters for more in-depth information on the methods.

Partitioning Methods

The main aim of dividing data objects into k distinct clusters is to identify groups where the members exhibit high similarity within their respective clusters but differ significantly from members in other clusters. The Partitioning Around Medoids (PAM) method, introduced by Kaufman and Rousseeuw [1990], aims to identify a representative object, known as a medoid, for each cluster. These medoids are the objects that are most centrally located within their respective clusters.

Initially, a set of k objects is chosen as the initial medoids. Then, during each step of the algorithm, all objects in the input dataset that are not currently designated as medoids are individually assessed to determine if they should replace one of the existing medoids. In other words, the algorithm checks if there is an object that would be a better fit as a medoid than the current ones. The decision to swap medoids with other non-selected objects is based on minimizing the total cost.

PAM's approach involves representing a cluster by its medoid, and as a result, it is also commonly referred to as the k-medoids algorithm. Algorithm 1 summarizes the PAM technique.

When evaluating the cost associated with swapping a non-medoid object (let's call it p_{rand}) with a medoid object (let's call it p_i), there are four scenarios to consider for each non-medoid object p. These scenarios are as follows:

Case 1: If p originally belongs to the medoid object p_i, and after the swap, p becomes closer to another medoid object p_j than to p_i, then p is reassigned to p_j.

Case 2: If p originally belongs to the medoid object p_i, and after the swap, p becomes closer to p_{rand} than to p_i, then p is reassigned to p_{rand}.

Case 3: If p originally belongs to one of the other medoid object p_j (where j is not equal to i), and after the swap, p remains closest to p_j, then there is no need to reassign p.

Case 4: If p originally belongs to one of the other medoid object p_j, and after the swap, p becomes closer to p_{rand} than to p_j, then p is reassigned to p_{rand}.

Algorithm 1: PAM algorithm for partitioning.

```
Input: $k$: the number of clusters, $D$: a data set containing $n$ objects
Output: A set of $k$ clusters
Arbitarily choose $k$ objects in $D$ as initial medoids
$C = -\infty$
while $C < 0$ do
    for each non-medoid object $p$ in $D$ do
        find the nearest medoid and assign $p$ to the corresponding cluster
    end
    randomly select a non-medoid $p_{rand}$
    compute the overall cost $C$ of swapping a medoid $p_i$ with $p_{rand}$
    if $C < 0$ then
        swap $p_j$ with $p_{rand}$ to form a new set of $k$ medoids
    end
end
return $k$ clusters
```

The cost function in this context is defined as the change in the value of the distortion function that occurs when a medoid object is replaced by a non-medoid object. The total cost C of making such replacements is calculated as the sum of the costs incurred by all non-medoid objects. If the total cost C is negative, it indicates that the replacement is permissible because it reduces the value of the distortion function.

Example 3.2. Consider a social network of objects located in a $2-$dimensional space, as shown in Figure 3.1. The user wants to divide the objects into 3 clusters, with $k = 3$.

The starting point of the k-medoids method is the dissimilarity matrix which is obtained by using distance or similarity measure. Clustering by k-medoids partitioning involves selecting three random objects as initial cluster centers (medoids) and assigning each object to a cluster based on its proximity to the cluster center. The cluster centers will then be updated. Figure 3.3 shows the resulting clusters $\{a,c,g,f\},\{b,d,e\}$, and $\{h,i,j\}$ by applying k-medoids clustering method. The Fruchterman-Reingold algorithm [Fruchterman and Reingold 1991] is used to visualize social networks by creating 2D representations based on the adjacency matrix, which is derived from the dissimilarity matrix.

By observing the triple $\{h,i,j\}$ closely, it is visible that the triple violates the TI property. Because there are edges between $\{h,i\}$ and between $\{i,j\}$, but there is no edge between $\{h,j\}$. Therefore, h and j must belong to different clusters to achieve meaningful clusters.

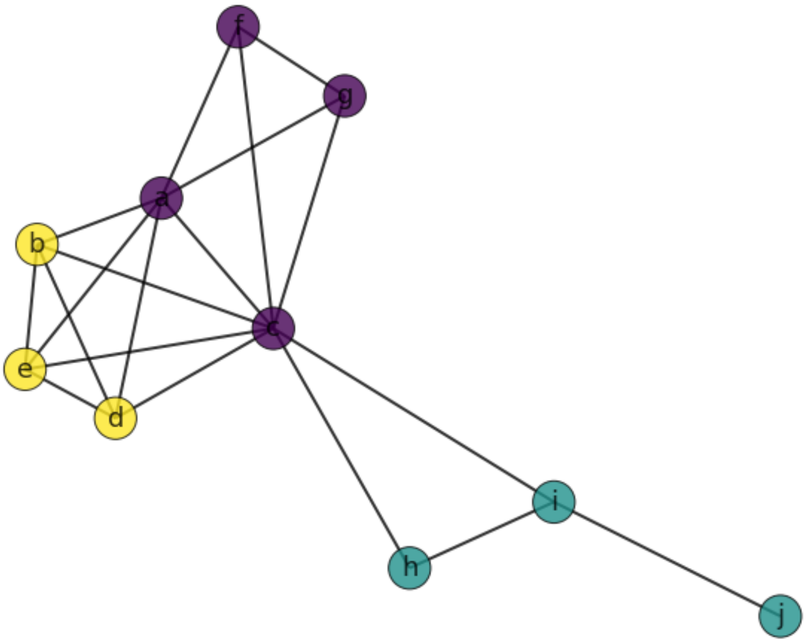


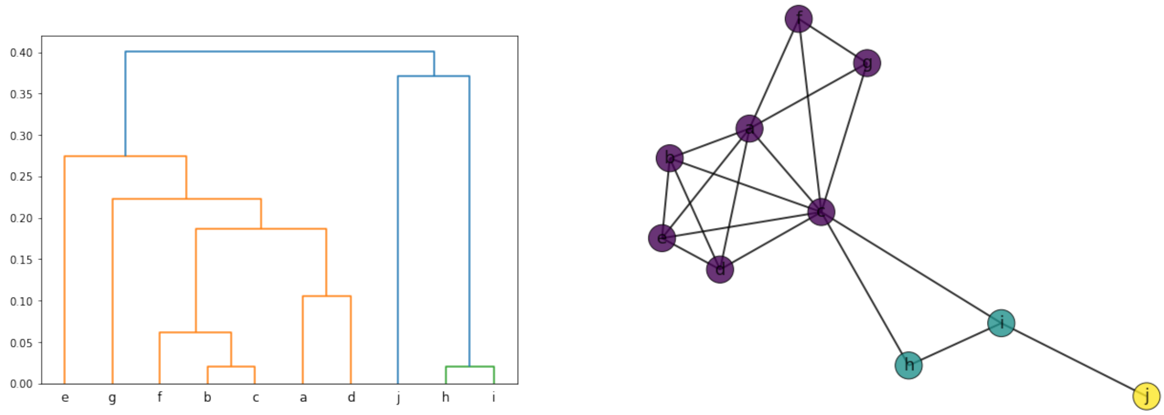
Fig. 3.3 Clustering of data objects $\{a,b,c,d,e,f,g,h,i,j\}$ using the k-medoids method.

Hierarchical Methods

Hierarchical clustering creates a cluster hierarchy, often known as a dendrogram, or a tree of clusters. Child clusters exist in every cluster node, while sibling clusters divide the points covered by their shared parent. This method enables data exploration at many levels of granularity. Hierarchical clustering methods are categorized into *agglomerative* (bottom-up) and *divisive* (top-down) [Jain and Dubes 1988]. An *agglomerative clustering* starts with one-point (singleton) clusters and merges two or more of the most appropriate clusters in a recursive manner. A *divisive clustering* starts with a single cluster of all objects and splits the most appropriate cluster recursively. The process continues until a stopping criterion (frequently, the requested number k of clusters) is achieved.

Example 3.3 (Agglomerative versus divisive hierarchical clustering.). Consider the social network of a data set of ten objects $\{a,b,c,d,e,f,g,h,i,j\}$, as depicted in Figure 3.1. The single linkage method is a well-known agglomerative hierarchical clustering technique. In this approach, the distance between two clusters is determined by measuring the distance between the closest pair of objects, with each object belonging to a different cluster. During the merging step, the algorithm identifies the nearest pair of clusters and combines them into a new, single cluster. Subsequently, it updates the distances between this newly formed cluster and the other clusters that remain unchanged. This merging process continues iteratively until the number of clusters ultimately reaches one.

The outcome of this algorithm is the creation of clusters in such a way that every member of a cluster shares a closer relationship with at least one other member of the same cluster than with any object outside of it. Additionally, this method has the capability to group together a chain of objects into a single cluster and can identify clusters with arbitrary shapes.



(a) Dendogram (single-linkage)

(b) Clusters ($k = 3$)

Fig. 3.4 Hierarchical clustering of data objects $\{a,b,c,d,e,f,g,h,i,j\}$.

In contrast, the divisive method works in a different way. To begin, all of the objects are combined into one cluster. The cluster is divided based on some criterion, such as the maximum Euclidean distance between the cluster's nearest neighbors. The process of cluster splitting continues until each new cluster contains only one object.

A dendrogram showing the clustering of objects $\{a,b,c,d,e,f,g,h,i,j\}$ is depicted in Figure 3.4a. The vertical axis represents the distance scale between the clusters. The algorithm merges two groups of objects into a single cluster when their distance is roughly 0.38, for example, $\{a,b,c,d,e,f,g\}$ and $\{h,i,j\}$

Figure 3.4b shows three clusters $\{a,b,c,d,e,f,g\}$, $\{h,i\}$, and $\{j\}$ by applying single-linkage hierarchical clustering method. It is clearly visible that the triple $\{b,a,f\}$ violates the TI property. Because there are edges between $\{b,a\}$ and between $\{a,f\}$ but there is no edge between $\{b,f\}$. But b and f must belong to different clusters.

Density-based Methods

A cluster in density-based clustering is a region with a high density of points surrounded by a low density region. DBSCAN, a density-based algorithm introduced by Ester et al. [1996], produces a partitional clustering and defines a cluster as a continuous area of any shape with greater density than its surroundings. The algorithm scans the data points in a dataset, computes neighborhoods with a defined radius and minimum number of points, and connects these dense neighborhoods to form clusters. A neighborhood with a defined radius and minimum number of points is referred to as a core point, while a data point without such a neighborhood is either considered a noise point or a border point if it is in the same neighborhood as a core point. The radius ε and the minimum number of points $MinPts$ serve as thresholds for determining the density of a neighborhood.

DENGRAPH is a graph clustering algorithm based on density, developed by Falkowski et al. [2007] to identify groups of similar nodes in graphs that contain numerous noise objects. Clusters in the graph are areas where nodes are densely packed and separated by low node density regions. DENGRAPH calculates neighborhoods by utilizing a specific radius (ε) and a minimum number of nodes ($MinPts$) to ensure that they are dense. A node with a neighborhood of this type is referred to as a core node. Nodes lacking such a neighborhood are classified as either border nodes if they are within a core node's neighborhood or noise nodes.

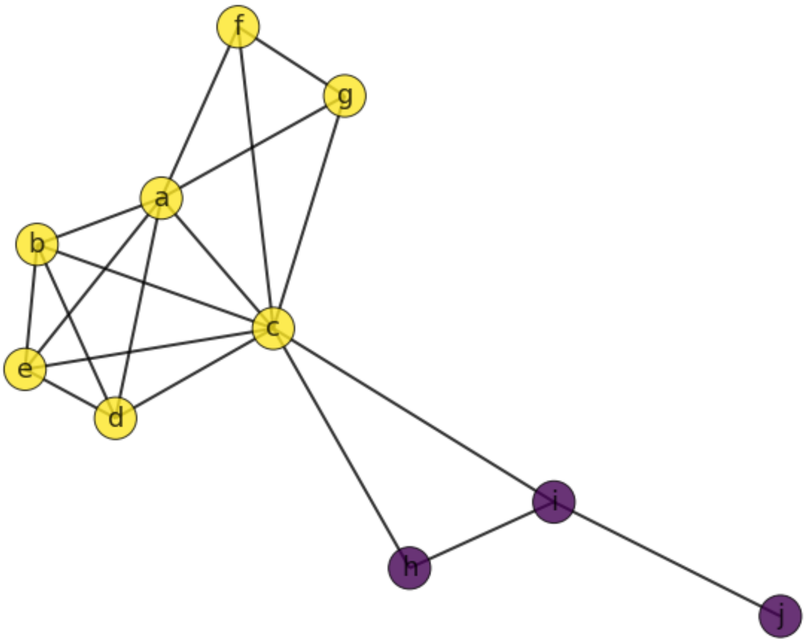


Fig. 3.5 DBSCAN clustering of data objects ($\varepsilon = 0.3, minPts = 3$).

To group similar data vertices, a measure of similarity or dissimilarity is needed. The similarity between two vertices is quantified by a numerical measure of how similar they are, and dissimilarity is quantified by the numerical difference between two objects. The higher the similarity, the more alike the objects are and the lower the dissimilarity. When referring to dissimilarity, the term "distance" is often used.

Figure 3.5 shows two clusters $\{a,b,c,d,e,f,g\}$ and $\{h,i,j\}$ by applying DENGRAPH. Unfortunately, the triples for example, $\{b,a,f\}$ and $\{h,i,j\}$ violates TI property. Thus may compromises the quality of resulting clusters.

Clique-based Methods

A clique is defined as a complete network on a set of nodes, with each pair of nodes connected by an edge in graph theory. If a clique can't be extended to a larger clique by including any nearby node, it's called maximal. Each community in a network should be a maximal clique in the ideal clustering outcome. In practice, this is difficult to achieve. It's considerably more difficult to guarantee that each community forms a clique in many real-world social networks. As a result, an acceptable measure for assessing a community's degree of connectivity is required.

Bron and Kerbosch [1973] presented a depth-first search algorithm for generating all the maximal cliques of an undirected graph as shown in Algorithm 2. That is, it lists all subsets of vertices with the two properties that each pair of vertices in one of the listed subsets is connected by an edge, and no listed subset can have any additional vertices added to it while preserving its complete connectivity.

Algorithm 2: Algorithm for searching all maximal cliques.

```
clique($P$,$R$,$X$)
if $P \cup X = \emptyset$ then
    report $R$ as a maximal clique
end
choose a pivot $u \in P \cup X$ to maximize $|P \cap N(u)|$
for each vertex $v \in P \setminus N(u)$ do
    clique($P \cap N(v)$,$R \cup \{v\}$,$X \cap N(v)$)
    $P \leftarrow P \setminus \{v\}$
    $X \leftarrow X \cup \{v\}$
end
```

The Bron–Kerbosch algorithm is a simple recursive algorithm that maintains three sets of vertices: a partial clique R, a set of candidates for clique expansion P, and a set of forbidden vertices X. In each recursive call, a vertex v from P is added to the partial clique R, and the sets of candidates for expansion and forbidden vertices are restricted to include only neighbors of v. If $P \cup X$ becomes empty, the algorithm reports R as a maximal clique, otherwise the algorithm chooses a vertex u in $P \cup X$ called a *pivot*. All maximal cliques must contain a non-neighbor of u (counting u itself as a non-neighbor), and therefore, the recursive calls can be restricted to the intersection of P with the non-neighbors.

Example 3.4. Consider the social network of ten objects shown in Figure 3.1. Algorithm 2 generates all maximal cliques $\{a,b,c,d,e\}$, $\{a,c,f,g\}$, $\{c,h,i\}$ and $\{i,j\}$ shown in Figure 3.6 where intra-cluster similarities of objects are higher and each cluster capture the natural structure of objects that reflects their relationship.

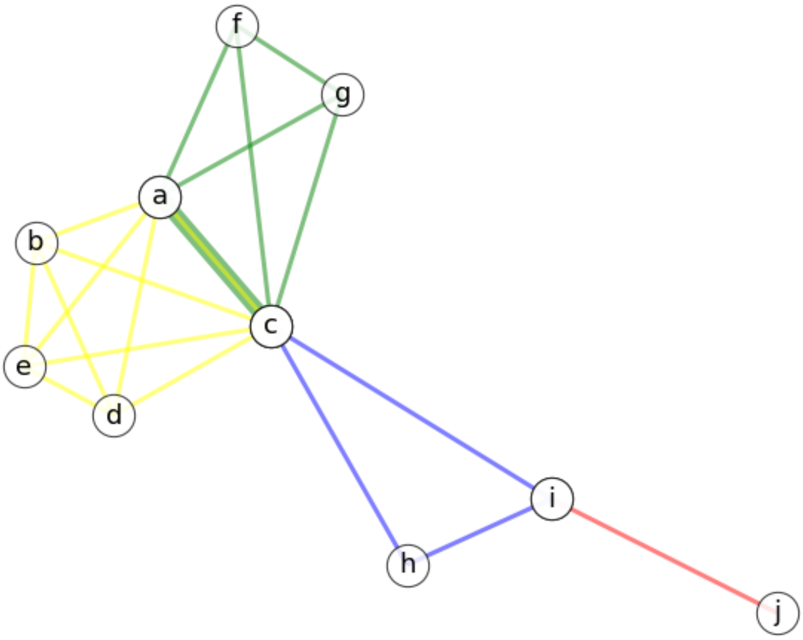


Fig. 3.6 Clique based clustering of data objects $\{a,b,c,d,e,f,g,h,i,j\}$.

Computing both the pivot and the vertex sets for the recursive calls can be done in time $O(|P| \cdot (|P| + |X|))$ within each call to the algorithm, using an adjacency matrix to quickly test the adjacency of pairs of vertices. This pivoting strategy, together with this adjacency-matrix-based method for computing the pivots, leads to a worst case time bound of $O(3^{n/3})$ for listing all maximal cliques.

3.5 Summary

To analyze experimental data in a variety of scientific disciplines, a large collection of clustering algorithms is available. In the scientific literature, new clustering programs are constantly appearing. The majority of these algorithms, however, are based on two popular clustering techniques: partitional clustering and agglomerative hierarchical clustering.

This chapter focused on the theoretical concepts of different clustering techniques and the measures used in these techniques: distance measure and similarity measure. It explained the violation of TI for detecting clusters in a social network using traditional clustering techniques and finally described a clique-guided approach to detect meaningful clusters.