Scientific Research Publishing

Journal of Computer and Communications, 2024, 12, 67-76
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

A Study of Triangle Inequality Violations in Social Network Clustering

Sanjit Kumar Saha[1], Tapashi Gosswami[2]

[1]Department of Computer Science and Engineering, Jahangirnagar University, Dhaka, Bangladesh
[2]Brandenburg University of Technology Cottbus-Senftenberg, Cottbus, Germany
Email: sanjit@juniv.edu, tapashi.cse@gmail.com

How to cite this paper: Saha, S.K. and Gosswami, T. (2024) A Study of Triangle Inequality Violations in Social Network Clustering. *Journal of Computer and Communications*, **12**, 67-76.
https://doi.org/10.4236/jcc.2024.121005

Received: December 16, 2023
Accepted: January 12, 2024
Published: January 15, 2024

Abstract

Clustering a social network is a process of grouping social actors into clusters where intra-cluster similarities among actors are higher than inter-cluster similarities. Clustering approaches, *i.e.*, *k*-medoids or hierarchical, use the distance function to measure the dissimilarities among actors. These distance functions need to fulfill various properties, including the triangle inequality (TI). However, in some cases, the triangle inequality might be violated, impacting the quality of the resulting clusters. With experiments, this paper explains how TI violates while performing traditional clustering techniques: *k*-medoids, hierarchical, DENGRAPH, and spectral clustering on social networks and how the violation of TI affects the quality of the resulting clusters.

Keywords

Clustering; Triangle Inequality Violations, Traditional Clustering, Graph Clustering

1. Introduction

A social network refers to a structure composed of individuals or entities or actors (nodes) connected by various types of relationships or interactions (edges). These networks can represent social relationships, professional connections, information flow, or any type of interconnected system where nodes and their interactions can be analyzed. Social networks provide a framework for understanding how individuals or entities are connected and how information or influence spreads within these structures.

Clustering of such a network is a process of grouping actors into disjoint clusters. Similarities among actors within the same cluster are higher compared with the similarities among actors between clusters.

Traditional clustering approaches use distance functions to measure similarities among actors. Among other properties, distance functions must adhere to the triangle inequality (TI). For example, for three actors *a*, *b*, and *c*, TI states that "if *a* is close to *b* and *b* is close to *c*, then *a* and *c* cannot be far away from each other". That means,

$$d(a,c) \le d(a,b) + d(b,c)$$

The significance of the triangular inequality in creating effective clustering methods was highlighted by Elkan [1], leveraging it to notably speed up the k-means algorithm. Additionally, Kryszkiewicz and Lasek [2], used the triangle inequality to enhance the search efficiency within the neighborhood space for the TI-DBSCAN version of DBSCAN.

But sometimes TI is violated. For example, consider a social network of seven actors $\{a,b,c,d,e,f,g\}$. **Figure 1** shows the resulting clusters $\{a,b,c,d,f\}$ and $\{e,g\}$. It is clearly visible in cluster $\{a,b,c,d,f\}$ that the triple $\{a,b,c\}$ has two edges among them. There is an edge between $\{a,b\}$ and an edge between $\{b,c\}$, but there is no edge between $\{a,c\}$. This indicates that the distance between *a* and *c* is greater than the combined distances from *a* to *b* and from *b* to *c*. Consequently, the closeness of *c* is affected as *a* is near *b* but distant from *c*. Thus, the triple $\{a,b,c\}$ violates the triangle inequality property. To get meaningful clusters, actors *a* and *d* must belong to different clusters.

2. Preliminaries

A social network can be seen as a graph. A graph, composed of vertices and edges, elucidate connections and structures within various systems. To quantify these connections, distance measures play a fundamental role, capturing the extent of separation or closeness between nodes. The adjacency matrix, a cornerstone in graph analysis, encapsulates these relationships succinctly, portraying the connectivity between nodes through a binary or weighted representation.

2.1. Graph

A graph $G=(V,E)$ comprises a finite set of *n* vertices $V=\{1,2,\cdots,n\}$ and a finite set of edges $E \subseteq V \times V$, representing pairs of distinct vertices. Vertices *u*

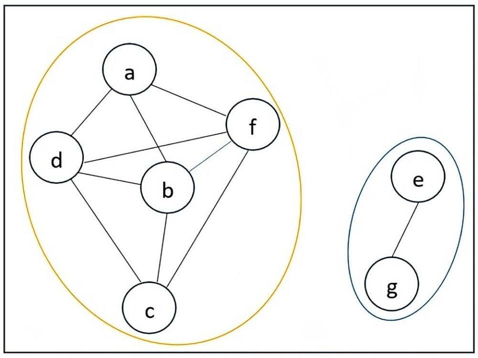

Figure 1. *k*-medoids clustering.

and v are considered *adjacent* (or *neighbors*) if $(u,v)\in E$. The *neighborhood* of a vertex *v*, symbolized as $N(v)$ (or $N_G(v)$ when specific mention of the graph is needed), is defined as $N(v)=u\in V \mid (u,v)\in E$.

When clustering social networks, distance metrics are pivotal for quantifying how alike or different nodes (representing individuals) are within the network. These metrics help identify clusters or communities of nodes that exhibit comparable connection or interaction patterns.

2.2. Distance Measure

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*.

Measures that meet all the specified properties are referred to as metrics. These properties can prove advantageous in specific applications; for instance, when the triangle inequality property holds, it enables more efficient clustering processes.

2.3. Adjacency Matrix

An adjacency matrix $A=\{a_{ij}\}$ with threshold *th* can be derived from a dissimilarity matrix where

$$a_{ij}=\begin{cases} 1 & \text{if } d_{ij}\leq th \\ 0 & \text{otherwise} \end{cases}$$

3. Clustering Approaches

The clustering techniques we'll explore next are versatile for graph adaptation, relying on a broad distance or similarity measure. We'll briefly assess their compatibility and guide readers toward comprehensive details about these methods.

3.1. Partitioning Methods

The primary objective behind partitioning data objects into *k* distinct clusters is to discern sets where members exhibit high similarity within their own cluster while demonstrating significant differences from members in other clusters. The Partitioning Around Medoids (PAM) technique, pioneered by Kaufman and Rousseeuw [3], strives to find a central and representative entity referred to as a "medoid" for every cluster. These medoids symbolize the entities holding the most central positions within their individual clusters. PAM's approach centers on representing clusters by their medoids, hence it's commonly known as the *k*-medoids algorithm.

At the outset, a set of *k* objects is chosen as the starting medoids. Throughout the algorithm's iterations, non-medoid dataset objects are assessed individually to potentially replace existing medoids. This process aims to find more suitable medoids, minimizing the overall cost by exchanging them with unselected objects.

3.2. Hierarchical Methods

Hierarchical clustering constructs a cluster hierarchy, often depicted as a dendrogram or a clusters tree. Each cluster node contains child clusters, and sibling clusters separate the data points covered by their shared parent. This method allows for multi-level data exploration. There are two main categories of hierarchical clustering: *agglomerative* (bottom-up) and *divisive* (top-down) (Jain and Dubes [4]). In *agglomerative clustering*, the process begins with individual (singleton) clusters and progressively merges the most compatible clusters. In *divisive clustering*, the process starts with a single cluster comprising all objects and recursively divides the most suitable clusters. This process continues until a stopping criterion, often the desired number of clusters, is reached.

The single linkage method stands as a widely acknowledged agglomerative hierarchical clustering technique. Initially, the distance between each pair of individual data points (or clusters, in the initial stage) is computed. This distance is usually defined by a chosen metric, like Euclidean distance or other appropriate measures based on the data. The method identifies the closest pair of points or clusters based on the shortest distance (single link) between any two points, each from a different cluster. These clusters are then merged into a new cluster. After merging, the distances between the newly formed cluster and other clusters or points are recalculated. It considers the distance between the new cluster and other clusters as the minimum distance between any point from the new cluster and any point from the other clusters. It iterates, continually identifying the closest pair of clusters or points and merging them until a stopping condition is met. This could be a predetermined number of clusters or when a specific criterion is satisfied.

3.3. DENGRAPH

DENGRAPH is a density-based graph clustering algorithm introduced by Falkowski *et al.* [5]. Its primary purpose is to detect clusters of similar nodes within graphs that may contain a significant amount of noise objects. Within the graph, clusters are delineated as regions exhibiting high node density, demarcated by regions characterized by lower node density. DENGRAPH identifies these neighborhoods by employing a specific radius (ε) and a minimum number of nodes (MinPts) to ensure their density. Nodes possessing such neighborhoods are known as “core nodes”. Nodes lacking these neighborhoods are classified either as “border nodes” if they fall within the neighborhood of a core node or as “noise nodes” otherwise.

3.4. Spectral Clustering

Spectral Clustering is a clustering method that relies on the interconnections between data points to create clusters. It utilizes the eigenvalues and eigenvectors of the data matrix to project the data into a lower-dimensional space for clustering. This approach is rooted in the concept of representing data as a graph, where data points are nodes, and the relationships between data points are denoted by edges as described by von Luxburg [6].

4. Experimental Results

To identify how triangle inequality is violated while performing traditional clustering approaches on social networks, several experiments are performed. A small social network of 10 people {*siam*, *ayon*, *alan*, *joe*, *eliza*, *diana*, *lina*, *ayuba*, *deba*, *kayle*} is considered, which is easy to understand. The Fruchterman-Reingold [7] algorithm is used to visualize a network that takes the adjacency matrix of a network as input.

After calculating dissimilarities among pairs of people using Euclidean distance function, a dissimilarity matrix D is obtained as:

$$D=\begin{bmatrix} 0 & 0.01 & 0.62 & 0.84 & 0.45 & 0.34 & 0.96 & 0.22 & 0.51 & 0.11 \\ 0.01 & 0 & 0.78 & 1.24 & 0.99 & 0.30 & 0.22 & 0.89 & 0.54 & 0.11 \\ 0.62 & 0.78 & 0 & 0.42 & 0.51 & 0.31 & 0.11 & 0.51 & 0.87 & 0.13 \\ 0.84 & 1.24 & 0.42 & 0 & 1.05 & 0.08 & 0.84 & 0.25 & 0.56 & 0.19 \\ 0.45 & 0.99 & 0.51 & 1.05 & 0 & 0.94 & 0.54 & 0.36 & 1.35 & 0.37 \\ 0.34 & 0.30 & 0.31 & 0.08 & 0.94 & 0 & 0.15 & 0.25 & 0.03 & 0.54 \\ 0.96 & 0.22 & 0.11 & 0.84 & 0.54 & 0.15 & 0 & 0.43 & 0.36 & 0.53 \\ 0.22 & 0.89 & 0.51 & 0.25 & 0.36 & 0.25 & 0.43 & 0 & 0.73 & 0.22 \\ 0.51 & 0.54 & 0.87 & 0.56 & 1.35 & 0.03 & 0.36 & 0.73 & 0 & 0.33 \\ 0.11 & 0.11 & 0.13 & 0.19 & 0.37 & 0.54 & 0.53 & 0.22 & 0.33 & 0 \end{bmatrix}$$

It is clearly visible inside the dissimilarity matrix that the triple, for example, {0.34, 0.15, 0.96} violates the triangle inequality property. That means

$$d\left(siam,lina\right)\le d\left(siam,diana\right)+d\left(diana,lina\right)$$
$$\Rightarrow 0.96\not\le 0.34+0.15\Rightarrow 0.96\not\le 0.49$$

An adjacency matrix Adj can be derived from the dissimilarity matrix for a threshold $th=0.7$ as:

$$Adj=\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Figure 2 shows the pictorial representation of the network using the adjacency matrix.

Several experiments have been performed on social network in Figure 2 using *k*-medoids, hierarchical, DENGRAPH and spectral clustering respectively and described how TI violates within resulting clusters during clustering.

After applying the *k*-medoids clustering approach for $k = 3$, three clusters, {*siam*, *diana*, *lina*, *kayle*}, {*ayon*, *deba*}, and {*alan*, *joe*, *eliza*, *ayuba*}, can be found as shown in Table 1. It is noted that the dissimilarity matrix D is the starting point of the *k*-medoids approach. It is visible in Figure 3 that within the cluster {*siam*, *diana*, *lina*, *kayle*}, there is no edge between *lina* and *siam*, which means they do not belong to the same cluster. Therefore, they must reside in two

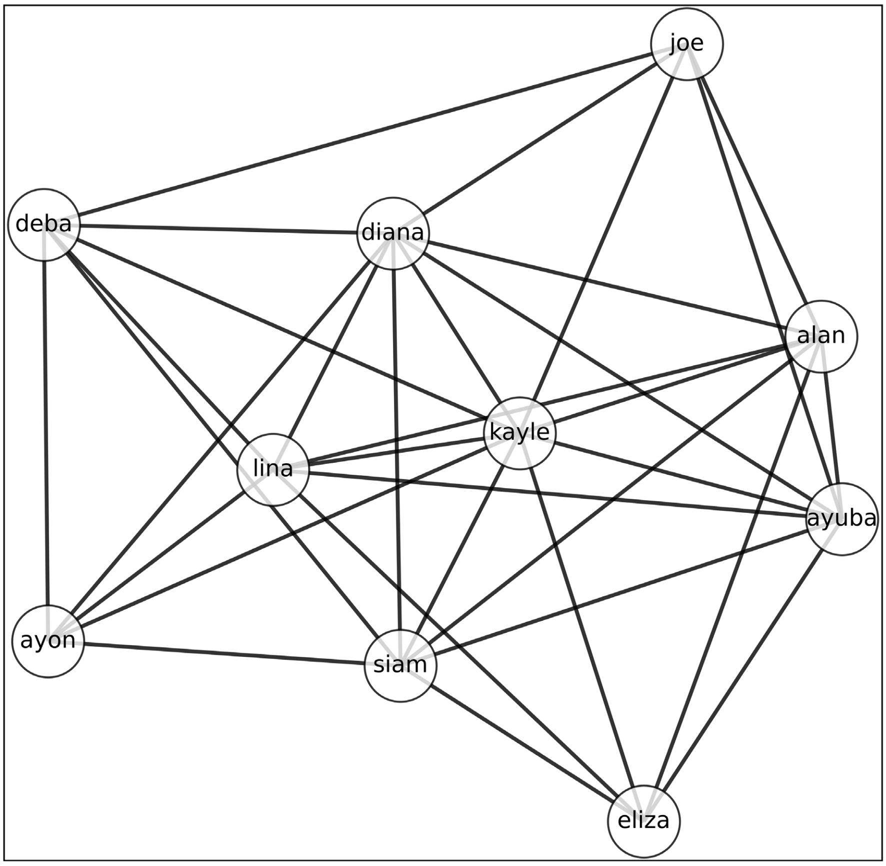

Figure 2. Social network.

Table 1. Clusters.

Methods	Clusters
k-medoids ($k = 3$)	{*siam, diana, lina, kayle*}, {*ayon, deba*}, {*alan, joe, eliza, ayuba*}
Hierarchical (*Single-linkage*)	{*siam, ayon, alan, joe, diana, lina, deba, kayle*}, {*eliza*}, {*ayuba*}
DENGRAPH	{*joe, diana, lina, deba*}, {*siam, ayon, alan, eliza, ayuba, kayle*}
Spectral	{*siam, joe, diana, lina, kayle*}, {*alan, eliza, ayuba* }, {*ayon, deba*}

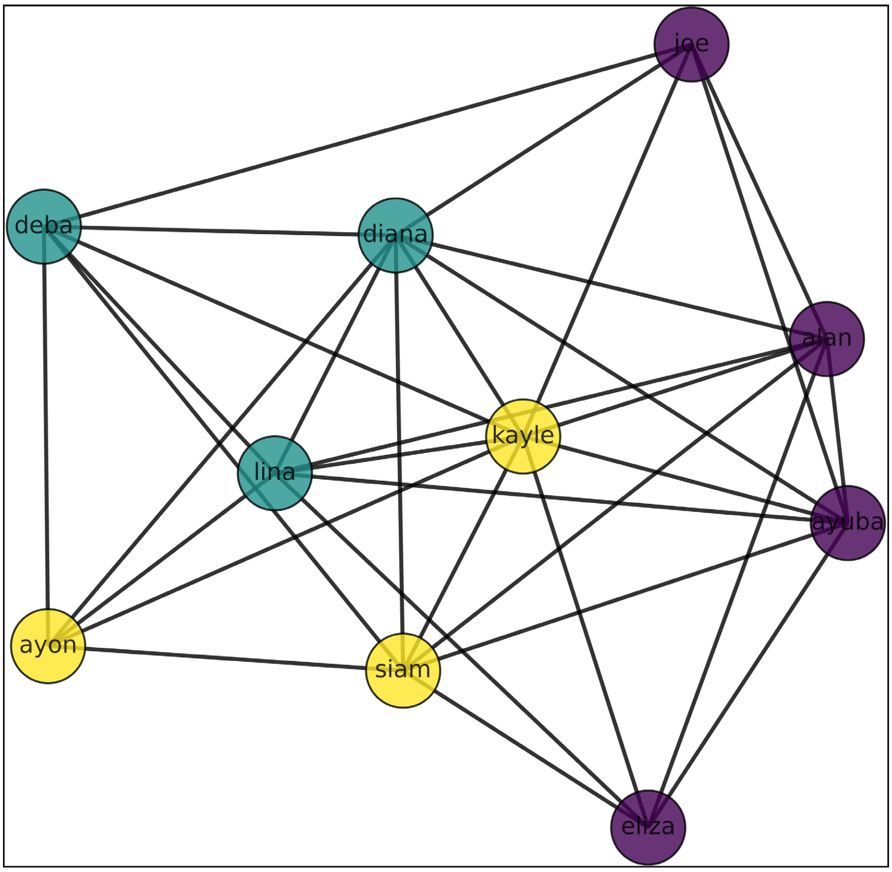

Figure 3. *k*-medoids clustering.

different clusters. Similarly, within the cluster {*alan*, *joe*, *eliza*, *ayuba*}, there is no edge between *joe* and *eliza*, but unfortunately they are within the same cluster. Thus, it violates the TI property and compromises the quality of the resulting clusters.

The next experiment shows how TI violates after applying a single-linkage hierarchical clustering approach to the social network. A dendogram is shown in Figure 4. Three clusters, {*siam*, *ayon*, *alan*, *joe*, *diana*, *lina*, *deba*, *kayle*}, {*eliza*}, and {*ayuba*}, can be found as illustrated in Figure 5. By closely observing within the cluster {*siam*, *ayon*, *alan*, *joe*, *diana*, *lina*, *deba*, *kayle*} in Figure 5, it is noticeable that several pairs of objects do not have edges between them; for example, there is no edge between *deba* and *joe*, *deba* and *alan*, and *ayon* and *joe*. In spite of this, they belong to the same cluster. Thus, it violates the TI property.

Two clusters, {*joe*, *diana*, *lina*, *deba*} and {*siam*, *ayon*, *alan*, *eliza*, *ayuba*, *kayle*}, can be found after applying the DENGRAPH approach for $\varepsilon = 0.2$ and $MinPts = 4$ as shown in Figure 6. Both clusters have some pairs of objects that do not have edges between them. There is no edge between *lina* and *joe* within the cluster {*joe*, *diana*, *lina*, *deba*}. That means they must reside in different clusters. Also, it is clearly visible within cluster {*siam*, *ayon*, *alan*, *eliza*, *ayuba*, *kayle*} that there are several pairs of objects that do not have edges between them. For example, there is no edge between *ayon* and *eliza*. In spite of this, they belong to the same clusters, which violates the TI property.

In the last experiment, the spectral clustering approach was applied to a social network to find clusters. Figure 7 shows the resulting clusters {*siam*, *joe*, *diana*,

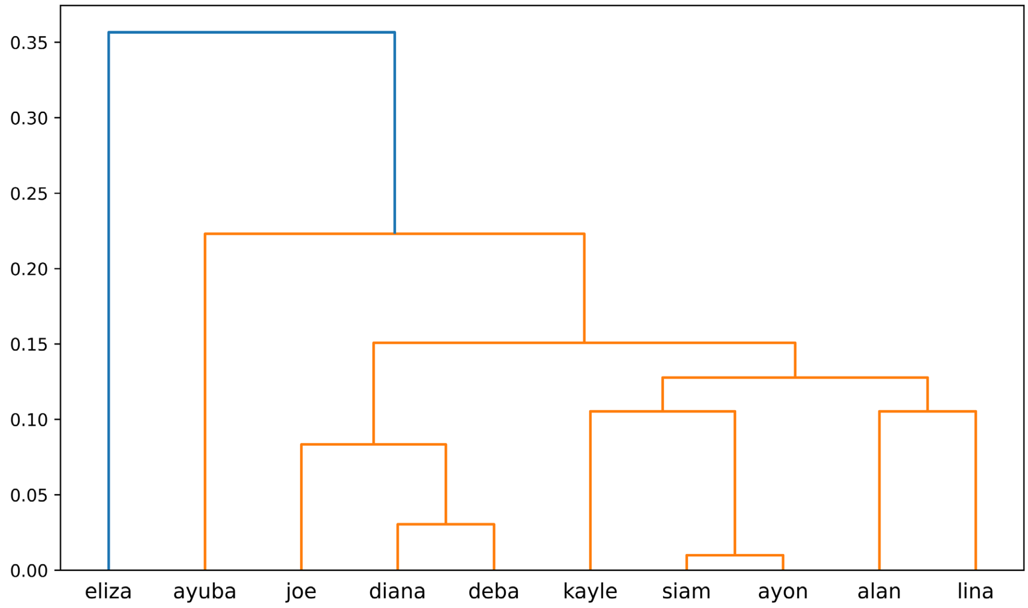

Figure 4. Dendogram.

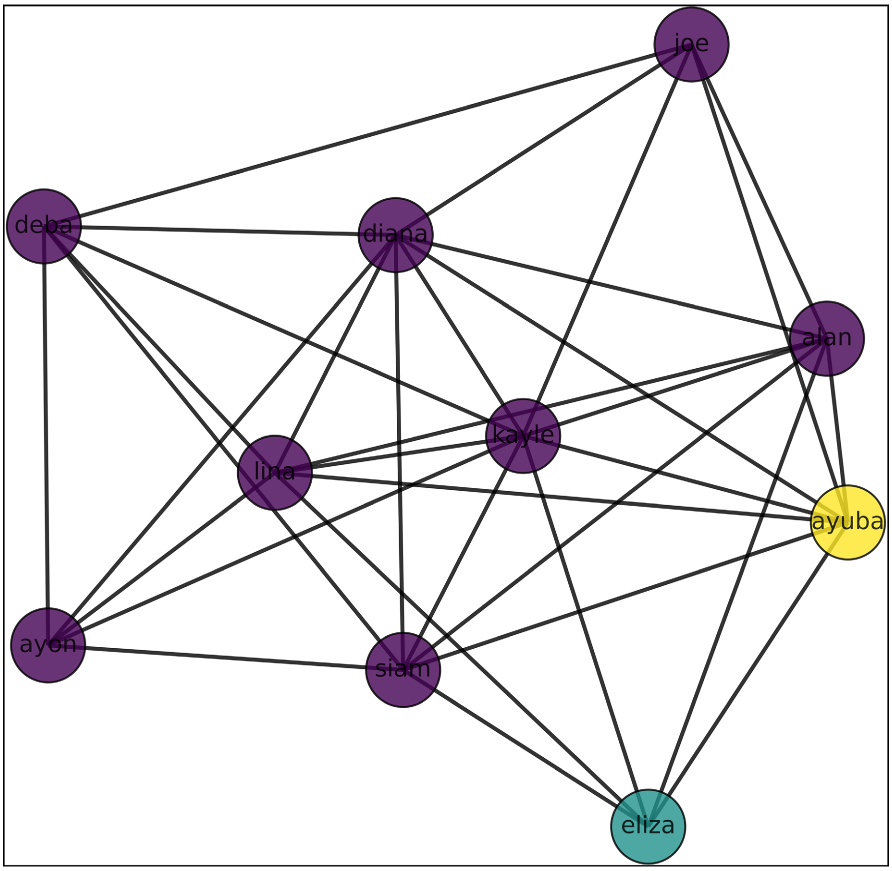

Figure 5. Hierarchical clustering.

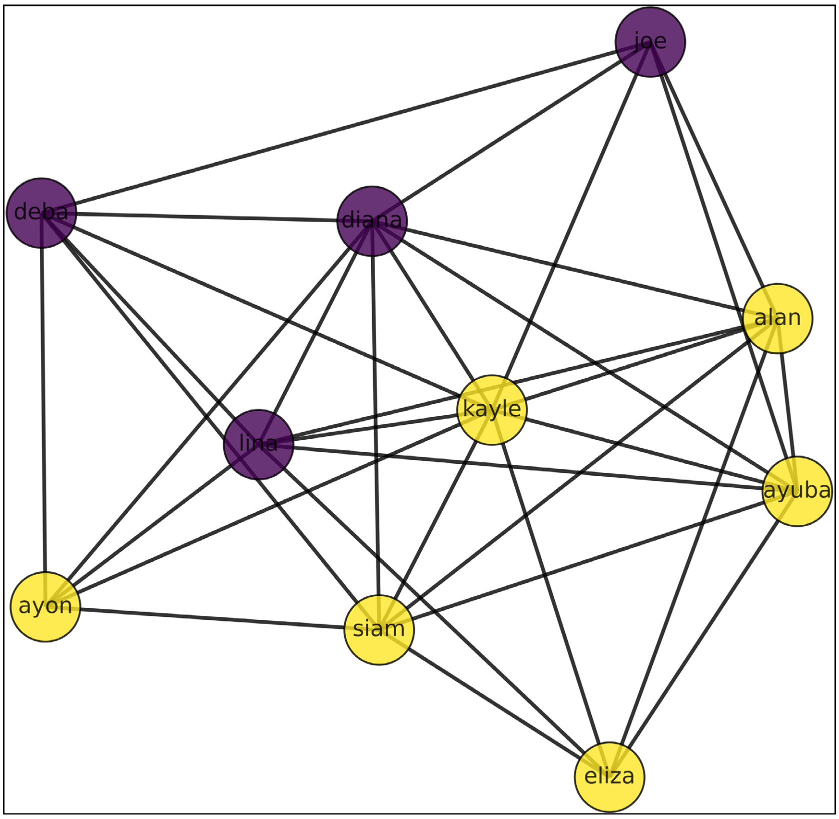

Figure 6. DENGRAPH clustering.

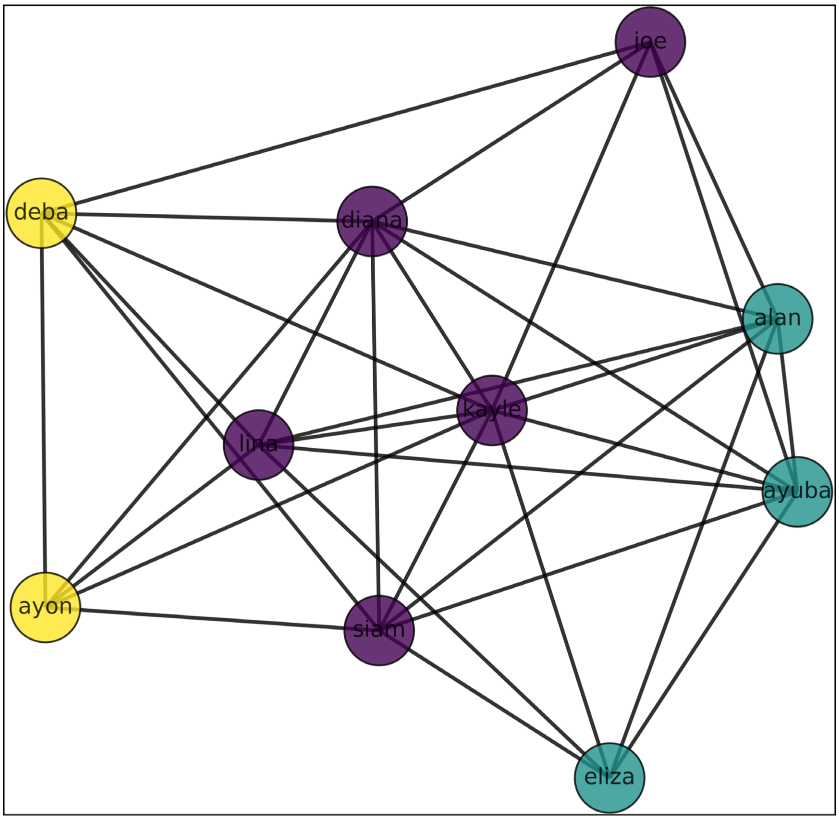

Figure 7. Spectral clustering.

lina, *kayle*}, {*alan*, *eliza*, *ayuba*}, and {*ayon*, *deba*}. TI violates within the cluster {*siam*, *joe*, *diana*, *lina*, *kayle*} where there is no edge between *siam* and *joe* and between *lina* and *joe*.

All four experiments by applying traditional clustering approaches to social networks explain clearly that sometimes TI violates within the resulting clusters and thus may compromise the quality of the clusters.

5. Conclusion

The main purpose of this paper is to explain with experiments how TI may violate within resulting clusters while performing traditional clustering approaches like *k*-medoid or hierarchical clustering approaches. Therefore, when the triangle inequality is violated, it undermines the quality of the clusters produced. The experiments elaborately explain that the violation of the TI property sometimes happens within clusters for different clustering approaches: *k*-mediods, hierarchical clustering, DENGRAPH, and spectral clustering. However, it's feasible to discover significant clusters where the objects within a cluster depict their relationships. In this scenario, two objects lacking direct edges between them might belong to separate clusters, and individual objects might exist in multiple clusters simultaneously.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Elkan, C. (2003) Using the Triangle Inequality to Accelerate k-Means, In: *Proceedings of the Twentieth International Conference on International Conference on Machine Learning* (*ICML*'03). AAAI Press, Washington DC, 147-153.

[2] Kryszkiewicz, M. and Lasek, P. (2010) TI-DBSCAN: Clustering with DBSCAN by Means of the Triangle Inequality, In: Szczuka, M., Kryszkiewicz, M., Ramanna, S., Jensen, R., Hu, Q., Eds., *RSCTC* 2010, LNCS, Vol. 6086, Springer, Heidelberg, 60-69. https://doi.org/10.1007/978-3-642-13529-3_8

[3] Kaufman, L. and Rousseeuw, P.J. (1990) Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons, New York. https://doi.org/10.1002/9780470316801

[4] Jain, A. and Dubes, R. (1988) Algorithms for Clustering Data. Prentice-Hall, Englewood Cliffs, NJ.

[5] Falkowski, T., Barth, A. and Spiliopoulou, M. (2007) DENGRAPH: A Density-Based Community Detection Algorithm. *IEEE*/ *WIC*/*ACM International Conference on Web Intelligence* (*WI*'07), Fremont, CA, 2-5 November 2007, 112-115. https://doi.org/10.1109/WI.2007.74

[6] von Luxburg, U. (2007) A Tutorial on Spectral Clustering. *Statistics and Computing*, **17**, 395-416. https://doi.org/10.1007/s11222-007-9033-z

[7] Fruchterman, T.M.J. and Reingold, E.M. (1991) Graph Drawing by Force-Directed Placement. *Software*: *Practice and Experience*, **21**, 1129-1164. https://doi.org/10.1002/spe.4380211102