Community Detection in Dynamic Face-to-Face Interaction Networks: A Louvain Algorithm Approach

Iroda Ibrohimova *

October 13, 2023

Abstract

In this paper, we present a study that evaluates the suitability of the Louvain Algorithm in the context of face-to-face interaction networks. Traditional community detection methods face challenges in this context, necessitating specialized solutions. Our research addresses this gap, offering a systematic approach that aggregates individual game data and applies the Louvain Algorithm. The results demonstrate the algorithm's effectiveness in consistently identifying the original 34 communities, demonstrating its relevance in face-to-face interaction networks.

1 Introduction

Social networks are interconnected individuals or entities characterized by relationships, interactions, and information flow. They indicate key webs of human interactions and help to understand these interactions' dynamics [New10]. These qualities make them potential grounds for applying machine learning techniques to discover patterns and structures within communities [WF94]. With its capacity to sift through vast datasets, machine learning offers a powerful tool to dissect and comprehend the complexities of social networks, enabling insights into human behavior, influence propagation, and community formation [Lea09], [CGP12].

Community detection is a widely employed technique in graph analysis aimed at partitioning vertices within a graph into coherent "communities" based on their relatedness. It serves as a valuable tool in various scientific and industrial fields, including biology, social networks, finance, and literature analysis, aiding in discovering meaningful structural patterns. Comprehensive reviews on the different formulations, methods, and applications of community detection can be found in Michele Coscia, Fosca Giannotti, Dino Pedreschi, and Santo Fortunato [For10], [KN11]. Various measures have been proposed to evaluate the goodness of partitioning produced by a community detection method [KS88], [NG04].

Among the methods used, modularity stands out due to its widespread application. Introduced by Newman [NG04], modularity quantifies the quality of community assignments by examining the proportion of edges within communities. However, modularity has constraints, including a resolution limit [FB07]. Nonetheless, it remains a popular choice for practitioners, and resolution-limit-free variations have been suggested.

Numerous efficient heuristics have been developed over the years, making the analysis of large-scale networks feasible in practice. The Louvain method, a highly efficient heuristic proposed by

*Advised by: Dr Maria Konte

Blondel et al. [BGLL08], has gained prominence for its speed and the quality of results it provides in practice.

Despite the widespread use of community detection methods, applying them to face-to-face interaction networks remains challenging. These networks, characterized by limited data and direct, physical interactions, differ significantly from virtual networks. This uniqueness necessitates tailored approaches. In the context of face-to-face interaction networks, the Louvain algorithm can be an effective method for community detection.

The objective of this study is to comprehensively assess the suitability of the Louvain Algorithm in the context of face-to-face interaction networks. These networks present distinctive challenges involving sparse data due to limited observation opportunities. Traditional community detection methods may not readily adapt to these conditions, necessitating specialized solutions.

This research addresses the need for effective community detection in face-to-face networks. The complexity arises from the need to extract meaningful patterns from a web of brief, physical interactions, a task conventional methods struggle with. Furthermore, the urgency of solving this problem stems from the increasing interest in comprehending real-world, physical interactions across diverse contexts, from workplaces to social gatherings.

To address the challenges described above, we have developed a systematic approach. We aggregated individual game data and applied the Louvain Algorithm to detect communities within each game. Subsequently, we used the results to assess whether the algorithm consistently identifies the original 34 communities. This process serves as a rigorous evaluation of the algorithm's effectiveness and underscores its relevance within face-to-face interaction networks.

Contributions: we make the following contributions in this paper: 1. Enhanced Resolution Parameter: Our study introduces a refined resolution parameter setting (8) specifically tailored to the dynamic nature of face-to-face interaction networks. 2. Methodological Framework: The research establishes a comprehensive methodological framework for analyzing face-to-face interaction networks. 3. Specialized Application of the Louvain Algorithm: This research employs the tailored application of the Louvain Algorithm to face-to-face interaction networks, a domain with distinct observational constraints and network characteristics.

2 Related Works

There have been notable efforts to address community detection within social networks, with a focus on various methodologies [BBPSV18], [PLR19], [PBM20], [MMP19], [BCT$^+$14]. For instance, Barrat et al. studied temporal multilayer networks, shedding light on the social structure of face-to-face interaction [BBPSV18]. Their work significantly advances our understanding of dynamic network interactions, primarily revolving around temporal aspects. Similarly, Peralta, Loslever, and Ramos contributed substantially to the field by examining community detection in social networks and provided critical insights into the intricate interplay of social dynamics, with a focus on broader virtual networks [PLR19].

In face-to-face proximity networks, Puglisi, Bullo, and Mantzaris [PBM20] dived deeper into the application of stochastic block models and community detection, providing a significant step toward understanding proximity-based social interactions. Their approach centers on a specific modeling framework. Conducting a data-driven study on community detection in these networks, Morone et al. [MMP19] offer valuable insights into the underlying structures, primarily focusing on the structural aspects. Additionally, Barrat and colleagues [BCT$^+$14] lay a significant groundwork by exploring community detection within temporal multilayer networks. This work provides a foundation for understanding social structures in face-to-face interactions, particularly temporal aspects.

In comparison, our study uniquely addresses the interplay of face-to-face interaction networks. By employing a systematic approach that encompasses data aggregation, algorithm application, and parameter optimization, we present a comprehensive framework tailored to this distinct domain.

3 Data Preparation

3.1 Dataset Description

The dataset used in this research paper was obtained from a series of face-to-face interaction games called Resistance, conducted as part of the research study [Lesnd]. Face-to-face interactions were extracted from videos of participants playing the game. Dynamically evolving networks were extracted from the free-form discussions using the ICAF algorithm. The extraction algorithm is a collective classification algorithm that leverages computer vision techniques for eye gaze and head pose extraction.

Each game had 5–9 participants and lasted 45-60 minutes. Each participant was part of exactly one game. In total, the dataset had 34 games and 232 participants [KBSL21].

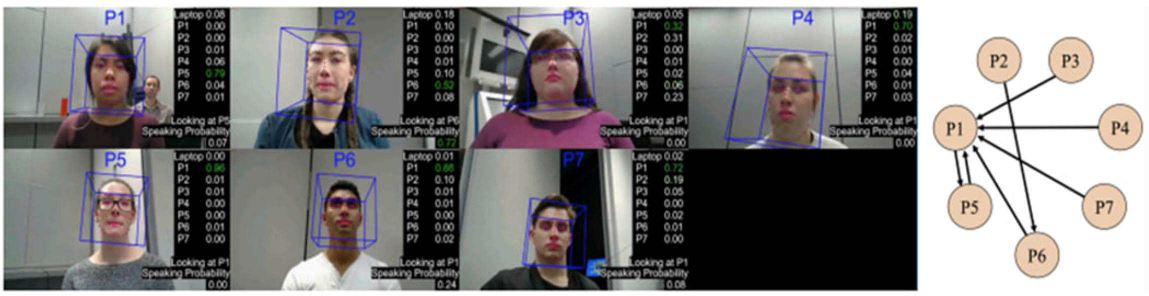

Figure 1: Given a group video conversation (left), we extracted face-to-face dynamic interaction networks (right), representing the instantaneous interactions between participants. Participants are nodes, and interactions are edges in the network.

The dataset is in CSV format, with one file for each game.

The networks are weighted, directed, and temporal. In the Excel spreadsheet, there are columns containing information about timestamps. The behavior was recorded for every 1/3 seconds, leading to a total of 9650 1/3 seconds. In addition, there are 65 rows in each file. Every third of a second, a line is drawn from one person (node 1) to another (node 2). The strength of this line is determined by how likely it is that person '1' is looking at person '2' or at the laptop.

3.2 Data Cleaning

The data was available in a zip file containing binary and weighted data. The latter type of data was most suitable, as it captures the strength and intensity of communication, which in our case is the likelihood of looking. Therefore, all files with binary data were extracted from the zip file.

It should be noted that the dataset did not have any instances of gaps in values that demonstrate the validity of the recorded data and simplified following analytical procedures.

1	TIME	P1_TO_LAPTOP	P1_TO_P1	P1_TO_P2	P1_TO_P3	P1_TO_P4	P1_TO_P5	P1_TO_P6	P1_TO_P7
2	0	0.077	0	0.109	0.099	0.101	0.134	0.137	0.07
3	1	0.1	0	0.134	0.127	0.124	0.153	0.19	0.08
4	2	0.15	0	0.12	0.119	0.12	0.156	0.139	0.087
5	3	0.3	0	0.076	0.104	0.08	0.091	0.071	0.141
6	4	0.629	0	0.046	0.043	0.073	0.04	0.057	0.044
7	5	0.546	0	0.064	0.053	0.066	0.067	0.063	0.06
8	6	0.644	0	0.024	0.031	0.07	0.033	0.059	0.056
9	7	0.654	0	0.017	0.034	0.07	0.03	0.057	0.057
10	8	0.653	0	0.017	0.034	0.07	0.027	0.059	0.061
11	9	0.623	0	0.02	0.033	0.073	0.039	0.063	0.06
12	10	0.564	0	0.051	0.06	0.069	0.07	0.071	0.04
13	11	0.637	0	0.034	0.03	0.074	0.033	0.06	0.041
14	12	0.624	0	0.037	0.037	0.069	0.04	0.06	0.054
15	13	0.671	0	0.016	0.03	0.071	0.026	0.057	0.044
16	14	0.27	0	0.077	0.114	0.079	0.081	0.084	0.17
17	15	0.221	0	0.074	0.127	0.079	0.09	0.091	0.189
18	16	0.27	0	0.096	0.103	0.084	0.111	0.081	0.159
19	17	0.557	0	0.051	0.056	0.079	0.053	0.061	0.056
20	18	0.444	0	0.08	0.083	0.08	0.087	0.07	0.059

Figure 2: In the figure, the main elements and variables of the dataset can be observed (TIME, P1_TO_P2 , etc.)

3.3 Data Processing

Several steps have been taken to test whether the Louvain Algorithm is adequate for identifying communities within dynamic face-to-face interaction networks, and the code was adopted to address them. First, data is presented in Excel spreadsheet format; in each, participants are always in order from 1 to 7, even though any participant can participate only once in a game. Therefore, having similar and repetitive names can interfere with the algorithm's work, so the code was adjusted to simulate interactions between different sets of participants across multiple games. This was done by keeping track of the participant numbers and incrementing them for each new group of participants. To illustrate, in the first file, participants were labeled from 1 to 7, while in the second file (2nd game), participants were labeled from 8 to 15, and this chain continued until it reached the last 232nd participant.

4 Methodology

4.1 Network Construction

Participants and their interactions were represented as nodes and edges in constructing the network. Each participant and the laptop were represented as nodes. Nodes were labeled according to the order number of the gamer. Edges were defined by the calculated probabilities of participants looking at each other.

This approach relied on face-to-face interactions, with visual attention being an essential aspect of engagement. To define edges, the weighted interaction values were considered, where if the nodes' interaction value equaled 0, no connection (edge) was created. In contrast, with higher values, the distance between nodes decreased, placing participants closer to each other and forming clusters.

4.2 Graph Analysis and Visualization

NetworkX and Matplotlib were used for analyzing and visualizing these dynamic networks.

NetworkX is a Python library designed to create, manipulate, and study complex networks. In our study, it served as a foundational tool for understanding the intricate web of interactions among participants. It allowed us to represent these interactions as nodes (representing participants) and edges (indicating connections between them). This facilitated a comprehensive analysis of the dynamic face-to-face interaction networks. [HSS08].

Matplotlib is a versatile data visualization library in Python. It allowed us to translate raw data into insightful visual representations. Our research used Matplotlib to render the network graphs derived from NetworkX. This enabled us to visually explore and communicate the patterns and dynamics inherent in the interactions, offering a clear and interpretable representation of the complex dataset. [Hun07]

4.3 The Louvain Community Detection Algorithm

The algorithm is divided into two phases that are repeated iteratively. In the first phase, each node is initially assigned to its community, resulting in as many communities as there are nodes in the network. Then, for each node, the algorithm considers its neighbors and evaluates the gain of modularity that would occur by removing the node from its community and placing it in the community of its neighbor. The node is then placed in the community, which results in the maximum gain of modularity, but only if this gain is positive. The node remains in its original community if no positive gain is possible. This process is repeated iteratively for all nodes until no further improvement in modularity can be achieved. [BGLL08]

The modularity formula is expressed as:

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

where:

$$\begin{aligned} Q &\text{ is the modularity,} \\ m &\text{ is the total number of edges in the network,} \\ A_{ij} &\text{ is the weight of the edge between nodes } i \text{ and } j, \\ k_i \text{ and } k_j &\text{ are the degrees of nodes } i \text{ and } j, \text{ respectively,} \\ c_i \text{ and } c_j &\text{ are the communities to which nodes } i \text{ and } j \text{ belong,} \\ \delta(c_i, c_j) &\text{ is the Kronecker delta function, which is equal to 1 if } c_i = c_j \text{ and 0 otherwise.} \end{aligned}$$

In simpler terms, the modularity formula measures the difference between the actual number of edges within communities and the expected number of edges if the edges were distributed randomly.

A higher modularity value indicates a better partition of the network into communities. The modularity formula is a critical component of the Louvain method, as it is used to evaluate the quality of the communities detected in every iteration of the algorithm. [BGLL08]

The second phase of the Louvain method involves building a new network whose nodes are the communities found during the first phase. To do this, the weights of the links between the new nodes are given by the sum of the weights of the links between nodes in the corresponding two communities. Links between nodes of the same community lead to self-loops for this community in the new network. Once this second phase is completed, the first phase of the algorithm is reapplied

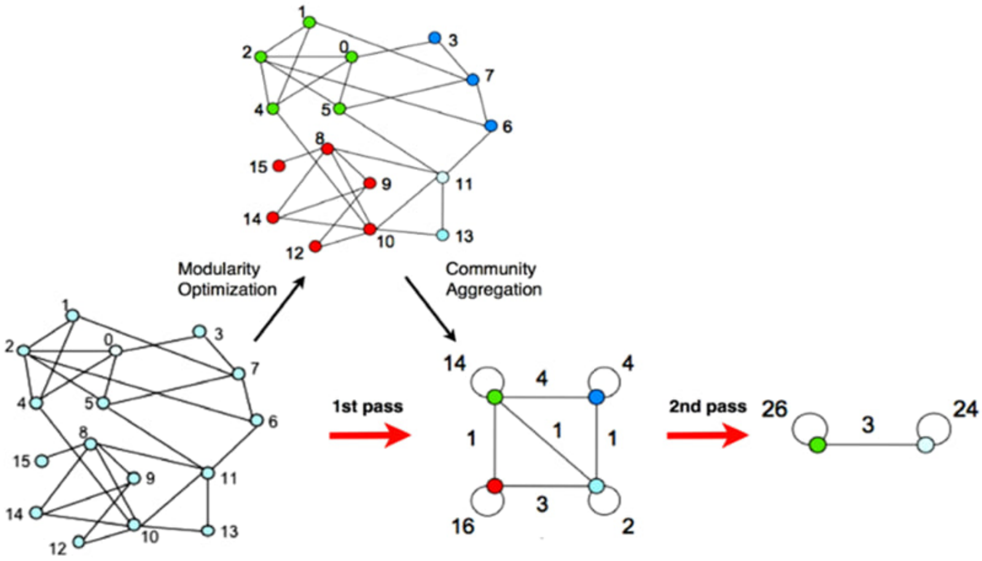

Figure 3: The illustration shows the stages involved in an overview of the Louvain Community Detection Algorithm.

to the resulting weighted network and iterated. A combination of these two phases is denoted as a 'pass.' By construction, the number of meta-communities decreases at each pass, resulting in a hierarchy of communities. [BGLL08]

Several advantages of the Louvain Algorithm fit the context of our research paper very well. In our dataset, edges represent the strength of interactions. Therefore, the Louvain Algorithm is a great fit because it can efficiently identify communities in networks with weighted connections [BGLL08].. Secondly, the Louvain Algorithm is adept at handling interaction data involving clear directions, ensuring that it accurately captures the interaction flow [BGLL08]. Finally, the Louvain algorithm performs better than many other algorithms in terms of modularity, which is crucial in identifying communities [CGar].

4.4 Resolution Adjustment and Refinement

Modularity optimization has a major disadvantage when it comes to identifying communities smaller than a particular scale [FB07]. When the algorithm encounters them, it can add these small groups to bigger communities, so the number of communities can be inaccurate. The resolution plays a crucial role in this process. It determines the level of granularity at which communities are discerned. A lower resolution value (e.g., 0.1) tends to identify larger, more encompassing communities. This implies that nodes with weaker connections may also be aggregated together. Conversely, a higher resolution value (e.g., 1.0) leads to the detection of smaller, more closely-knit communities, where nodes with stronger connections are more likely to be grouped.

As our dataset contained very unstable data points, in which unusually low values were also present, it was important to choose the relevant resolution. We systematically varied the resolution value through iterative experimentation to ascertain its impact on community detection within the dynamic face-to-face interaction networks. Notably, an enhanced resolution value of 8 yielded markedly improved outcomes: this refined resolution allowed for a more granular delineation of communities, capturing nuanced interaction patterns previously obscured. The adjustments facilitated the identification of more distinct groups and provided a deeper understanding of participant affiliations.

4.5 Setting up the process

In order to verify the validity and relevance of the Louvain Algorithm, it was applied to our dataset collected from a face-to-face interaction network.

In the context of the paper, each participant was analogous to a vertex in the algorithm. For every participant u, the algorithm scrutinized each connected participant v. It assessed the potential gain in modularity (ΔQ) if u were to shift from its current community c to v's community c' - the participant u was placed in the community $\hat{c}$ that offered the maximum ΔQ. If no positive gain in modularity was observed, u stayed within its original community (Lines 7–15, Figure 4). This process continued until no further enhancement was possible, resulting in the determination of community information (C') and modularity Q (Lines 19–22, Figure 4).

The second phase involved constructing a fresh graph wherein the vertices represent the communities identified in the prior phase. The new vertex set V' comprised the latest communities, and the edge weights between these new vertices were determined by aggregating the weights of the edges between participants in the respective communities.

In our case, edges linking participants within the same community resulted in loops for that community within the new graph (Lines 24–26, Figure 4). This iterative process was repeated until the communities stabilized. [BGLL08]

4.6 Visualizing

After applying the Louvain Algorithm, we visualized and analyzed the identified communities. This involved generating a graphical representation of the network. Each node (participant) was displayed, with edges (interactions) connecting them. This visual layout provided an intuitive overview of how participants interact. Similarly, nodes were assigned distinct colors based on the communities they belong to. This visual cue helps distinguish different groups of participants who exhibit similar interaction patterns.

4.7 Clustering

In these dynamic networks, participants' interactions are complex and nuanced. The Louvain Algorithm identifies communities based on how participants interact more frequently with each other compared to those outside their communities. However, this raw community information can still be quite intricate, especially given the nature of our data.

Clustering takes the detected communities a step further by grouping participants who exhibit similar interaction behavior into distinct and coherent clusters. This process provides a more precise, more intuitive representation. By employing clustering, we effectively organized participants into manageable groups, each characterized by shared interaction patterns.

5 Results

The Louvain Algorithm for Community Detection was applied to the dataset obtained from the game that involved face-to-face interactions. The dataset was cleaned from binary data, participants were labeled in a continuous order, and the inconsistencies, such as resolution, were addressed and maintained to yield accurate data.

The application of the Louvain Algorithm revealed a total of 34 (from 0 to 33) distinct communities within the dynamic face-to-face interaction networks. These communities exhibited varying sizes, ranging from 5 to 9 participants.

Community Order Number	Participant Number
0	[1, 2, 3, 5, 6, 7, 4]
1	[LAPTOP]
2	[176, 177, 179, 180, 181, 182, 178]
3	[183, 184, 185, 186, 187, 188, 189]
4	[190, 191, 192, 193, 194, 195, 196, 197, 201]
5	[8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
6	[198, 199, 200, 202, 203, 204]
7	[71, 72, 73, 74, 75, 76, 77]
8	[22, 23, 24, 25, 26, 27, 28]
9	[57, 58, 59, 60, 61, 62]
10	[170, 171, 172, 173]
11	[163, 164, 165, 166, 167, 168, 169]
12	[156, 157, 158, 159, 160, 161, 162]
13	[148, 149, 150, 151, 152, 153, 154, 155]
14	[142, 143, 144, 146]
15	[134, 135, 136, 137, 138, 139, 140, 141, 145]
16	[233, 234, 235, 236, 237, 238, 239]
17	[127, 128, 129, 130, 131, 132, 133]
18	[120, 121, 122, 123, 124, 125]
19	[114, 115, 116, 118, 119]
20	[108, 109, 110, 111, 112, 113]
21	[99, 100, 101, 102, 103, 104, 105, 106, 107]
22	[92, 93, 94, 95, 96, 97, 98]
23	[226, 227, 228, 229, 230, 231, 232]
24	[85, 86, 87, 88, 89, 90, 91]
25	[78, 79, 80, 81, 82]
26	[64, 65, 66, 67, 68, 69, 70]
27	[219, 220, 221, 222, 223, 224, 225]
28	[51, 52, 53, 54, 55]
29	[211, 212, 213, 214, 215, 216, 217, 218]
30	[43, 44, 45, 46, 47, 48, 49, 50]
31	[37, 38, 39, 40, 41, 42]
32	[205, 206, 207, 208, 209, 210]
33	[29, 30, 31, 32, 33, 34, 35, 36]

Table 1: Community Order Numbers and Participant Numbers

6 Conclusion

After close examination, it was observed that communities exhibited different interaction patterns. Some communities demonstrated a higher density of interactions, indicating a stronger cohesion among members. In contrast, others exhibited a pattern characterized by irregular exchanges, indicative of a more diffuse and loosely knit network structure.

Also, it should be noted that resolution 8 in the Louvain Community Detection Algorithm was the most suitable resolution level for our data.

As the results confirmed that the Louvain Algorithm, given the right resolution, can accurately detect communities in face-to-face interaction networks, we can say that this study provides a pivotal step toward advancing our understanding of dynamic face-to-face interaction network communities.

6.1 Future Prospect

Looking forward, our research can lay the foundation for a number of potential future investigations. Analyzing communities within these networks could reveal deeper insights into the dynamic nature of human interactions in the platforms with both face-to-face and virtual communication elements. A good example of further research could be studying the evolution of the communities based on their participants' face-to-face interaction in online settings such as games. Furthermore, the scope of the study can be extended to incorporate multimodal data, such as audio and non-verbal cues, so that a more holistic view of this type of interaction can be achieved. For instance, integrating speech analysis and body language recognition could provide additional insights into community formation patterns.

However, it should be noted that there are limitations in the research study: the dataset's size and diversity. Although gained insights from the provided data are substantial, a more extensive dataset covering various contexts could offer a more comprehensive account of community detection in face-to-face interactions.

Despite its current limitations, we believe that our study not only evaluates whether the Louvain Algorithm is appropriate for detecting communities in face-to-face communication networks but also contributes to the field by paving the way for further research on a wider scope.

References

[BBPSV18] A. Barrat, M. Barthelemy, R. Pastor-Satorras, and A. Vespignani. Community detection in temporal multilayer networks, revealing the social structure of face-to-face interaction. In *Multilayer Networks*, 2018.

[BCT+14] A. Barrat, C. Cattuto, A. E. Tozzi, P. Vanhems, and N. Voirin. High-resolution temporal networks of face-to-face human interactions. In *Advances in Neural Information Processing Systems*, pages 2268–2276, 2014.

[BGLL08] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, (10):P10008, 2008.

[CGP12] Michele Coscia, Fosca Giannotti, and Dino Pedreschi. A classification for community discovery methods in complex networks. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 4(5):512–546, 2012.

[CGar] P. Chejara and W. W. Godfrey. Comparative analysis of community detection algorithms. Technical report, Department of ICT, ABV-Indian Institute of Information Technology and Management Gwalior, India, Year.

[FB07] Santo Fortunato and Marc Barthelemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2007.

[For10] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

[HSS08] A. A. Hagberg, D. A. Schult, and P. J. Swart. Exploring network structure, dynamics, and function using networkx. In *Proceedings of the 7th Python in Science Conference (SciPy2008)*, pages 11–15, Pasadena, CA, USA, 2008. NASA Ames Research Center.

[Hun07] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[KBSL21] S. Kumar, C. Bai, V. S. Subrahmanian, and J. Leskovec. Deception detection in group video conversations using dynamic interaction networks. In *Proceedings of the 15th International AAAI Conference on Web and Social Media (ICWSM)*, 2021.

[KN11] Brian Karrer and Mark EJ Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107, 2011.

[KS88] David Krackhardt and Robert N Stern. Informal networks and organizational crises: An experimental simulation. *Social Psychology Quarterly*, pages 123–140, 1988.

[Lea09] David Lazer and et al. Computational social science. *Science*, 323(5915):721–723, 2009.

[Lesnd] Jure Leskovec. Dynamic face-to-face interaction networks [dataset], n.d. SNAP: Stanford Network Analysis Project. Retrieved August 2, 2023, from `http://snap.stanford.edu/data/comm-f2f-Resistance.html`.

[MMP19] F. Morone, H. A. Makse, and M. T. Pisabarro. Community detection in face-to-face proximity networks: A data-driven study. *Journal of Statistical Physics*, 175(6):1270–1285, 2019.

[New10] Mark Newman. *Networks: An Introduction.* Oxford University Press, 2010.

[NG04] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113, 2004.

[PBM20] S. Puglisi, F. R. Bullo, and A. V. Mantzaris. Stochastic block models and community detection in face-to-face proximity networks. *Applied Network Science*, 5(1):1–24, 2020.

[PLR19] D. N. Peralta, E. S. Loslever, and A. Y. Ramos. Community detection in social networks. *Applied Network Science*, 4(1):1–21, 2019.

[WF94] Stanley Wasserman and Katherine Faust. *Social Network Analysis: Methods and Applications.* Cambridge University Press, 1994.

Algorithm 1: Sequential Louvain Algorithm

Input: $G=(V,E)$: graph representation.
Output: C: community sets at each level;
Q: modularity at each level.
Var: $\hat{c}$: vertex u's best candidate community set.

Loop ***outer***
  $C \leftarrow \{\{u\}\}, \forall u \in V$;
  $\Sigma_{in}^{c} \leftarrow \sum w_{u,v},\ e(u,v) \in E,\ u \in c$ and $v \in c$;
  $\Sigma_{tot}^{c} \leftarrow \sum w_{u,v},\ e(u,v) \in E,\ u \in c$ or $v \in c$;
  // Phase 1.
  Loop ***inner***
    for *$u \in V$ and $u \in c$* **do**
      // Find the best community for vertex u.
      $\hat{c} \leftarrow \underset{\forall c', \exists e(u,v) \in E, v \in c'}{\operatorname{argmax}} \Delta Q_{u \to c'}$;
      if $\Delta Q_{u \to \hat{c}} > 0$ **then**
        // Update Σ_{tot} and Σ_{in}.
        $\Sigma_{tot}^{\hat{c}} \leftarrow \Sigma_{tot}^{\hat{c}} + w(u)$; $\Sigma_{in}^{\hat{c}} \leftarrow \Sigma_{in}^{\hat{c}} + w_{u \to \hat{c}}$;
        $\Sigma_{tot}^{c} \leftarrow \Sigma_{tot}^{c} - w(u)$; $\Sigma_{in}^{c} \leftarrow \Sigma_{in}^{c} - w_{u \to c}$;
        // Update the community information.
        $\hat{c} \leftarrow \hat{c} \cup \{u\}$; $c \leftarrow c - \{u\}$;
    if *No vertex moves to a new community* **then**
      exit ***inner*** **Loop**;
  // Calculate community set and modularity.
  $Q \leftarrow 0$;
  for $c \in C$ **do**
    $Q \leftarrow Q + \frac{\Sigma_{in}^{c}}{2m} - (\frac{\Sigma_{tot}^{c}}{2m})^2$;
  $C' \leftarrow \{c\}, \forall c \in C$; print C' and Q ;
  // Phase 2: Rebuild Graph.
  $V' \leftarrow C'$;
  $E' \leftarrow \{e(c,c')\},\ \exists e(u,v) \in E,\ u \in c,\ v \in c'$;
  $w_{c,c'} \leftarrow \sum w_{u,v},\ \forall e(u,v) \in E,\ u \in c,\ v \in c'$;
  if *No community changes* **then**
    exit ***outer*** **Loop**;
  $V \leftarrow V'$; $E \leftarrow E'$;

Figure 4: The Sequential (standard) Louvain Algorithm code template.

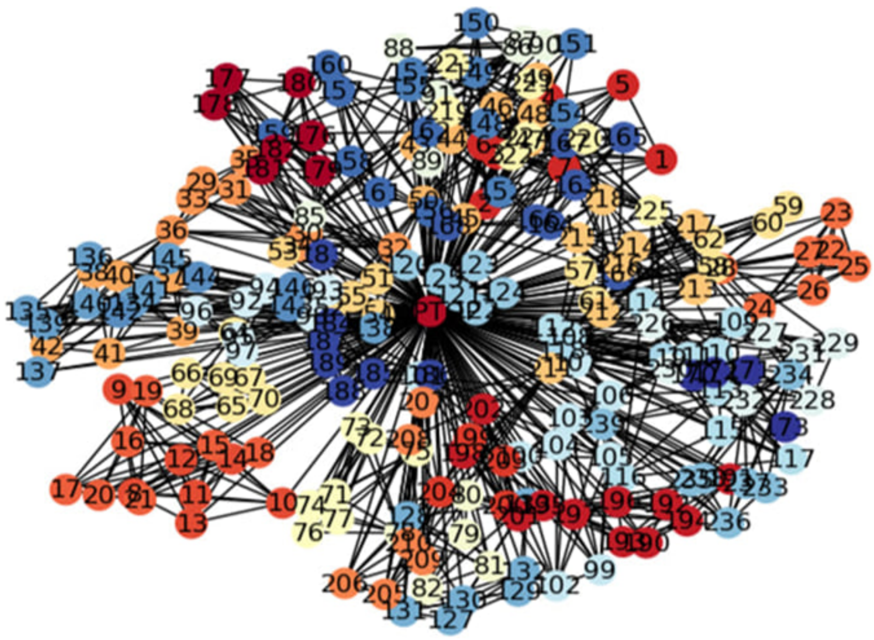

Figure 5: In this image, there is a visualization that was obtained right after applying the code and before clustering the data. Each color determines a particular community.

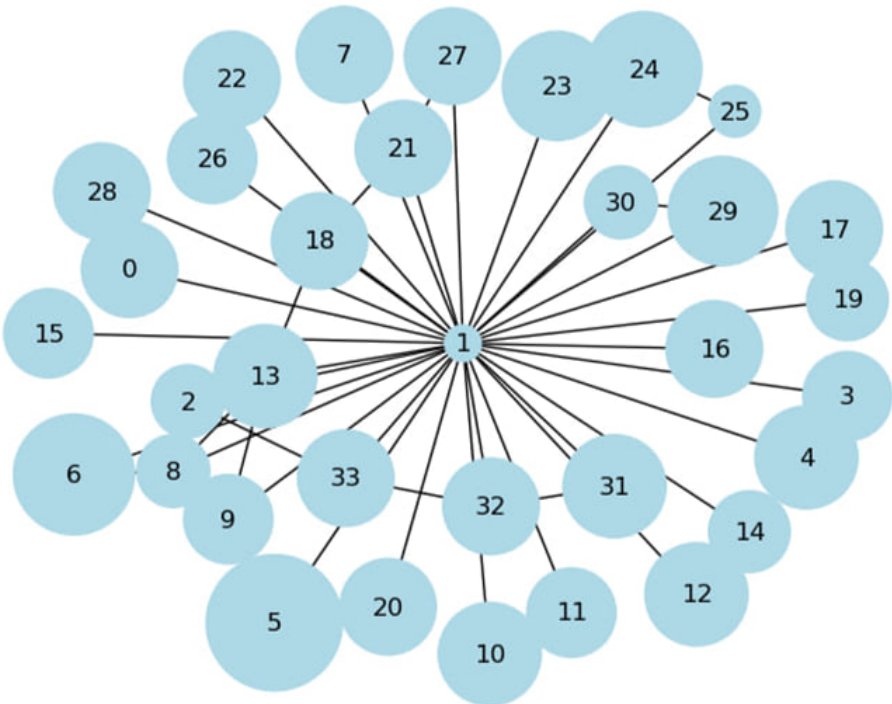

Figure 6: There, we are presented with a final graph that clearly shows all 34 communities (as they were initially grouped). This visualization was obtained after employing clustering to the data points.