Development of Overlapping Community Detection Algorithms in Temporal Networks

Mohanad A. Elagan

STEM High School for Boys - 6th of October, Egypt; mohanad.elagan1@gmail.com

Mentor: Dr. Patrick Shepherd - Berea College; shepherdp@berea.edu

Reviewers: Prof.Dr. Elnomery Zanaty - the dean of faculty computer and information at Sohag Univerisity

Prof. Hossam Abdelsalam - Menofia University and The Applied College

Prof. Hani Ibrahim - Taibah University

Prof. Abdelalim Farag - Bow Valley College

Abstract

The difficulty with approaches for overlapping community detection is effectively mining both topological and temporal structures. Community and change point detection are the most effective approaches for studying such developing structural relationships. Therefore, we propose an algorithm that uses the so-called change points within communities to find overlapping community structures. The suggested technique combines two methods that can effectively identify stable communities at various time scales, in contrast to existing dynamic community identification approaches. The first method involves adding a community to the given dynamic graphs to detect communities on a multi-temporal scale. In order to track the expanded community and improve the algorithm, the second method involves computing the normalized mutual information (NMI) scores and plotting the found communities in conjugate with the first method. The efficacy of the technique is tested on high-resolution time-varying and artificial contact networks made from existing social networks.

Keywords: overlapping community structure, network science, dynamic networks, community detection, temporal networks

1 Introduction

Interacting with people has become more well-facilitated in the society we currently live in than any other time in history. Large-scale social networks like Facebook and Twitter have enabled people to engage with one another easily, quickly, and often profoundly. Often, interactions in these networks are represented by *graphs*, in which a social network's users are represented as *nodes*, and their communications with one another are represented as *edges* between them, as depicted in Figure 1.

A *community* is a collection of nodes with a high frequency of interactions. The number of communities a person may join on online social networks is practically limitless because a person can concurrently connect with as many groups as he wants, such as the millions of people who follow a research institute on Facebook. Such networks exhibit fluctuations in community membership on an annual, monthly, daily, hourly, and even shorter time scale. Therefore, analyzing communities at several temporal scales is a necessary approach to understanding networks of interactions. This approach depicts real-time interactions that can accurately capture the underlying temporal information.

Several snapshots of the network, or timestamps, are used by algorithms that identify communities in topologically dynamic networks. These algorithms consider the network's changes that naturally occur between successive timestamps, and from there, they also detect communities. The majority of algorithms are designed to find disconnected groups. Since communities in online social networks may naturally overlap, limiting potential communities to those that are disjoint produces results that are similarly limited. This paper presents a technique that draws inspiration from change point detection and the study of dynamic communities or groupings of nodes that maintain a cohesive community throughout time at a particular temporal scale.

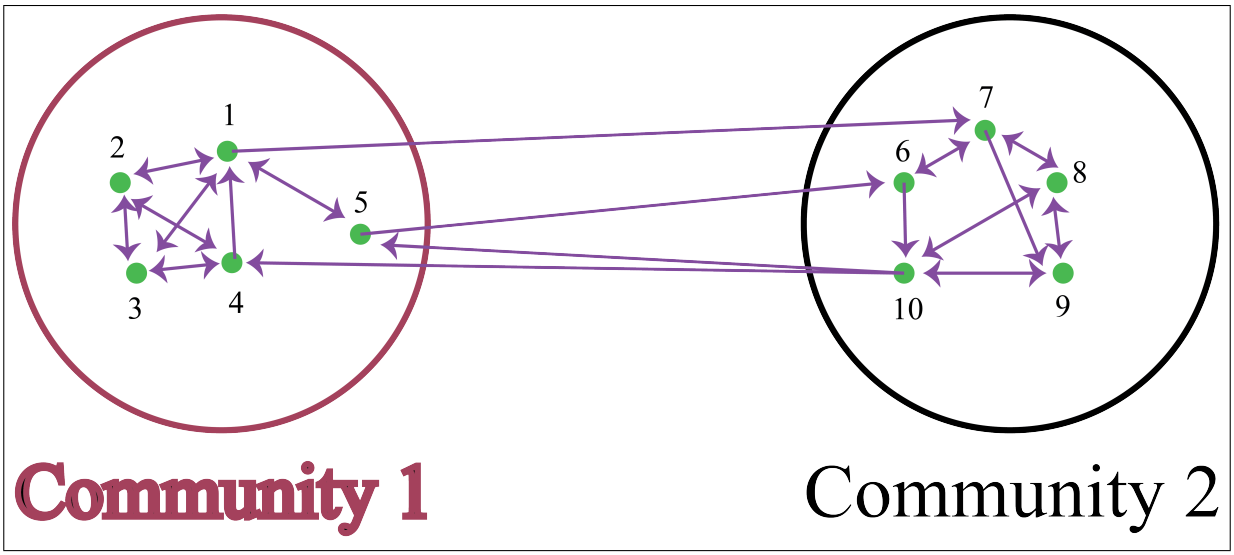


Figure 1: Ten nodes comprising 2 communities, with five nodes each.

2 Related Work

2.1 Community Detection

Research on community detection has been ongoing, and several community detection techniques exist. The Louvain technique for community recognition offers a quick algorithm for locating communities in static networks [2]; however, it cannot, without modification, identify overlapping communities. Another method that has the benefit of locating high-quality communities more quickly is the Leiden [15] community detection method. Leskovec et al. [8] define the *conductance* (the ratio of the number of edges between nodes inside the community and nodes outside the community to the sum of the degrees of the nodes inside the community) of a set of nodes S in a static graph G ($G = (V, E)$) as:

$$\phi(S) = \frac{\sum_{i \ in S, j \notin S} A_{i,j}}{min(A(S), A(\bar{S}))}$$

where A is the adjacency matrix of graph G and

$$A(S) = \sum_{i \in S, j \in V} A_{ij}$$

to represent a community's quality. $A(S)$ is the sum of all the points in S. The index of the summation is $i \in S, j \in V$, where it carries over A_{ij}.

2.2 Change Point Detection

The change-point detection method was first introduced by Peel and Clauset, in which they develop networks using generative network models and statistical hypothesis testing [12]. It looks for instances when a stochastic process or time series' probability distribution changes, and it pinpoints moments when one model transforms into another (e.g., a change in the mean value). In a recent study, Masuda et al.[9] employed hierarchical clustering and graph distance measurements to pinpoint system state dynamics sequences. A hierarchical change point identification technique was also put out by Wang et al. [16] to identify intra-community and inter-community development.

The proposed approach varies in two ways: first, it looks for stable individual communities rather than stable graph intervals, and second, it looks for stable structures at various temporal granularity levels.

2.3 Louvain Algorithm

The Louvain Method [2] of community detection begins with the original network, with each node representing a separate community of size 1, and then finds the single-node aggregation that maximizes the network's modularity score. Modularity is a measure of how well a set of communities describes the structure of connections in a network. Although the Louvain algorithm can locate high-quality clusters in most networks, this method has a few significant drawbacks. Since the Louvain method continually switches nodes between clusters, eventually, it could switch a bridge node to a different cluster, causing the original cluster's link to fail. Therefore, it may assign nodes to clusters incorrectly.

3 Methodology

The approach aims to effectively and precisely identify communities of various sizes without redundancy. A three-stage procedure is employed at each of various temporal scales: seed finding, pruning, and expansion.

We define a stable dynamic community by the triplet

$$c = (N, p, \gamma)$$

where $c.N$ is a list of the community's nodes, $c.p$ is an interval representing the community's existence, and $c.\gamma$ is the temporal granularity. N stands for the distinct nodes and p for the probability of edge existence.

First, the interval $c.p$ is used to create a sequence of snapshots of interaction graphs over varying time periods. These snapshots are then used to find a set of seeds S to be used in a community detection algorithm. The original set of seeds is then filtered into a new seed set FS containing only those seeds that developed relatively high quality communities over multiple time periods. Finally, a process called seed expansion is used to assess the quality over time of these filtered seeds.

3.1 Seed Finding

A seed node is a particular node that permits the integration of new nodes to the network; different detected communities are the result of various choices of seed nodes. High-quality seeds for community detection are nodes that can detect communities that are similar to the real-world communities. The search for intriguing seeds is conducted for each time scale. Starting from time t_0, a cumulative graph (snapshot) is constructed for each period

$$[t_0, t_0 + \gamma], [t_0 + \gamma, t_0 + 2\gamma], [\dots]$$

where granularity is defined as a time, until every interaction is a part of a single cumulative graph. $G_{t_0,\gamma}$ is a cumulative snapshot of link stream G for the period commencing at t_0 and of duration γ. The expression $c.p = [t_1, t_2]$ indicates that the community c exists from t_1 to t_2. G_γ is a list of every single such graph. This approach essentially produces a series of static graphs.

Given a static community detection algorithm CD that produces a collection of communities and a function to evaluate community quality CQ, CD is applied to each snapshot and uses CQ to select potential seeds. A collection X of random sets of seeds are picked and tested on each snapshot g in the cumulative graph. As a result, the definition of the set of seeds S is as follows:

$$S = \{s : s \in X; \forall g \in G_\gamma, c \in CD(g, s), CQ(g, c) > \theta_q\}$$

θ_q is a cutoff for a community quality, and g is a single snapshot.

S is supposed to be a constant set of nodes that produces high quality communities across time steps. Moreover, it is crucial for scalability because community detection at each stage may be conducted in parallel on all phases as each step's implementation is independent.

It should be noted that adopting an algorithm that produces communities with high quality in accordance with the chosen CQ would be significant; nevertheless, this method is employed because it is necessary to use the most common algorithms and quality functions in order to demonstrate the general approach.

3.2 Seed Pruning

The multi-scale community detection process is sped up, and redundancy is reduced through the seed pruning stage. The less interesting seeds are removed using a measure of structural similarity CSS. We develop a set FS of filtered seeds as follows. FS begins empty, and we progressively fill it by examining the seeds discovered at increasing time granularities. For each granularity γ we collect a set of discovered seed sets S_γ. For each individual seed set $s \in S$, we let $FS = FS \cup \{s\}$ if s for which $\nexists s' \in S : CSS(s.n, s'.n) > \theta_s$ to avoid redundancy.

The interval corresponding to the snapshot at which this seed was found is denoted by $s.p$, and a threshold of similarity is denoted by θ_s; it is the minimum degree of similarity between two data records in the same cluster.

The CSS approach calculates how similar two groups of nodes are to one another. FS is the set of all seed sets s such that, for every community c that was detected, either s and c have a Jaccard index exceeding $theta_s$ or s and c occupy time intervals that do not overlap. The Jaccard Index is used as a reference function [5]:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

3.3 Seed Expansion

This step determines if a seed belongs to a dynamic society. Since the dynamic community detection field's infancy, the instability issue has been recognized [1]. It implies that after being subjected to small random alterations, the same algorithm will usually provide different outcomes when run repeatedly on the same network. As a result, it is unknown if the variations through multiple realizations in community structure on the interval $[t, t+1]$ result from algorithmic structural changes. Smoothing methods are typically introduced to address this issue [13]. The technique takes a similar approach, but instead of comparing communities discovered at step $[t, t+1]$, it recursively determines if a community established at step t is relevant.

As long as the quality $\mathrm{CQ}(s.N, G_{t_i,\gamma})$ of the community formed by the nodes $s.N$ on the graph at $G_{t_i,\gamma}$ is sufficient, it increases the duration of each seed $s \in FS$, where $s.d = [t, t+\gamma]$ discovered on the graph $G_{t_i,\gamma}$, at each step t_i in both temporal directions.

$$t_i \in (\dots [t-2\gamma, t\text{-}\gamma], [t\text{-}\gamma, t], [t+\gamma, t+2\gamma], \dots)$$

The extended seed is is appended to the set of filtered seeds if its ultimate duration period $|s.p|$ exceeds a duration $\theta_p\gamma$, where θ_p is a stability criterion. If not, it is eliminated. Algorithm 1 formalizes this action.

Algorithm 1 Forward seed expansion. A seed's forward temporal expansion is discovered at the moment of granularity γ.

Input: $s, \gamma, \theta_p, \theta_s$
$t \leftarrow t^{start}$
$s.p = [t^{start}, t^{end}]$
$g \leftarrow G_{t,\gamma}$
$p \leftarrow [t, t+\gamma]$
while $CQ(s.N, g) > \theta_s$ **do**
  $s.p \leftarrow s.p \cup p$
  $t \leftarrow t + \gamma$
  $p \leftarrow [t, t+\gamma]$;
  $g \leftarrow G_{t,\gamma}$
end
if $|s, p| \geq \theta_p\gamma$ **then**
  $D \leftarrow D \cup \{s\}$
end

Seeds are considered in descending order of their CQ score to choose the most pertinent stable communities. A community of the lowest quality may be pruned by a community of best quality at the same granularity γ because of the pruning technique.

3.4 Iterative Process at Multiple Scales

The array of examined scales is created in order to identify communities at various scales Γ. The maximum scale is denoted by the expression

$$\gamma^{max} = \frac{|G.d|}{\theta_p}$$

where $|G.d|$ represents the duration of the dynamic graph as a whole. The sorted list is what defines Γ the set of time scales under consideration as:

$$\Gamma = [\gamma^{max}, \frac{\gamma^{max}}{2^1}, \frac{\gamma^{max}}{2^2}, \ldots, \frac{\gamma^{max}}{2^k}]$$

The parameter k, which has the form

$$\frac{\gamma^{max}}{2^k} > \theta_\gamma \geq \frac{\gamma^{max}}{2^{k+1}}$$

corresponds to the temporal granularity to assess and is hence data-dependent; algorithm 2 formalizes this action.

Algorithm 2 Iterative Process at Multiple Scales. The threshold settings are θ_p, θ_γ, θ_q, and θ_s, and G is the connection streams to be analysed.

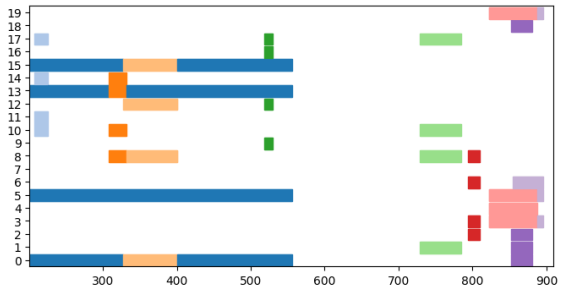
Input: $G, \theta_p, \theta_\gamma, \theta_q, \theta_s$
$D \leftarrow \{\emptyset\}$
$\Gamma \leftarrow$ studiedScales(G, θ_γ)
$p \leftarrow [t, t + \gamma]$
for $\gamma \in \Gamma$ **do**
  $S \leftarrow$ seedFinding$(\gamma, CD, CQ, \theta_q)$
  $FS \leftarrow$ seedPruning(S, CSS, θ_s)
  for $s \in FS$ **do**
    seedExpansion$(s, \gamma, \theta_p, \theta_s)$
  end
end

4 Results and Tests

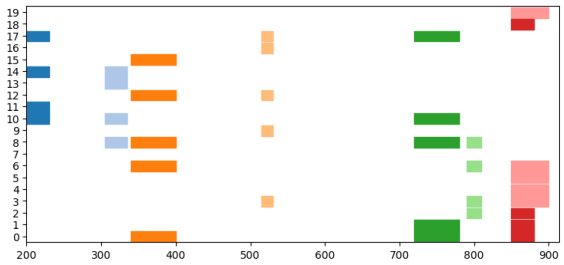
There are two different types of tests: on actual networks, where both qualitative and quantitative assessment were used to verify the approach, and on synthetic data, where planted ground-truth is used to compare the findings numerically.

4.1 Parameters

1. The value of θ_s establishes the point at which two communities are deemed redundant. The more communities that are obtained, the higher this value is. In all experiments, θ_s is set to 0.3 [0 to 1 is the best scale.]. It depends on the CSS function that is selected.
2. θ_γ depends on the data. It is set to 1 for synthetic networks and 20 seconds for the SocioPatterns dataset [14] (the minimum length of time required to capture a contact).
3. A seed must be expanded for at least θ_p consecutive periods in order for it to be regarded as a stable community. In every experiment, θ_s is set to 3. The number should not be lower to prevent erroneous detections brought on by random chance. (if it was selected to be 2, for instance, there will be a risk to have errors while the detection process because the random chance will be higher.) To reduce the number of results, choose higher values.
4. The minimum quality for a seed to be kept and grown is determined by θ_q. In all experiments, θ_q is set at 0.7, which is the minimum quality [0.7 to 1 is the best scale].



(a) Plotting the communities on the generator to be seeded. communities.



(b) The suggested approach led to the discovery of stable communities.

Figure 2: Comparison of the communities that were seeded and those that were found. Nodes on the vertical axis, time steps on the horizontal axis. Colors are selected at random and relate to communities. The majority of communities are detected accurately in terms of nodes and duration.

4.2 Verification of Synthetic Data

An Erdos-Renyi random graph[4] is formed at each step with p, considering the dynamic network's T stages and N. Then, a certain SC of randomly selected stable communities is included. A set of

$$n \in [4, \frac{N}{4}]$$

nodes we start with, a duration

$$d \in [10, \frac{T}{4}]$$

and a commencement date

$$s \in [0, T - d]$$

are assigned at communities, then expanding the number of edges to match that. In order to increase variability within the community, n and d are selected with a logarithmic probability. The likelihood of witnessing an edge between any two of the community's nodes during its existence, set at $\frac{10}{d}$, determines the community's temporal scale. Therefore, a community with a lifetime of 10 will have edges between all its nodes at every step, but a community with a duration of 100 will only have edges between any two nodes on average every ten steps.

The problem is contrasted with a baseline: communities are detected using a static method on each window for various window sizes because there is no technique to detect communities at numerous temporal scales. However, the current method will be able to do estimations for detecting communities at different scales. As specified in [7], the overlapping NMI is calculated at each phase to compare the outcomes. The NMI between our communities and communities found by another dynamic community detection algorithm is computed. T is set to 5000, N to 200, and p to $\frac{20}{N}$ for those tests. SC is a variable set.

The synthetic communities to be discovered for SC = 10 are shown in Figure 2. Table 1 presents the comparison's findings together with baselines. In terms of average NMI, it can be seen that the suggested strategy performs better than the baseline at every scale in every situation.

4.3 Verification of Real Datasets

In order to confirm the findings, two real datasets are used. In order to find patterns in dynamic datasets and compare the features of the communities revealed with those of other current algorithms, a quantitative technique is utilized to examine the method's scalability. The following datasets were utilized:

- Math overflow: an extensive network to assess scalability using the stack exchange interaction dataset [11]. (506550 interactions and 24818 nodes)
- SocioPatterns: data from primary schools[14], interactions between students in person (125773 interactions and 323 nodes)

Table 1: Comparison between the average NMI scores obtained using the suggested approach and each temporal scale.

$t_{scale}(\gamma)$	5	20	40
Proposed	0.91	0.69	0.62
1666	0.41	0.24	0.15
416	0.39	0.36	0.32
104	0.47	0.44	0.45
26	0.35	0.38	0.42
6	0.17	0.19	0.20
1	0.05	0.04	0.05

4.3.1 Quantitative Evaluation

The aspect used in the quantitative evaluation to assess the outcomes is scalability:

- The Mucha et al. [10] approach of detecting community structure in time-dependent multiplexed and multi-scale networks. A customized implementation was employed, and each snapshot's detection was carried out using the networkx library's implementation [6].
- The Identify and Match approach was introduced by Greene *et al.* [5]. The approach is a model for following the evolution of communities through time in a dynamic network, where each community is defined by a number of key evolutionary events; it serves as motivation for a community-matching technique that effectively finds and monitors dynamic communities. This approach was used to measure the θ_q by adding it to the proposed algorithm. With a 0.7 minimum similarity criterion, it was developed using the Louvain technique for community discovery and the Jaccard coefficient to match communities. With this technique, community detection was shared.

The community detection step for Identify and Match, D-CPM, and the suggested technique are carried out concurrently for all snapshots. Mucha et al. cannot do it since the procedure runs concurrently on all snapshots. In addition to finding communities, it must also prevent overlap across communities over a range of temporal periods. **However, the approach scales to networks with hundreds of thousands of interactions and thousands of nodes. Although it is slower than the Identify and Match strategy, it does not have the same scaling issues as the other two.**

Table 2 summarises the number of communities discovered using each approach and each community's persistence, size, stability, density, and Conductance. It can be shown that the other approaches produce far more communities than the proposed method, generally at the cost of overall inferior quality. Analyzing the data further reveals that alternative approaches produce several noisy communities on a single snapshot for unprocessed methods and frequently with low density.

Table 2: Average community characteristics for each approach #Communities: total number of communities discovered. The number of successive pictures is persistent. Size is measured in nodes. Stability is measured by the average Jaccard coefficient between nodes in the same community over time. Average degree size is the density. Q = 1 - Conductance.

Method	# Communities	Persistent	Size	Stability	Density	Q
Proposed	179	3.44	**10.89**	**1.00**	**0.50**	0.91
CD/MATCH	**29846**	1.21	5.50	0.97	0.42	**0.96**
MUCHA	1097	**15.48**	9.72	0.62	0.38	0.85
CPM	3259	1.87	5.37	0.51	0.01	0.53

4.3.2 Qualitative evaluation

The SocioPatterns partnership employed RFID technology to collect elementary school data [14]. At a pace of one capture every 20 seconds, they record people wearing them in close proximity to one another. The data was gathered

over two days in October 2009 and included direct encounters between 10 instructors and 323 students. This school has ten courses, including five levels, with each level having two classes.

The visualization was confined to one day and the analysis to four classes to provide an appropriate study of the data (1B, 2B, 3B, 4B). On this dataset, 120 communities have been found in total. Three alternative numbers were produced, each representing a different community length: less than a half-hour, between a half-hour and two hours, and longer than two hours. Figure 3 shows the outcomes. The following observations can be made:

- The majority of medium-duration communities take place at lunch. That can be explained by the fact that kids share a playground and a canteen. Only two or three classes have breaks, and lunches are eaten in two consecutive rounds of an hour since the playground and canteen are not big enough to accommodate all the pupils at once. The data show that around half of youngsters return home for lunch [14], and some kids do not belong to any communities during the lunch hour.
- Communities with longer durations have a definite relationship with the class system.
- Some communities include students from various classes during class breaks and lunchtime, as demonstrated by the dark green community around lunchtime or the pink community around 10:00 for short communities when classes 2B and 3B are likely taking a break at the same time. That demonstrates how a study performed at coarser scales alone can be deceptive since it only identifies the more robust class structure while excluding the communities across classes for shorter periods. The majority of communities with a shorter lifespan are discovered during what are likely breaks in class.

5 Conclusion

The algorithm finds communities that remain unchanged over different timescales, rather than finding initial communities, and then tracking their evolution at multiple temporal scales. Given fast-growing social networks, analyzing communities using dynamic algorithms is challenging. The method demonstrated finds overlapping community structures in dynamic networks. The algorithm finds communities that remain unchanged over different timescales rather than finding initial communities and then tracking their evolution at multiple temporal scales. Given fast-growing social networks, analyzing communities using dynamic algorithms is challenging. The method demonstrated finds overlapping community structures in dynamic networks. The initial method put out involved finding structures that had been verified on both synthetic and real-world networks. Then it computes the average NMI at each step in the process. Finally, it plots the found communities with different duration to give accurate results. The technique is presented as a generic, expandable method.

6 Future Work

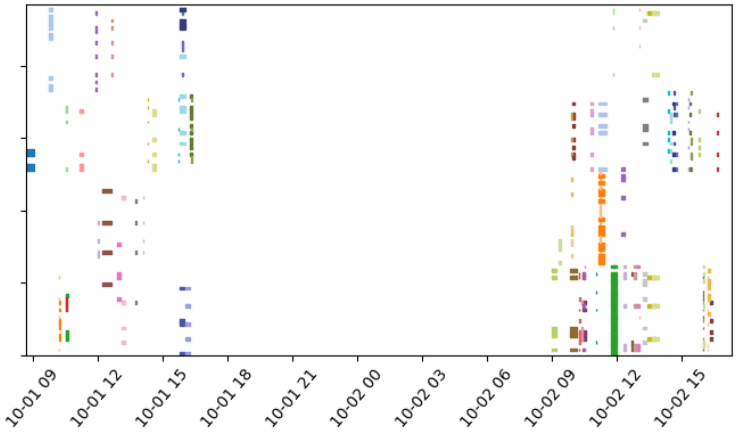
The general approach can be applied to other methods. For instance, it is intended to improve and expand the approach to assessing the function and significance of nodes in dynamic communities using algorithms like Clauset-Newman-Moore and Leiden. Statistical selection procedures may also be used to lessen the computing complexity.

6.1 Clauset-Newman-Moore Algorithm (FastGreedy) [3]

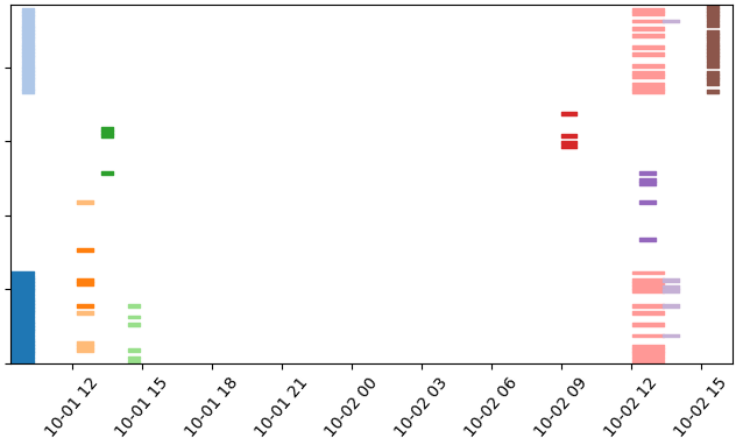
The process is started with the original network, with each node representing a separate community. The algorithm then begins merging communities, choosing two communities to combine each time, producing the greatest improvement in graph modularity. The method terminates, and a dendrogram is produced when merging any two communities no longer results in a positive gain in modularity. This technique might not always produce the best results compared to other algorithms since it takes the biggest modularity gain; each loop does not ensure that a global optimum will be attained. According to an analysis by Yang et al. [17], the time complexity of the algorithm is $O(Nlog^2(N))$. That indicates that while Fast Greedy is incredibly time-effective, there is less assurance regarding the quality of communities produced.

6.2 Leiden Algorithm

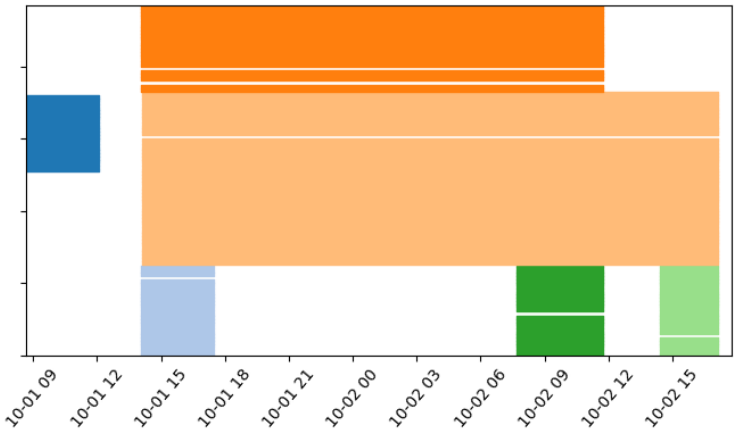
The algorithm is based on the Louvain algorithm, but in addition to merging clusters, it also can separate clusters. It ensures cluster connectivity by breaking clusters in a certain way. Additionally, the approach ensures that it will



(a) Second day, length < 30 minutes.



(b) Second day, 30 minutes $<$ length < 2 hours.



(c) Second day, length > 2 hours.

Figure 3: Long-lasting communities on the SocioPatterns Primary School Dataset. Children are on the vertical axis while time is on the horizontal axis. Colors are assigned at random.

acquire a subset of optimum clusters if continually executing the procedure. As a result, switching one or more nodes from one cluster to another cannot enhance the quality of the clusters, which exists in the Leiden algorithm.

7 Source Code

The entire Python implementation is available at https://github.com/Mohanad-Elagan/Development-of-Overlapping-Com-Detection-in-tnetwork.

References

[1] Thomas Aynaud and Jean-Loup Guillaume. Static community detection algorithms for evolving networks. In *8th international symposium on modeling and optimization in mobile, ad hoc, and wireless networks*, pages 513–519. IEEE, 2010.

[2] 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*, 2008(10):P10008, 2008.

[3] A. Clauset, M. E. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70:066111, 2004.

[4] P. Erdos. R enyi, a.: On random graphs i. *Publicationes Mathematicae Debrecen*, 6(290):297, 1959.

[5] D. Greene, D. Doyle, and P. Cunningham. Tracking the evolution of communities in dynamic social networks. In *2010 international conference on advances in social networks analysis and mining*. pp. 176 (183). IEEE, 2010.

[6] A. Hagberg, P. Swart, S. Chult, and D.: Exploring network structure. *dynamics, and function using networkx*. Tech. rep, Los Alamos National Lab.(LANL), Los Alamos, NM (United States, 2008.

[7] A. Lancichinetti, S. Fortunato, and J. Kertesz. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):033015, 2009.

[8] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6:29–123, 2009.

[9] N. Masuda and P. Holme. Detecting sequences of system states in temporal networks. *Scientific Reports*, 9(1):1, 2019.

[10] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J. P. Onnela. Community structure in time-dependent, multiscale, and multiplex networks. *Science*, 328:5980, 2010.

[11] A. Paranjape, A. R. Benson, and J. Leskovec. Motifs in temporal networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. pp. 601 (610). ACM, 2017.

[12] L. Peel and A. Clauset. Detecting change points in the large-scale structure of evolving networks. *CoRR abs/*, 1403:0989, 2014.

[13] G. Rossetti and R. Cazabet. Community discovery in dynamic networks: a survey. *ACM Computing Surveys (CSUR)*, 51(2):35, 2018.

[14] J. Stehle, N. Voirin, A. Barrat, C. Cattuto, L. Isella, J. Pinton, and M. Quaggiotto. Van den broeck, w., regis, c., lina, b., vanhems, p.: High-resolution measurements of face-to-face contact patterns in a primary school. *PLOS ONE*, 6:8, 2011.

[15] V. Traag, L. Waltman, and N. J. van Eck. From louvain to leiden: guaranteeing well-connected communities, arxiv. preprint, 2018.

[16] Y. Wang, A. Chakrabarti, D. Sivakoff, and S. Parthasarathy. Fast change point detection on dynamic social networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, AAAI Press, 2017. pp. 2992 (2998). IJCAI'17.

[17] Z. Yang, R. Algesheimer, and C. J. Tessone. A comparative analysis of community detection algorithms on artificial networks. *Scientific Reports*, 6:30750, 2016.