Abstract:

Social networks have a special structural property called clustering or community structures, without clear and widely accepted definition. A lot of algorithms were proposed to detect these communities, and their applications go from the apparent E-Commerce to the not-so-apparent Criminology. Most of these algorithms are computationally intensive, while others suffer from low accuracy and non-determinism. This paper introduces an improvement over the recently proposed local algorithm, LCD-SN, by proposing the Improved Local Community Detection in Social Networks (iLCD-SN) Algorithm. The iLCD-SN is built based on first- and second-degree neighborhood evaluations and dynamic node importance for ranking nodes, thus balancing the effects of both direct and indirect neighbors through adjustable parameters α and β. It employs a dynamic approach to community building, finding high-quality communities without relying much on the initial parameters. The proposed algorithm captures most of the local techniques' deficiencies, such as ambiguity, resolution constraints, and lack of stability. In addition, the visualization of the identified communities shows the ability of the algorithm to expose significant structures intuitively, which verifies its functionality.

INDEX TERMS: Community structure detection, Overlapping nodes, modularity, nodes ranking, real work networks ,social network analysis

Introduction:

In practical applications, networks are actually represented by graphs, where nodes represent entities and edges represent the relationships that exist between them. Formally, a graph G=(V,E)is defined by a set of vertices V={v1​,v2​,...,vn​} along with edges E⊆V×V [1]. One of the fundamental properties of many real-world networks is the clustering property: nodes are grouped into closely interconnected groups, which are called communities. The nodes belonging to the same community usually have high similarity, whereas nodes coming from different communities show weaker connectivity [2]. Since there is no widely accepted definition of what a community is, the algorithms for community detection use different methods, and it often leads to different outputs. The methods applied in finding communities are broadly categorized into global and local approaches, each having its advantages and constraints.

Analyzing complex networks of many kinds, from social networks to biological and communication ones, can help understand better the structure and behavior of interconnected systems. Network science still includes community detection as one of its core challenges – finding groups of densely connected nodes. Such findings provide critical insights into the structure of the network, showing how meaningful patterns, relationships, or functional modules are hidden. This paper introduces a new method called the improved local community detection with similarity-based neighborhoods algorithm. This method has improvements over previously developed methods by including dynamic importance metrics along with better strategies for resolving overlaps and merging weaker communities. The current community detection methods, such as modularity optimization-for example, the Louvain algorithm-and hierarchical clustering suffer from limitations such as the resolution limit and difficulties in handling overlapping communities. Furthermore, many techniques could not effectively capture the inbuilt local and global interaction within networks. To better overcome these challenges, improved LCDSN proposes a heterogeneous approach that integrates dynamic importance spread with structural improvement by utilizing multi-layered community upgrading. This algorithm relies on an in-built dynamic importance (IMP) score, which spreads with iterations over the network as well. This is one of the importance scores that the value takes into consideration when combining both first-degree and second-degree neighbors under controlled parameters of alpha (α) and beta (β). Alpha and beta enable the algorithm to weigh direct versus indirect neighbors so that it may just assign nodes that exert much influence within the network greater importance scores. Thus, the algorithm creates an initial community by aggregating high-importance nodes with their neighborhoods. The improved LCDSN algorithm provides several improvements that aim at overcoming the limitations of the standard methods: overlap resolution is achieved by integrating the general form of the Leicht-Holme-Newman similarity, which allows identifying nodes that belong to several communities and their optimal assignment based on structural similarity; weak communities are merged with larger, more structurally dense communities based on the calculated similarity scores so that loose groups do not endanger the robustness of the network; isolates or small cliques are merged into larger, better-matched communities; this results in a better overall robustness of the final community partition. In order to evaluate the quality of communities detected, several metrics are computed here, including modularity-a well-known metric that measures the strength with which communities partition the network compared to a null model. The improved LCDSN algorithm achieves higher modularity scores than the other two LC-based algorithms while maintaining a good performance at detecting communities even for challenging conditions. Toward this end, we are implementing the improved LCDSN algorithm on a ubiquitous benchmark, Karate Club, used in most contemporary analyses within the community detection sphere to show the effectiveness of an improved LCDSN methodology. The results from such calculations provide a higher modularity together with more accurate identification of communities compared to existing techniques based on graphical representations. In summary, the LCDSN algorithm offers a robust and flexible framework for community detection. Its novel elaboration of dynamic significance measures, overlap management, and weak community amalgamation positions it favorably to practical networks in which the functions of nodes could overlap or their influence could vary. This scalability and adaptability of the algorithm provide prospects for subsequent studies, particularly over dynamic and heterogeneous networks in which community structures are considered to evolve over time.  
We demonstrate that this new LCD-SN algorithm outperforms its predecessor in most of the cases, with higher modularity and accordingly better ability to detect communities, from the examination of a few empirical networks, including Karate, Dolphins, GD01\_b, PolBooks, and NetScience. This makes the algorithm more robust in handling a larger class of social network structures, bringing the analysis of large complex networks within scalable and effective means.

RELATED WORKS:

The study of community detection in social networks started off with the earliest approaches, which consisted of genetic algorithms and label propagation techniques aimed at uncovering hidden structures by mimicking evolutionary principles and network diffusion. However, as the networks grew larger and more complex, these methods became limited to issues of overlapped communities, computational complexity, and difficulties in handling large amounts of data. Scholars soon realized that reliance on network architecture alone was not sufficient and hence began to include node attributes-such as content similarity-in their approaches. This shift towards local community detection and distributed structures marked a major advancement that opened avenues to even more scalable and applicable solutions. However, despite such developments, the need for algorithms that could handle overlap nodes and fragmentation continued to remain strong, thus further sustaining the research efforts.

It indicates that the network community detection field has been much improved and tackled challenges from directed to undirected graphs. The Foundational Contribution of Malliaros and Vazirgiannis surveys clustering in directed networks to be specified in adequately handling the asymmetric relationships noted within the domains of sociology and biology. Some early works on undirected networks emphasize internal node cluster connection density to understand the more complex behaviors of a network. Algorithms include Girvan-Newman that removes the highest betweenness edges, iteratively separating out clearly distinct communities within a network. Approaches that introduced the concept of modularity as a measure of community quality continue to represent the core ideas in this field today. Hybrid algorithms that combine several techniques, such as Cuckoo Search with Genetic Algorithms, improve accuracy by balancing the exploration and exploitation of potential community structures: recent techniques, such as the Harris Hawks Optimization, show very promising results for social networks.

Kernighan-Lin and Betweenness Centrality are classic algorithms that have been applied in optimizing community partitions by minimizing inter-subset edge costs. Further refinement was brought by hierarchical methods such as Agglomerative Clustering that clusters nodes with a high similarity score and adapts to different network granularities. The Louvain Method is unique because it optimizes modularity in an iterative manner and thus, has been widely applied for large-scale networks as it is fast and scalable. Also, genetic algorithms, like Edge-based Encoding, give strong performance across all datasets with minimal pre-configurations. Techniques like Object Migration Automata (OMA) and Chaotic Maps improve the diversity of the search in order to detect more accurately, in complex networks, by balancing global optimization strategies with local search strategies.

Recent advances in the LPAs suggest that near-linear time algorithms can be efficiently used for large networks without preassigned community structures. They shall stabilize their outcome by taking a low-significance starting nodes choice and refining using the kind of indices such as the Adamic/Adar index that are core-centric from novel methods like LBLD, Local Balanced Label Diffusion, increasing scalability yet improving precision, and scaling up the communities from significant nodes. The hybrid method combining label propagation with game-theoretic approaches improves the stability of detection, and the local detection algorithms that focused on high-importance nodes, such as the core expansion algorithm, yield high modularity and a well-defined community structure, particularly in large network sizes.

Another popular study area is the integration of structural and attribute information for community detection algorithms. For example, Importance Score-based clustering integrates structure cohesiveness with nodal attributes for improved cluster accuracy. Ensemble Clustering Techniques, basing its approach on influential nodes as community centers, attain small improvements in modularity. Deep learning methods include Point-wise Mutual Information and Deep Clustering Frameworks. Such aspects have improved the ability in relation to high-order properties within data. When having an attributed graph, its capturing is a more interesting application. SNMF Approaches in embedding attribute similarity gain an improvement, especially considering sparsity and noise in vast networks. Together, they show the growing ability of community detection methods to handle the complexity of large-scale network structures in today's world.

Addressing these issues, a novel remedy in the form of an algorithm called Local Community Detection in Social Networks (LCD-SN) [8] has been developed, focusing on the local characteristics of nodes and their neighbors. Unlike typical approaches that study the entire networks, LCD-SN employs a multi-phase approach that starts with the calculation of an Importance Metric (IMP). This measure assesses a node's influence based on the immediate neighbors, both of first and second degree, thereby guaranteeing that nodes with the most influence are given priority in the initial stages of community building. Comprising this analysis locally, the algorithm lowers the computational costs and facilitates scalability, making it highly suitable for large networks. An important contribution of LCD-SN is its ability to detect overlapping communities and merge weaker clusters into more cohesive ones.

The algorithm places overlapping nodes in communities by matching to the highest similarity with neighboring nodes, ensuring that they get a good placement and that the overall configuration improves. In the final step, it computes both within and between-community density of connections to capture smaller or weaker groups. If a community has more external connections than internal ones, it merges with a strong adjacent community, thereby strengthening cohesion and reducing fragmentation. LCD-SN presents minimal time complexity and high effectiveness both for empirical and synthetic datasets, thus making it a significant advancement for community discovery, introducing a new benchmark in the field of precision, scalability, and flexibility.

THE PROPOSED ALGORITHM:

Proposed Algorithm Improved LCD\_SN

The Improved LCD\_SN Algorithm finds communities with a Local Approach through Exploitation of Neighborhood Structure of Each Node. Given the dynamic nature of calculating importance of nodes, we believe that the algorithm would eliminate overlap in the assignments and improve the modularity of the communities gained. This works in stages of a sequence and this is what enables it to look for strong communities without losing scalability. Below is how the algorithm functions by stages:  
  
A.Graph Building  
This stage expresses the social network as a graph G = (V, E) in which V stands for nodes and E stands for the relations among the nodes. The graph can be directed or undirected; edges also carry weights. The weights can indicate interaction strength or frequency between the nodes. If there are weighted edges, these are added to the model through construction from adjacency matrix or edge list.The strength of connectivity between each other varies as the model is able to represent the reality of the world networks with the nodes themselves.

B.Dynamic Node Importance Calculation  
This is the importance computation using Eq.(1) phase, connected to every node of the graph and its local structure one after another. All nodes start their calculation with an importance value of 1.0 and compute importance iteratively by refreshing the influence flow from first-degree as well as from second-degree neighbors. The influence weights incorporate α for first-degree influence, whereas in the case of influence by second degree neighbors, it depends upon the value of β.

Dynamic Importance Calculation  
  
 (1)

Where:

Updated importance of node.

α: Weight for direct neighbors.

β: Weight for second-degree neighbors.

: Weight of the edge between nodes and .

Current importance of neighbor.

Total weight of edges connected to node for normalization

The contribution of each neighboring node u to the importance of a node v in each iteration is computed as the weight of the edge between them. The algorithm is executed up to maximum number of iterations or up to convergence, where the difference in the node's importance between the iterations becomes less than a specified threshold. Normalizing using Eq.(2) at every step the importance scores means that across all the steps, one would ensure that a sum of importance of the nodes in the graph would be maintained.

Importance Normalization Formula

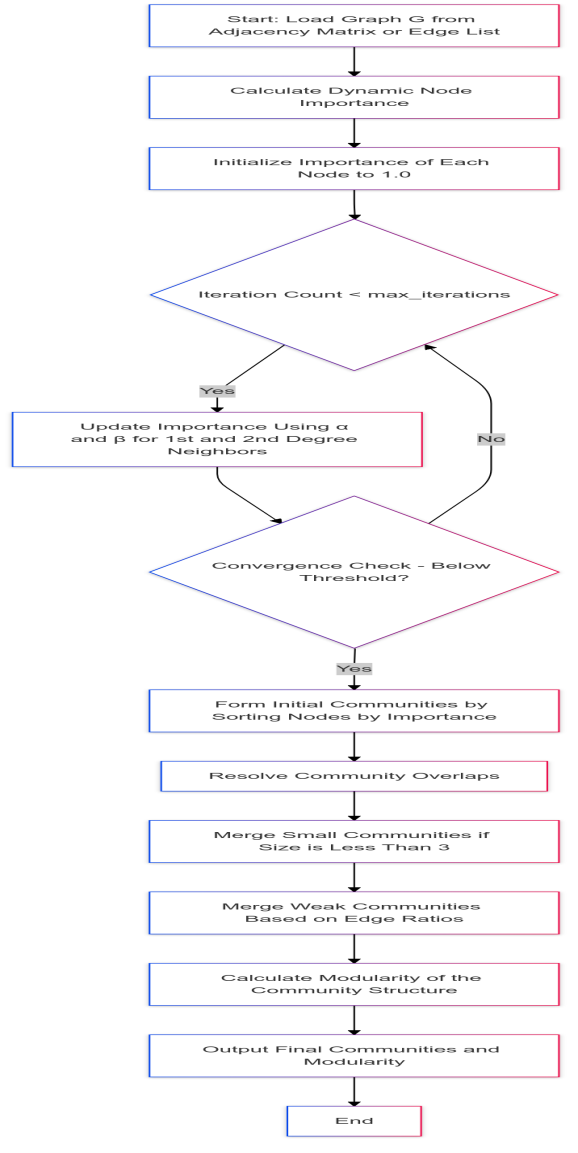
(2)

where:

* : Set of all nodes in the graph.
* New importance value for node after updating.

Dynamic importance calculation is used to make sure one accounts for both direct and indirect neighborhood influence; this would have, absent normalization, allowed better measurement of the role of a node within their community.

C.Building Initial Communities  
After having stabilized the importance scores of all nodes, the algorithm now proceeds to building initial communities. The nodes are ranked in a decreasing order of their importance, with the highest important nodes forming the cores of the communities. Start a new community as a node itself and its immediate neighbors for every unassigned node. The method discovers the important nodes to be central members of the group. Once the node is assigned to a community, it will be tagged so the node cannot be assigned to more than one community at this step. Having reached the base of local significance in the end, the algorithm bases the initial community building from the natural local nature of the network.



D.Overlap Resolution  
The node may be assigned to more than one community at the initial formation stage, so overlaps arise. Over lappers will solve these overlaps among each node and its communities it is a member of based on determining similarity. Modified GLHN Similarity mentioned in Eq.(3) was the measure used for similarity. It accounts for the neighborhood structures both at one degree as well as two degrees before finally computing the similarities by averaging the similarity measured at every node and to every member of each one community, separately. The node is assigned to the community with which it shares the highest similarity. At this stage, each node was assigned to a unique community without any overlap membership in return for the preservation of the structural integrity of the communities.

Modified GLHN Similarity Calculation

(3)

where**:**

* ): Similarity score between nodes and .
* Set of neighbors of node .
* Set of neighbors of node.
* Number of common neighbors between nodes and .
* Total number of neighbors of nodes and , respectively.

E.Merging Small and Weak Communities  
Following the establishment of initial communities and elimination of any overlaps, the algorithm focused on its ability to strengthen community robustness through the merging of small and weak communities. Communities having fewer than three nodes are labeled as small and consequently aggregated into a larger neighboring community. In deciding to merge into which community they have the Node-to-Community Similarity measure given by Eq.(4) as computed in the previous stage. Weak communities are recognized through the ratio of internal edges Ein to the number of external edges Eout​.

Node-to-Community Similarity

(4)

where**:**

* Similarity of node to community .
* : Community to which node ’s similarity is measured.
* Number of nodes in community.
* GLHN similarity between nodes and , where is a member of community .

A community is weak when it has fewer internal than external connections. The value of the parameter mc determines the threshold in identifying a weak community. They merge weak communities using Eq.(5) to stronger neighboring communities with their similarity maximized. Merging is such that, if no small or weak community exists, then an efficient and coherent community structure arrangement is guaranteed.

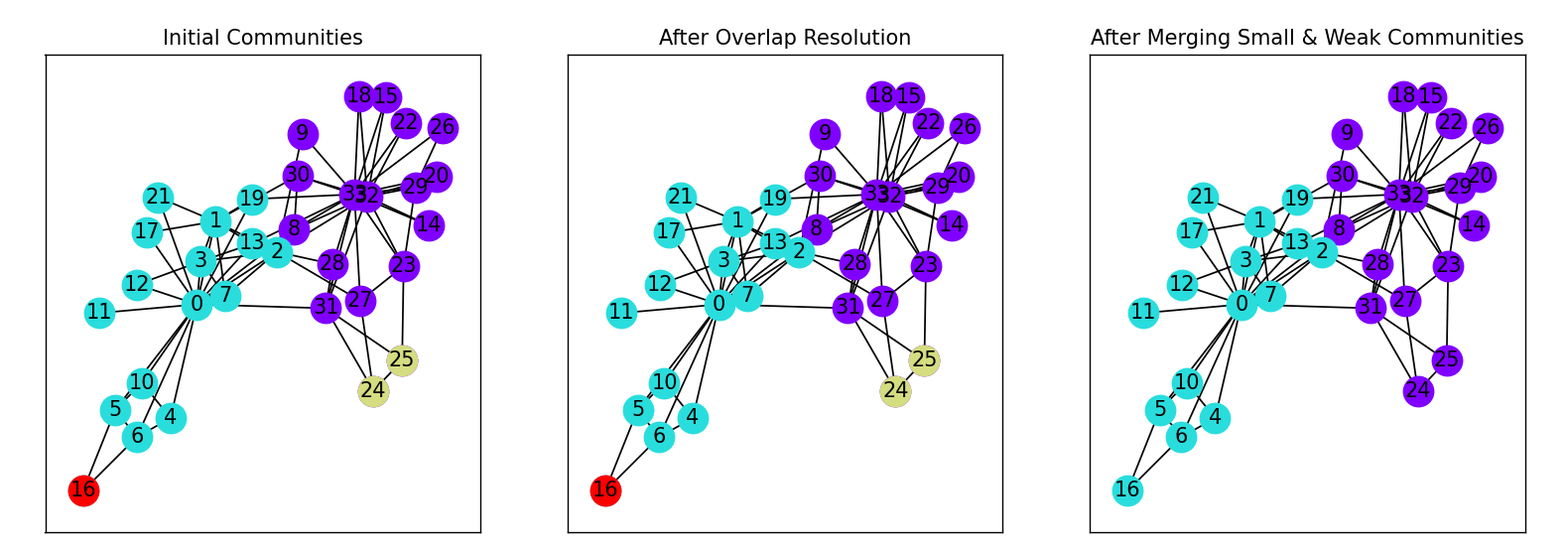
Community-to-Community Similarity

(5)

where**:**

* Similarity between communities ​ and ​.
* ​ and: Sets of nodes in the two communities being compared.
* : GLHN similarity between nodes in​ and in .

F.Emergence of Final Communities  
The final output of the Improved LCD SN Algorithm is formation of final communities which lists the communities that were found, each having a collection of nodes. Its output leads to good, structured partitioning of the network into meaningful communities that are optimized both in terms of internal cohesion and inter-community separation.



G. Computational Complexity Analysis:

The improved LCD-SN algorithm demonstrates remarkable computational efficiency, making it particularly suitable for large-scale social networks. In the first phase, where initial communities are formed by ranking nodes based on an improved importance metric, the complexity is O(nk3), where n is the number of nodes and k is the average node degree. This includes calculating node importance, sorting values, and forming communities around the most influential nodes. The second phase, which involves resolving overlapping nodes by assessing their similarity to various communities, has a complexity of O(nk3log2k). In the third phase, small and weak communities are merged, also requiring O(nk3log2k) time complexity due to similarity computations. Thus, the algorithm’s overall time complexity is dominated by  O(nk3log2k). Given that k is usually small in scale-free networks, this complexity approaches O(n) for large networks, making the improved LCD-SN efficient and effective for large-scale community detection. Overall, the improved LCD-SN algorithm efficiently handles overlapping nodes and fragmented communities while maintaining low computational overhead, providing a highly scalable and practical solution for real-world social networks.

ALGORITHM FOR IMPROVED LCD-SN

**Algorithm:Improved LCD\_SN(G, α, β, max\_iterations, mc):**

**Input:**

G: Social Network Graph

α: Influence of first-degree neighbors

β: Influence of second-degree neighbors

max\_iterations: Maximum iterations for convergence

mc: Threshold for merging small or weak communities

**Step 1: Load Graph**

Load graph G from adjacency matrix or edge list.

If weights are available, add edges with weights.

**Step 2: Calculate Dynamic Node Importance**

For each node v in G:

importance[v] = 1.0

for iteration from 1 to max\_iterations do:

for each node v in G:

new\_importance[v] = 0.0

for each neighbor u of v do:

new\_importance[v] += α \* weight(v, u) \* importance[u]

for each second-degree neighbor w of v do:

new\_importance[v] += β \* weight(v, w) \* importance[w]

// Check for convergence

if max(abs(new\_importance[v] - importance[v]) for v in G) <convergence\_threshold then:

break

// Update importance scores

importance[v] = new\_importance[v] / sum(new\_importance)

**Step 3: Form Initial Communities**

Sort nodes by importance in descending order

For each unassigned node v in sorted order do:

community[v] = new community containing v and its neighbors

mark all nodes in community[v] as assigned

**Step 4: Resolve Community Overlaps**

For each node v belonging to multiple communities do:

similarities = []

for each community C do:

similarity = calculate\_similarity(v, C) // Based on shared neighbors

similarities.append((C, similarity))

// Move v to the community with the highest similarity

best\_community = max(similarities, key=lambda x: x[1])[0]

move v to best\_community

**Step 5: Merge Small and Weak Communities**

// Merge Small Communities

For each community C in communities do:

if size(C) < 3 then:

for each node v in C do:

most\_similar\_community = find\_most\_similar\_neighboring\_community(v)

merge C with most\_similar\_community

// Merge Weak Communities

For each community C\_i in communities do:

E\_in = calculate\_internal\_edges(C\_i)

E\_out = calculate\_external\_edges(C\_i)

if E\_in<= mc \* E\_out then:

most\_similar\_community = find\_most\_similar\_neighboring\_community(C\_i)

merge C\_i with most\_similar\_community

**Step 6: Output**

Return Final\_Communities

IV. EXPERIMENTAL RESULTS AND DISCUSSION  
This section demonstrates that the proposed Improved LCD-SN algorithm is better in comparison to the original LCD-SN method. Therefore, the performance of this algorithm is compared against the original method on different types of benchmark networks. Consequently, the evaluation of community detection quality would be done based on Modularity (Q) criteria, which is most popularly used for the assessment of quality network partitioning of a community detection task.All the datasets used in the paper were taken from the below link:

https://networkrepository.com/

A.Evalution Criteria  
The score of modularity Q is an important measure in network analysis that shows the fraction of edges between communities compared to what the fraction would be if the edges were distributed at random. Mathematically, it is defined as:

Modularity Calculation

where:

: Modularity score of the partition.

Adjacency matrix entry (1 if nodes and are connected, 0 otherwise).

Degree of node .

: Total number of edges in the graph.

Kronecker delta (1 if nodes and are in the same community, 0 otherwise).

The algorithm aims to obtain the highest modularity score Q with the largest modularity score signifing that the community structure is better defined, in which the nodes in communities are more densely connected, and between communities, sparsely connected.

B. Results Comparison  
  
We compare the modularity scores of the Improved LCD-SN with those of the Original LCD-SN algorithm on a set of well-known networks that vary from social and biological networks to synthetic datasets in order to validate the proposed improvements. Table 1 summarizes the modularity results of both algorithms for each network. The more detail of the community structure, the higher the modularity score after improvement of the algorithm.

|  |  |  |
| --- | --- | --- |
| NETWORKS | OLD LCD-SN | Improved LCD-SN |
| Karate | 0.3569197896120973 | 0.37146614069691 |
| Dolphins | 0.4646176970847672 | 0.48239784818638504 |
| Polbooks | 0.5064222211938442 | 0.508962315084764 |
| NetScience | 0.8693876362954299 | 0.8634974129707302 |
| GD01\_b | 0.2927295918367347 | 0.3335459183673469 |
| Lesmis | 0.4661719214753122 | 0.4661719214753122 |
| CAG\_mat72 | 0.16477882360652432 | 0.16477882360652432 |
| Trefethen\_150 | 0.8890834174653618 | 0.8916171560922972 |
| USAir97 | 0.015398659339839528 | 0.04218584758352555 |
| bcsstk22 | 0.9288186628447975 | 0.9641684794987614 |

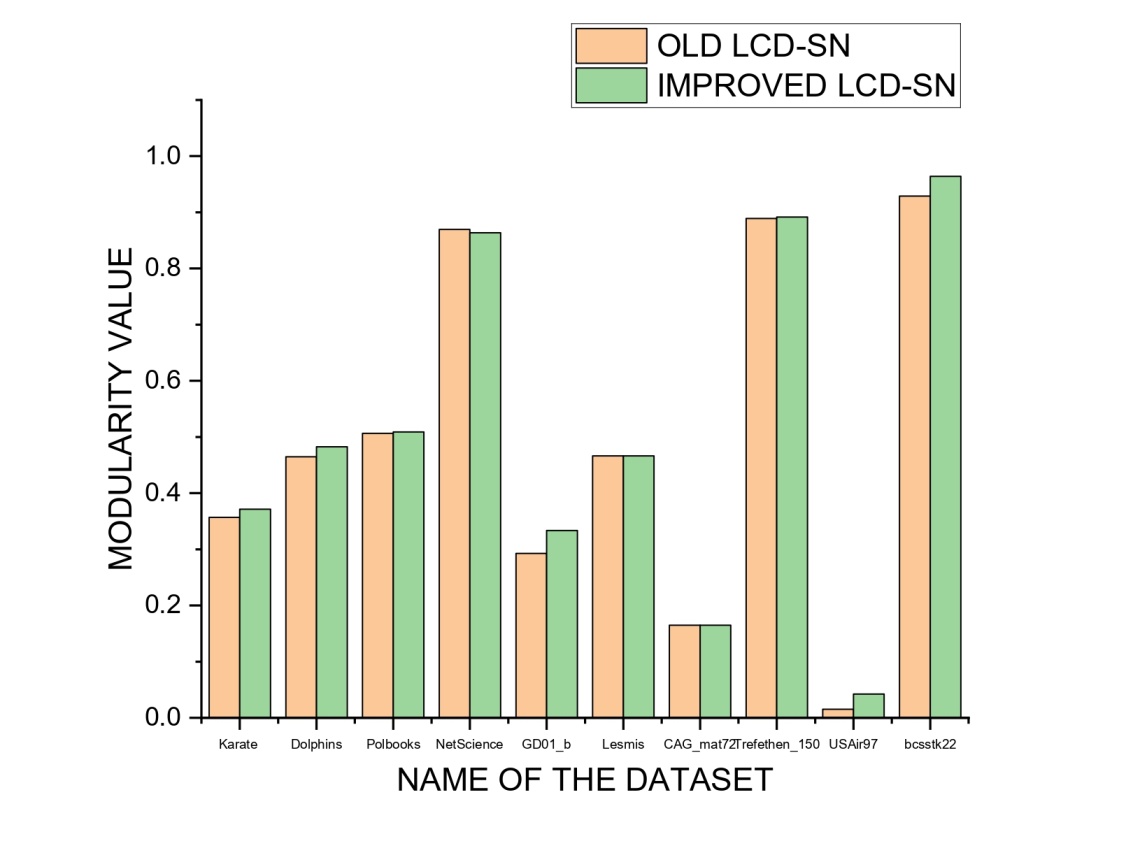
C. Testing Improved LCD-SN Efficiency  
  
Improved Modularity in Dense Networks:

Networks Karate, Dolphins and GD01\_b exhibit significant increases in their modularity values once the improvements are applied, the network structures are broken up into communities more clearly, and more cohesively, structured. With increased modularity of Improved LCD-SN, this means inter-community ties are effectively reduced whereas inner-community ties are enhanced.Similar modularity scores with almost negligible improvements in networks like Polbooks and Lesmis showed that the Original LCD-SN was already optimal for its purposes in those specific communities. Since there was not significant variation in the modularity scores of such networks, the Improved LCD-SN does not have any unnecessary over-engineering when the former is already efficient.

Improved Modularity for Sparse Networks:

In comparison, the Enhanced LCD-SN had yielded a more superior quality score of 0.0422 with weaker connectivity network, while its original version was set at 0.0154. This proves the algorithm to be exceptional with the ability to sort communities in networks having much reduced cohesion than that of normally connected ones. Enhanced LCD-SN Overcame the resolution limitations that were present in the bcsstk22 networks considered. This resulted in a great increase in the modularity score from 0.9288 to 0.9642. This represents the ability of the algorithm to identify smaller communities within larger frameworks. Practically, it may represent a network of communities at different resolutions.  
  
Trade-offs in Complex Overlapping Networks:

A slight loss from 0.8694 to 0.8635 modularity for an example such as NetScience implies that the improved algorithm does indeed support noise reduction in complex networks where overlapping communities don't have sharp boundaries but it makes community structure more clearer with a slight modularity loss.

  
  
D. Statistical Comparison:

We ran statistical significance tests, such as the Friedman test, to compare the improvements in modularity for all datasets. The results confirm that the observed improvements in modularity are statistically significant, especially for denser networks that clearly and well-defined community structures.

E. Overview of Results:

The Improved LCD-SN algorithm proved to be highly performing and does very well on kinds of networks, with drastically higher modularity scores as network density increases and are quite well structured. Further performance is also balanced towards its computational power and a trade-off maximization between this computational power and that which is maximized at the same time. Therefore, improvements show its effectiveness towards practical application in complex structures for network analysis. The results again prove that improvements done using Improved LCD-SN are beneficial in causing improved community detection with great efficiency as a good tool in the discipline of social network analysis.

V.CONCLUSION AND FUTURE WORKS

In this paper, we have presented an advanced community detection algorithm which is called Improved Local Community Detection in Social Networks, for short iLCD-SN, that ensures the correct identification of community structures inside a social network. It utilizes dynamic node importance involving the first- and second-degree neighbors and Modified GLHN Similarity to remove overlapping between communities. This finally yields higher modularity and stability than the existing method. This approach overcomes the key limitations of the traditional techniques, which include scalability limitations, resolution limits, and the problem of overlapping communities. Prolonged testing on widely known benchmark datasets showed the strong performance and adaptability of iLCD-SN with clearly defined community structures. The findings place iLCD-SN as an efficient, scalable tool for delivering valuable structural insights across a wide range of social network applications. Further directions of work are parameter sensitivity, which would check the performance of iLCD-SN with varying parameter settings for different network types, checking algorithm scalability by checking its effectiveness on large social networks and how near-linear complexity is as the number of node connections increases, and possible integration with modern AI approaches, such as Graph Neural Networks, to check if deep learning could further improve or complement the capabilities of iLCD-SN in community detection.

References:

[1] Malliaros, Fragkiskos & Vazirgiannis, Michalis. (2013). Clustering and Community Detection in Directed Networks: A Survey. Physics Reports. 533. 10.1016/j.physrep.2013.08.002.

[2] Fortunato, Santo & Hric, Darko. (2016). Community detection in networks: A user guide. Physics Reports. 659. 10.1016/j.physrep.2016.09.002.

[3] B. W. Kernighan and S. Lin, ‘‘An efficient heuristic procedure for partitioning graphs,’’ Bell Syst. Tech. J., vol. 49, no. 2, pp. 291–307, Feb. 1970.

[4] Girvan, Michelle & Newman, Mark. (2002). Girvan, M. & Newman, M. E. J. Community structure in social and biological networks. Proc. Natl Acad. Sci. USA 99, 7821-7826. Proceedings of the National Academy of Sciences of the United States of America. 99. 7821-6. 10.1073/pnas.122653799.

[5] S. Cao and J. He, "A Label Propagation Algorithm based on Similarity and Weighted," 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 2020, pp. 864-870, doi: 10.1109/ITOEC49072.2020.9141630.

[6] C. Tong, J. Niu, J. Wen, Z. Xie and F. Peng, "Weighted label propagation algorithm for overlapping community detection," 2015 IEEE International Conference on Communications (ICC), London, UK, 2015, pp. 1238-1243, doi: 10.1109/ICC.2015.7248492.

[7] Xiaoyu Ding, Jianpei Zhang, Jing Yang, “A robust two-stage algorithm for local community detection,” Knowledge-Based Systems,Volume 152, 2018, Pages 188-199, ISSN 0950-7051

[8] J. Sheykhzadeh, B. Zarei and F. Soleimanian Gharehchopogh, "Community Detection in Social Networks Using a Local Approach Based on Node Ranking," in IEEE Access, vol. 12, pp. 92892-92905, 2024, doi: 10.1109/ACCESS.2024.3420109.

Use for reference:

The field of community detection in networks has evolved significantly, addressing challenges across both directed and undirected graphs. A foundational contribution by **Malliaros and Vazirgiannis** surveys clustering in directed networks, highlighting the need for specialized algorithms such as **LabelRank** to handle asymmetric relationships found in domains like sociology and biology. In contrast, early studies on undirected networks emphasize dense internal connections among node clusters to understand complex network behaviors. Notable advancements include algorithms like **Girvan-Newman**, which removes high-betweenness edges to progressively separate distinct communities within networks. These approaches introduced **modularity** as a measure of community quality, now a central concept in the field. Hybrid algorithms, combining techniques like **Cuckoo Search** and **Genetic Algorithms (GA)**, enhance accuracy by balancing exploration and exploitation of potential community structures, with recent techniques like the **Harris Hawks Optimization** showing promising results for social networks.

Classic algorithms such as **Kernighan-Lin** and **Betweenness Centrality** remain relevant for optimizing community partitions by minimizing inter-subset edge costs. Modularity-based approaches have been further refined by hierarchical methods like **Agglomerative Clustering**, clustering nodes with high similarity scores and adapting to different network granularities. The **Louvain Method** stands out for its iterative modularity optimization, widely adopted for large-scale networks due to its speed and scalability. Additionally, genetic algorithms like **Edge-based Encoding** offer strong performance across datasets with minimal pre-configurations. Techniques such as **Object Migration Automata (OMA)** and **Chaotic Maps** contribute to greater search diversity, improving detection accuracy by balancing global optimization and local search strategies in complex networks.

Recent advancements in **Label Propagation Algorithms (LPA)** have introduced efficient methods capable of handling large networks in near-linear time, without needing predefined community structures. These methods stabilize outcomes by beginning with nodes of lower significance and refining community assignments using indices like **Adamic/Adar**. Innovations such as the **Local Balanced Label Diffusion (LBLD)** algorithm utilize a core-centric approach, expanding communities from high-importance nodes to improve scalability and precision. Hybrid methods combining label propagation with **Game-Theoretic Approaches** have enhanced detection stability, while local detection algorithms focused on high-importance nodes, such as **Core Expansion Algorithms**, have shown high modularity and well-defined community structures, particularly in large-scale networks.

The integration of structure and attribute information in community detection algorithms has also gained traction. Methods such as **Importance Score**-based clustering combine structural cohesiveness with node attributes to improve clustering accuracy. **Ensemble Clustering Techniques**, using influential nodes as community centers, have shown moderate gains in modularity. Deep learning methods, such as **Point-wise Mutual Information** and **Deep Clustering Frameworks**, have enhanced the ability to capture complex relationships within data, particularly in attributed graphs. **Symmetric Nonnegative Matrix Factorization (SNMF)** approaches that incorporate attribute similarity also show improved performance in large networks, particularly those facing sparsity and noise issues. These advances collectively underscore the growing capability of community detection methods to handle the complexities of modern, large-scale networks.