Assignment-1

Question: Create a random, unweighted, undirected, simple (no self-loop and parallel edge) connected graph G(V, E),
where |V| = 1000. The edge (u,v) exists in the graph or not -- decide randomly, but the graph should be
connected, i.e., |E|>=999.
Use the following measures to understand the centrality (c_v) of a vertex (v):
(i) Normalized degree centrality of a node: the ratio of the degree of the node with the maximum possible
degree.
(ii) Closeness Centrality of a node: the average shortest path lengths between the node and all other nodes in
the graph.
(iii) Betweenness centrality of a node: the ratio of the count of shortest paths that pass through the node and the
total number of shortest paths in the graph. Initially, count for two nodes, s and t, and use the sum considering all
s to t shortest paths.
Compute c_v for all vertices in the graph using a combined formula made from (i), (ii), and (iii). Print the top 10
vertices in terms of the c_v scores. Centrality defines the eligibility/impact of a node in a graph as a central node.

Computation of Centrality Measure:

- **Normalized degree Centrality of a node:** $C_D(v)$ = degree(v) / max possible degree(N-1) where degree(v) is the number of edges incident on v
- **Closeness Centrality of a node:** $C_C(v) = (N-1) / \sum_{u \in V} d(v,u)$ where d(v, u) is the shortest path distance between v and u.
- **Betweenness Centrality of a node:** $C_B(v) = \sum \sigma_{st}(v) / \sigma_{st}$ (s!= v!= t) where σ_{st} is the number of shortest paths between s and t, and $\sigma_{st}(v)$ is the number of those paths passing through v.
- **EigenVector Centrality of a node:** $x^{(k+1)} = A.x^k / || A.x^k||$ [A: Adjacency Matrix] where $x^{(k)}$ is the centrality vector at iteration k and the process until convergence with respect to predefined tolerance.

Computation of Combined Centrality Score:

$$\mathbf{C_V(v) = w1.\ C_D(v) + w2.\ C_C(v) + w3.\ C_B(v) + w4.x^k}$$

where **w1, w2, w3 and w4** are weights that can be adjusted to fine-tune the score.

Introduction

Graph analysis plays a crucial role in understanding the structure and importance of nodes in a network. In this study, we implemented various centrality measures and introduced a fine-tuning method called **Entropy weighted and Page Rank Fine-Tuning Approach** to improve the accuracy of identifying influential nodes. This report outlines the methodology, significance of centrality and fine-tuning, and the impact of using them in refining centrality scores.

Methodology

Our approach consists of the following steps:

1. **Graph Generation**: A **random connected graph** is generated with a mix of structured connectivity (spanning tree) and randomness to mimic real-world networks.
2. **Centrality Measures**: The following centrality metrics are computed:
    - **Degree Centrality**: Measures node importance based on direct connections.
    - **Closeness Centrality**: Evaluates the efficiency of a node in reaching others.
    - **Betweenness Centrality**: Captures nodes acting as bridges in shortest paths.
    - **Eigenvector Centrality:** measures a node's influence in the network based not only on the number of connections but also on the quality of those connections.
3. **Fine-Tuning (using Entropy weighted and Page Rank Approach)**:This approach improves centrality scores by calculating entropy-based weights for different centrality measures, including **PageRank**. The final centrality is a weighted combination of these measures, with top nodes visualized based on the improved scores.
4. **Graph Visualization**: The input and output graphs are being displayed, allowing for comparison before and after centrality analysis.

Significance of Fine-Tuning

While traditional centrality measures provide a general ranking of node importance, they often overlook indirect influences. **Fine-tuning using Entropy weighted and Page Rank Approach** ensures that we capture:

- **Hidden influential nodes**: Nodes that may not be directly central but play a key role in extending connectivity.
- **More accurate ranking**: Helps differentiate between nodes that initially have similar scores but contribute differently to network flow.
- **Better resilience analysis**: Helps in understanding which nodes are vital for network robustness.

Why Entropy Weighted and Page Rank Fine-Tuning?

To improve accuracy, we refine centrality scores using an additional step:

- **Balanced Contribution**: Entropy-based weights ensure both centrality measures (final centrality and PageRank) contribute proportionally based on their significance.
- **Adaptability**: Dynamically adjusts to network structure, reflecting variability in node importance.
- **Enhanced Node Identification**: Highlights key nodes more effectively by incorporating entropy-driven weights.

Results & Observations

To analyze the impact of fine-tuning, we generate three visualizations using **matplotlib**:

Input Graph

- Displays the randomly generated network before centrality computations.
- Nodes and edges are uniformly sized and colored.

Top 10 Nodes Without Fine-Tuning

- Highlights the **top 10 nodes based on first-level centrality**.
- These nodes are marked in **red**, while others remain **light blue**.

Top 10 Nodes After Fine-Tuning

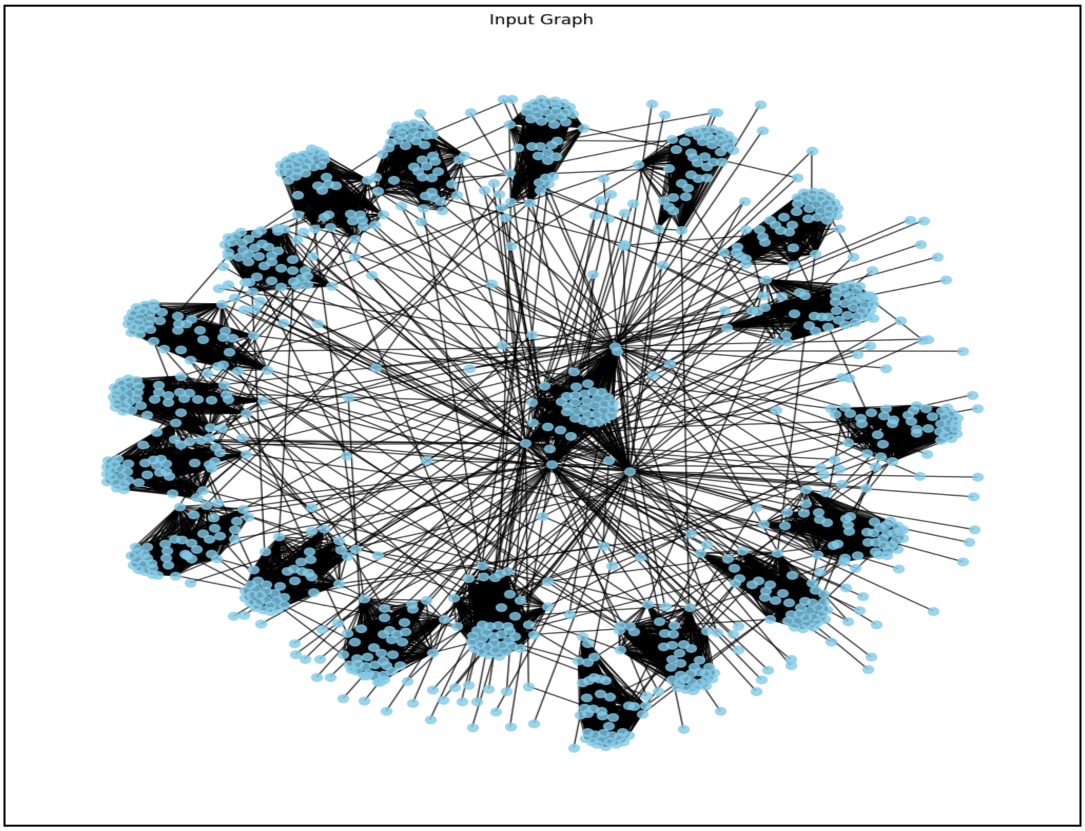
- Highlights the **top 10 nodes after incorporating Fine-tuning**.
- This comparison reveals how fine-tuning **adjusts the ranking of key nodes**, ensuring a more refined selection.

Conclusion

In conclusion, fine-tuning centrality measures with entropy-based weighting improves node importance assessments, providing a more balanced and accurate ranking. This approach adapts to network structures, enhancing the identification of key nodes. **Applications** include **social network analysis (identifying influencers), recommendation systems and infrastructure networks (critical node identification)**. It offers a robust method for improving analysis across various domains.

Output Visualization:

Input Graph-



->Degree Centrality (Top 10 nodes)-

Top 10 Nodes by Degree Centrality:

Node 291: 0.0991

Node 724: 0.0891

Node 982: 0.0871

Node 941: 0.0801

Node 132: 0.0440

Node 17: 0.0430

Node 138: 0.0420

Node 196: 0.0420

Node 504: 0.0420

Node 639: 0.0420

->Closeness Centrality (Top 10 nodes)-

Top 10 Nodes by Closeness Centrality:

Node 291: 0.3684

Node 724: 0.3470

Node 132: 0.3441

Node 982: 0.3429

Node 941: 0.3421

Node 963: 0.3326

Node 17: 0.3295

Node 642: 0.3293

Node 396: 0.3292

Node 494: 0.3291

->Betweenness Centrality (Top 10 nodes)-

Top 10 Nodes by Betweenness Centrality:

Node 291: 1.0000

Node 982: 0.7362

Node 724: 0.6448

Node 941: 0.5812

Node 132: 0.4235

Node 504: 0.2220

Node 196: 0.2014

Node 46: 0.1732

Node 298: 0.1610

Node 727: 0.1575

->EigenValue Centrality (Top 10 nodes)-

Top 10 Nodes by Eigenvector Centrality:

Node 132: 0.0559

Node 291: 0.0544

Node 724: 0.0529

Node 982: 0.0528

Node 941: 0.0526

Node 642: 0.0521

Node 494: 0.0520

Node 17: 0.0520

Node 103: 0.0520

Node 396: 0.0519

->Final Centrality Before Fine-tuning (Top 10 nodes)-

Top 10 Nodes by Final Centrality Score:

Node 291: 0.9925

Node 982: 0.8483

Node 724: 0.8439

Node 941: 0.7955

Node 132: 0.6862

Node 963: 0.5780

Node 17: 0.5755

Node 396: 0.5703

Node 642: 0.5665

Node 494: 0.5505

->Final Centrality After Fine-tuning (Top 10 nodes)-

Top 10 Nodes by Improved Centrality:

Node 291: 0.5059

Node 982: 0.4326

Node 724: 0.4306

Node 941: 0.4050

Node 132: 0.3461

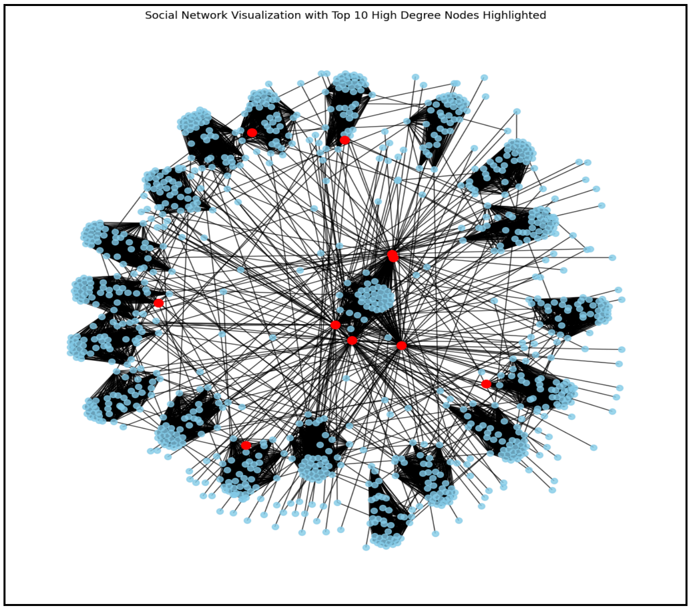
Node 963: 0.2917

Node 17: 0.2907

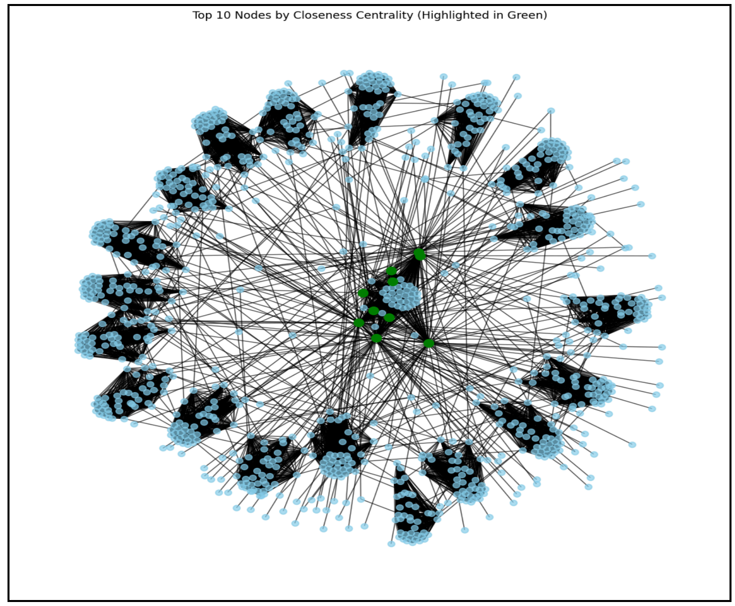
Node 642: 0.2878

Node 494: 0.2859

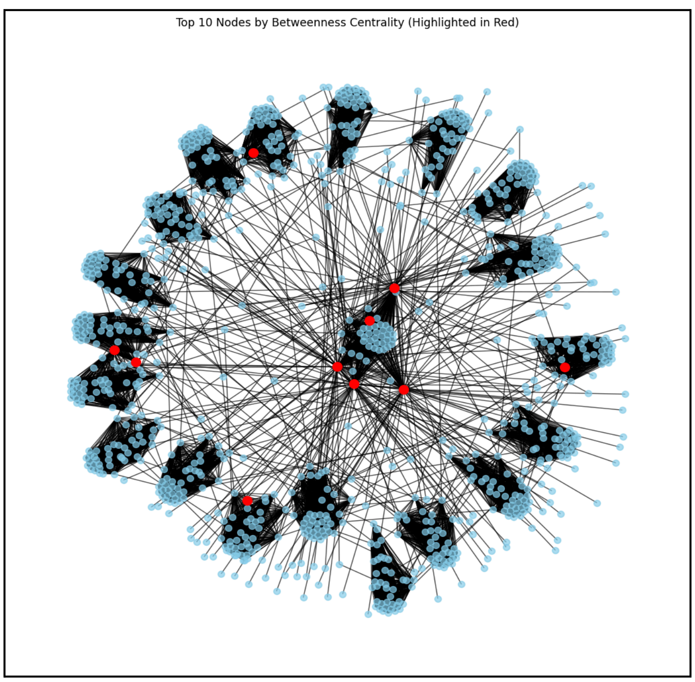
Node 396: 0.2779



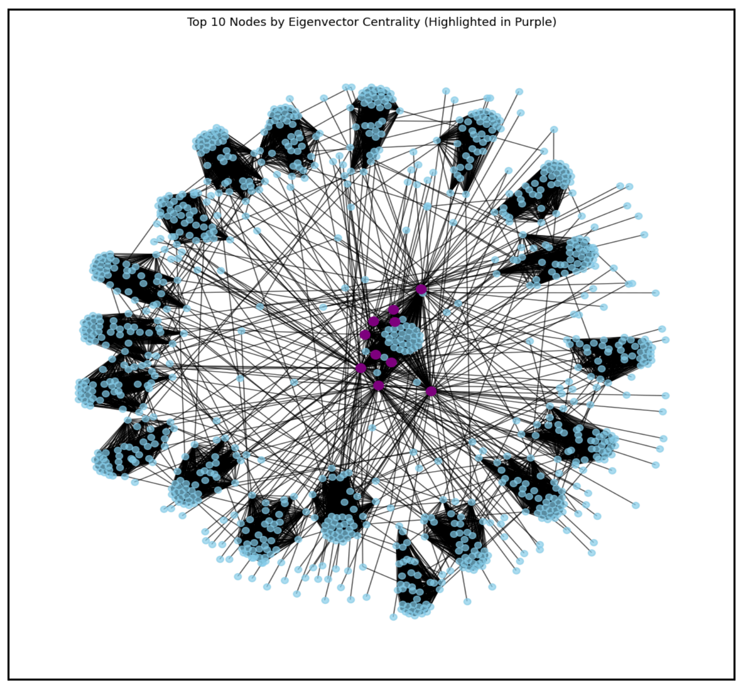
Degree Centrality



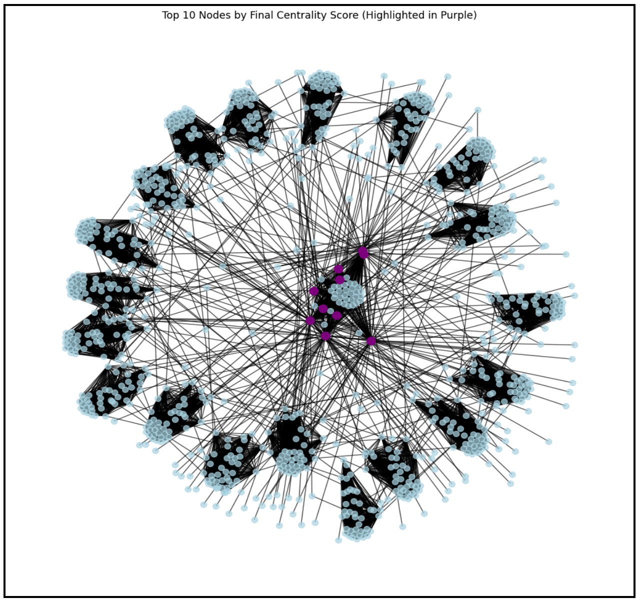
Closeness Centrality



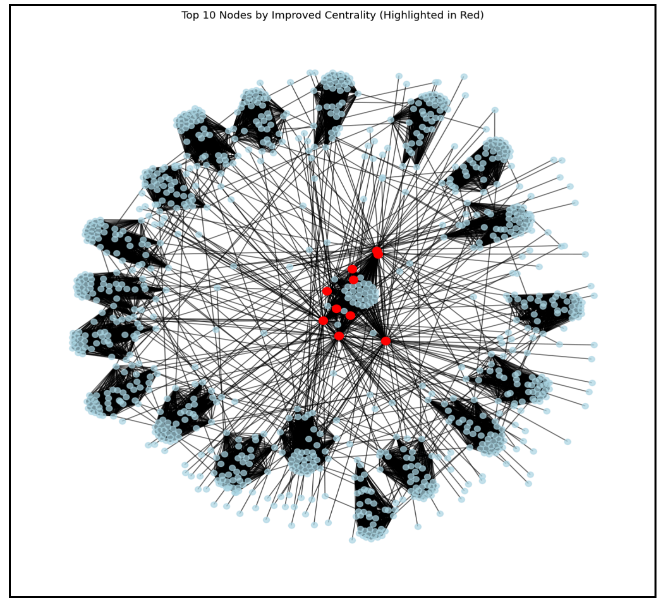
Betweenness Centrality



EigenVector Centrality



Final Centrality Before Fine-tuning



Final Centrality After Fine-tuning

The final centrality scores in the network were computed by combining multiple centrality measures with entropy-based weighting.

Entropy quantifies the randomness in a distribution, with higher entropy indicating more variability and uncertainty in the importance of nodes.

$$\mathbf{H(X) = -\sum p(x_i) * \log_2(p(x_i)) \qquad [from\ i = 1\ to\ n]}$$

Where:

- **H(X) is the entropy of the distribution XXX.**
- **$p(x_i)$ is the probability of the i-th event (in this case, the normalized centrality score of each node).**
- **n is the total number of events or values (in this case, the total number of nodes).**
- **The logarithm is taken to the base 2 because we typically measure entropy in bits.**

Here in this case as an example - w1: 0.266, w2: 0.267, w3: 0.201, and w4: 0.266 represent the entropy-based weights assigned to the respective centrality measures. These weights indicate the relative importance of each centrality measure in the final score.

The weights w1, w2, and w4 (around 0.266) are nearly identical, suggesting that the centrality measures they correspond to have similar significance in identifying influential nodes.

The weight w3 (0.201) is slightly smaller, indicating that the centrality measure corresponding to this weight is less influential or carries more uncertainty compared to the other measures.

For Fine-tuning also this entropy weighted approach is used upon Final centrality measure and PageRank and the corresponding weights were -> α: 0.502, γ: 0.498

From these results, we can conclude the following:

1. **Reduction in Centrality Scores After Fine-tuning**: The final centrality scores of the top nodes have decreased significantly after fine-tuning, which suggests that the fine-tuning process has adjusted the node rankings, possibly by balancing multiple centrality measures and incorporating entropy-based weights.

2. **Impact of Combining Centrality Measures**: The fine-tuning process, which combines different centrality measures (using entropy), appears to have redistributed centrality values, reducing the dominance of any single centrality measure. This indicates that the improved centrality reflects a more holistic view of node importance, factoring in various perspectives.

3. **Refinement of Node Rankings**: Although the top nodes by final centrality remain similar (e.g., Node 291, Node 982), their relative ranking has shifted, indicating that fine-tuning has led to a more refined distinction between nodes. For example, the difference in centrality scores between nodes is now smaller, which may help in more accurately identifying influential nodes in complex networks.

4. **Possible Normalization Effect**: The decrease in centrality values could also suggest that the centrality values are now normalized after the fine-tuning, making them more comparable across different nodes and centrality measures.

In summary, fine-tuning has refined the centrality values, providing a more balanced and nuanced view of node importance in the network. It likely reduced overemphasis on certain nodes and adjusted rankings to reflect a more comprehensive evaluation of node significance.