GRAPH CLUSTERING ALGORITHMS

CST 302
Algorithm Analysis and Design

TEAM MEMBERS

- Angel Sebastian (11) , IDK22CS011
- Ansu Kurian(14) , IDK22CS014
- Devika Saji(24) , IDK22CS024
- Sunisha KS(75) , IDK22CS075

Faculty: Prof.Anish Abraham

PROBLEM DEFINITION

Problem: Graph Clustering

Given a graph, partition the vertices into clusters such that vertices within the same cluster are densely connected, while vertices in different clusters are sparsely connected.

Goal: Identify communities or groups in the graph.

Applications:

- Social network analysis (community detection)
- Recommendation systems

ALGORITHM-1: K - Means Clustering

A centroid-based clustering algorithm that partitions data into k clusters by minimizing the variance within each cluster.

Adapted for graph clustering by using graph embeddings or node features.

Best for: Graphs with node features or embeddings.

PSEUDOCODE:

```
KMeansClustering(G, k):
    Initialize k centroids randomly
    while not converged:
        Assign each node to the nearest centroid
        Update centroids as the mean of the nodes in each cluster
    return the final clusters
```

COMPLEXITY ANALYSIS

Time Complexity

$O(n \cdot k \cdot d \cdot I)$

n: Number of nodes in the graph. k: Number of clusters.

d: Dimensionality of the node features or embeddings. I: Number of iterations until convergence.

Explanation

- In each iteration, the algorithm computes the distance between each node and each centroid ($O(n \cdot k \cdot d)$).

- This process is repeated for I iterations until convergence.

- The complexity is linear in terms of the number of nodes (n) but scales with the number of clusters (k) and dimensions (d).

COMPLEXITY ANALYSIS

Space Complexity:

- O(n.d)

Explanation:

- The algorithm stores the node features or embeddings, which require O(n · d) space.

- Additional space for centroids and distances is negligible compared to the node data.

ALGORITHM-2: Louvain Method

A greedy optimization method that maximizes modularity, a measure of the quality of a graph partition.

Iteratively moves nodes between communities to improve modularity.

Best for: Large-scale graphs with hierarchical community structures.

PSEUDOCODE

```
Louvain Method(G):
  Initialize each node as its own community
  while modularity improves:
    for each node:
      Move node to the community that maximizes modularity
    Aggregate communities into super-nodes
  return the final partition
```

COMPLEXITY ANALYSIS

Time Complexity

- O(V log V) (on average)

- V: Number of vertices (nodes) in the graph.

Explanation

- The Louvain method iteratively optimizes modularity by moving nodes between communities.

- Each node is processed in O(log V) time on average due to the greedy optimization and hierarchical aggregation.

- The algorithm is efficient for large-scale graphs because it avoids expensive matrix operations.

COMPLEXITY ANALYSIS

Space Complexity:

- O(V + E)

- V: Number of vertices.

- E: Number of edges.

Explanation

- The graph is stored using an adjacency list, which requires O(V + E) space.

- Additional space is needed for storing community assignments and modularity gains, but this is typically linear in terms of V.

ALGORITHM-3 Spectral Clustering

Uses the eigenvalues and eigenvectors of the graph Laplacian matrix to perform dimensionality reduction before clustering.

Converts the graph into a lower-dimensional space where clustering is easier.

Best for: Graphs with clear community structures.

PSEUDOCODE:

Spectral Clustering(G, k):

Compute the Laplacian matrix L of the graph G

Compute the first k eigenvectors of L

Use k-means clustering on the eigenvectors to partition the graph into k clusters

COMPLEXITY ANALYSIS

Time Complexity

$O(V^3)$

V Number of vertices.

Explanation

- The most expensive step is computing the eigenvalues and eigenvectors of the Laplacian matrix, which takes $O(V^3)$ time for a dense matrix

- If the graph is sparse, specialized algorithms (e.g., Lanczos method) can reduce this to $O(k \cdot E)$, where k is the number of eigenvectors computed.

- The subsequent k-means step adds $O(V \cdot k \cdot I)$ time, but this is dominated by the eigenvalue decomposition.

COMPLEXITY ANALYSIS

Space Complexity

$O(V^2)$

Explanation

- The Laplacian matrix requires $O(V^2)$ space for a dense graph.

- For sparse graphs, this can be reduced to $O(V + E)$ using sparse matrix representations.

- The eigenvectors also require $O(V \cdot k)$ space.

ALGORITHM-4: Affinity Propagation

A message-passing algorithm that identifies exemplars (cluster centers) based on the similarity between nodes.

Does not require the number of clusters to be specified in advance.

Best for: Graphs with small to moderate size and no prior knowledge of cluster count.

PSEUDOCODE

```
AffinityPropagation(G):
  Initialize similarity matrix S
  Initialize responsibility and availability matrices
  while not converged:
    Update responsibility matrix
    Update availability matrix
  Extract exemplars and assign nodes to clusters
```

COMPLEXITY ANALYSIS

Time Complexity:

- $O(V^2 \cdot I)$

- V: Number of vertices.

- I: Number of iterations until convergence.

Explanation

- The algorithm computes and updates two matrices: responsibility (R) and availability (A), each of size V × V.

- In each iteration, all pairs of nodes are processed, leading to $O(V^2)$ time per iteration.

- The total time depends on the number of iterations (I) required for convergence.

COMPLEXITY ANALYSIS

Space Complexity:

- $O(V^2)$

Explanation

- The similarity matrix, responsibility matrix, and availability matrix each require $O(V^2)$ space.

- This makes the algorithm memory-intensive for large graphs.

COMPARISON OF GRAPH CLUSTERING ALGORITHMS

ALGORITHM	TIME COMPLEXITY	SPACE COMPLEXITY	BEST FOR	APPROACH
K-Means Clustering	O(n.k.d.I)	O(n.d)	Graphs with node features/embedding	Centroid-based clustering
Lovain Method	O(V log V)	O(V+E)	Large-scale hierarchical graph	Greedy modularity optimization
Spectral Clustering	$O(V^3)$	$O(V^2)$	Graphs with clear community structures	Eigenvalue decomposition +k-means
Affinity Propagation	$O(V^2.I)$	$O(V^2)$	Small to moderate graphs	Message passing

References

1) **MacQueen, J. (1967).** Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability.*
2) **Blondel, V. D., et al. (2008)**. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment, 2008(10), P10008.*
3) **Von Luxburg, U. (2007)**. A tutorial on spectral clustering. *Statistics and Computing, 17(4), 395–416.*
4) **Frey, B. J., & Dueck, D. (2007)**. Clustering by passing messages between data points. *Science, 315(5814), 972–976.*

THE END