GRAPH CLUSTERING ALGORITHMS

Kishore Kumar(M21CS058), P Sandhya Gayatri(M21CS060), Tejaswee A(M21CS064)

IIT Jodhpur

ABSTRACT

Graph clustering algorithms partition the vertex set present in a graph into disjoint groups or clusters, such that the elements of each cluster are highly similar among the clusters and simultaneously very dissimilar to the elements of other clusters. Given the constraints, optimizing these graph clustering algorithms is an NP-hard problem. Few Graph Clustering approaches include the Top-down, the Bottom-up, and the Local Optimization approach. We aim to explore some of these best strategic graph clustering algorithms, which have wider range of applications. These include k-means, Greedy Agglomeration, Markov Clustering Algorithm, Local Moving and Multilevel Algorithm, and Clique Percolation Algorithm.

Index Terms— Graph Clustering, Top-Down, Bottom-Up, Local Optimization

1. INTRODUCTION

1.1. *Problem Statement*

Social Networking Sites(SNS), these days have become a powerful way to keep people connected irrespective of their geographic locations, difference in time, or other context specific barriers. Some popular social networking sites are Facebook, Instagram, Twitter, etc. and people on these platforms are united by their common interests, common communities and so on. The data so obtained from these sites is gigantic and if used properly, can help us to extract meaningful knowledge about the user/person in particular.

1.2. *Solution*

The interactions between people using social networking sites can be modeled as a graph, which abstracts people and their relations as vertices and their corresponding edges. To understand these relations, we find clusters. Each cluster depicts the vertices that are highly similar to one another simultaneously being highly dissimilar with vertices of other clusters. Let us consider the the graph shown below. Graph shows contacts of a brother-sister (Ross-Monica). Our main concern is to find friends of Ross and Monica individually.

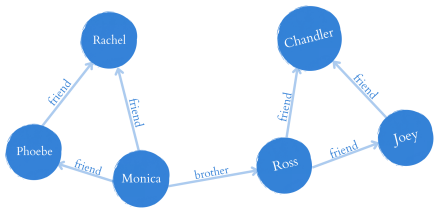

Fig. 1. Social Network Graph

To identify friends of Ross and Monica, we use Graph Clustering Algorithms. These help us to find clusters. This is as shown in the graph below. The yellow cluster is known to have Ross's friends, while blue cluster is known to have Monica's friends.

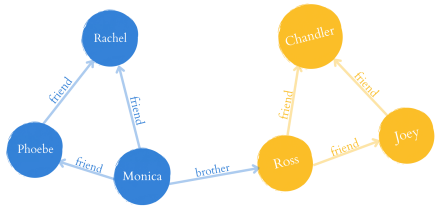

Fig. 2. Clusters in Social Network Graph

Further with an aim to understand these graph clustering algorithms, references that have been used in our learning process are mentioned below.

- Research papers are as cited [1] [2]
- The Articles that have been referred are as cited [3] [4] [5] [6] [7] [8]
- The Git repository containing algorithms can be referred by navigating through the link provided in [9]

Section 2 gives a brief introduction to clustering measures, Section 3 discusses the Top-down clustering algorithms, Section 4 discusses the Bottom-up clustering algorithms, Section 5 discusses the Local Optimization clustering algorithms and Section 6 focuses on the Clique Percolation Clustering Algorithm. Section 7 mainly focuses on the applications where the graph clustering algorithms can be used. A consolidated discussions has been put forward in the Section 8 and Section 9 deals with the summary of the project.

2. DEFINITIONS

2.1. *Graph Clustering Definition*

Graph clustering is the process of partitioning a set of nodes present in a graph into disjoint groups or clusters so that the elements of each cluster are highly similar, but are very dissimilar to the elements of other clusters.

2.2. *Graph Clustering Measures*

The quality measures of a clustered graph are identified as follows:

1. Combination of less inter-cluster edges and more intra-cluster edges gives better and higher quality
2. Inseparable Cliques
3. Connected Clusters
4. Disjoint Cliques approaching maximum quality
5. Modularity measures the strength of division of a network into modules

2.3. *Impossibility Theorem for Clustering*

Given set S. Let $f : d \rightarrow \Gamma$ be a function on a distance function d on set S, returning a clustering Γ. No function f can simultaneously fulfil Scale Invariance, Richness and Consistency.

3. TOP-DOWN APPROACH

Top-down approach iteratively splits the initial cluster, containing all the vertices, into smaller clusters.

3.1. *k-Means*

k-Means stores the centroids corresponding to k-clusters. For finding the centroids using k-Means, we iteratively choose the nearest data points by calculating Euclidean distance with the centroid and assigning these data points to a cluster by calculating the mean. Clustering terminates when either the number of centroids are not changing further or it has reached the maximum number of iterations.

3.2. *k-Center*

In k-Center Clustering, for each cluster, farthest sample data from the centroid is considered. On the clusters level, it takes into consideration of the worst cluster whose data point has the maximum distance from the centroid in comparison to other clusters. Since k-Center Clustering is an NP-Hard problem, therefore Greedy approximation algorithms are preferred.

3.3. *k-Median*

k-medians approach aims at minimizing the 1-norm distances (Manhattan-distance) of every data point and its closest center to its cluster. Mean is highly vulnerable to outliers, even a single outlier can change the value of mean making it distant from other data points whereas median is highly resistant to outliers. This algorithm may require more than 50% of the data points to be differed in order to change the value of median.

4. BOTTOM-UP APPROACH

The Bottom-up approach starts with singleton set of individual vertex and merges them to form clusters.

4.1. *Greedy Agglomeration*

Algorithms under bottom-up approach consider each data point as a singleton cluster and iteratively agglomerates to form a single cluster containing all the sample points. Similar pair of clusters are grouped together as they go up in hierarchy. Based on the distance measurement generated, we use linkage function to group the clusters.

4.2. *Markov Clustering Algorithm*

Markov Clustering

Markov Clustering Algorithm is based on Random Walks.

Random Walks: If we start our random walk at a node, then we are more likely to stay within a cluster to which the node belongs, than travelling between clusters. To know where the flow trends gather, we use these random walks on graphs, which results in finding the clusters. These random walks are calculated using Markov Chains.

In MCL, the following two processes are alternatively repeated:

1. Expansion (taking the Markov Chain transition matrix powers)
2. Inflation (re-normalizing a single column when expanded by a parameter r)

Algorithm

Following are the steps that describe MCL Algorithm:

1. Undirected graph, e (power), and r (inflation parameter) are given as inputs to the algorithm.
2. Matrix is initialized with the corresponding edge weights. Self loops are then added to each node.
3. Then column-wise Normalization of the matrix is done.
4. We multiply the matrix by e times, to itself.
5. Then the matrix is inflated by a parameter r.
6. Steps 4 and 5 are repeated until convergence is achieved.
7. We thereafter find for clusters so formed.

5. LOCAL OPTIMIZATION APPROACH

In Local Optimization, using random clustering, we migrate towards the nodes.

5.1. *Local Moving and Multilevel Algorithm*

To improve clustering and increase modularity, we use local search heuristics. Following are the classes of local search heuristics:

1. Cluster Joining (CJ), which iteratively joins two clusters
2. Vertex Moving (VM) approach is known to form different clusters by iteratively moving through individual vertices

Local Moving(LM) Algorithm

The Local Moving Algorithm is known to perform the best possible move for each and every vertex by repeatedly iterating through all vertices in a randomized fashion. Amongst all the modularity-increasing vertices, the vertex with highest priority over all the target classes is called best move. It iterates until there is no modularity-increasing vertex.

Multilevel Clustering Algorithm

Phases in the Multilevel Clustering Algorithm:

1. Coarsening Phase
2. Refinement Phase

Coarsening Phase

A sequence of graphs called the coarsening levels are produced by the Coarsening Phase, in which the first coarsening level is the input graph. Using coarseners (either of CJ or VM algorithms), clustering is done. Initially each single vertex is a cluster for the coarsener and it starts clustering. The coarsener runs until:

1. it terminates
2. the number of clusters decrease by a certain percentage, as compared with the start, where the percentage is called reduction factor.

The next coarsening level is formed where each cluster is contracted to result into a single vertex. When there are no further changes in clusters in 2 subsequent layers, the coarsening phase is known to reach a fixed point and it ends.

Refinement Phase

The algorithm then visits all the coarsening levels in a reverse order, from the coarsest graph to the original graph. It computes clustering at each level. This phase is called the Refinement Phase. The final clustering of a level is projected to its next subsequent level and this forms the initial clustering of the subsequent level, on which a refiner is applied to compute the final clustering. Here the refiner is preferred to be a VM algorithm, since CJ algorithms may not find join-able clusters resulting to an optimal clustering.

An abbreviation of MLx is be used for this Multilevel Refinement, where x represents the Reduction Factor that is used in the coarsening phase.

6. CLIQUE PERCOLATION ALGORITHM

In social networking graphs, it is important to find the communities which consist of similar type of people having similar thought process. Hence, the key task becomes to find the strongly connected subgraphs in a graph. These subgraphs are called cliques. It is so possible to have a single person belonging to multiple communities i.e., a node in a community may be shared with certain other number of communities. Therefore, clique percolation algorithms aims at identifying the overlapping communities. It initiates with identifying k-cliques and maintaining the adjacent clique (having k-1 nodes) in a community. In this process, k-cliques are considered to be different entities when it is not possible to group furthermore as a community.

7. APPLICATIONS OF GRAPH CLUSTERING ALGORITHMS

This section mainly focuses on the some of the real-time scenarios where the above discussed methods can be applied. These are as listed below:

1. k-Means clustering algorithm is generally applied to numeric data that is continuous with smaller dimensions such as document clustering, identification of crime prone areas, insurance fraud detection.

2. General application of k-Center clustering algorithm is in finding the placement of an item in the best possible and appropriate place like placement of an ATM center in a city so that it covers a certain range.

3. k-Median clustering algorithm's applications include software fault prediction, formal software verification, software defect analysis.

4. Application of Greedy agglomeration algorithm is mainly seen in constructing Connectomes or Neural wiring diagrams that shows how neurons are connected together.

5. Applications of Multi-level clustering include Hierarchical FGPA (Field Programmable Gate array) mapping. In each of it's hierarchy level, cluster of logic blocks or logic blocks are connected together along with the switch blocks.

6. Markov clustering algorithm is majorly used in clustering Bioinformatics data like to cluster protein sequences and to cluster genes from co-expression data.

7. Clique Percolation method is generally used to detect communities from the studies of cancer metastasis, social networks, and economical networks.

8. DISCUSSIONS

Among the well-known graph clustering approaches, the Top-Down approach iteratively splits the initial cluster containing all the vertices into smaller clusters, the Bottom-Up approach starts with singleton set of individual vertex and merges them to form clusters, and the Local Optimization approach are the approaches that have been explored. The Clique Percolation Algorithm, a Min-cut based approach has also been studied.

The methods such as k-Means, k-Centers, k-Medians, fall under the category of the Top-Down approach, but these require an initial input of the number of clusters to be formed, represented by k. Prior knowledge of k from a graph is not known and might not yield good results as well. This motivated to explore the other set of clustering algorithms.

Under the Bottom-Up approach Greedy Agglomeration and Markov Clustering Algorithms have been discussed. The Greedy Agglomeration Algorithm is mainly used in Connectomes or Neural wiring diagrams that show how neurons are connected together, while the Markov Clustering Algorithm's application is found in the field of Bioinformatics, requiring the graphs to be of sparse natured. Studies suggest that the MCL Algorithm finds overlapping clusters in some restricted cases only.

Taking the above points into consideration, the Local Optimization approach was further explored. Multilevel clustering algorithm, which comes under the local optimization category has been discussed and it was observed that this algorithm is best suited for applications that include Hierarchical FGPA(Field Programmable Gate array).

Thereafter, Min-cut clustering approach has been studied. The Clique Percolation Method is a popular Min-cut clustering approach and is widely used when there is a need of detecting community clusters. One of major applications of this method is in social networks and hence serves the problem statement so considered.

9. SUMMARY

The interactions between people in social networking sites can be modeled as a graph, abstracting people and their relations as vertices and their corresponding edges. To get the essence of underlying relations among the nodes, we find clusters such that there are dense connections among intra-cluster nodes and sparse connections among inter-cluster nodes.

The problem of finding the relations among social network graphs are solved by graph clustering algorithms. To name a few from the proposed approaches, algorithms like k-Means, k-Center, k-Medians, Greedy Agglomeration, Markov Clustering, Local Optimization and Clique Percolation. Starting with the top-down approach, these algorithms do not efficiently reveal the relations among nodes and hence algorithms of bottom-up approach were studied. But the said approach does not satisfy the need of finding relations in social networking graphs. Therefore, the clique-percolation method was further explored, which is better suited for social networking graphs as it works in the direction of linkage of nodes for a given graph.

10. REFERENCES

[1] K. Macropol, "Clustering on graphs: The markov cluster algorithm (mcl)," *University of Utrecht: Utrecht, The Netherlands*, 2009.

[2] R. Rotta and A. Noack, "Multilevel local search algorithms for modularity clustering," *Journal of Experimental Algorithmics (JEA)*, vol. 16, pp. 2–1, 2011.

[3] "Modularity." [Online]. Available: https://neo4j.com/blog/graph-algorithms-neo4j-louvain-modularity/

[4] "K-center." [Online]. Available: https://www.cs.upc.edu/~mjserna/docencia/grauA/T18/Greedy-approx.pdf

[5] "K-medians." [Online]. Available: https://www.coursera.org/lecture/cluster-analysis/3-5-the-k-medians-and-k-modes-clustering-methods-pShI2

[6] A. Marino, "Graph clustering algorithms." [Online]. Available: https://pages.di.unipi.it/marino/cluster18.pdf

[7] "Data clustering algorithms." [Online]. Available: https://sites.google.com/site/dataclusteringalgorithms/home?authuser=0

[8] "Clique percolation algorithm." [Online]. Available: https://cran.r-project.org/web/packages/CliquePercolation/vignettes/CliquePercolation.html

[9] "Git repository." [Online]. Available: https://github.com/tejasweeA/GTA_project