GRAPH CLUSTERING ALGORITHMS

Shruti Sureshan (M21CS015), Dhriti Prasanna Paul (M21CS056)

IIT Jodhpur

ABSTRACT

Graph clustering refers to the methods by which we can do a grouping of data in the form of graphs. It has received lot of attention from the research community in the recent years. Here we have done a detailed study on graph clustering, the methods by which they are classified, and studied certain algorithms that play a vital role in clustering. We have performed and shown different results of the popular clustering algorithms. These graph clustering algorithms can be used in various fields, including bioinformatics, community detection, computer network analysis, social network analysis etc.

Index Terms— Clustering, similarity, spanning tree, markov.

1. INTRODUCTION

Graph clustering is a way of grouping data by forms of graph in a way that the elements are related by some similarity measure. Graph clustering falls under the category of graph mining algorithms. A graph is clustered in one of the two ways, either by clustering its nodes by their edges, or by their edges' distances. A vertex cluster is formed by grouping associated edges together based on their edge weights. A second method involves considering the graphs as points to be clustered and grouped on the basis of their similarity. In this project, we have provided an insight into graph clustering for which we have done the following:

- We read 9 research papers for this project
- We have understood the Minimum Spanning Tree(MST) based clustering algorithm, Hierarchical Agglomerative Clustering algorithm, Markov clustering algorithm, Spectral Clustering, Shared Nearest Neighbor(SNN) Clustering, Betweenness Centrality, Maximal Clique Enumeration and Kernel K-means clustering
- We have also implemented some of the algorithms like Minimum Spanning tree based clustering, Markov clustering, Spectral Clustering and here is the Colab Link of the implementation
- Section 2 discusses related work, Section 3 defines terms, Section 4 provides the detailed explanation of the algorithms and Section 5 provides the summary.

2. RELATED WORKS

Clustering has always been a hard problem and an active topic of research[3]. There have been a few recent research on graph clustering. Minimum Spanning Tree Based Clustering Algorithm which are quite popular nowadays can even discover clusters of heterogeneous nature which have non-uniform borders[1]. Machine learning researchers have spent a lot of time on hierarchical clustering[2]. It gives better performance than k-means algorithm. A widely popular approach is the Markov Clustering algorithm[9]. The Markov chain is used to calculate random walks through a large graph simulating a long walk-through. Using spectral approaches for clustering is a viable option that has lately developed in a number of disciplines. The top eigenvectors of a matrix formed from the distance between locations are used here. Recently, a new approach has started to get a lot of attention namely spectral methods[3]. Spectral clustering techniques have seen an explosive development and proliferation over the past few years and they promise to become strong competitors for other clustering methods[3]. Another popular algorithm is Shared Nearest Neighbor Algorithm which strongly works on the concept of Shared Nearest Neighbor similarity[4]. Betweenness Centrality is another algorithm which takes into consideration vertex and edges of a graph[5]. Girvan and Newman Algorithm is used to have a better understanding about the clustering. Maximal Clique Enumeration uses the concept of clique for clustering. Naturally they NP-complete clique decision problem[6]. Maximal clique enumeration(MCE) is the fastest graph clustering method for finding all vertices that have the most influence in a graph[7]. Kernel clustering is more accurate than linear clustering, which makes kernel k-means more accurate for graph clustering[8]. Thus, Graph clustering has evolved as a viable approach for modelling complicated patterns in graph-structured data in the recent years.

3. DEFINITIONS

Data mining: It is the practise of extracting and detecting patterns from large datasets.

Graph mining: Graph mining is the process of extracting interesting patterns from graphs that represent the underlying data and can be used for classification or grouping.

Clustering: Clustering is an unsupervised task of grouping the data points into clusters based on similarity.

Graph clustering: Graph clustering is grouping data by forms of graph in a way that the elements are related by some similarity measure.

4. DISCUSSIONS

Minimum spanning tree based Clustering: Minimum spanning tree based clustering algorithm starts with Kruskal's algorithm using which we will construct a minimum spanning tree. Once the minimum spanning tree is created we can then start with the clustering. After building the minimum spanning tree, we compute a value which is the threshold along with the step size. Now the edge lengths which are greater than the computed threshold can be eliminated. Ratio between the distance of the intra-cluster and the distance of the inter-cluster is also computed. Advance the step size and thus update the computed threshold and repeat the same steps again in a loop. When no MST edges can be removed and the threshold is highest, we stop the loop. Thus, we compute the lowest Intra-inter ratio to form the clusters based on the threshold to get the optimum threshold. The optimum threshold lies in the middle of the extreme values. The extreme cases here are: First is when the threshold is zero then the point lies within one cluster and second case is when the threshold is highest then all the points lie within one cluster. To decrease the total iterations, we will never set the starting threshold to 0.

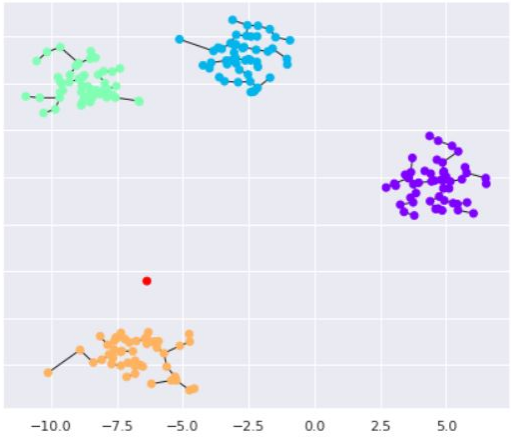

Fig. 1. Minimum spanning tree based clustering

Hierarchical Agglomerative Clustering: In this algorithm, we begin with small groups and then combine them to form larger clusters. To link the data points, we can use any type of linkages like average, ward, single and complete. In the case of average linkage, we use the average distance of the data. In the case of single linkage, we use the minimum of the distances while in complete linkage, we use the maximum of the distances. Ward linkage, which is most often used, reduces the variance. To get an idea about the actual path, we can also create its dendrogram. Dendrogram gives an idea about the similarity level among the different points. Thus, it provides a visibility on the closeness of the data points via construction of dendrograms.

Markov Clustering Algorithm: Markov clustering algorithm(MCL) is based on Random walks. A random walk is if you begin at a particular node and start randomly traveling to other connected nodes, then you will probably remain inside a group than travel between. By doing so, it is possible to locate where the flow is prone to be meet and thus where the clusters are. In this algorithm, we follow two basic processes one after the other alternatively: Expansion and Inflation. The expansion operator is in charge of permitting flow to connect the graph's various areas. The inflation operator is in charge of both present strengthening and weakening. Amid the prior powers of the Markov Chain, the edge weights will be higher in links that are inside clusters, and lower between the clusters. This implies there is a correlation between the distribution of weight over the columns and the clusterings. The input to this algorithm is a power parameter e, an inflation parameter r and an undirected graph G. Then we compute an associated matrix depending on the associativity of the nodes. Next step is optional which is to include the self loops. Next step is matrix normalization. After this step, we expand by including the matrix to the power of e and then using the r parameter, computing the inflation of the matrix and again repeating this step till convergence. The final matrix obtained can be useful to find new clusters.

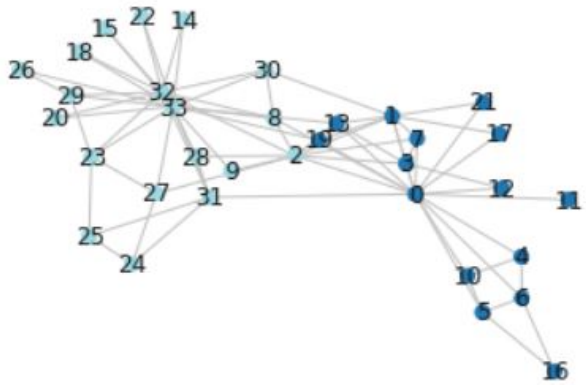

Fig. 2. Markov clustering

Spectral Clustering: The eigenvalues (spectrum) of special matrices constructed from the graph or data set are used in spectral clustering. The graph's edges represent the correla-

tion among the points. The Graph Laplacian's eigenvalues can then be used to determine the optimal number of clusters. The actual cluster labels can be determined from the eigenvectors. Constructing a similarity network, projecting data into a lower-dimensional space, and grouping the data are the three primary steps in spectral clustering. First we form a distance matrix and then transform it into affinity matrix A. Now we compute the Laplacian matrix which is L=D-A where D is degree matrix. For this laplacian matrix we will compute eigenvalue and corresponding eigenvectors. A matrix is formed by the eigenvectors of the k greatest eigenvalues obtained in the preceding phase. The vectors should be normalized. In k-dimensional space, group the data points.

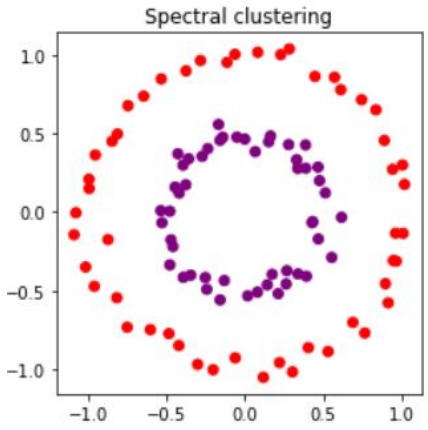

Fig. 3. Spectral clustering

Shared Nearest Neighbor(SNN) Clustering: A clustering technique that uses the concept of similarity based on the sharing of near neighbors. Shared Nearest Neighbor is the measure that represents the number of neighbor nodes which are common between any given pair of nodes. It works by firstly finding the k-nearest neighbors of all the available nodes. Than we take pair of nodes and find out the similarity between them. Similarity is equal to number of shared neighbors if two nodes(points) are among the k-nearest neighbors of each other else, similarity is zero. Jarvis Patrick Clustering Algorithm is based on the concept of SNN. Let us understand it in more depth. Let us consider the graph as shown in Fig. 4 The undirected graph in Fig. 4 is having it's nodes connected where we can calculate the common neighbors among pair of nodes. Nodes x_4 and x_5 are having the maximum number

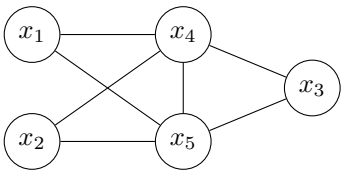

Fig. 4. SNN Clustering

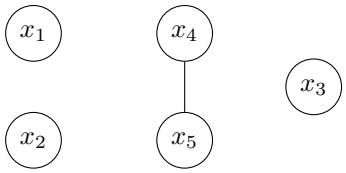

Fig. 5. SNN Clustering

of common neighbors which is 3(x_1, x_2, x_3). Now, in Jarvis Patrick Clustering algorithm, we place two nodes in a cluster if they have more than equal to t neighbors, such that t is the user input that we, give. Here, if we consider t to be 3, the cluster that we would get would be like the below graph. Nodes x_4 and x_5 are forming a cluster here.

The space complexity is $O(km)$, it's because of the fact that the complete similarity matrix is not stored. The time complexity is $O(m^2)$, since k-nearest neighbor requires a computation of $O(m^2)$. It works very well incase of high dimensional data. The below figure shows the clusters which where formed with various values of k_t. Clearer clusters are visible with larger k_t.

Betweenness Centrality: Betweenness centrality represents a measure to which a vertex or an edge occurs on the shortest path between all the other possible pairs of nodes in a graph.

Vertex betweenness: in a graph indicates the highly central nodes in a network. Centrality here refers to the numbers or rankings to nodes corresponding to their network position. For a node d, it refers the number of shortest paths that run through node d.

Edge betweenness: It refers the number of shortest paths that run through a given edge connecting two nodes.We will discuss Girvan and Newman Algorithm here.

Vertex Betweenness Clustering : Given a graph G, we first calculate the betweenness of all existing vertex in the graph and than select the vertex with the highest betweenness. Highest betweenness is generally represented in a range from [0-1]. We now select the vertex with highest betweenness and disconnect the graph. Now, we copy the vertex to the disconnected components which are made. Lastly, we repeat until highest vertex betweenness is less than equal to p, where p is value from 0 to 1.

Edge Betweenness Clustering : Here, we will use Girvan and Newman Algorithm to have a better understanding. Firstly, we will calculate the edge betweenness for each edge. If multiple shortest path exists between between pair of nodes each path is assigned equal weight such that total weight is equal to one. The edge with the highest betweenness is removed and the betweenness of all edges affected by the removal is calculated. We repeat it until highest edge betweenness is less than equal to q, where q is value from 0 to 1, which would finally result in a *dendogram*, a diagram representing a tree.

Maximal Clique Enumeration: Let us consider a graph G.

Clique(C) refers to a subgraph C of a graph G with edges between all pairs of nodes. A maximal clique is a clique that cannot be extended by including one more adjacent vertex. Now, we understand the Bron and Kerbosch Algorithm for better understanding. The algorithm is used for finding maximal cliques in an undirected graph. It takes into consideration the vertices in current clique(C), vertices that can be added to C(P) and vertices that cannot be added to C(N). Our main aim is to find the maximal clique. Initially, both P and C are empty. The output is a maximal clique. It takes time complexity of $O(3^{n/3})$ and space complexity of $O(n^2)$ in worst cases.

Kernel k-means clustering : K-means is a clustering algorithm that uses datapoints having both magnitude and direction. Firstly, we consider the number k which would decide the number of clusters. Than we select random k points, and start assigning data points to their closest centroid which would form the predefined k clusters. Now we calculate again and place a new centroid of each cluster. Now, we repeat the steps and reassign each datapoint to the new closest centroid of each cluster. Now, in Kernel k-means we simply use the within-graph kernel function to calculate the inner product of a pair of vertices in a user-defined feature space. Within-graph kernel function is used inplace of distance measures of k-means. Kernel k-means is able to find "complex" clusters.

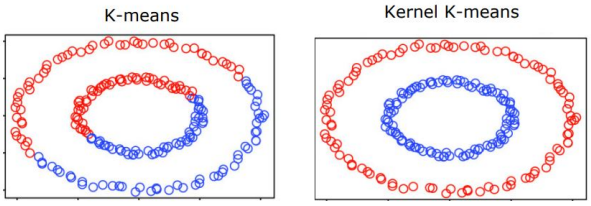

Fig. 6. Kernel k-means clustering

5. SUMMARY

Many graph clustering techniques have been developed in the past. Each algorithm has its own set of advantages and disadvantages. Because of its superiority, Markov Clustering has been highlighted frequently nowadays. However, it is biassed towards discovering a large number of extremely small clusters and fails to detect many larger clusters. Experiments on data sets have shown that the MST based Clustering and Spectral Clustering method outperforms the kmeans clustering algorithm by a wide margin. The drawback of MST based Clustering is that they have high computational complexity. Spectral Clustering algorithm is effective for different shapes of cluster and for the sparse data it is computationally faster. Hierarchical Agglomerative Clustering is quite straightforward and easy to understand. Also it is not necessary to know the number of clusters ahead of time in this algorithm but the disadvantage of this approach is that it is too slow when working with large datasets. Shared Nearest Neighbor(SNN) Clustering with the functioning of Jarvis Patrick Clustering algorithm is exceptionally good. In Betweenness Centrality we have seen that the role of centrality and shortest path passing through the node or vertex helped us get the clusters in an easier way. The Bron and Kerbosch Algorithm made it possible to find the maximal cliques followed by the Kernel k means which helped us calculate complex clusters using inplace of distance measures of k-means.

6. REFERENCES

[1] O. Grygorash, Y. Zhou and Z. Jorgensen, "Minimum Spanning Tree Based Clustering Algorithms," 2006 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06), 2006, pp. 73-81, doi: 10.1109/ICTAI.2006.83.

[2] M. Makrehchi, "Hierarchical Agglomerative Clustering Using Common Neighbours Similarity," 2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI), 2016, pp. 546-551, doi: 10.1109/WI.2016.0093.

[3] Verma, Deepak, and Marina Meila. "A comparison of spectral clustering algorithms." University of Washington Tech Rep UWCSE030501 1 (2003): 1-18.

[4] R. A. Jarvis and E. A. Patrick, "Clustering Using a Similarity Measure Based on Shared Near Neighbors," in IEEE Transactions on Computers, vol. C-22, no. 11, pp. 1025-1034, Nov. 1973, doi: 10.1109/T-C.1973.223640.

[5] Daniel, C., Furno, A., Goglia, L. et al. Fast cluster-based computation of exact betweenness centrality in large graphs. J Big Data 8, 92 (2021). https://doi.org/10.1186/s40537-021-00483-1

[6] Eblen, J.D., Phillips, C.A., Rogers, G.L. et al. The maximum clique enumeration problem: algorithms, applications, and implementations. BMC Bioinformatics 13, S5 (2012). https://doi.org/10.1186/1471-2105-13-S10-S5

[7] Antoro, S. Sugeng, Kiki Handari, Bevina. (2017). Application of Bron-Kerbosch algorithm in graph clustering using complement matrix. AIP Conference Proceedings. 1862. 030141. 10.1063/1.4991245.

[8] Chitta, Radha. Kernel Clustering. http://www.cse.msu.edu/ cse902/S14/ppt/kernel-Clustering.pdf. PowerPoint Presentation.

[9] Van Dongen, Stijn. "Performance criteria for graph clustering and Markov cluster experiments." NATIONAL RESEARCH INSTITUTE FOR MATHEMATICS AND COMPUTER SCIENCE IN THE. 2000.