Graph Clustering of Social Networks on Map Reduce

Jinyu Zhou

Abstract— **Graph clustering is an important technique in graph data mining. It plays an important role to reveal community clusters, hubs and outliers in social networks. There are several graph clustering algorithms have been proposed based on the SCAN algorithm. For large graph, SCAN and its improved algorithms are slow due to the limitations on a single node. Therefore, in this paper, we proposed a graph clustering algorithm based on SCAN using the widely used MapReduce framework on social networks. Moreover, we conducted experiments on real world data sets to justify the feasibility of the proposed graph clustering algorithm.**

Index Terms—**Graph clustering, data mining, social networks, MapReduce.**

I. Introduction

In the current era of increasingly developing technology, many things in our life can be compressed into data information, such as weather information, digital video from surveillance cameras, mobile phone signals, internet pages and user posts on social media. Graph is a general model used to represent objects and their relationships in many applications, such as social networks, road maps, bioinformatics, etc.

With the rapid development of graphic applications such as social networks, information networks, collaboration networks, communication networks and biology and the explosive growth of data volume, the processing volume of graph data information is also increasing [1]. Efficient and effective management and analysis of social graph data has become the focus of research and has been widely studied. Thus, graph clustering [2][3] has important theoretical and practical significance. It can solve practical problems in many fields.

The network structure clustering algorithm (SCAN) [4] is a widely used social network graph algorithm. The advantage of this algorithm is that in addition to finding clusters in the graph, it can also find hubs and outliers in the graph. This is meaningful for community discovery in large social networks. But SCAN also has some disadvantages. The algorithm takes a long time to run on a graph with a large number data. And some improved algorithms also have the similar problems.

Hadoop cluster is the main distributed system used by many enterprises to process big data. It has remarkable scalability, the flexibility of distributed batch processing of structured and unstructured data, the fault-tolerance ability of computer system faults, and the convenience of MapReduce [5] parallel programming model. The idea of MapReduce computing model is "divide and conquer", that is, to solve problems by dividing large and complex tasks into several small tasks. They are independent and can be distributed to multiple nodes to achieve high parallel computing capability.

Based on the above practical background and existing problems, this paper proposes the graph clustering of social networks algorithm on MapReduce to speed up the graph clustering.

II. Related Works

Network, also known as graph, its earliest research began in 1736 with Euler's seven bridges in Konesburg. In fact, most of the actual networks are not completely random. An important research was put forward by Girvan and Newman [6] in an article published in 2002: there is a universal clustering feature in complex networks, and each class is called a community (that is, the cluster in the clustering algorithm). Therefore, a large number of scholars have carried out a lot of research on the proposed community discovery problem and proposed many algorithms, which have been applied to many disciplines at the same time.

Graph clustering is a basic method to discover the underlying hidden structure of the network. It has been applied in various industries and fields, such as identifying cross-collaboration networks [7], identifying communities that are filling social networks, and so on. Firstly, several basic graph clustering algorithms are introduced, including graph partitioning algorithm, module-based algorithm and density-based algorithm. Graph partitioning is to divide a graph into several subgraphs. This method needs to specify the size of the segmented subgraph in advance and needs to continuously dichotomy to obtain multiple subgraphs. The graph partitioning method has many applications, such as software and hardware co-design, task allocation in parallel computing and other fields. The algorithm based on modularity divides vertices into clusters to minimize the number of links between clusters and make the number of links in clusters more dense. The algorithm has low time complexity, stable partition results, and can find irregular network shapes without specifying the number of communities in advance. However, the algorithm still has limitations, such as the lack of global objective function is easy to cause overfitting, and is sensitive to a single point. The density-based clustering method can find clusters of various shapes and sizes in noisy data. This method can find clusters of various sizes and shapes, and has certain anti-noise characteristics.

The algorithms described above are several common basic clustering algorithms at present. We will briefly introduce the SCAN algorithm used in this paper and the related work of its improved algorithm. SCAN is a representative structural graph clustering algorithm. It can not only find clusters in the graph, but also identify hubs and outliers. The algorithm uses the neighborhood of vertices as the clustering criterion, rather

Manuscript received March 11, 2023

Jinyu Zhou, School of Computer Science and Technology, Tiangong University, China

than just using their direct connection. This is meaningful when considering the detection of communities in social networks [8]. However, the algorithm is time-consuming, so some scholars have proposed relevant algorithms to solve the problem of its slow operation. Below are some popular improved structure graph clustering algorithms. Hiroaki et al. proposed the SCAN++ algorithm. Since adjacent nodes have large clustering coefficients, they only define vertex sets with vertices that are two hops away, thereby reducing the structural similarity assessment of vertices. The pSCAN algorithm maintains an upper bound and a lower bound for the number of similar neighbors of each vertex. There are also improved algorithms that apply beyond undirected network graphs. For example, the index-based approach graph structural clustering algorithm, which can be extended to dynamic network graphs. Chen et al. proposed DirSCAN and the corresponding distributed PDirSCAN for the directed interactivity and large-scale social networks. However, These algorithms are still limited by insufficient computing resources on a single node.

III. Method

In this section, we propose a clustering algorithm based on SCAN and MapReduce. We designed this algorithm to deal with social graphs using distributed storage on multiple machines. The graph clustering algorithm on MapReduce is mainly implemented in the following steps.

Step 1: The parallel computing process can be highly abstracted into two functions: *map* and *reduce*. The function *map* uses *key-value* pairs as input. The *key* and *value* in this step are two vertices in the adjacent list, and the output is the same as the input. The *reduce* function takes the result of *map* as input, and then we can obtain each vertex and all its connected neighbors through *reduce* process.

Step 2: Calculate the structural similarity of each edge and find the vertices that may be clustered together. We use the output of the previous step as the input in this step: *key* is a vertex, and *value* is its corresponding neighbor list. Each vertex in the *value* will be paired with its *key* through the *map* function, and the *key-value* is sorted according to the sequence number to get the unique edge. After the *reduce* aggregation, the two neighbor sets corresponding to the unique edges and vertices will be generated, and then the structural similarity will be calculated through the definition of SCAN structural similarity. Delete the unqualified edges, leaving the vertices and edges that meet the conditions.

Step 3: In order to further expand the cluster, we should find the core vertices. The output of the previous step is also used as the input in this step. In the *map* process, *key* is the edge that satisfy the structural similarity, and *value* is the similarity of the edge. Then, we mapping the two vertices in the *key* to form a new *key* and *value*. Therefore, in the *reduce* function, each vertex can get their structural connected vertex list. If the number of vertices in the vertex list meets the core vertex condition, then the vertex is a core. Then the core vertices and their corresponding structural neighbor list are output as the *key* and *value*.

Step 4: Output the vertex pair *key-value* (ascending order) as *key*. In the *reduce* function, the sets corresponding to the edges will be combined into a new set, and then the pairs are output as new *key*, and the new set is output as new *value*. In this step, preliminary clusters will be formed.

Step 5: In this step, we only use the *key-value* pairs to merge the set circularly. In the *map*, each vertex in the input *key* is used as the output *key*, and the input pairs are used as the output *value*; In the *reduce*, merge vertices and output the merged collections. Then iteratively execute the merge process until the result is exactly the same as the previous process. When completing each iteration, there are some duplicate values in the results, so these vertices need to be deduplicated after each iteration.

IV. Evaluation

In order to verify the feasibility of the algorithm, the clustering quality of the proposed algorithm and the existing SCAN algorithm is evaluated. This experiment proves the accuracy of our proposed algorithm in finding clusters by Adjust Rand index (ARI). The experiment is conducted on the data sets including soc-delicious, doc-gowalla and Facebook respectively. The ARI scores of the experimental results are shown in TABLE 1. From the results, we can see that the proposed graph clustering algorithm has good clustering quality and feasibility.

TABLE 1. Clustering Scores

Datasets	**ARI**
soc-delicious	0.97
doc-gowalla	0.97
Facebook	0.98

V. Conclusion

This paper proposes a social network clustering algorithm based on MapReduce by analyzing the shortcomings of existing clustering algorithms. Since the algorithm is implemented on a distributed platform, it greatly speeds up the running efficiency compared to the original algorithm. At the same time, the effectiveness of the proposed algorithm in this paper can also be verified through the evaluation of clustering quality. We will continue to improve the algorithm in the following work to make it more efficient for clustering on large social networks.

References

[1] C. Sammut and G. I. Webb, *Encyclopedia of machine learning and data mining*. Springer Publishing Company, Incorporated, 2017.

[2] S. E. Schaeffer, “Graph clustering,” *Computer Science Review*, vol. 1, no. 1, pp. 27–64, 2007.

[3] M. Chen, J. Lin, X. Li, B. Liu, C. Wang, D. Huang, and J. Lai, “Representation learning in multi-view clustering: A literature review,” *Data Science and Engineering*, pp. 1–17, 2022.

[4] X. Xu, N. Yuruk, Z. Feng, and T. A. Schweiger, “Scan: a structural clustering algorithm for networks,” in *SIGKDD*, 2007, pp. 824–833.

[5] J. Dean and S. Ghemawat, “Mapreduce: Simplified data processing on large clusters,” in OSDI, E. A. Brewer and P. Chen, Eds. USENIX Association, 2004, pp. 137-150.

[6] Girvan M, Newman M J . Community structure in social and biological. networks.[J]. Proceedings of the National Academy of Sciences of the United States of America, 2002(12):99.

[7] W. Zhao, V. Martha, and X. Xu, “Pscan: a parallel structural clustering algorithm for big networks in mapreduce,” in 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA). IEEE, 2013, pp. 862–869.

[8] Newman, M. E. (2006). Modularity and community structure in networks. *Proceedings of the national academy of sciences*, *103*(23), 8577-8582.