Parallel Graph Clustering Based on Minhash

Byoungwook Kim[1(✉)], Jaehwa Chung[2], Joon-Min Gil[3], and JinGon Shon[2]

[1] Department of Computer Engineering, Dongguk University, Gyeongju, Korea
bwkim@dongguk.ac.kr
[2] Department of Computer Science, Korea National Open University, Seoul, Korea
{jaehwachung, jgshon}@knou.ac.kr
[3] School of Information Technology Engineering, Catholic University of Daegu, Daegu, Korea
jmgil@cu.ac.kr

Abstract. Graph clustering is a technique for grouping vertices having similar characteristics into the same cluster. It is widely used to analyze graph data and identify its characteristics. Recently, a large-capacity large-scale graph data is being generated in a variety of applications such as a social network service, a world wide web, and a telephone network. Therefore, the importance of clustering technique for efficiently processing large capacity graph data is increasing. In this paper, we propose a clustering algorithm that efficiently generates clusters of large capacity graph data. Our proposed method efficiently estimates the similarity between clusters in the graph using Min-Hash and generates clusters according to the calculated similarity. In the experiment using real world data, we show the efficiency of the proposed method compared with the proposed method and existing graph clustering methods.

Keywords: Graph clustering · Spark

1 Introduction

The graph consists of vertices and edges and has been regarded as an important data structure by modeling the relationship between the vertices constituting the graph. Graph data is used in various fields such as social network (SNS), telephone network, bio-network, world wide web (WWW) and road network [1]. Among the various techniques for analyzing graph data and extracting meaningful information, graph clustering is a technique that measures the similarity of the vertices constituting the graph and classifies the entire graph into multiple clusters so that the vertices related to each other belong to the same cluster, And graph clustering is an important technique used in various applications such as social network analysis and community detection [2], image segmentation, and protein-protein correlation.

Recently, a large amount of graph data with large number of vertices and trunks has been generated due to low price of storage media, activation of social networks, development of web technology, and high availability of various data [1]. Large-scale graph data consists of tens of small to large billions of vertices and corresponding

J. J. Park et al. (Eds.): CUTE 2018/CSA 2018, LNEE 536, pp. 393–395, 2020.
https://doi.org/10.1007/978-981-13-9341-9_67

trunks. For example, the World Wide Web (WWW) contains 50 billion web pages and more than one trillion links, and the social network Facebook also contains more than 800 million peaks and over 100 billion friends. As the size of the generated graph data increases, the clustering methods proposed in the past have a problem in that the time required for clustering a large amount of graph data increases (Fig. 1).

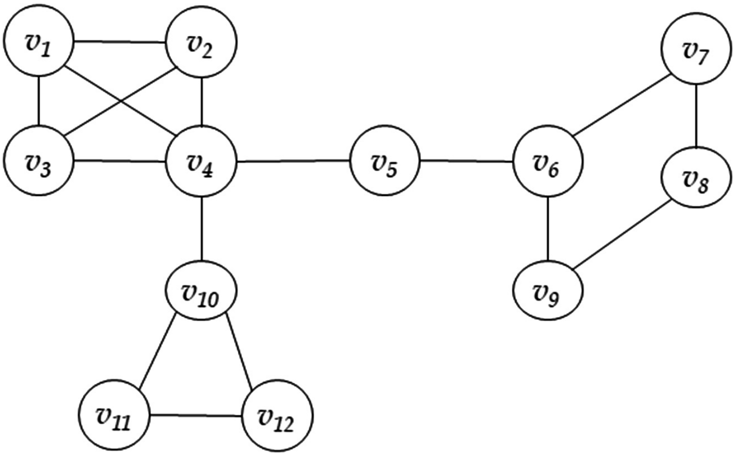

Fig. 1. A simple graph

In this paper, we use Min-Hash, a kind of locality sensitive hashing (LSH) technique, to approximate similarities between vertices in a non-directional and non-weighted large- If the similarity between vertices is greater than the user-defined threshold, the two vertices are merged into the same cluster. We also propose a technique to calculate the similarity between clusters in which vertices are merged and to merge two clusters if the similarity between clusters is larger than the user - defined threshold.

2 Related Works

2.1 Minhash

The Minhash technique is a kind of Locality Sensitive Hash (LSH), which approximates how two sets are similar. The similarity of two sets can be defined as Jaccard similarity [9] as in Eq. (1).

$$\mathrm{Sim}(\mathrm{S},\mathrm{T}) = (\mathrm{S} \cap \mathrm{T})/(\mathrm{S} \cup \mathrm{T}) \tag{1}$$

The Min-Hash technique approximates Jaccard similarity of two sets S and T, and the basic principle is to map the set S to one of the elements belonging to the set S. To map a set to an element, a hash function is used in the Min-Hash technique. The value mapped to one element is called the Min-Hash value, and all the elements belonging to the set S are substituted into the hash function, and the smallest value is the Min-Hash value. At this time, the probability that the Minhash value of both sets S and T is the same is equivalent to Jaccard similarity.

3 Parallel Graph Clustering

In this paper, we propose a method to efficiently calculate similarity using Min-Hash and to cluster graph data based on it. The first step is the preprocessing step (line 3 in Fig. 3) to generate signatures to perform clustering. The second step (line 4 in Fig. 3) performs clustering using the signatures generated in the preprocessing step.

Various clustering techniques have been proposed to extract meaningful information from graph data. However, the conventional techniques require many operations in order to determine the degree of similarity between vertices, and thus clustering can not be efficiently performed in a large amount of graph data.

Initially, each cluster is considered to contain only one vertex, and it is judged whether or not it can be merged into the same cluster through calculation of similarity between clusters and clusters.

4 Conclusions

In this paper, we propose an efficient graph clustering technique using Min-Hash. Since the min-hash technique approximates Jaccard similarity and the similarity between vertices in graph data can be represented by Jaccard similarity, we propose a technique to efficiently calculate similarities between vertices using Min-Hash. In addition, we show that the merged clusters can effectively generate the signatures of the merged clusters. Experiments show that the proposed algorithm performs clustering at a high speed, while the quality of clusters shows good results.

References

1. Kang, U., Faloutsos, C.: Big graph mining: algorithms and discoveries. ACM SIGKDD Explor. Newslett. **14**(2), 29–36 (2012)
2. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E **69**(2), 026113 (2004)