Data & Knowledge Engineering 104 (2016) 17–31

Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

Editorial

# Parallel community detection on large graphs with MapReduce and GraphChi

Seunghyeon Moon [a], Jae-Gil Lee [b,*], Minseo Kang [b], Minsoo Choy [b], Jin-woo Lee [b]

[a] KAIST Institute for IT Convergence, 291 Daehak-ro, Yuseong-gu, Daejeon 305-701, Republic of Korea
[b] Department of Knowledge Service Engineering, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 305-701, Republic of Korea

ARTICLE INFO

Available online 8 May 2015

*Keywords:*
Clustering, classification, and association rules
Mining methods and algorithms
Community detection
Social networks
MapReduce
Vertex-centric model

ABSTRACT

Community detection from social network data gains much attention from academia and industry since it has many real-world applications. The Girvan–Newman (GN) algorithm is a divisive hierarchical clustering algorithm for community detection, which is regarded as one of the most popular algorithms. It exploits the concept of *edge betweenness* to divide a network into multiple communities. Though it is being widely used, it has limitations in supporting large-scale networks since it needs to calculate the shortest path between *every pair* of vertices in a network. In this paper, we develop two parallel versions of the GN algorithm to support large-scale networks. First, we propose a new algorithm, which we call *Shortest Path Betweenness MapReduce Algorithm* (SPB-MRA), that utilizes the MapReduce model. Second, we propose another new algorithm, which we call *Shortest Path Betweenness Vertex-Centric Algorithm* (SPB-VCA), that utilizes the vertex-centric model. An approximation technique is also developed to further speed up community detection processes. We implemented SPB-MRA using Hadoop and SPB-VCA using GraphChi, and then evaluated the performance of SPB-MRA on Amazon EC2 instances and that of SPB-VCA on a single commodity PC. The evaluation results showed that the elapsed time of SPB-MRA decreased almost linearly as the number of reducers increased, SPB-VCA outperformed SPB-MRA just on a *single* PC by 4–6 times, and the approximation technique introduced negligible errors.


## 1. Introduction

As social networking services (SNSs) such as Facebook and Twitter are getting more popular, analyzing social network data has become one of the most important issues in various areas [1]. Among those analysis jobs, community detection from social network data gains much attention from academia and industry since it has many real-world applications such as friend recommendation and target marketing [2,3].

*Community detection* is to partition the set of network vertices into multiple groups such that the vertices within a group are connected densely, but connections between groups are sparse [4]. There have been many studies regarding community detection [5]. The Girvan–Newman (GN) algorithm proposed by Girvan and Newman [6] exploits the concept of *edge betweenness*, which is a measure of the centrality and influence of an edge in a network. Though the GN algorithm is being widely used, it has limitations in supporting large-scale networks since it needs to calculate the shortest path between *every pair* of vertices. The number of vertex pairs in a large-scale network is really prohibitive.

* Corresponding author.
*E-mail addresses:* myth624@kaist.ac.kr (S. Moon), jaegil@kaist.ac.kr (J.-G. Lee), minseo@kaist.ac.kr (M. Kang), minsoo.choy@kaist.ac.kr (M. Choy), jinwoo.lee@kaist.ac.kr (J. Lee).

http://dx.doi.org/10.1016/j.datak.2015.05.001
0169-023X/

In the era of Big Data, the amount of available data is growing unprecedentedly. Thus, data analysis calls for very scalable methods that can cope with huge data sets. MapReduce is a programming model for processing large data sets with a parallel, distributed algorithm on a cluster. MapReduce has been widely used owing to its scalability and ease of use [7–11]. It has been the driving force behind big data analysis in recent years. In addition, as graph (or network) data become more prevalent, totally new parallel computing platforms based on the vertex-centric model are being developed especially for graph data [12–14].

In this paper, we develop two parallel versions of the GN algorithm to solve its scalability issues. First, we propose a new algorithm, which we call *Shortest Path Betweenness MapReduce Algorithm* (SPB-MRA), that utilizes the MapReduce model. Second, we propose another new algorithm, which we call *Shortest Path Betweenness Vertex-Centric Algorithm* (SPB-VCA), that utilizes the vertex-centric model [12]. In addition, we suggest an approximation technique to further speed up community detection processes. Our preliminary work [15][1] contains the former algorithm only, and this paper contains the latter algorithm as well.

Our first algorithm, SPB-MRA, consists of four major stages, and all operations are executed in parallel *on a cluster*. In the first stage, all-pair shortest paths on a network are calculated. In the second stage, the edge betweennesses of all edges in the network are calculated. In the third stage, $k_{iter}$ edges are selected by edge betweenness, and they get removed. In the final stage, the network is updated, and this new network is provided to the next iteration. These four stages repeat until the quality of communities does not improve any more. SPB-MRA is implemented on top of Apache Hadoop [16].

Our second algorithm, SPB-VCA, consists of three major stages without the last one of SPB-MRA. That is, the update of a network after an edge removal is *not* explicitly necessary. SPB-VCA is implemented on top of GraphChi [14] and thus runs on a *single* PC.

SPB-VCA has two main advantages over SPB-MRA by virtue of the vertex-centric model. (i) Since the algorithm is represented directly on a graph, which is the underlying data structure, it looks more natural and intuitive. On the other hand, in SPB-MRA, a graph is decomposed into its constituent edges, and the algorithm manipulates the edges (i.e., key-value pairs). Accordingly, there is no need to generate and merge key-value pairs in SPB-VCA. (ii) As a result, the performance of SPB-VCA is dramatically higher than that of SPB-MRA.

The major contributions of this paper are as follows.

- We propose two algorithms—SPB-MRA and SPB-VCA—and demonstrate performance improvement. The results of performance tests showed that the elapsed time of SPB-MRA decreased almost linearly as more reducers were added to a cluster. In addition, we confirmed that the vertex-centric model, which is dedicated to graph data, allows us to achieve much higher performance than the MapReduce model. In fact, SPB-VCA was shown to outperform SPB-MRA by 4–6 times just on a *single* PC.
- We suggest an approximation technique to further speed up community detection processes. Instead of removing a *single* edge per iteration, we remove *multiple* edges that have the top-$k_{iter}$ highest edge betweenness at once. The results of accuracy tests showed that a negligible error was introduced by the approximation.

The rest of this paper is organized as follows. Section 2 summarizes the background knowledge required for this study. Sections 3 and 4 propose SPB-MRA and SPB-VCA respectively. Section 5 presents the results of performance tests. Finally, Section 6 concludes this study.

## 2. Background and related work

### 2.1. MapReduce and Hadoop

MapReduce is a programming model for processing large-scale data in a parallel way [7]. Users can easily implement distributed, parallel processing software by writing only two functions: *map* and *reduce*. Fig. 1 shows the control flow of MapReduce. The map function processes a sub-problem for input data and emits intermediate ⟨*key*, *value*⟩ pairs. The reduce function combines the values associated with the same key and produces the final output. Apache Hadoop [16] is the most popular open-source implementation of MapReduce. However, despite its popularity for big data processing, MapReduce is known to be awkward at supporting iterative graph algorithms [17].

### 2.2. Vertex-centric model and GraphChi

The vertex-centric model is a programming model for iterative graph computation. It is easy to program since a programmer just needs to "think like a vertex." Thus, this model has been adopted by parallel graph computing platforms, including Pregel [12], GraphLab [18], and GraphChi [14]. In the vertex-centric model, every vertex and edge is associated with a value, and computation is performed on a vertex by a user-specified function.

GraphChi is a disk-based system exploiting the vertex-centric model for processing graph computations on just a single machine [14]. A novel parallel sliding windows (PSW) method enables GraphChi to execute graph mining algorithms on very large graphs. The PSW method splits vertices in a graph into multiple intervals, where each interval is associated with a shard which stores all edges whose destination vertex is contained in that interval. The subgraphs divided by the PSW method are processed in three steps: 1) loading a subgraph from disk; 2) updating the values of vertices and edges; and 3) writing the updated results to disk.

Fig. 2 illustrates the operation of the PSW method on a toy graph. In this example, a toy graph with four vertices, which is split into two intervals, is used. First, the subgraph with **Vertex 1, 2** for the execution interval, **Interval 1**, is stored in a memory-shard (**Shard 1**)

[1] This work received the best paper award at BigComp 2014.