ORIGINAL ARTICLE

Communities Detection for Advertising by Futuristic Greedy Method with Clustering Approach

Ali Bakhthemmat,¹ and Mohammad Izadi²

Abstract

Community detection in social networks is one of the advertising methods in electronic marketing. One of the approaches to find communities in large social networks is to use greedy methods, because these methods perform very fast. Greedy methods are generally designed based on local decisions; thus, inappropriate local decisions may result in an improper global solution. The use of a greedy improved index with a futuristic approach can, to some extent, prevent inappropriate local choices. Our proposed method determines the influential nodes in the social network based on the followers and following and new futuristic greedy index. It classifies the nodes based on the influential nodes by the density-based clustering algorithm with a new distance function. The proposed method can improve clustering precision to detect communities by the futuristic greedy approach. We implemented the proposed algorithm with the map-reduce technique in the Hadoop structure. Experimental results in datasets show that the average of the rand index of clusters was accomplished by 99.32% in the proposed method. In addition, these results illustrate that there is a reduction in execution time by the proposed algorithm.

Keywords: futuristic greedy; similarity opinions; clustering; communities detection

Introduction

The social network is a platform based on an internet structure that several people form. People can share scientific ideas and content and news and advertising, and other information on social networks. Sharing all kinds of information can be accomplished to inform everyone for different purposes. These goals can have different types, such as campaigning for the presidency, advertising and selling products, disseminating ideas, social relations, hobbies, etc. To achieve these goals, forming groups with the same desires can be very useful. Everyone in social networks with friends and family relatives and acquainted followers can create a group. This group of acquaintances can be in a community on a social network with several shared features. One of the essential cases that people join social networking is maintaining previous relationships and forming new ones to develop their social network. People join each community, because they care about the shared interests of members of that community. Other people

join the community, because they have shared goals. Other people join the community due to the benefit of community members. People inside the community have some shared opinions. However, there is no proper definition of community because different researchers define community by comparing people based on various research topics.

Communities hold people together due to shared interests. These interests can be of people, and family relations followers create them. Others follow influential and famous people in each community; hence, influential people can result in shared tendencies.

From a social perspective, a community in social networks is a group of dependent individuals who are less dependent on individuals in other communities. Dependence between people can be in three ways. First, people can follow other people after viewing personal pages. If a person understands that he has the same interests as another person by viewing personal pages, he follows that person. Second, the number of

¹Kish International Campus, Sharif University of Technology, Tehran, Iran.

²Department of Computer Engineering, Sharif University of Technology, Tehran, Iran.

*Address correspondence to: Mohammad Izadi, Department of Computer Engineering, Sharif University of Technology, Azadi Street, Tehran 11888, Iran, E-mail: izadi@sharif.edu

shared followers between the two people can lead one person to follow another person because they are more dependent on the other. Third, influential and famous people in a community can influence different people so that these people come together in the same community (like fans of a football team, they fall into one group). People usually endorse influential person behaviors in each community. We can advertise in each community by influential people. Every influential person on a social network is an individual who has a large number of followers versus the following. This person has a significant influence on others in the community because of his popularity. For detecting these communities, it is valuable to determine one or more influential people. Hence, scientists try to discover influential people and then find communities based on these influential people and shared interests of people and shared followers for advertising.¹

In this study, we introduce the community for advertising based on people in the graph of social networks. In addition, we consider the community as a group of dependent people with shared ideas that they follow the same popular and influential people. Hence, there is a good chance that sellers might sell more products and services in that community. Therefore, one of the most important community detection applications in social networks is to find people with the same interests in marketing and advertising.

From an algorithmic perspective, community detection finds clusters in a social network graph an NP-complete problem. Scientists investigate various approximate intelligent algorithms to solve the problem of community detection in recent years. Scientists compared different community detection methods with runtime index and machine learning indexes such as accuracy, precision, etc.

Several algorithms for solving community detection have some disadvantages. For example, algorithms based on edge betweenness have no accuracy for the detection of overlapping communities.² In addition, Graph theory algorithms and hierarchical algorithms are not suitable for detecting overlapping communities and dynamic communities. Bayesian models are not efficient due to high computational time, and the clique percolation methods are not used for dynamic communities. Therefore, we present an algorithm that can solve some of the challenges in the previous methods.

We considered the following cases in designing the proposed method by the experiences of previous algorithms.

- (a) Many algorithms detect communities only based on connections between people (graph nodes) and the density of their relations, such as graph partitioning algorithms, density-based clustering algorithms, algorithms based on edge betweenness, hierarchical algorithms, etc. These algorithms efficiently detect people's connections density. We considered this approach in the proposed algorithm.
- (b) Several algorithms are not designed based on social networks in the real world. We developed the proposed algorithm based on the real world. In this world, just knowing many influential people's connections is not suitable for detecting the community because robots may have created these relations. They may be fake, such as increasing the number of views and followers on instagram social network by robots. Therefore, considering the relationship between individuals in communities and recognizing influential people is significant. Shared followers can detect the relationship between individuals of a community among those people.³ In these methods, the behavior of people cannot be traced based on the following of other people. Discovering shared ideas between people, along with shared followers, can create communities with the same ideas.

Having the same opinions and following influential people in these communities can be a reasonable goal for advertising products or services. This is because buying products by one member of each community encourages other members to buy that product. These communities create trust in the product's use, because they create the need to use the product by having shared interests. This approach satisfies the two main conditions of advertising, trust, and feeling the product's need. Therefore, we used a greedy method with this approach for advertising. In some clustering methods, it is necessary to determine the number of clusters before implementing the algorithm. Therefore, our proposed algorithm forms communities based on shared followers and influential people. Members of these communities have the same approximated interests due to shared followers between them.

- (c) Some methods do not need to determine the number of clusters before the clustering operations. The proposed method tries to build communities by finding influential people and their

followers related to each other. Therefore, the proposed method gradually forms communities. In this method, there is no need to determine the number of communities at the beginning of the algorithm.⁴

- (d) The previous greedy methods are very fast for big social networks, but they sometimes result in improper approximate solutions. Some local choices obtain improper solutions. The futuristic approach to greedy decisions eliminates inappropriate local choices to some extent. Therefore, futuristic greedy indexes (FGs) can be useful for greedy methods. As a result, the proposed method avoids greedy inappropriate local choices.
- (e) In several methods, with the growing number of members on social networks, the volume of data associated with social networks increases, and community detection becomes more complex.⁵ There are two main challenges in community detection with big data. The first challenge is the precision decreasing of the discovery of independent communities by the person's interests. In this challenge, there may be people with similar interests in two or more communities. Another challenge is increasing the execution time of algorithms for detecting communities, because the processing of big data from social networks is not possible with traditional methods. For this reason, we used map-reduce programming in the proposed algorithm with the Hadoop structure.⁶

Related Work

Communities can be detected by using different approaches. The accuracy and execution time of community detection depended on approaches to solving the problem; for example, the cut-based approach deals with balanced group segregation. Clustering approaches are implemented based on the density of groups. Other approaches were performed by influential nodes and similar profiles between nodes.^{7,8} In this section of this article, we have classified community detection methods based on the problem-solving approach.

Graph partitioning methods

A community is a sub-graph of a social network whose nodes have the approximated similar interests. Hence, community structure analysis helps in relationship de-

termination between nodes in the social network. We can define the relationship between nodes by edge weight. Partitioning of edges can create sub-graphs (community) with high edge density. It can also create a very low density of the edges of the communication between the subgroups. This partitioning structure makes the sub-graphs known as communities. The graph-based approach with the cut-based problem function is used to detect communities. Graph-based techniques are often applied for undirected networks. This approach results in community detection that is based on the similarity between nodes in a community. In addition, it minimizes related external edges between communities by minimizing the cut-based problem. In this approach, the internal number of edges in the community is very high.^{9,10}

An approach of graph partitioning is investigating the relationship between the graph structure and clustering effectiveness, and developing a heuristic partitioning algorithm that is suitable for modularity-based algorithms.¹¹

Another approach of Graph partitioning is applied for finding overlapping communities by the Split betweenness (e.g., CONGA algorithm). This algorithm is the development of the basic Newman Girvan algorithm. It considers the index of edge betweenness of all edges and nodes and vertex betweenness of all those vertices. Then, it deletes the edge with the highest edge betweenness or splits the vertex with the highest split betweenness.⁴

Clustering methods

The density-based clustering approach is designed based on grouping the nodes in clusters such that nodes in a cluster are distant from other nodes in other clusters. In addition, in this approach, the densities inside the clusters are high, and the distance between each node of one cluster and that of the other clusters is very high. One of the methods for density-based clustering is the use of hierarchical structure. In this method, similar nodes are determined by the hierarchical structure of node features. Hierarchical clustering was used by edge-weighted similarity for community detection. First, this algorithm exchanges user tags and then compares the tags based on edge-weighted similarity, and the similarity of the users is diagnosed based on followers and followings.¹²

Another method of clustering is known as spectral clustering. In this method, continuous learning of node features in the social network, and mapping nodes to

other nodes are accomplished in low-dimensional space. This learning can calculate the similarity between two nodes. As a result, communities are detected by the grouping of similar nodes.¹³ A modern spectral clustering algorithm consists of three steps: (1) regularization of an appropriate adjacency or laplacian matrix, (2) a form of spectral truncation, and (3) a k-means type algorithm in the reduced spectral domain.¹⁴

Several recent research works improve clustering speed. One of these methods was to improve the clustering speed based on the online storage and access scheme. In this method, increasing the speed of access to information and reducing the number of links computation between the two nodes decreased the execution time.¹⁵ Another method to improve the speed is consensus utilization in clustering. This method is based on the consensus matrix, which calculates some of the nodes instead of computing the total nodes, and therefore runtime reduces.¹⁶

Another algorithm is designed based on link clustering. Since links represent unique connections among nodes, the link clustering will detect groups of links with the same features. Thus, nodes may be placed in multiple communities.^{17,18} Other research proposes a hybrid method for community detection by improved label propagation and fuzzy C-means. The results illustrated that this method could achieve better accuracy on a real network.¹⁹ KLFCM model fuzzy C-means is one of the clustering methods that is used in community detection. The fuzzy C-means algorithm cannot guarantee good clustering results in some special conditions. Later, Pirim et al. suggested an improved fuzzy objective function KLFCM.²⁰

Greedy-based method

The greedy method for community detection is grouped into two categories: traditional and new greedy methods. In recent years, new greedy-based methods have been applied, such as heuristic algorithms, crowd-based greedy methods etc.

The use of heuristic methods to detect communities is designed based on modularity. There are three categories of greedy approaches for community detection methods—divisive, agglomerative, and diffusive. Divisive approaches initially consider all the nodes in a network to be in the same community and then recursively split them into smaller communities based on a criterion. The divisive approach detected communities in networks by using edge betweenness as a critical measure for clustering. The edges connecting nodes from

different communities have a higher betweenness value than those connecting nodes within a community.

On the other hand, each node can be considered as a separate community at the outset by agglomerative clustering approaches. Then, agglomerative clustering approaches iteratively combine pairs of closest communities until there is no improvement in the chosen performance measure (such as modularity). Generally, agglomerative clustering algorithms are computationally expensive. In diffusive methods, labels are assigned to different nodes in the social network. Labels to form communities group the nodes.^{4,21}

In crowd-based greedy methods, node communication is used to form communities. These methods are evaluated by using the concept of modularity of constructed communities. Some methods, such as Particle Swarm Optimization, in addition to crowding, determine the basis for influential nodes in communities.²² Several methods apply to crowd trends for identifying the community (such as the artificial fish swarm algorithm).²³

Innovative greedy algorithms designed in recent years, for example, the fast greedy algorithm, are designed based on an agglomerative hierarchical clustering method that merges the pair of communities at each step to produce large communities. This algorithm has good complexity. Likewise, Tandon et al. developed an agglomerative heuristic, the Louvain method, based on maximizing modularity. Louvain algorithm is another greedy algorithm applied on a weighted and unweighted graph.¹⁶ It presents the high density of the links inside the communities compared with connections between the communities. Louvain algorithm has two phases, which are executed repeatedly until the maximum modularity is gained. In phase 1, it locally optimizes the modularity by searching a “small” community. In phase 2, it combines the nodes in the same community and creates a new network whose nodes are communities. Phase 1 and phase 2 are executed repeatedly. Once the maximum of modularity is obtained, the algorithm terminates.²⁴ Subsequently, many researchers used block modeling for improving greedy algorithms. The goal of block modeling is to reduce a big complex network to a smaller one. It cannot detect groups of highly linked units in a social network until considering a lot of global structure information.²⁵

User behavior-based

Network topology cannot explain social network behavior. For this reason, the behavior of users on social networks became important to detect communities.

The behavior of the users develops communities by influential users (nodes). Therefore, influential nodes make network behaviors and communities. In addition, influential nodes as leaders form communities. Also, people's behavior in a social network, such as following, viewing, and so on, can relate people to each other.^{26,27}

Some user-behavior methods consider content information in addition to social network links. A multi-view of the community is formed by identifying similar nodes based on link and content information. For example, combining two methods, agent-based modeling and social network analysis, is a proper hybrid method for detecting human interactions in social networks. Integrating these methods presents a proper view of human behavior.^{3,28}

Several methods present approaches for marketing based on user behavior in the graph. These methods developed the recommendation system. They introduced nodes as users on social networks. In the graph of social networks, similar nodes are as near-nodes. Also, the group of similar nodes introduces as a community. These communities were based on the behavior of users' requests and were used to recommend the purchase of other goods or services or advertisements. The community detection is implemented by finding nodes with the highest centrality and forming communities based on them. The new method is designed based on a parallel algorithm (shared + distributed memory), because the parallel structure in social networks can significantly reduce the processing time of big data.²⁹

Community quality optimization-based methods

Optimization-based methods detect a community structure by exploring the optimal solution by maximizing or minimizing a goal function while satisfying a set of constraints imposed on the social network. In this method, the problem converts to an integer-programming model. This model can explain a clustering model of a social network. The model output comprises partitions of a social network graph that maximizes or minimizes goal function (e.g., modularity maximization). This method does not recommend large graphs due to high execution time.^{2,30} Some community search methods are metric-based. They are classified according to different criteria as k-core, location, k-truss, k-clique, K-ECC, etc.⁸

We can consider influence maximization for goal function in optimization-based methods.³¹ Influence maximization is an optimization problem to discover

a group of nodes in a social network that has a maximum influence considering a propagation model. Kempe and colleagues have illustrated that this problem is an NP-hard problem. The TI-SC algorithm is introduced for the problem of influence maximization. The TI-SC algorithm selects the influential nodes by exploring the relationships between the central nodes and the scoring ability of other nodes. This algorithm has proper performance in high Rich-Club networks.^{32,33}

Innovative methods

Hybrid approaches can sometimes detect communities better than previous methods. A hybrid method of spectral clustering operates by using the distance graph and node feature. In this way, nodes are grouped by their features and their relationship to communities. In this method, the similarities of features are calculated by the Gaussian kernel function.^{30,34} Other hybrid approaches detect communities in GEO-Social networks by spatial and social relationships among users.³⁵

Another hybrid method is ISOFDP. The hybrid method ISOFDP is designed based on FDP. The FDP has little ability to detect cluster centers and several clusters. The ISOFDP method first reveals the similarity of different pairs, and then communities are detected. In this method, the partitioning function improves clustering by using the density function, which automatically finds the appropriate number of communities.³⁶ Another hybrid method performs large graph partitioning. This algorithm forms subgroups based on topology and node features with the attraction-force similarity structure of the communities. In this model, the clustering process uses the k-means algorithm.²¹

Innovative methods are presented in recent years as link prediction methods. Link prediction methods anticipate the likelihood of a future connection between two nodes in a given network. The methods are essential in social networks to infer social interactions or to suggest possible friends to the users. Rapid social network growth triggers link prediction analysis to be more challenging, especially with the significant advancement in complex social network modeling. Researchers implement numerous applications related to link prediction analysis in different network contexts such as dynamic network, weighted network, heterogeneous network, and cross-network. However, link prediction applications, namely, recommendation system, anomaly detection, influence analysis, and community detection, become more strenuous due to network diversity and complex and dynamic network contexts. Several

reviews on link prediction were published to discuss the algorithms, state-of-the-art applications, challenges, and future directions of link prediction research.³⁷

Another innovative method is the affinity propagation (AP)-based method. The AP-based method depicts good results for community detection in social networks. It does not need any predefined information, such as the number of communities. The development AP method uses an adaptive similarity matrix, namely AP with adaptive similarity matrix. This algorithm is efficient for large networks.³⁸

Big graph methods

Data related to large social networks cannot be processed in a single machine due to inadequate resources in memory and processor. We can process big data of social networks by distributed computation platforms. One of the distributed computation methods is parallel processing in the Hadoop structure by the map-reduce technique. In this method, the big data process is divided into two phases. In the first phase, big data is classified as mappers. Then, mappers are assigned to reducers. In the second phase, reducers process data. Traditional algorithms for community detection in large social networks run in high execution time. In one of the last researches, Fast algorithms compare with each other in large-scale graphs. These algorithms are Walk Trap, Edge betweenness, Random Walk Fast Greedy, Infomap, and Leading Eigen Vector. Various metrics such as Modularity, Normalized Mutual Information, Adjusted Rand Index, and Rand Index are applied to discuss the performance. Results illustrate that the Random Fast Greedy algorithm has enhanced performance in comparison with the other techniques.⁶

In recent years, indexing to reduce the time to identify communities in large graphs has been considered; for example, D-Forest investigates a community search over directed graphs query to be finished within proper execution time. It proposed appropriate index construction methods. Experiments illustrated that an index-based query algorithm performs efficiently.²⁵

In other research, among the algorithms (SLPA, TOPGC, SVNET, CFINDER, LE, LPA, walktrap, Fast greedy, DSE, and Conclude) for community detection, TopGC has the fastest execution time. CFinder is very computationally expensive and thus may not be appropriate for large-scale networks. The execution time of each algorithm depended on using the number of edges and vertices in algorithm complexity. Datasets

can vary the complexity of the algorithm execution time due to different sizes of edges and nodes.³⁹

In other research, researchers investigate more than 30 research articles, which consider community search, published between 2010 and 2019. They discuss different metrics. If graphs are stored in a single machine's memory, then the algorithm often runs on low execution time. We can process big data by developing algorithms based on distributed computation platforms to process it in a cluster. To make parallel algorithms more efficient, use the main memory properly, and use as little I/O transfer possible.⁸

By summarizing the several methods for community detection, we try to use a new greedy approach with a Hadoop structure for big data processing by indexing due to low execution time. For indexing, we determine a group of people with shared opinions by a density-based clustering approach. Therefore, we presented a hybrid method that consists of a behavior-based approach, a greedy approach, and a density-based clustering approach. In the proposed method, we improved the greedy algorithm by a futuristic approach and performed big data processing on the Hadoop structure. Besides, we examined the effect of a futuristic greedy approach in reducing execution time in reducers and obtained good results.⁴⁰ Table 1 introduces the used notations in this research.

Problem Definition by Integer Programming Approach

We can classify the methods of community detection by integer programming. Each integer-programming model consists of an objective function. Integer programming models for clustering have applications in several fields, such as community detection and market segmentation, and facilities location. They are flexible in implementation by changes of constraints or objective function in the several clustering problems. Pirim

Table 1. Notations

Notation	Description
$N(i)$	i th node of social networks graph
$f^j(i)$	j th follower of $N(i)$
$f(i)$	Followers of $N(i) = \{f^1(i), f^2(i), \dots\}$
follower(i)	Number of followers of $N(i)$
following(i)	Number of people that are followed by $N(i)$
follow(i , centroid)	Node i follows centroid
follower(centeroid)	Number of followers of influential node (centroid)
FI(i)	Followers index of $N(i)$
FI($f^j(i)$)	Followers index of $f^j(i)$
FG(i)	Futuristic greedy index of $N(i)$
Centroid	Central node or influential node

et al. presented a novel integer-programming model for the clustering of community detection. Their model is applied to several social network datasets to show its ability to detect community detection. This model developed clustering relation in networks. That model is designed to create high density and separated clusters. In a network, the distance between any two clusters (community) is measured by the number of links that connect these two clusters. Therefore, we present the integer-programming model for community detection for advertising based on the proposed method of Pirim et al.^{4,20}

We present an Integer programming model with a clustering approach. The objective function illustrates the definition of clustering. It minimizes the number of output connections of the cluster with other clusters (r_m^o). Therefore, (r_m^o) is the number of connections of each node of a cluster with the maximum number of connections to nodes in other clusters. Also, the distance between members in each cluster decreases, and the density of clusters increases (L_m). Therefore, (L_m) is the maximum diameter length of all cluster diameters. The minimization of maximum diameter in a cluster indicates that the total distance between the nodes is minimal.

$$\text{Minimize } (L_m + r_m^o)$$

Subject to:

$$l_m \geq l_{ij}(x_{it} + x_{jt} - 1) \quad \forall i, j, t, (i < j)$$

$$\sum_{t=1}^C x_{it} = 1 \quad \forall i$$

$$\sum_{i=1}^n x_{it} \geq 1 \quad \forall t$$

$$\sum_{j=1}^n (P_{ij}x_{jt}) \geq \left(\frac{\sum_{j=1}^n P_{ij}}{2} \right) x_{it} \quad \forall i, t$$

$$\sum_{j=1}^n (P_{ij}x_{jt}) \geq \left(\sum_{j=1}^n P_{ij} \right) x_{it} - r_m^o \quad \forall i, t$$

$$x_{it} = 0 \text{ or } 1 \quad \forall i, t$$

x_{it} is the decision variable. It composes of two values, 0 or 1. If x_{it} is 1, it means that node i is placed in cluster t . In this model, the number of nodes (n), the number of clusters (c), the shortest path distances between nodes i and j (l_{ij}) are considered. Also, the

connection status of nodes i and j shows (P_{ij}). If the value of (P_{ij}) is 1, it means that nodes i and j are connected directly.

The first constraint of the proposed mathematical model depicts l_m as the maximum diameter among other diameters of each cluster. The second constraint ensures each cluster has at least one node and each node is placed only in one cluster. The third constraint explains that each cluster has at least one node. The fourth constraint depicts that a node has at least as many connections with nodes inside its cluster as well as the outside. Therefore, a node has at least as many connections with nodes inside its cluster as well as the outside, that is, the term on the left is the number of connections that node i has with other nodes in the same cluster, and the term on the right is half the total number of connections for nodes i . The fifth constraint shows that r_m^o is greater than the number of connections that any node consists of other than external nodes in its own cluster. The sixth constraint explains that x_{it} is a binary value. The values of this variable specify which cluster is assigned to each node such that the objective function is minimized.

In this research, nodes and clusters are considered as people and communities, respectively. The center of clusters is considered to the influential individuals with the number of followers. In the proposed method, (L_m) is considered by similar opinions index (SO). The SO is designed based on the relationship between people with the number of shared followers and the not-shared followers. This index causes clusters with a high density to be selected. Also, (r_m^o) is considered by the FG. This index selects an influential person with noninfluential followers. This index causes clusters to be selected approximated independently of each other. The SO and FG concepts are fully explained later. We introduce SO and FG by using a clustering approach in mathematical programming and behavior in social networks so that real communities for advertising can be detected in the real world.

Proposed Method

People grouped by their same opinions and friends can form a community. Some people in one community may be in other communities, because they have shared opinions in different communities. These shared people have a joint boundary between the communities.⁴¹ Changing the different opinions of these shared people to one community's opinions can create an approximated independent community, because the joint boundary of similar opinions between the communities is diminished.

Communities can be formed based on the connections of both nodes, and gradually other nodes join them. These two-node connections can determine the different conditions for the formation or nonformation of communities. These conditions can create different situations for two nodes. In any social network of nodes, the following situations may occur between the two nodes:

- (a) The two nodes have a shared follower, but the nodes do not follow each other.

In this case, the two nodes do not have much in shared interests. They do not follow each other due to behavior reasons, or the two nodes could not find each other on the social network. These nodes are not suitable for forming an advertising community, because they do not have a good relationship with each other through social networking.

- (b) The two nodes that follow each other do not have shared followers.

In this case, one node may be a fake follower for another node. These nodes do not have shared followers, so it is unlikely that they have much in shared interests. Forming these nodes in a community is not suitable for advertising, because they have very little shared interests. Therefore, communities with these nodes are improper for advertising products or services.

- (c) The two nodes follow each other and have many shared followers. Such nodes can be community members, and there are likely many shared behaviors and interests between these nodes. We use these nodes in this research for community detection.
- (d) The two nodes do not follow each other and have no more shared followers. Such nodes must be placed in different communities, because there is nothing in shared cases between them.

One method for grouping people to determine connections between two nodes is to use influential people. People in a community trust influential people, so they endorse influential people's opinions.

Influential people can affect other people in a community, because they affect forming the same ideas between people. Influential people can be joined in several communities as the center of advertising, because they can be famous and trustworthy. These communities have an influential central node that other nodes follow it, but they do not necessarily have many shared

interests. These communities are completely not suitable for marketing and advertising for two reasons. The first reason is that follower nodes usually follow the influential node's interest in a few topics; these interests do not necessarily indicate many shared interests between the follower node and the influential central node. For example, a singer as an influential node of a community has many followers. Hence, most followers are usually interested in the singer's songs. They typically do not follow other interests of influential nodes. The second reason is that there may be fake followers for influential central nodes on each community that they follow the influential node, but they do not have any shared interests with influential nodes. This study intends to detect communities through influential nodes that have shared interests between them and their followers. Also, the nodes of each community have minimal interests with the nodes of other communities. Community detection by finding influential nodes and their followers is a greedy approach based on follower index (FI). To improve the greedy approach to community detection, we consider two futuristic approaches. In the first approach, we place the influential central node and its followers in a community so that its followers are not influential. In the second approach, followers have properly shared interests with their central nodes. In this research, communities are discovered through influential nodes with many followers and a greedy futuristic approach.

Later, we introduce the FG by the clustering approach and then discuss the nodes' similarity through the shared followers with the futuristic greedy approach. Then, we calculate the distance between the nodes through similarities. Finally, using these defined indexes, we present the proposed algorithm.^{41,42}

Density-based clusters can consider communities in the graph of social networks with a centrality of influential people. People are regarded as nodes in clusters. The peoples (nodes) in these clusters are almost close together (approximately similar opinions) or are very distant (about different opinions) from other people in other clusters. Thus, we can consider several influential people for a number of clusters. Identifying influential people can improve some challenges of density-based clustering, such as determining the unknown number of clusters at the beginning of the density-based clustering algorithm because communities are extended based on influential people.

Influential people are known for their popularity with others. A person's popularity can identify based on the number of followers. This popularity [following(*i*)] is a

greedy index that we do not consider followers (futuristic greedy approach). Influential people cannot follow all their followers, because they do not have enough time to follow too many people. So the FI is determined by the Formula (1). $\text{Follower}(i)$ denotes the number of followers of the i th node, and $\text{following}(i)$ indicates that the i th node follows what number of the other nodes. This greedy index can determine influential nodes, because the higher FI for each person (node) indicates that the person has become more popular on the social network.

$$\text{FI}(i) = \frac{\text{follower}(i)}{\text{following}(i)} \quad (1)$$

We can sort the nodes in a set based on the FI. Selecting each node from this sorted set can identify approximated influential nodes. An influential node with its follower nodes can form a community. The ordered set of influential nodes can detect communities until all nodes in the social network are covered. This method is similar to the conventional greedy method, because it chooses local selections based on the FI greedy index.

In the FI, inappropriate local choices are selected when highly influential nodes are selected to cover part of a social network. These inappropriate influential nodes are selected when they have many shared followers with other selected influential nodes. This problem often occurs when the followers of an influential node are also influential. Therefore, in choosing the greedy index, we should consider a futuristic approach to use in the FI. The futuristic approach selects influential

(popular people). Similar nodes can be detected based on shared followers. The selection of similar nodes by influential nodes is achieved by the shared follower index (SFI), because nodes with shared opinions often follow each other or other nodes due to family, friends, partners, etc. The SFI between person i and person j in social networks is shown in formula (3).^{39,43}

$$\text{SFI}(i, j) = \text{follower}(i) \cap \text{follower}(j) \quad (3)$$

Detected communities should be approximately independent with a few shared followers. We can improve the SFI with the futuristic approach. In the new greedy index, we try to select a similar node to the influential node (centroid node) in a community, with many shared followers in the community and a few followers in other communities. Hence, for calculating similar opinions, we try to avoid selecting nodes with sharing followers in other communities. Also, the relationship between two nodes (persons) shows similar opinions. Hence, we introduce neighbor relation (NR). The NR of a node means whether its neighbor follows that node. The NR is calculated in terms of dividing $\text{follow}(i, \text{centroid})$ by the total of following every two nodes (TF); $\text{follow}(i, \text{centroid})$ means that node i follows the centroid. Also, the total of the two nodes is summated as $\text{follow}(\text{centroid}, i)$ and $\text{follow}(i, \text{centroid})$. We use the SO for the calculation of the similarity of two nodes. The SO is calculated by the number of shared followers and not-shared followers. The SO between node, i , and the centroid is illustrated in formula (4).

$$\begin{aligned} \text{SO}(i, \text{centroid}) = & \frac{\text{follower}(i) \cap \text{follower}(\text{centroid})}{\text{follower}(i) + \text{follower}(\text{centroid}) - 2 * (\text{follower}(i) \cap \text{follower}(\text{centroid}))} \\ & + \frac{\text{follow}(i, \text{centroid})}{\text{follow}(i, \text{centroid}) + \text{follow}(\text{centroid}, i)} \end{aligned} \quad (4)$$

nodes whose followers are less influential. Therefore, the FG is the determining Equation (2); accordingly,

$$\text{FG}(i) = \frac{\text{FI}(i)}{\sum_{j=1}^{\text{follower}(i)} \text{FI}(f^j(i))} = \frac{\frac{\text{follower}(i)}{\text{following}(i)}}{\sum_{j=1}^{\text{follower}(i)} \frac{\text{follower}(f^j(i))}{\text{following}(f^j(i))}} \quad (2)$$

Community detection needs to find similar nodes (people with similar opinions) with influential nodes

If the SO value for each node and influential node (central node) of each community increases, then the dependence between nodes in the community increases. The dependency of each node of a community with nodes of other communities also decreases.

We introduce the proposed algorithm based on three phases. In the first phase, a set of relatively influential nodes with a greedy futuristic index based on the number of followers is obtained. Then, in the second phase, influential nodes with the followers create communities provided there are many shared followers between the

influential nodes and their followers. In the third phase, border nodes (outliers) in clusters are placed in almost suitable clusters.

The proposed method's structure in Figure 1 is based on FG, SO, and it is composed of three main phases.

In the first phase (sorting nodes by futuristic greedy), the FG for each node is calculated in the Hadoop structure. Then, results are saved in HDFS (Hadoop Distributed File System). Finally, a sorted list of nodes based on FG (G-List) is obtained.

In the second phase (cluster centroid determination by the futuristic greedy method), according to the FG-List, the influential nodes (cluster centers) and communities (clusters) are determined based on the F index. The SO and FG calculate the F index. The output of this phase is the approximated clusters and their centers (C) and their outliers.

In the third phase (clustering improvement), this algorithm discovers the improper nodes (outliers) by knowing the clusters and centroids of the phase 2. Then, outliers are placed closer to existing clusters. Finally, clusters and their centers are assigned to communities and influential nodes, respectively.

Phase 1: sorting nodes by futuristic greedy

This phase consists of four steps. In the first step, each node's profile is stored in a data point, including the number of followers, the number of following, and the number of followers. Algorithm 1 depicts how the data points are stored. The complexity of Algorithm 1 is $O(N)$, that N is the number of nodes. In the second step (Algorithm 2), the data points are stored almost equally in the mappers, and the FI value is calculated for each data point. The complexity of Algorithm 2 is $O(N)$. In the third step (Algorithm 3), followers of the i th node are placed in a reducer, and thus the follower's index of the i th node is calculated. Then, FG is computed based on $FI(i)$ and $FI(f^j(i))$. Later, FG-List results from nodes based on FG by the Hadoop structure. The complexity of Algorithm 3 is $O(N)$. In the fourth step, FG-List is sorted, and Algorithm 4 saves the results of FG-List in the G-list. In this step, data are divided by R parts (R is the number of reducers). Reducers sort data in parallel. Then, the reducers merge, and the output of this step is a sorted list (G) based on FG. The complexity of Algorithm 4 is $O(N/R \log(N/R))$, where R is the number of reducers.

Algorithm 1. Storing Operations in Hadoop ($O(N)$)

```

While (node of social networks exists) //N
  number of the input data=loop iteration,
  ( $O(N)$ )
  {
     $i=i+1$ 
     $V_i = \text{Read}(N(i), \text{follower in Hadoop},$ 
      following,  $f(i)$ ) //  $V_i$  is a data point
      ( $V_1, V_2, V_3, \dots, V_n$ )
  }

```

Algorithm 2. FI Calculation of Nodes // $O(N)$

```

Number_node_mapper =  $N/M$  //  $M$  is the number
of Mappers, and  $N$  is the number of nodes
 $i=1$ 
For each Mapper ( $t=1$  to  $M$ ) //  $O((N/M) * M)$ 
  For cnt=1 to Number_node_mapper
    {
      Assign ( $V_i$ ) to Mapper( $t$ )
       $i=i+1$ 
    }
For each Mapper in parallel
  {
    Calculate  $F(i)$  for  $N(i)$  in each
    Mapper //  $O(N/M)$ 
    Store  $F(i)$  to  $N(i)$  in HDFS
  }

```

Algorithm 3. Determining FG-List // $O(N)$

```

Number_node_reducer =  $N/R$  //  $R$  is the number
of Reducers, and  $N$  is the number of nodes
 $i=1$ 
for each Reducer ( $t=1$  to  $R$ ) //  $O(R)$ 
  for cnt=1 to Number_node_reducer //  $O(N/R)$ 
    {
      Assign ( $f_j(i)$ ) to reducer ( $t$ )
       $i=i+1$ 
    }
For each reducer in parallel
  {
    Classify each reducer based on Nodes
    ( $N(i)$ ) //  $O(N/R)$ 
    Calculate ( $f_j(i)$ ) based on  $N(i)$  //  $O(N/R)$ 
    Calculate FG( $i$ ) based on  $N(i)$  //  $O(N/R)$ 
    Store FG( $i$ ) to  $N(i)$  in HDFS
  }

```

Algorithm 4. Sorting FG-List $O(N/R \log(N/R))$

```

For each mapper,
  Classify Nodes based on intervals
   $\frac{\text{Max}(FG)}{R}(i-1) < FG(i) < \frac{\text{Max}(FG)}{R}(i)$  //  $O((N/M) * M)$ 
For each reducer in parallel
  Assign classified Nodes from Mappers to
  one Reducer //  $O(N/R)$ 
For each reducer in parallel,
  Sort nodes of each reducer //  $O(N/R)$ 
  log( $N/R$ )
G=Merge(sorted list of reducers) //
G={G(1), G(2), G(3), ...} //  $O(N)$ 

```

Phase 2: determination of clusters and their centroids by futuristic greedy method

In this phase (Algorithm 5), centroids are selected from the G-list, respectively. According to Algorithm 5, we select the first G-list member with similar nodes for detecting communities. Then, that node and its similar node are deleted from the G-list.

When we delete a first member and its followers from the Main List, the G-list and the Main List are updated. We repeat these steps each time until all of the members of the Main List are deleted. Selected nodes of the G-list in these iterations are kept in a centroid array. The centroid array presents cluster centers, where these cluster centers represent influential nodes. In each of these steps, a cluster is created based on the selected centroid and its similar nodes.

Nodes of social networks are clustered based on similarity to cluster centers (SO) by the Hadoop structure's parallel clustering method (Algorithm 5). Also, this method does not need to specify the number of clusters before the algorithm running. Therefore, these cases decrease execution time. Density-based clustering methods usually require two parameters (Epsilon and MinPoints), but our proposed methods do not need these parameters. Our proposed method determines centroids when the algorithm runs; thus, the execution time of clustering decreases. Also, it can detect clusters with different shapes. Thus, different communities have resulted in these clusters. Each node's distance and its centroid in social networks are determined based on the SO in Formula (5). If the SO (similarity) between two nodes increases, the distance between two nodes decreases.

$$\text{distance}(i, \text{centroid}) = \frac{1}{\text{SO}(i, \text{centroid})} \quad (5)$$

It may be a tradeoff between FG and SO. The importance of FG or SO in detecting communities on different social networks varies. These changes are based on detecting communities and the types of social networks and their individuals. Therefore, we consider futuristic between effects of FG and SO. These are calculated by the Formula (6). α, β coefficients can be varied due to the goal of communities and personality type and their distribution in social networks. α, β can be calculated in different social networks by machine-learning methods. This discussion is out of this research scope and should be researched separately for future research on social network types.

$$F = (FG)^\alpha (\text{distance})^\beta = (FG)^\alpha \left(\frac{1}{SO} \right)^\beta \quad (6)$$

In this research, we suppose that FG and SO have the same effects. Therefore, both α, β are to be considered as equal to 1. We used the F index to select the central nodes' neighbors for forming clusters in Algorithm 5. Thus, we expect the values of FG and the distance based on SO to be less than one and to neutralize each other's side effects. If the value of F between each node and influence node is less than one, then that node is placed in a cluster with the influence node. The number of iterations while in the loop in Algorithm 5 depends on the number of selected central node neighbors. In the worst case, in each iteration, two nodes are selected. Therefore, almost every iteration while in the loop is $N/2$. As a result, the complexity of Algorithm 5 is $O(N)$.

Algorithm 5. Determination of Clusters and Their Centroids by Futuristic Greedy $//O(N)$

```

MainList=all of social network nodes
Count=1
While (Main list!=0)  $//O(N/2)$ 
{
  K=Select from G  $//G$  is sorted
  list based on FG
  Centroid(count)=k
  {neighbors}=Find neighbor nodes of k in
  parallel  $//O(N/M)$ 
  For each {neighbors} in parallel
  {
    If (F(neighbors(k),k) < 1 in parallel
    then  $//O(neighbors(k))$ 
    {
      Cluster(count)=K+
      {neighbors(k)}  $//O(neighbors(k))$ 
      G=G-k-{neighbors(k)}  $//(k)$  and NFS
      are deleted from G  $//O(neighbors(k))$ 
      MainList=MainList--k-
      {neighbors(k)}  $//O(neighbors(k))$ 
    }
  }
  Count=count+1
}

```

Phase 3: improvement of clustering by SO

In previous steps, almost all nodes are placed in a cluster based on shared follower and similarity opinions. There may be nodes separately (border nodes or outliers). Phase 3 assigns outliers to existing clusters by Algorithm 6. If each outlier has maximum similarity (minimum distance) to one cluster, it is placed in that cluster. These operations are accomplished in par-

allel for all clusters in the Hadoop structure. The complexity of Algorithm 6 is $O(N)$.

Algorithm 6. Improving Clusters and Centroids, $O(N)$

```

 $//$  clustering improvement
Clusters={C1,C2,C3,...}
{Outliers}={outlier(1),outlier(2),...}
 $//$ outliers nodes
For each outlier i
{
  For each cluster in parallel
  {
    k=Find
    (shorest(Distance(i,cluster(j)))
    Assign outlier(i) to cluster(k)  $//O(N)$ 
  }
  i=i+1
}
Assign clusters and their centroids to
communities and their influence on people

```

Example 1: We consider a simple graph in Figure 2. Results of the phase 1 of the proposed algorithm are shown in Table 2. In phase 1, the proposed method calculates FG. Then, it obtains the G-list. In phase 2, node 2 from G-List is selected. Then, Algorithm 5 investigates its neighbors with the F index. The results of step 1 of the phase 2 are shown in Table 3. In the F-List, node1 or node 4 is the smallest. There is no difference between the two nodes 1 and node 4. We randomly select node 1 as an influential node (central node). Node 1 and some of its neighbors form a cluster provided that the F index between node 1 and the other node is less than one. Therefore, cluster (1), new graph (G) is built (Fig. 3). These steps of phase 2 are repeated until there are only isolated nodes in the graph, which are called an outlier (Table 4 and Fig. 4) in phase 3; node 7 (outlier) is assigned to a cluster (2) due to less distance (more similarity) by Algorithm 6. Finally, clusters (community) are formed (Fig. 5).

The proposed method can be more efficient than many algorithms for the following reasons. The first reason for the algorithm's low execution time is the Hadoop structure's use. The second reason for this method is not to specify the number of communities before executing the algorithm. The third and most important reason is that the algorithm is designed based on people's behavior and relationships. For this reason, in the steps of greedy choices using the F -index and identifying influential people, the similarity of other people to influential people and the similarity of people to each other are considered. Using a greedy index in

FIG. 2. Simple example of graph.

Table 2. Results of phase 1 for simple example

Node	Followers	FI	FG
1	4,2	$FI(1) = \frac{2}{3}$	$FG(1) = \frac{\frac{2}{3}}{\frac{2}{3} + \frac{1}{3}} = \frac{1}{2}$
2	1,3	$\frac{2}{3}$	$\frac{2}{11}$
3	1,4,2	$\frac{3}{1}$	$\frac{3}{2}$
4	2,1	$\frac{2}{3}$	$\frac{1}{2}$
5	4,6	$\frac{2}{1}$	$\frac{3}{4}$
6	5,7	$\frac{2}{1}$	1
7	—	—	—

G-List = {2,1,4,5,3,6,7}.

FG, futuristic greedy index; FI, follower index.

conventional greedy methods such as SO may not satisfy other greedy indexes (such as FG). This method, with a futuristic approach, causes a balance between different indexes. Of course, our method's main challenge is to consider the same effect of SO and FG on various social networks. For being more accurate in identifying communities, the effects of SO and FG in determining the F index's value must be done more accurately. Experimental results illustrate these cases.

Experimentations

The complexity of phases of the proposed algorithm comprises $O((N/R)\log(N/R))$, $O(N)$, $O(N)$, respectively. Therefore, the complexity of the proposed method is

$O((N/R)\log(N/R))$, where R is the number of Reducers. This complexity is compared with several algorithms in Table 5.³⁹

The experimental platform is implemented by Map-reduce programming in the Hadoop framework. Hardware is composed of one master machine and eight slave machines. All machines are homogenous, and each of them is composed of the Intel Xeon E7-2850 @ 2.00 GHz (Dual cpu) and 8.00 GB RAM. All of the tests were performed on Ubuntu 16.04 with Hadoop 2.9.1 and JDK 1.8. Java implemented the codes of the program in the Hadoop environment.

Table 6 presents datasets applied in this research for the computation of execution time in big data. These social networking datasets are relatively big, because it is impossible to load these data in the main memory of an existing machine (except the Orkut-dir dataset). Therefore, an available computer cannot execute it. Also, datasets in this research are too big not to be processed by traditional algorithms. Sinaweibo dataset is an online social network. It is a micro-blogging service with millions of users in China. Also, the soc-Friendster dataset is the online gaming social network. This dataset contains all links among users. The Orkut-dir dataset contains a list of all of the user-to-user links in the Orkut social network.

Table 3. Results of phase 2 in step 1 for simple example

Node	1	2	3	4	5	6	7
SO(2,i)	$0 + 1/2 = 1/2$	Infinity	$\frac{1}{2+3-2*1} + \frac{1}{2} = 5/6$	$1/2 + 0 = 1/2$	0	0	0
FG(i)	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{3}{3}$	$\frac{2}{3}$	2	2	—
$F(i) = FG(i)/SO(2,i)$	$\frac{4}{3}$	Infinity	$\frac{18}{5}$	$\frac{4}{3}$	—	—	—

SO, similar opinions index.

Twitter2010 dataset is a network of follower relationships from a snapshot of Twitter in 2010. We use other datasets in Table 7 for ground-truth network communities in social and information networks. The Rand Index, F1 measure, Normalized Mutual Information (NMI), and execution time evaluated these datasets with several algorithms.

Com-LiveJournal and com-Orkut are an online blogging community where people declare their friendship with each other. They also permit users to form a group that other members can then join. We consider such user-defined groups as ground-truth communities. Also, com-Youtube is a video-sharing website that includes the fact that users and their friends can build groups that other users can join. We consider such user-defined groups as ground-truth communities. The com-DBLP computer science bibliography provides a series of research papers in computer science. We form a co-authorship network where two authors are connected if they publish at least one paper together. Publication venue, for example, journal or conference, defines an individual ground-truth community.

The experiments' results are compared with other methods (SLPA, Fast greedy, LPA, walk trap, TopGc, MCD, LPA-FCM). Table 8 illustrates the execution time for these algorithms. The proposed method runs

Table 4. Results of step2 in phase 2 for simple example

Node	5	6	7
SO(5,i)	Infinity	$0 + 1/2 = 1/2$	0
FG(i)	1/1	2/1	0
$F(i) = FG(i)/SO(2,i)$	Infinity	1	0

less time than other algorithms for two reasons. The first reason is that big data is classified and assigned to mappers and reducers based on similar nodes based on SO and influential nodes. The clustering operations are performed on each mapper or each reducer in low execution time, because interrelated data (similar nodes) are allocated in a machine (mapper or reducer). Accordingly, the communication cost for calculations of similar nodes between machines in the Hadoop framework decreases because most similar nodes are allocated in one reducer or one mapper. Therefore, communication and coordination cost between machines is reduced. Hence, the coordinator (master machine) is performed fast. Thus, the total execution time is decreased in the parallel structure of Map-Reduce.

Accordingly, the computation of distance in density-based clustering is decreased. Besides, several similar operations, the same as FI, are calculated only once, and its result can regularly be used. Hence, the iterations of operations and execution time will be decreased significantly.

Parallelization in computations at different phases of the proposed method has led to improved

Table 5. Comparison of algorithms' complexity

	Complexity	Year
SLPA	$O(Tm)$	2012
Fast greedy	$O(n*(\log n)^2)$	2004
LPA	$O(m)$	2007
Walktrap	$((m(n^2)) O((n^2)\log n)$	2005
TopGC	—	2010
MCD	$O(nk)$	2012
LPA-FCM	$O((2+m)n + t*(3m+2)n), m \leq E $	2019
Proposed method	$N/R \log(N/R)$	2020

m , the number of edges; N , the number of nodes; R , the number of reducers; t , the number of iterations.

implementation of the proposed method. The speedup index is used for this comparison. The speedup is defined as the serial runtime ratio (T_s) of the best sequential algorithm to the time taken by the parallel algorithm (T_p) to solve the same problem on p processors. Table 9 illustrates the comparison execution time between sequential and parallel in the proposed algorithm. Figure 6 shows the speedup between datasets. Sequence execution time in the Orkut-dir dataset is better than parallel because of the coordinator's very little processing (master machine). Sequential processing performs faster than parallel processing when the dataset space is entirely stored in one machine's RAM. Speed up is depicted by Formula (7):

$$\text{Speedup} = \frac{T_s}{T_p} \quad (7)$$

Table 10 shows the results of the execution time in ground-truth communities. The reduction in the algorithm's execution is quite visible due to the parallelization and the reduction in the number of iterative calculations in the mappers and reducers.

Clustering algorithms based on the greedy approach often do not detect appropriate cluster centers, because

they may detect centroids with maximum-shared opinions. These centroids are placed in one cluster and not in separate clusters. The proposed method has partially solved this problem. The proposed method creates similar nodes in neighborhoods of dense areas. In the proposed method, the clusters and their centers are selected. The nodes with almost opposite ideas are placed in different clusters, because centroids with maximum similar opinions are placed in one cluster by Algorithm 5. Hence, nodes with similar opinions of the centroid are placed in one cluster. Also, Algorithm 6 improves clustering due to the allocation of outliers. These factors increase the Rand Index of clusters. We can compare clustering methods with the Rand Index. The Rand Index in data clustering is a measure of the similarity between two clusters. It brings to view the measure of the percentage of correct decisions made by the algorithm. It depicts detected communities by clustering, which consists of similar opinions of people. The Rand Index is calculated by using the Formula (8)⁴⁴:

$$\text{Rand Index} = \frac{TP + FN}{TP + FP + TN + FN} \quad (8)$$

Rand Index is calculated for every two clusters. Subsequently, we consider the average of Rand Indexes and compare clusters with them. TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives. Rand Index can calculate clustering accuracy.

Table 11 shows that TopGC has the least Rand Index, and our proposed method has the greatest Rand Index. It illustrates that the proposed method performs more efficiently than the other methods in

Table 6. Big datasets for execution time

Data set	Vertices	Edges	Size	Description
Sina Weibo	21M	261M	7G	Sina Weibo online social network micro-blogging service
Twitter2010	58M	265M	23G	The network of follower relationships from a snapshot of Twitter in 2010
Orkut-dir	3.1M	117.2M	1.6G	Orkut social network list of all of the user-to-user links.
soc-Friendster	41M	1468M	27G	Online gaming social network all links among users

Table 7. Datasets with ground-truth communities for clustering quality

Name	Type	Nodes	Edges	Communities	Description	Capacity
com-LiveJournal	Undirected, Communities	3,997,962	34,681,189	287,512	LiveJournal online social network	150 MB
com-Orkut	Undirected, Communities	3,072,441	117,185,083	6,288,363	Orkut online social network	0.8 GB
com-Youtube	Undirected, Communities	1,134,890	2,987,624	8,385	Youtube online social network	10 MB
com-DBLP	Undirected, Communities	317,080	1,049,866	13,477	DBLP collaboration network	6 MB

Table 8. Comparison of execution time for big datasets

Datasets	LPA	MCD	Fast greedy	LPA-FCM	Proposed method
Sina Weibo	537.81	527.24	563.14	653.31	514.47
Twitter2010	2047.21	1997.71	2103.78	2641.19	1174.747
Orkut-dir	131.47	109.37	123.97	173.61	101.37
soc-friendster	1904.49	1883.13	1907.67	22578.31	1736.86

Table 9. Comparison of parallel and sequential execution time in proposed algorithm

Datasets	Speedup*100	Total time (serial)	Total time (parallel)
Sina Weibo	2.203	1237.45	561.48
Twitter2010	2.652	4317.48	1627.43
Orkut-dir	0.981	97.98	99.83
soc-friendster	2.851	5141.83	1803.43

creating clusters. This efficiency is the result of the usage of FG and SO. These indexes detect communities of nodes so that nodes in one community are similar and nodes in one community are far from nodes in another community. This process creates clusters by the Rand Index definition, which has high internal similarity between members in the cluster and low similarity between members in different clusters. Therefore, detected communities are the density of a similar person based on influential people. Thus, detected communities are approximated independent communities because people in each community are less similar to people in other communities. Rand Index shows that detected communities are accurate for advertising, because they include people with similar opinions about advertising.

FIG. 6. Speedup of parallelizing.**Table 10. Comparison of execution time with ground-truth communities**

Execution time (seconds)	SLPA	Fast greedy	LPA	Walktrap	TopGC	MCD	LPA-FCM	Proposed method
com-Youtube	19.18	0.31	0.41	0.73	3.26	0.37	0.47	0.35
com-DBLP	41.46	0.51	0.78	13.62	12.72	0.39	0.87	0.49
com-LiveJournal	78.21	1.23	4.01	58.31	27.93	0.87	2.97	0.97
com-Orkut	22.37	0.73	0.56	0.96	5.17	0.49	0.38	0.67

Table 11. Comparison of Rand Index with ground-truth communities

Rand Index	SLPA	Fast greedy	LPA	Walktrap	TopGC	MCD	LPA-FCM	Proposed method
com-Youtube	0.875	0.911	0.908	0.817	0.35	0.709	0.971	0.983
com-DBLP	0.912	1	1	0.997	0.73	0.879	1	1
com-LiveJournal	0.983	0.995	0.995	0.977	0.79	0.576	0.998	0.993
com-Orkut	0.989	0.999	0.997	0.977	0.013	0.997	0.999	0.997

Table 12. Comparison of NMI with ground-truth communities

<i>NMI</i>	<i>SLPA</i>	<i>Fast greedy</i>	<i>LPA</i>	<i>Walktrap</i>	<i>TopGC</i>	<i>MCD</i>	<i>LPA-FCM</i>	<i>Proposed method</i>
com-YouTube	0.565	0.354	0.538	0.663	0.043	0.592	0.607	0.639
com-DBLP	0.932	0.635	0.918	0.635	0.073	0.856	0.931	0.947
com-LiveJournal	0.979	0.750	0.975	0.974	0.109	0.797	0.983	0.978
com-Orkut	0.975	0.967	0.981	0.956	0.73	0.984	0.978	0.988

NMI, Normalized Mutual Information.

We use two common criteria: F (Formulas 9, 10, and 11), NMI (Formulas 12, 13, and 14) for comparisons.⁴³

$$\text{Precision} = \frac{TP}{TP + FP} \quad (9)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (10)$$

$$F = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (11)$$

C and C' are the actual cluster (community) and the predicted cluster (community) in the dataset. MI is a symmetric measure that is used to quantify the statistical information shared between two distributions, which indicates the shared information between two partitions. $I(C, C')$ denotes MI; $E(C)$ and $E(C')$ show entropy C and C' , respectively. Formulas 12, 13, and 14 denote the calculation of NMI.

$$I(C, C') = \sum_{k=1}^{|C|} \sum_{l=1}^{|C'|} \frac{n_{kl}}{M} \log \left(\frac{M n_{kl}}{n_k m_l} \right) \quad (12)$$

$$E(C) = - \sum_{k=1}^{|C|} \frac{n_k}{M} \log \left(\frac{n_k}{M} \right) \quad (13)$$

$$\text{NMI}(C, C') = \frac{I(C, C')}{\sqrt{E(C)E(C')}} \quad (14)$$

Table 12 shows NMI indexes among several methods. In social networks, where two indexes of SO and FG have the same approximated effect on each other, our method performs very well. In these social net-

works, the NMI index has the best value. Table 13 depicts the effects of several methods on the F1 measure. Due to the use of FGs, our method avoided inappropriate local selections so that the selected nodes in each cluster are far from the other nodes of other clusters. The F index improves indexes of SO and FG simultaneously. This futuristic approach chooses a decision based on several indexes. The effect of indexes SO and FG for identifying communities varies in different types of social networks. To improve community detection, the effect of these two indexes on the type of social network should be examined by machine-learning methods. Then, the F -index should be calculated according to these effects. Thus, the calculated F -index can provide good improvement for identifying communities on that type of social network. The futuristic greedy method performs effectively when we can calculate effective indexes with regard to each other. The futuristic greedy method performs well when calculating the opposite effects of the greedy indexes on each other. The futuristic greedy index can establish a tradeoff among different greedy indexes in problem solving. For example, in our previous research, this index could obtain a suitable distance for clustering, which ultimately reduced the execution time of reducers in the Hadoop structure.⁴⁰

Conclusions

In this research, we detect influential people based on the greedy index SO and FG. At the beginning of the proposed algorithm, influential people are considered as cluster (community) centers in social networks. Hence, density-based clustering runs fast because it

Table 13. Comparison of F measure with ground-truth communities

<i>F1 measure</i>	<i>SLPA</i>	<i>Fast greedy</i>	<i>LPA</i>	<i>Walktrap</i>	<i>TopGC</i>	<i>MCD</i>	<i>LPA-FCM</i>	<i>Proposed method</i>
com-YouTube	0.06	0.078	0.028	0.077	0.066	0.587	0.09	0.097
com-DBLP	0.932	0.631	0.926	0.442	0.131	0.863	0.931	0.971
com-LiveJournal	0.984	0.782	0.976	0.983	0.147	0.690	0.781	0.773
com-Orkut	0.992	0.997	0.996	0.986	0.026	0.997	0.993	0.997

does not need to discover the number of clusters and their centers. Also, our proposed method improved clustering detection accuracy due to users' similarity to the SO. Community detection is created based on these clusters. These communities are proper for advertising and marketing due to the many similarities of people in each community. In the proposed method, each node with its followers' information can be placed in a mapper or reducer. Since the information of these nodes varies, the capacity of this information in a mapper or reducers changes. The varying capacity of mappers or reducers creates a load imbalance between the machines, leading to increased execution time. Introducing new methods to improve load balancing can be a good idea for future research. Increasing the accuracy of community detection can be determined by the F index. This index is calculated based on distributing SO and FG in different social networks. Research about determining the SO and FG effects by machine-learning methods can be recommended for future works.

Author Disclosure Statement

No competing financial interests exist.

Funding Information

No funding was received for this article.

References

- Fernandes A, Gonçalves PCT, Campos P, et al. Centrality and community detection: A co-marketing multilayer network. *J Bus Ind Mark*. 2019;34:1749–1762.
- Newman MEJ, Girvan M. Finding and evaluating community structure in networks. *Phys Rev E*. 2004;69:26113.
- Will M, Groeneveld J, Frank K, et al. Combining social network analysis and agent-based modelling to explore dynamics of human interaction: A review. *Socioenviron Syst Model*. 2020;2:16325.
- Srinivas S, Rajendran C. Community detection and influential node identification in complex networks using mathematical programming. *Expert Syst Appl*. 2019;135:296–312.
- Jaho E, Karaliopoulos M, Stavrakakis I. Iscode: A framework for interest similarity-based community detection in social networks. In: 2011 IEEE Conference on Computer and Communication Work (INFOCOM WKSHPS), IEEE, 2011. pp. 912–917.
- Dabas C, Nigam GK, Nagar H. Investigating large-scale graphs for community detection. In: *Innovations in Computational Intelligence and Computer Vision*. Singapore: Springer, pp. 122–129.
- Khan BS, Niazi MA. Network community detection: A review and visual survey. *arXiv Preprint*. 2017;arXiv1708.00977.
- Fang Y, Huang X, Qin L, et al. A survey of community search over big graphs. *Vldb J*. 2020;29:353–392.
- Phyu KS, Min MM. Graph-based community detection in social network. In: 2019 IEEE/ACIS 18th International Conference on Computer and Information Science, IEEE, 2019. pp. 12–17.
- Carusi C, Bianchi G. Scientific community detection via bipartite scholar/journal graph co-clustering. *J Informetr*. 2019;13:354–386.
- Li J, Cai T, Deng K, et al. Community-diversified influence maximization in social networks. *Inf Syst*. 2020;92:101522.
- Li C, Bai J, Wenjun Z, et al. Community detection using hierarchical clustering based on edge-weighted similarity in cloud environment. *Inf Process Manag*. 2019;56:91–109.
- Hu F, Liu J, Li L, et al. Community detection in complex networks using Node2vec with spectral clustering. *Phys A Stat Mech Appl*. 2019;545:123633.
- Zhou Z, Amini AA. Analysis of spectral clustering algorithms for community detection: The general bipartite setting. *J Mach Learn Res*. 2019;20:1–47.
- Singh A, Garg S, Batra S, et al. Probabilistic data structure-based community detection and storage scheme in online social networks. *Future Gener Comput Syst*. 2019;94:173–184.
- Tandon A, Albeshri A, Thayanathan V, et al. Fast consensus clustering in complex networks. *Phys Rev E*. 2019;99:42301.
- Shi C, Cai Y, Fu D, et al. A link clustering based overlapping community detection algorithm. *Data Knowl Eng*. 2013;87:394–404.
- Doreian P, Batagelj V, Ferligoj A. *Advances in Network Clustering and Blockmodeling*. United States: John Wiley & Sons, 2020.
- Deng Z-H, Qiao H-H, Song Q, et al. A complex network community detection algorithm based on label propagation and fuzzy C-means. *Phys A Stat Mech Appl*. 2019;519:217–226.
- Pirim H, Eksioglu B, Glover FW. A novel mixed integer linear programming model for clustering relational networks. *J Optimiz Theory App*. 2018;176:492–508.
- Zhou H, Xi B, Zhang Y, et al. A graph clustering algorithm using attraction-force similarity for community detection. *IEEE Access*. 2019;7:13683–13692.
- Mishra KK, Bisht H, Singh T, et al. A direction aware particle swarm optimization with sensitive swarm leader. *Big Data Res*. 2018;14:57–67.
- Wang Y, Xu W, Kang Q. AFSMA: An enhanced artificial fish swarm algorithm based on mutuality for community detection. In: *Proceedings of the 2nd International Conference on Big Data Research*. ACM, 2018. pp. 107–113.
- Ghosh S, Halappanavar M, Tumeo A, et al. Distributed louvain algorithm for graph community detection. In: 2018 IEEE International Symposium on Parallel and Distributed Processing. IEEE, 2018. pp. 885–895.
- Chen Y, Zhang J, Fang Y, et al. Efficient community search over large directed graphs: An augmented index-based approach. In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, AAAI*, 2020.
- Chamberlain J, Turpin B, Maged A, et al. Designing for collective intelligence and community resilience on social networks. *Hum Comput* (in press).
- Hosseini-Pozveh M, Zamanifar K, Naghsh-Nilchi AR. A community-based approach to identify the most influential nodes in social networks. *J Inf Sci*. 2017;43:204–220.
- He C, Tang Y, Liu H, et al. A robust multi-view clustering method for community detection combining link and content information. *Phys A Stat Mech Appl*. 2019;514:396–411.
- Sharma R, Oliveira S. Community detection algorithm for big social networks using hybrid architecture. *Big Data Res*. 2017;10:44–52.
- Chunaev P. Community detection in node-attributed social networks: A survey, *arXiv Preprint*. 2019;arXiv1912.09816.
- Taha K. Static and dynamic community detection methods that optimize a specific objective function: A survey and experimental evaluation. *IEEE Access*. 2020;8:98330–98358.
- Agarwal G, Kempe D. Modularity-maximizing graph communities via mathematical programming. *Eur Phys J B*. 2008;66:409–418.
- Yuan Y, Chen B, Yu Y, et al. An influence maximisation algorithm based on community detection. *Int J Comput Sci Eng*. 2020;22:1–14.
- Tang F, Wang C, Su J, et al. Spectral clustering-based community detection using graph distance and node attributes. *Comput Stat*. 2019;35:1–26.
- Yao K, Papadias D, Bakiras S. Density-based community detection in geo-social networks. In: *Proceedings of the 16th International Symposium on Spatial and Temporal Databases, ACM*, 2019. pp. 110–119.
- You T, Cheng H-M, Ning Y-Z, et al. Community detection in complex networks using density-based clustering algorithm and manifold learning. *Phys A Stat Mech Appl*. 2016;464:221–230.
- Daud NN, Ab Hamid SH, Saadoon M, et al. Applications of link prediction in social networks: A review. *J Netw Comput Appl*. 2020;166:102716.

38. Taheri S, Bouyer A. Community detection in social networks using affinity propagation with adaptive similarity matrix. *Big Data*. 2020;8:189–202.
39. Harenberg S, Bello G, Gjeltema L, et al. Community detection in large-scale networks: A survey and empirical evaluation. *Wiley Interdiscip Rev Comput Stat*. 2014;6:426–439.
40. Bakhthemmat A, Izadi M. Decreasing the execution time of reducers by revising clustering based on the futuristic greedy approach. *J Big Data*. 2020;7:1–21.
41. Beni HA, Bouyer A. TI-SC: Top-k influential nodes selection based on community detection and scoring criteria in social networks. *J Ambient Intell Humaniz Comput*. 2020;11:4889–4908.
42. Pagare R, Khare A, Chaudhary S. (T-ToCODE): A framework for trendy topic detection and community detection for information diffusion in social network. In: *Data Management, Analytics and Innovation*. Singapore, Singapore: Springer, 2020. pp. 613–632.
43. Chattopadhyay S, Basu T, Das AK, et al. Towards effective discovery of natural communities in complex networks and implications in e-commerce. *Electron Commer Res*. 2020;1–38.
44. Sammut C, Webb GI. *Encyclopedia of Machine Learning*. Boston, MA: Springer Science & Business Media, 2011.

Cite this article as: Bakhthemmat A, Izadi M (2021) Communities detection for advertising by futuristic greedy method with clustering approach. *Big Data* 9:1, 22–40, DOI: 10.1089/big.2020.0133.

Abbreviations Used

AP = affinity propagation
 FG = futuristic greedy index
 FI = follower index
 HDFS = Hadoop Distributed File System
 NMI = Normalized Mutual Information
 NR = neighbor relation
 SFI = shared follower index
 SO = similar opinions index