Social Network Analysis and Mining (2023) 13:103
https://doi.org/10.1007/s13278-023-01109-5

ORIGINAL ARTICLE

PCMeans: community detection using local PageRank, clustering, and K-means

Wafa Louafi[1] · **Faiza Titouna**[1]

Received: 25 March 2023 / Revised: 27 July 2023 / Accepted: 28 July 2023 / Published online: 16 August 2023
© The Author(s), under exclusive licence to Springer-Verlag GmbH Austria, part of Springer Nature 2023

Abstract
With the rise of social networks, the task of community detection in networks has become increasingly difficult in recent years. In this study, we introduce a novel approach for community detection named PCMeans, which combines PageRank, hierarchical clustering, and k-means algorithms to tackle the community detection problem on the entire network. Our technique employs Local PageRank to identify the most influential nodes within a local subgraph, followed by an overlapping hierarchical clustering strategy that determines the optimal number of clusters on the entire network. While our approach uses Local PageRank, which operates locally on each node, the clustering itself is performed globally on the entire network. K-means learning is then applied to swiftly converge to the final community structure. PCMeans is an unsupervised method that is easy to implement, efficient, and simple, and it addresses three common problems, including the random selection of the initial central node, specification of the number of classes K, and slow convergence. Experiments show that our algorithm not only has improved influence but also effectively reduces time complexity and outperforms other recent approaches on both real networks and synthetic benchmarks. Our approach is versatile and can be applied to a wide range of community detection problems, including those with non-convex shapes and unknown numbers of communities.

Keywords Community detection · Local PageRank · Clustering · K-means

1 Introduction

A network is a collection of nodes that are connected by links, and this definition is applicable to many systems in today's world, such as social networks, web networks, and biological networks. The analysis of these networks originated from the work of mathematicians on graphs and has since attracted the attention of other fields, owing to its applicability to a wide range of applications in different fields, including community detection, which is one of the most fundamental and classic problems. Most networks of interest display community structure, where the nodes are organized into groups, called communities, clusters, or modules (Fortunato and Hric 2016). Community detection involves dividing a network into a number of groups or modules, with each group representing a community. To be considered a community, a group or module must not be empty, its nodes must be part of the graph, each community must be different (i.e., the set of nodes in two different communities must not be identical), and the union of all communities must return the set of nodes in the graph (Kumar et al. 2020).

The objective of community detection is to identify hidden communities in a network based on information provided by the network, such as its topology or the characteristics of its nodes and edges (Guo et al. 2022). Community structures are critical to understanding not only the network topology but also how the network functions. Various methods have been proposed to detect community structures from different perspectives (Fortunato 2010).

In this work, we propose a novel approach for community detection, named PCMeans, which combines PageRank, hierarchical clustering, and k-means algorithms to tackle the community entire detection problem on the network. Our technique involves three phases: (1) identifying the most important nodes in the graph using Local PageRank, (2) partitioning the data points into groups based on their

✉ Wafa Louafi
louafiwafa@yahoo.fr

Faiza Titouna
f.titouna@univ-batna2.dz

[1] LaSTIC Computer Sciences Laboratory, University of Batna2, Batna, Algeria

similarity using hierarchical clustering, and (3) refining the clusters using k-means clustering to obtain disjoint clusters from the overlapping clusters obtained in the second phase. While our approach employs Local PageRank, which operates locally on each node, the clustering itself is performed globally on the entire network. We adopt the initial number of communities and nodes, and this algorithm has low time complexity, simple operation, and easy implementation. The main objective of this approach is to select the initial nodes in the subgraph network with Local PageRank. Then, a hierarchical classification is conducted to create overlapping communities. Finally, the K-means algorithm is applied to the centers of these communities to find disjoint communities. According to the experimental results and comparative analysis of PCMeans with other state-of-the-art community detection algorithms, PCMeans generates consistently high-quality results for real-world and synthetic networks that are competitive with other algorithms in terms of accuracy. PCMeans is also more efficient than some other algorithms, as it addresses the problem of slow convergence and has low time complexity, simple operation, and easy implementation. Therefore, PCMeans is an effective and efficient method for community detection.

The rest of this paper is organized as follows: the related work is presented in Sect. 2, the proposed method in Sect. 3, the experimental results in Sect. 4, and finally, the concluding remarks in Sect. 5.

2 Related work

There are many methods proposed for detecting communities in social networks. Some of these methods allow for disjoint communities, and examples include:

The Newman–Girvan algorithm (Newman and Girvan 2004) is a method proposed for detecting community and sub-community structures in social networks. This algorithm involves iteratively removing edges from the network to split it into communities. The edges are identified using one of several possible "betweenness" measures, and these measures are recalculated after each removal. This process continues until the network is completely divided into communities. The algorithm has two definitive features: the iterative removal of edges and the recalculation of betweenness measures, which are crucial for its effectiveness. The algorithm has been shown to perform well in detecting community structures in a variety of real-world social networks.

The Louvain method (Blondel et al. 2008) is an iterative algorithm that aims to maximize the modularity of a network by moving nodes between communities. During the first phase, each node is initially assigned to its own community, and then the algorithm iteratively moves nodes between communities to maximize the modularity. In the second phase, the communities found in the first phase are collapsed into "super-nodes" and a new network is constructed, with the weights of the edges between the super-nodes representing the sum of the weights of the edges between nodes in the corresponding communities. The algorithm then starts again with the new network, repeating the two phases until no further improvement in modularity can be achieved.

K'-means (Žalik 2008): A modification of the K-means algorithm for non-overlapping community detection that allows the number of clusters to vary.

Vilcek (2014) proposed a new graph clustering approach for network community detection called the deep K-means algorithm. This algorithm is based entirely on K-means clustering and aims to improve upon traditional spectral clustering. Instead of using eigenvector decomposition in spectral clustering, Vilcek proposed using a multilayer autoencoder pipeline implemented with recursive K-means clustering. This approach was shown to outperform traditional spectral clustering in terms of accuracy, as measured by normalized mutual information. However, the algorithm has a disadvantage in that its execution time increases faster than spectral clustering as the size of the dataset increases. Additionally, the algorithm requires prior knowledge of the number of communities to be found, which is not always practical. Incorporating a technique such as modularity function optimization to automatically choose the number of communities would be an interesting future direction for this approach.

CLPSO-DE (Pourkazemi and Keyvanpour 2017): A hybrid algorithm that combines particle swarm optimization and differential evolution for community detection.

Frequent Pattern (Moosavi et al. 2017): A frequent pattern mining-based approach for community detection that extracts frequent subgraphs as communities.

Cai et al. (2019) proposed that DDJKM (Density, Degree, Jaccard, and K-means) algorithm is a clustering-based overlapping community detection method proposed in 2019. The algorithm uses the uncertainty of nodes, calculated based on their density, degree, and similarity, to describe their membership to different communities. K-means clustering is then applied to the uncertainty matrix to identify overlapping communities. The algorithm has shown promising accuracy and efficiency in various real-world networks.

CMCM (Geng et al. 2019): A multi-objective optimization-based algorithm that simultaneously optimizes both the modularity and the conductance for community detection.

Sheng et al. (2020) in 2020 proposed a new community detection method called IACD (Inter-node Attraction Community Detection), which is based on the attraction of internal nodes. The algorithm consists of three main steps: evaluating node importance, selecting pairs of attractive nodes, and dividing the community. First, they calculate the node influence (IF) and global influence (GIF) of each node,

and then identify the most attractive node for each node to form pairs of attractive nodes. Finally, the algorithm creates a two-dimensional array to output the community division result. IACD considers the importance of nodes in the network, and uses a physics-inspired approach to represent the forces between nodes.

Hajij et al. (2020) proposed a novel approach to finding initial centers for the K-means clustering algorithm by using the PageRank vector as a centrality measure. This method was found to be efficient and provided several key benefits. One such advantage is that the PageRank vector can be calculated for both direct and indirect graphs. Additionally, since the PageRank vector was designed to be computed on large graphs, it offers improved speed. Finally, this method can be applied to other domains by easily generalizing to metric spaces.

Yuan et al. (2020) developed an influence maximization algorithm for social networks that selects the most influential node as the initial active node and assigns it the maximum number of nodes. They proposed the edge betweenness algorithm, based on community detection, to maximize node influence. The algorithm uses the K-means algorithm to divide the community and selects the optimal community segmentation result based on modularity. It then calculates the edge betweenness of each community and selects important nodes to form the set of starting nodes for the influence maximization algorithm. Finally, the independent cascade model is used to simulate the propagation of influence and maximize its effect.

Li et al. (2021) proposed an improved algorithm, called LPA-MNI (label propagation algorithm based on modularity and node importance), for detecting community structure in complex networks. The LPA-MNI algorithm aims to address the randomness inherent in the original LPA algorithm by combining modularity and node importance. Initially, LPA-MNI uses modularity optimization procedures to identify initial communities, and all nodes within a community are assigned the same label. In iterative label propagation processes, LPA-MNI updates labels in a decreasing order of node importance. When multiple labels are assigned to the same maximum number of nodes, LPA-MNI calculates the importance of each node and selects the label of the most influential node to update. Overall, the LPA-MNI algorithm improves upon the original LPA algorithm by leveraging modularity and node importance to more accurately detect community structure in complex networks.

Chaudhary and Singh (2021) concluded that unsupervised machine learning techniques can be used for community detection in COVID-19 datasets, which can help in understanding the spread of the virus and identifying potential transmission hotspots. They also noted that further research is needed to improve the accuracy of community detection algorithms and to account for factors such as the time-varying nature of social interactions during the pandemic.

FPPM (Wu et al. 2021): A frequent pattern-based approach for community detection that uses a pattern mining algorithm to extract communities.

Akbar et al. (2021) suggested that the use of modularity maximization for community detection in social networks can provide valuable insights for businesses and help them make more informed decisions.

Entropy gap (Liu et al. 2022): An algorithm that uses the concept of entropy to detect communities by identifying the gap between the entropy of the original network and that of a null model.

CDBNE (Zhou et al. 2023): presented a community detection algorithm based on unsupervised attributed network integration (CDBNE) to solve problems. They propose a framework that simultaneously learns representation based on network structure and attribute information and clustering-oriented representation.

There are also many community detection algorithms proposed by various researchers, allowing for overlapping communities. Some of the algorithms are:

The clique percolation method (CPM) is a community detection algorithm that was proposed by Palla et al. (2005). It is unique in that it allows for overlapping communities to be detected in a network. The algorithm works by finding all k-cliques in the network and creating a clique graph where each k-clique is represented by a node. Overlapping communities are then identified as sets of nodes that are connected by k-cliques. The size of the communities detected is determined by the value of k, with larger values detecting smaller and more tightly knit communities. The CPM algorithm has been effective in identifying overlapping communities in various types of networks, including social, biological, and technological networks.

The Infomap algorithm (Rosvall and Bergstrom 2008) is a popular method for detecting communities in networks proposed in 2008. It uses a random walk approach to identify communities based on information theory principles. The basic idea is to treat the network as a flow of information and to try to identify communities that represent clusters of nodes with high information flow within the cluster, but low information flow between clusters.

RCE (Musdar and Azhari 2015): A randomized clustering-based algorithm for overlapping community detection that uses a random walk-based approach.

LED (Ma et al. 2016) uses a community detection algorithm called the label propagation algorithm (LPA), proposed in 2016, which assigns each vertex to one or more communities based on the network weights and the current community assignments of its neighbors. The algorithm iterates between computing the network weights and applying LPA until convergence is reached.

EMc and PGDc (Van Laarhoven and Marchiori 2016): Two algorithms based on the expectation maximization algorithm and the projected gradient descent method, respectively, for overlapping community detection.

WalkSCAN is a popular algorithm for detecting overlapping communities. It was proposed by Tong and Cao in 2017 (Hollocou et al. 2016). The algorithm is based on a probabilistic generative model that uses a combination of Bayesian and expectation maximization techniques to identify overlapping communities. The idea behind the algorithm is to model the network as a random walk process, where each node represents a state in the process, and the edges represent the probabilities of transitioning from one state to another.

Kumar et al. (2020) proposed a clustering-based overlapping community detection method called NSGA-II that uses a multi-objective optimization approach to balance the trade-off between maximizing intra-community density and minimizing inter-community density. The algorithm starts by randomly selecting initial individuals, representing communities, and evaluating their fitness based on the two objectives. It then generates new individuals using a genetic algorithm-based approach to potentially improve fitness values. The process repeats iteratively until a set of non-dominated solutions, representing the overlapping communities in the network, are identified.

LCDNN (Luo et al. 2020): A deep neural network-based algorithm for overlapping community detection that uses both local and global information.

LGIEM (Ma et al. 2020): A likelihood-based algorithm for overlapping community detection that uses a generative model to describe the community structure.

While overlapping community detection methods offer certain advantages, they may not always be the best choice for detecting complex community structures or handling noisy data. In this article, we propose a new method for detecting communities in social networks. The method is primarily based on identifying important nodes and creates a set of overlapping communities that subsequently converge to a set of disjoint communities.

3 PCMeans: a novel community detection algorithm

In this paper, we proposed a novel approach for community detection called PCMeans algorithm, it is a novel approach for detecting communities in a graph. It consists of three stages: detect influenced nodes by calculate of the Local PageRank, overlapping hierarchical clustering, and K-means clustering. These three phases are successive.

In the first stage, the algorithm calculates the Local PageRank score for each node in the graph and sorts them in descending order.

In the second stage of the PCMeans algorithm, communities are refined by grouping overlapping ones. This involves starting with the node that has the highest Local PageRank score and its neighbors, and adding them to a new class. This process continues until all nodes in the graph have been assigned to at least one classes. Then, the similarity between pairs of communities is calculated based on the number of common nodes they share. Communities that have more than 50% of their nodes in common are merged. This step is repeated until no further merging is possible.

In the third and final stage, PCMeans applies K-means clustering to the communities identified in the second stage. The algorithm uses the centers of these communities as the initial centroids for K-means clustering. The number of clusters is set to be equal to the number of communities identified in the second stage. We use k-means clustering to obtain disjoint clusters from the overlapping clusters obtained in the second phase. This addresses the issue of overlapping data points in the clusters.

Figure 1 illustrates the flowchart of the PCMeans algorithm.

The PCMeans algorithm presented globally in Algorithm 1; thereafter, it will be detailed in the following sessions.

Algorithm 1 PCMeans Algorithm

Require: *networks* G
Ensure: C1,C2,..., CK
1: Calculate Local PageRank score for each node and sort them
2: Create initial overlapping community detection
3: Regrouping clusters according to the similarity until no further merging is possible
4: Set the number k is equal to the number of communities identified and use their centers as the initial centroids for K-means clustering
5: Run K-means clustering on the communities

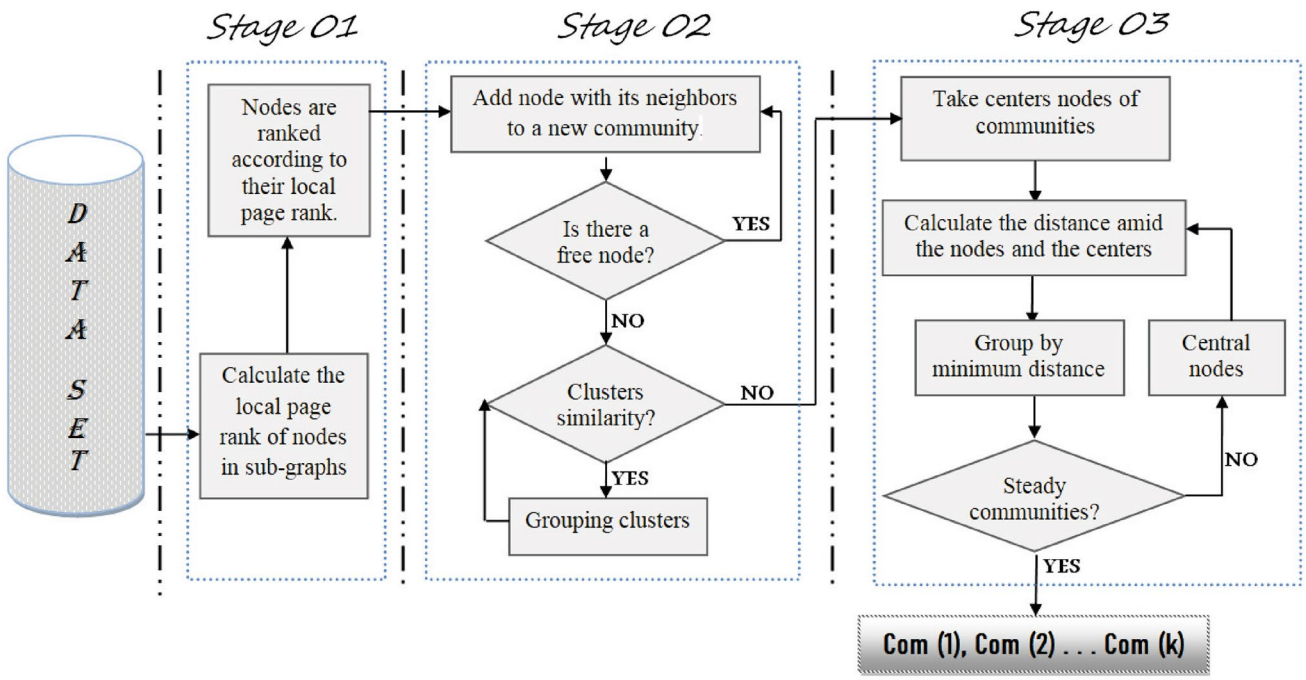

Fig. 1 Framework of PCMeans algorithm

Overall, the PCMeans algorithm for community detection and can be used to analyze social networks, identify key players in a network, and understand the structure of complex systems.

The steps involved in PCMeans algorithm are:

1. Influenced nodes by Local PageRank Stage: The first stage of PCMeans involves community detection, which identifies influential nodes in a graph.
   (a) Calculate the PageRank score for each node in the subgraph
   (b) Sort the nodes in decreasing order of their score.
2. Overlapping hierarchical clustering Stage: The second stage of PCMeans involves refining the nodes found in the first stage by order for grouping overlapping communities. The steps involved in this stage are:
   (a) Take the node with the highest Local PageRank score, and add it and its neighbors to a new cluster.
   (b) Continue adding nodes to clusters until all nodes in the graph have been assigned to at least one cluster.
   (c) Calculate the similarity between pairs of clusters based on the number of common nodes between them.
   (d) If the number of common nodes between two clusters is greater than 50% of the size of the smaller cluster, then the two clusters are merged.
   (e) Repeat the previous step until no further merging is possible.
3. K-means clustering stage: The final stage of PCMeans involves applying K-means clustering to the clusters identified in the first two stages. The steps involved in this stage are:
   (a) Use the centers of the clusters identified in the second stages as the initial centroids for K-means clustering.
   (b) Set the number of communities to be equal to the number of clusters identified in the last stages.
   (c) Run K-means clustering on the communities to group nodes within the same community together

3.1 Stage 01: influenced nodes by local PageRank

PCMeans is a community detection algorithm that begins by calculating the Local PageRank (Brin and Page 1998; Bar-Yossef and Mashiach 2008) for each node in the graph. Local PageRank is a measure of the importance of a node within its local neighborhood, which is defined as the set of nodes that are reachable within two hops from the node using a modified breadth-first search algorithm with $h = 2$.

This algorithm creates a subgraph that includes only the vertices and edges that are within two hops of the starting vertex, which can be useful for analyzing the local structure of a larger graph. We use Local PageRank to calculate the importance of nodes within this subgraph.

However, it is important to note that PCMeans is a global community detection method, which means that it takes into account the entire graph, not just the local subgraph. While Local PageRank provides a useful measure of node importance within a specific neighborhood, it is not the only factor taken into consideration by PCMeans. The algorithm also considers the edge weights between nodes and their corresponding cluster assignments to determine the final community structure of the graph.

The nodes in the graph are then sorted based on their Local PageRank values, from highest to lowest. Formula 1 for Local PageRank is defined as:

$$\mathrm{PR}(v_i) = \frac{1-d}{N} + d\sum_{v_j \in M(v_i)} \frac{\mathrm{PR}(v_j)}{L(v_j)} \tag{1}$$

where d is the damping factor, v_i and v_j are the nodes under consideration in G', $M(v_i)$ is the set of nodes that have a link with v_i where v_i in G', $L(v_i)$ is the number of links in subgraph G', and N is the total number of nodes of G'.

All nodes in the network are arranged in descending order based on their Local PageRank values computed using Formula 1. In cases where nodes have the same Local PageRank, these nodes are arranged in ascending order of node number. We store the nodes with their Local PageRank, establishing a ranked list of nodes based on their importance within their local neighborhood.

Algorithm 2 presents this step.

highest Local PageRank score found in the previous stage, and adding it and its neighbors to a community. The similarity between every pair of communities is then computed using a measure based on the percentage of common nodes, and two communities are grouped if their similarity exceeds a threshold of 50% compared to the smallest one. The algorithm repeats this process, gradually reducing the level of overlap between communities until no more grouping is possible.

In this paper, we use a measure of similarity based on the percentage of common nodes of two communities. This similarity measure is non-commutative. The formula for the similarity measure is presented as follows:

$$\mathrm{Similarity}(C_i) = \frac{|C_i \cap C_j|}{|C_i|} \tag{2}$$

$$\mathrm{Similarity}(C_j) = \frac{|C_i \cap C_j|}{|C_j|} \tag{3}$$

here C_i and C_j are communities detected in the graph G; $|C_i|$ and $|C_j|$ are the number of nodes in C_i and C_j, respectively.

What interests us in these two measures is the maximum between them, we called SimMax. The formula for SimMax is presented as follows:

$$\mathrm{SimMax}(C_i, C_j) = \max(\mathrm{Similarity}(C_i), \mathrm{Similarity}(C_j)) \tag{4}$$

An advantage of this approach is its ability to assign nodes to multiple communities, reflecting the intricate and interconnected nature of real-world networks. The hierarchical arrangement of the resultant groupings can offer valuable understanding of the connections between communities at different levels of detail. However, we believe that using

Algorithm 2 Algorithm for Local PageRank

Require: *networks* G
Ensure: List of nodes stored by their Local PageRank Local
1: create subgraph G' using BFS with two hops from each node
2: compute Local PageRank for each node in G' using formula 1
3: Store the nodes with their Local PageRank in the list

3.2 Stage 02: overlapping clustering hierarchical

This stage presents an overlapping hierarchical clustering algorithm. The algorithm is based on selecting the node with the

disjoint communities with inclusion of nodes would provide more precise results. Therefore, we plan to utilize the number and centers of these communities in the next stage.

The algorithm for this stage is presented as Algorithm 3:

Algorithm 3 Overlapping Hierarchical Clustering

Require: List of nodes stored by their Local PageRank Local networks
Ensure: Overlapping community detection
1: Select node with highest Local PageRank score
2: Add nodes with their neighbors to communities until nodes have been assigned to at least one community
3: Compute the similarity between every pair of classes using the SimMax presented in formula 4
4: Group two classes if their similarity exceeds 50%
5: Repeat steps 3 and 4 until no more grouping is possible

3.3 Stage 03: K-means clustering

K-means clustering is a widely used method for partitioning a set of data points into K clusters. In the context of community detection, K-means can be used to group nodes in a network into K communities based on their similarity or distance. It is important to note that the quality of the community detection heavily depends on the choice of K, which is often determined by trial and error or by using external validation measures. Additionally, K-means can be sensitive to the initial choice of centroids and can sometimes converge to suboptimal solutions. Therefore, it is recommended to run the algorithm multiple times with different initializations and choose the best solution based on some criteria such as the minimum sum of squared distances between nodes and their assigned centroids.

In PCMeans, the number k of clusters to be equal to the number of communities identified in the previous stage and the centers of this communities are the initial centroids. The resulting K clusters generate the final community structure. We employ K-means clustering as a post-processing step after hierarchical clustering. Since K-means is faster and can partition data points into non-overlapping groups, we apply it to the cluster centers obtained from hierarchical clustering. The steps involved in the K-means clustering algorithm are present in algorithm 4:

Algorithm 4 k-means Algorithm

Require: Communities identified, k: the number of communities.
Ensure: C1,C2,..., CK
1: Set the number k is equal to the number of communities identified
2: Use the centers of the communities identified as the initial centroids
3: Assign each object to the nearest centroid of the cluster.
4: Recalculate the centroids of the newly formed clusters
5: Repeat steps 3 to 4 until the algorithm converges.

3.4 The time complexity

The time complexity of the PCMeans algorithm depends on the time complexity of the individual steps involved. The time complexity of the first stage is the time for calculating the Local PageRank score for each node is $T1$. The time complexity of overlapping hierarchical clustering is $T2$. The time complexity of running K-means clustering on the communities to group nodes into K non-overlapping clusters is $T3$.

where $T1$ = Time complexity(first stage) = $O(n \log n)$. $T2$ = Time complexity(second stage) = $O(K^2\, n \log n)$. $T3$ = Time complexity(third stage)= $O(\text{it} * K * n)$.

where n: The number of nodes in the network. It: The number of iterations required for convergence, K: The number of communities.

Overall, the time complexity of PCMeans algorithm can be expressed as:

$$O(nlogn + K^2 nlogn + it * K * n). \tag{5}$$

The time complexity of an algorithm is an important factor to consider when evaluating its performance. PCMeans has low time complexity compared to some other community detection algorithms. This is mainly due to the use of Local PageRank to identify the most influential nodes, which reduces the search space and computation time. Additionally, the overlapping hierarchical clustering strategy used in PCMeans can reduce the number of iterations required to converge to the final community structure, as it can quickly identify the optimal number of clusters. Finally, the K-means algorithm used in PCMeans is known for its rapid

Table 1 Modularity and NMI of different algorithms with real networks

		Karaty	Dolphin	PolBooks	Football
Reference		Zachary (1977)	Lusseau et al. (2003)	Krebs (2008)	Jiang and McQuay (2012)
Nodes		34	62	105	115
Edges		78	159	440	612
Number C		2	2	3	12
Number C	Louvain (Blondel et al. 2008)	3	5	5	10
	Newman (Newman and Girvan 2004)	4	5	4	8
	INFOMAP (Rosvall and Bergstrom 2008)	3	5	6	12
	Fast-Greedy (Parés et al. 2017)	4	4	4	7
	LPA (Raghavan et al. 2007)	3	4	3	8
	PCMeans	2	2	3	10
Quality measures					
Q	Louvain (Blondel et al. 2008)	**0.4151**	0.5176	**0.5267**	**0.6045**
	Newman (Newman and Girvan 2004)	0.3943	0.4912	0.4671	0.4926
	INFOMAP (Rosvall and Bergstrom 2008)	0.4020	**0.5247**	0.5228	0.6005
	Fast-Greedy (Parés et al. 2017)	0.3806	0.4954	0.5018	0.5784
	LPA (Raghavan et al. 2007)	0.3990	0.4939	0.4565	0.5944
	PCMeans	0.3582	0.5191	0.4285	0.5081
NMI	Louvain (Blondel et al. 2008)	0.4899	0.5176	0.5368	0.8849
	Newman (Newman and Girvan 2004)	0.5791	0.4912	0.4671	0.6986
	INFOMAP (Rosvall and Bergstrom 2008)	0.5683	0.5247	0.5228	**0.9241**
	Fast-Greedy (Parés et al. 2017)	0.5398	0.4954	0.5018	0.7623
	LPA (Raghavan et al. 2007)	0.5683	0.4939	0.4986	0.8507
	PCMeans	**1.0000**	**0.8140**	**0.6750**	0.9160
ARI	Louvain (Blondel et al. 2008)	0.3922	0.2708	0.6421	0.8034
	Newman (Newman and Girvan 2004)	0.4351	0.3901	0.5466	0.4640
	INFOMAP (Rosvall and Bergstrom 2008)	0.5905	0.2830	0.5360	**0.8966**
	Fast-Greedy (Parés et al. 2017)	0.5351	0.4954	0.6378	0.5363
	LPA (Raghavan et al. 2007)	0.5905	0.4647	0.6745	0.6839
	PCMeans	**1.0000**	**0.7659**	**0.7623**	0.8543

The best and most significant results are indicated in bold

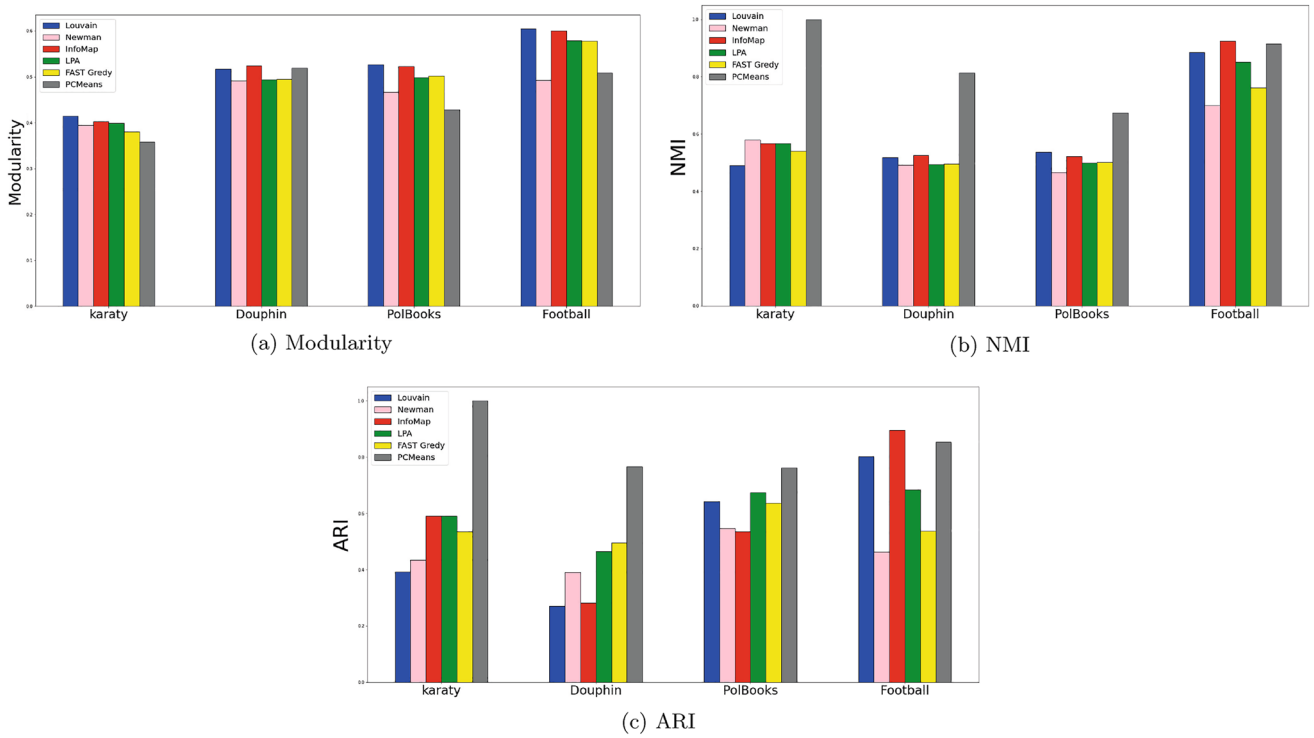

Fig. 2 Comparison of performance metrics between PCMeans and other algorithms

convergence and effective classification in large-scale datasets, which further contributes to the algorithm's low time complexity. Overall, the low time complexity of PCMeans makes it a practical and efficient method for community detection in large networks.

4 Experiments

PCMeans algorithm as an effective tool for community detection in complex networks. It has been compared to other state-of-the-art algorithms (Louvain algorithm (Blondel et al. 2008), Girven-Newman (Newman and Girvan 2004), INFOMAP Fluid Communities algorithm (Parés et al. 2017), and label propagation algorithm (LPA) (Raghavan et al. 2007)).and found to be competitive in terms of accuracy and efficiency. However, the effectiveness of the algorithm may depend on the specific characteristics of the network being analyzed and the parameter settings used in the algorithm. Therefore, it is recommended to experiment with different parameter values and compare the results to other algorithms to determine the effectiveness of PCMeans in a given application. The effectiveness of the PCMeans algorithm was evaluated by conducting experiments on both synthetic and real social networks using Python implementation. The experiments were performed on a computer with an Intel Core i7 processor and 16GB of RAM. The approach was evaluated using four real network datasets and the Lancichinetti Fortunato Radicchi benchmark network.

4.1 Evaluation measures

Modularity (Q) Newman and Girvan (2004) A metric for measuring the quality of communities in a network. The higher the value of Q (closer to 1), the better the result.

Normalized mutual information (NMI) Danon et al. (2005) and Li et al. (2021) A performance measure based on information theory that compares the true partition of a network with the partition obtained by an experimental community detection algorithm. The higher the value of NMI, the better the performance of the algorithm.

Adjusted Rand Index (ARI) A measure of similarity between two partitions that corrects for chance agreement between the partitions. The ARI ranges from -1 to 1, with 1 indicating perfect agreement, 0 indicating agreement no better than chance, and -1 indicating complete disagreement.

These measures are commonly used to evaluate the performance of PCMeans community detection algorithms.

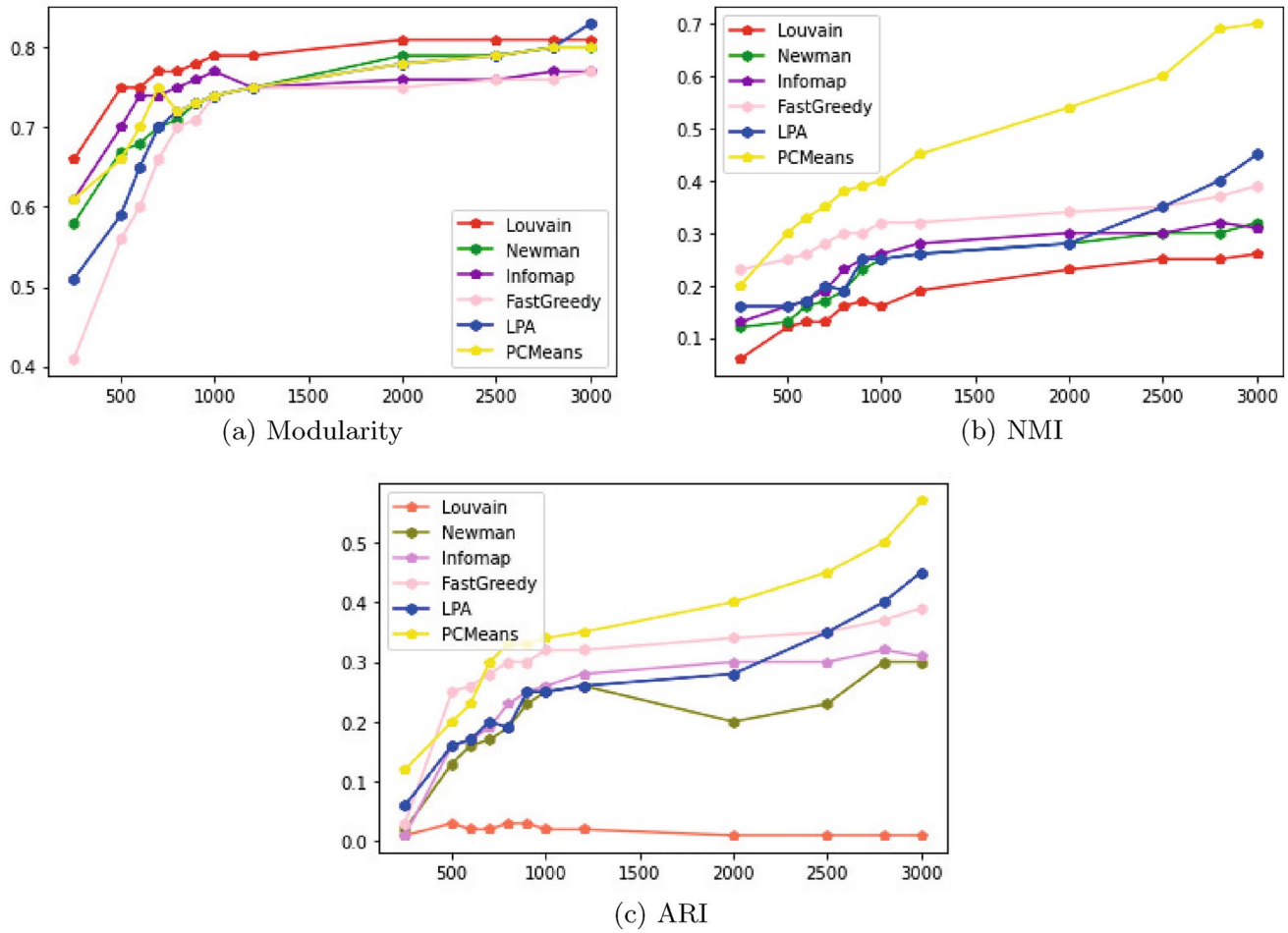

Fig. 3 Comparison of performance metrics in artificial networks

4.2 Experimental results and analysis in networks

To validate the performance of the proposed algorithm, we used four real-world networks: Zachary Karate (Zachary 1977), Dolphin (Lusseau et al. 2003), Political Books (Krebs 2008), and Football (Jiang and McQuay 2012), along with five artificial networks from the LFR benchmark.

4.2.1 Experiments on real-world networks

Calculate the modularity Q, NMI, and ARI for different algorithms. Each result is an average of 30 repetitions, with the best result highlighted. The descriptions of these networks and all results can be found in Table 1.

Results of this table are presented in Fig. 2a–c for modularity, NMI, and ARI by order.

Moreover, while several algorithms exhibit unstable results, our algorithm remains stable. In summary, PCMeans produces similar partitions as compared to the other algorithms, while maintaining stability in its results.

The proposed algorithm achieves remarkable results on various real-world networks. On Zachary's karate club network, the algorithm converges to the global optimal, as indicated by the NMI score of 1. This means that the communities identified by the algorithm are identical to the actual communities. Similar excellent results are also observed on the Football club network, the Political book network, and the Dolphin network, with average NMIs of 0.9160, 0.6750, and 0.8140, respectively. The proposed algorithm outperforms all other experimental algorithms in terms of NMI and ARI indicators on all real-world networks. It also achieves the best ARI score for all experiments. Furthermore, on the Political book network, the algorithm identifies the second-best modularity score of 0.5191, which is comparable to the performance of INFOMAP. Despite not achieving the best modularity score,

the NMI and ARI scores demonstrate that our algorithm's performance is very close to the real results. This highlights the reliability and effectiveness of the proposed algorithm in identifying community structures in real-world networks.

4.2.2 Experiments on generated network "LFR benchmark"

To assess the accuracy of community detection algorithms, synthetic networks are often utilized due to the ability to control the network's properties and obtain the corresponding real community structure through tunable parameters. The LFR benchmark network, developed by Lancichinetti, Fortunato, and Radicchi, is a widely adopted synthetic network model that exhibits similar properties to real-world networks. Our study focuses on LFR benchmark networks with 250 to 3000 nodes and average degree between 5 and 10. The degree distribution and community size distribution exponents are set to $t1 = 1.5$ and $t2 = 3$, respectively. Additionally, we consider two ranges of community sizes: $s = (10{:}50)$ and $b = (20{:}100)$, and maximum degrees of 20 or 50. The mixing parameter nu, which represents the expected fraction of links through which a node connects to other nodes in the same community, is set to 0.1 or 0.3. We evaluate the performance of different algorithms on all artificial networks using modularity, NMI, and ARI metrics.

All the results are present in Fig. 3a for modularity, in Fig. 3b for NMI and in Fig. 3c for ARI.

PCMeans achieved the best NMI and ARI results on LFR artificial networks, indicating its superior performance compared to other algorithms. Although its modularity results were not the best, they were still satisfactory

5 Conclusion

In this study, we introduced the concept of Local PageRank through a community detection algorithm that combines K-means clustering and overlapping hierarchical clustering. Our proposed PCMeans algorithm leverages hierarchical clustering to study the similarity between nodes and reduces the number of iterations required for community splitting, resulting in improved efficiency. By initializing the population according to Local PageRank and using the proposed neighbor-based clustering operator, PCMeans outperforms traditional approaches with random initialization, as evidenced by our experiments on both real-world and synthetic networks. Specifically, our method achieves higher NMI and ARI values, indicating that it produces community structures that closely resemble real community structures, and the results are stable. In the future, we plan to explore PCMeans for use in very large dynamic complex networks.

Author contributions LW wrote the main manuscript text and FT prepared figures and algorithms.

Declarations

Conflict of interest The authors declare no competing interests.

References

Akbar Z, Liu J, Latif Z (2021) Mining social applications network from business perspective using modularity maximization for community detection. Soc Netw Anal Min 11:1–19

Bar-Yossef Z, Mashiach L-T (2008) Local approximation of pagerank and reverse pagerank. In: Proceedings of the 17th ACM conference on information and knowledge management, pp. 279–288

Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. J Stat Mech Theory Exp 2008(10):10008

Brin S, Page L (1998) The anatomy of a large-scale hypertextual web search engine. Comput Netw ISDN Syst 30(1–7):107–117

Cai B, Zeng L, Wang Y, Li H, Hu Y (2019) Community detection method based on node density, degree centrality, and k-means clustering in complex network. Entropy 21(12):1145

Chaudhary L, Singh B (2021) Community detection using unsupervised machine learning techniques on covid-19 dataset. Soc Netw Anal Min 11:1–9

Danon L, Díaz-Guilera A, Duch J, Arenas A (2005) Comparing community structure identification. J Stat Mech Theory Exp 2005(09):09008–09008. https://doi.org/10.1088/1742-5468/2005/09/p09008

Fortunato S (2010) Community detection in graphs. Phys Rep 486(3–5):75–174

Fortunato S, Hric D (2016) Community detection in networks: a user guide. Phys Rep 659:1–44

Geng J, Bhattacharya A, Pati D (2019) Probabilistic community detection with unknown number of communities. J Am Stat Assoc 114(526):893–905

Guo K, Huang X, Wu L, Chen Y (2022) Local community detection algorithm based on local modularity density. Appl Intell 52(2):1238–1253

Hajij M, Said E, Todd R (2020) Pagerank and the k-means clustering algorithm. arXiv preprint arXiv:2005.04774

Hollocou A, Bonald T, Lelarge M (2016) Improving pagerank for local community detection. arXiv preprint arXiv:1610.08722

Jiang JQ, McQuay LJ (2012) Modularity functions maximization with non negative relaxation facilitates community detection in networks. Phys A Stat Mech Appl 391(3):854–865

Krebs V (2008) A network of co-purchased books about us politics. October 20(1), 0–03

Kumar A, Barman D, Sarkar R, Chowdhury N (2020) Overlapping community detection using multiobjective genetic algorithm. IEEE Trans Comput Soc Syst 7(3):802–817

Li H, Zhang R, Zhao Z, Liu X (2021) Lpa-mni: an improved label propagation algorithm based on modularity and node importance for community detection. Entropy 23(5):497

Liu X, Fu L, Wang X, Zhou C (2022) On the similarity between von Neumann graph entropy and structural information: interpretation, computation, and applications. IEEE Trans Inf Theory 68:2182–2202

Luo W, Lu N, Ni L, Zhu W, Ding W (2020) Local community detection by the nearest nodes with greater centrality. Inf Sci 517:377–392

Lusseau D, Schneider K, Boisseau OJ, Haase P, Slooten E, Dawson SM (2003) The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. Behav Ecol Sociobiol 54(4):396–405

Ma T, Wang Y, Tang M, Cao J, Tian Y, Al-Dhelaan A, Al-Rodhaan M (2016) Led: a fast overlapping communities detection algorithm based on structural clustering. Neurocomputing 207:488–500

Ma T, Liu Q, Cao J, Tian Y, Al-Dhelaan A, Al-Rodhaan M (2020) Lgiem: global and local node influence based community detection. Futur Gener Comput Syst 105:533–546

Moosavi SA, Jalali M, Misaghian N, Shamshirband S, Anisi MH (2017) Community detection in social networks using user frequent pattern mining. Knowl Inf Syst 51(1):159–186

Musdar IA, Azhari S (2015) Metode rce-kmeans untuk clustering data. IJCCS (Indonesian Journal of Computing and Cybernetics Systems) 9(2):157–166

Newman ME, Girvan M (2004) Finding and evaluating community structure in networks. Phys Rev E 69(2):026113

Palla G, Derényi I, Farkas I, Vicsek T (2005) Uncovering the overlapping community structure of complex networks in nature and society. Nature 435(7043):814–818

Parés F, Gasulla DG, Vilalta A, Moreno J, Ayguadé E, Labarta J, Cortés U, Suzumura T (2017) Fluid communities: a competitive, scalable and diverse community detection algorithm, 229–240. Springer

Pourkazemi M, Keyvanpour MR (2017) Community detection in social network by using a multi-objective evolutionary algorithm. Intell Data Anal 21(2):385–409

Raghavan UN, Albert R, Kumara S (2007) Near linear time algorithm to detect community structures in large-scale networks. Phys Rev E 76(3):036106

Rosvall M, Bergstrom CT (2008) Maps of random walks on complex networks reveal community structure. Proc Natl Acad Sci 105(4):1118–1123

Sheng J, Liu C, Chen L, Wang B, Zhang J (2020) Research on community detection in complex networks based on internode attraction. Entropy 22(12):1383

Van Laarhoven T, Marchiori E (2016) Local network community detection with continuous optimization of conductance and weighted kernel k-means. J Mach Learn Res 17(1):5148–5175

Vilcek A (2014) Deep learning with k-means applied to community detection in networks. CS224W Project Report

Wu Z, Wang X, Fang W, Liu L, Tang S, Zheng H, Zheng Z (2021) Community detection based on first passage probabilities. Phys Lett A 390:127099

Yuan Y, Chen B, Yu Y, Jin Y (2020) An influence maximisation algorithm based on community detection. Int J Comput Sci Eng 22(1):1–14

Zachary WW (1977) An information flow model for conflict and fission in small groups. J Anthropol Res 33(4):452–473

Žalik KR (2008) An efficient k'-means clustering algorithm. Pattern Recognit Lett 29(9):1385–1391

Zhou X, Su L, Li X, Zhao Z, Li C (2023) Community detection based on unsupervised attributed network embedding. Expert Syst Appl 213:118937

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.