symmetry

MDPI

Article

An Overlapping Community Detection Approach in Ego-Splitting Networks Using Symmetric Nonnegative Matrix Factorization

Mingqing Huang [1], Qingshan Jiang [1,*], Qiang Qu [1] and Abdur Rasool [1,2]

[1] Shenzhen Key Lab for High Performance Data Mining, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China; mq.huang@siat.ac.cn (M.H.); qiang@siat.ac.cn (Q.Q.); rasool@siat.ac.cn or arsultan300@outlook.com (A.R.)

[2] Shenzhen College of Advanced Technology, University of Chinese Academy of Sciences, Shenzhen 518055, China

* Correspondence: qs.jiang@siat.ac.cn; Tel.: +86-755-8639-2340

Abstract: Overlapping clustering is a fundamental and widely studied subject that identifies all densely connected groups of vertices and separates them from other vertices in complex networks. However, most conventional algorithms extract modules directly from the whole large-scale graph using various heuristics, resulting in either high time consumption or low accuracy. To address this issue, we develop an overlapping community detection approach in Ego-Splitting networks using symmetric Nonnegative Matrix Factorization (ESNMF). It primarily divides the whole network into many sub-graphs under the premise of preserving the clustering property, then extracts the well-connected sub-sub-graph round each community seed as prior information to supplement symmetric adjacent matrix, and finally identifies precise communities via nonnegative matrix factorization in each sub-network. Experiments on both synthetic and real-world networks of publicly available datasets demonstrate that the proposed approach outperforms the state-of-the-art methods for community detection in large-scale networks.

Keywords: overlapping community detection; ego-splitting network; nonnegative matrix factorization; graph symmetry theory; priori information embedding

Citation: Huang, M.; Jiang, Q.; Qu, Q; Rasool, A. An Overlapping Community Detection Approach in Ego-Splitting Networks Using Symmetric Nonnegative Matrix Factorization. *Symmetry* **2021**, *13*, 869. https://doi.org/10.3390/sym13050869

Received: 15 March 2021
Accepted: 1 May 2021
Published: 13 May 2021

1. Introduction

Since the ground-breaking advent of online social networking, complex network analysis tools have been developed in the last decades for excerpting insights from the various relationships between participants [1]. Network analysis also has become a research hotspot to uncover critical patterns that facilitate the understanding of phenomena for a variety of applications. As can be learned from recent literature, such knowledge can be extracted for a myriad of practical purposes, such as the detection of impersonation [2], the inference of extremist propaganda [3], and the identification of child abuse [4].

Communities indicate similar opinions, functions, objectives, etc., which are ubiquitously and naturally present as basic modules in real-world networks. Community detection is a fundamental problem in complex networks, consisting of the unsupervised division of elements into densely knitted and highly related clusters, where the connectivity between different groups is relatively loose. Revealing the clustering structure of real-world networks has emerged as a basic protocol in many data mining tasks, such as human seizure tracking [5], society influence maximization [6], cancer tissue phenotyping [7], and semantic trajectory clustering [8]. Consequently, research on the topology of real-world networks and their modular structure is at the core of network analysis.

In regard to measuring the cohesiveness of communities, different metrics have been designed in the literature for assessing the quality of any partition of a given graph [9]. Newman et al. conceived the famous modularity Q [10], leading to many algorithms

that optimize modularity Q or modularity density Q_D [11]. In recent years, a large quantity of generative-model-based methods also have been presented to capture clustering structure [12], which assume that edges are generated with probabilities based on their community memberships.

Since the convenient availability of large datasets, the general size of large networks such as the world-wide web, social networking services, or mobile phone networks now counts in millions of vertices if not billions and these scales require new approaches to extract comprehensive information from their topological structure. Consequently, one of the persistent challenges in recent years is how to design a fast algorithm for precisely retrieving communities from large-scale networks [13].

Unfortunately, most previously proposed schemes capture the community structure at a macroscopic level with low precision and high time consumption, due to the lack of a distinct macroscopic clustering view in real-world networks. By contrast, the community detection mission becomes easy at a microscopic level, especially when we restrict our attention to local structures [14]. Inspired by this observation, we propose an overlapping community detection algorithm in Ego-Splitting networks using symmetric Nonnegative Matrix Factorization (ESNMF) in this research, which primarily divides the large-scale graph into many sub-networks preserving clustering attributes, and then accurately discovers the communities via nonnegative matrix factorization [15], along with priori information embedding.

The main contributions and characteristics of our present paper are itemized as follows.

- The overlapping clustering issue in global networks is transformed into a partitioning problem using the ego-splitting framework without changing community property, to scale up with the increase of network size.
- High-quality groups of vertices are retrieved as priori information to incorporate into clustering algorithms properly, not only improving the accuracy of these methods but also accelerating their speed.
- By integrating partial supervision and nonnegative matrix factorization, a semi-supervised clustering scheme is proposed for enhancing accuracy obviously without increasing time complexity.

The remainder of the paper is organized as follows. Section 2 reviews the contribution of some related work by analyzing the state of the art of the main idea. The overall framework of the proposed approach and the corresponding functionality of major components are expounded in depth in Section 3. The experimental evaluation on synthetic and real-world networks of four open datasets is illustrated and discussed in Section 4. Finally, we present the conclusion briefly with several prospects for future work in Section 5.

2. Related Work

A short glance at the recent literature reveals the increasing effectiveness of community detection technology in scientific panorama. The widespread societal influence of online social networks lit the wick of promoting interest in this field, causing the valuable knowledge that can be extracted from community structures to be strongly regarded. This statement is supported by vast quantities of comprehensive surveys published in related investigations [16].

Community detection, also entitled *network clustering*, is an unsupervised learning technique for partitioning vertices into groups in consideration of topological structure [17]. Individuals within each cluster are tightly linked, whereas external connections are relatively sparse. Various community detection schemes have been proposed to cater toward diverse application requirements, which, loosely speaking, can be divided into several categories as follows.

2.1. Global Topological Analysis

Taking all the connections of the network into consideration simultaneously, graph characteristic optimization through statistical analysis has been widely used in modern solvers [18].

Internal Density [19] can discover non-overlapping communities based on the predefined modularity, and this metric is effective and robust for identifying community. *Diffusion* [20] is a propagation process in which the spread of influence is used to detect communities. In spectral clustering, a modularity matrix is built from the original network, and then the community is distinguished based on the eigenvector analysis of the constructed matrix. *Information Discovery Using Community Detection* [21] identifies overlapping communities of authors from the big scholarly data, where the interactions between authors are modeled as a novel graph by combining document metadata with semantic information. *Structure Mining* [22] can be regarded as graph searching, aiming to seek the maximal structures that conform to some constraint rules. The clique percolation method searches for the maximal cliques in the network, and these maximal cliques are then leveraged to form the connected subgraphs of k-cliques deemed as communities.

With the current explosive increase in network size, these global strategies are computational infeasible in practical applications due to their low efficiency.

2.2. Local Seed Expansion

To tackle the problem of low efficiency in global methods, many greedy algorithms have recently been designed [23], where the procedure consists of selecting influential vertices as a seed set, forming initial clusters, and greedily adding homeless vertices into clusters, relying on a local benefit function. This greedy expansion iteratively executes until the value of the benefit function stops increasing.

Ameliorated Local Fitness Maximization [24] discovers overlapping communities using initial community set expansion and optimization relying on a local fitness function, which attains linear time complexity without loss of effectiveness via multiple-vertex removal and addition on the premise of prohibiting community drift. *Two Expansions of Seeds* [25] distinguishes the local maximum vertices as the seeds by adapting the topological feature of the network, and then twice expands seeds based on the fitness function and the gravitational degree. The distance between correlative communities is computed to merge similar communities. *Neighborhood-Inflated Seed Expansion* [26] presents a seed expansion approach for overlapping community detection, where seeds are transformed to represent their entire vertex neighborhood. *Local and Global Influence Expanding and Merging* [27] conceives a metric to identify influential vertices as seeds based on global and local topology, using a novel strategy that calculates the similarity and distance between unsigned vertices and existing communities in the expansion stage.

The drawback of overemphasizing local information and ignoring/weakening global structure in these strategies leads to low effectiveness, which can be improved substantially.

2.3. Deep Learning Transformation

Because the majority of vertices are unlabeled, and there is little to no prior knowledge about clustering in many real-world scenarios, deep learning is an excellent choice for unsupervised learning tasks. Deep learning is also much more resilient to the sparsity present with large-scale networks [28].

Inspired by the mighty representation power of deep neural networks, *Modularity-Based Deep Learning* [29] brings forward a novel nonlinear reconstruction strategy by adopting deep neural networks for representation of realistic scenarios, and then applies the strategy to a semi-supervised community detection method by incorporating pairwise constraints between graph vertices. *Deep Community Detection* [30] puts forward a graph embedding method combining an auto-encoder with a convolutional neural network, which reconstructs the adjacency matrix with spatial proximity based on the opinion leader and nearer neighbors, for extracting higher spatial features with lower dimensions. *Game*

Theory [31] models the process of community formation as a game by representing each vertex with a playing actor, which makes predictions about behaviors of the actor in an interdependent scenario. *Network Structure Transformation* [32] engages a denoising auto-encoder to nonlinearly map the probability transfer matrix, which is calculated from the network adjacency matrix, into a new subspace. Network vertices are then clustered via *k*-means clustering to obtain communities.

There still exist challenges in deep learning that need better solutions, such as dealing with networks that contain an unknown number of communities, network heterogeneity, signed information on edges, hierarchical networks, community embedding.

As evinced by the above description, community detection has been conducted over various formulations of the underlying combinatorial optimization. However, we note that there are several drawbacks in existing methods, such as one-sided consideration of local topology or global structure and applying machine learning technology mechanically. To remedy these limitations, a community detection algorithm is proposed in this study, which fragments the whole network into small pieces on the premise of preserving clustering structure, and extracts communities in each subnet for performance enhancement in aspects of efficiency and effectiveness.

3. Proposed Approach

In this section, we first display the framework of our ESNMF approach. Next, we describe in detail the procedures corresponding to functional modules. Finally, we discuss the time complexity by theoretical analysis.

3.1. System Framework

Throughout this paper, we focus our discussions on undirected and weighted networks, which manifest as symmetric matrices. However, our proposal can be easily extended to deal with directed networks with modest adjustments.

To achieve an accurate and efficient solution for community detection, the procedure consists of three crucial steps, namely, ego-splitting partitioning, priori information embedding, and nonnegative matrix factorization, which are illustrated in Figure 1 with different panes surrounding them, where network datasets have been plugged as the necessary prerequisite.

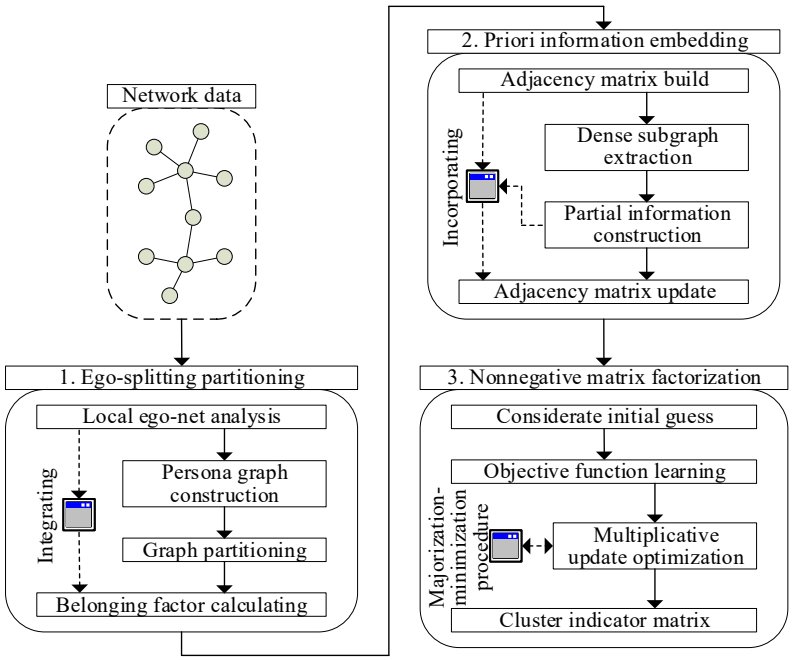


Figure 1. Structure overview of the proposed approach.

All the components are explained in detail in the upcoming sections, where the general process is outlined as follows.

(i) The global network is divided into many connected sub-graphs through the ego-splitting process, which preserves strictly the clustering attributes of the original graph.

(ii) Instead of directly factorizing the adjacency matrix, the well-connected motifs (sub-networks) are then extracted via a greedy algorithm as priori information to supplement the underlying data.

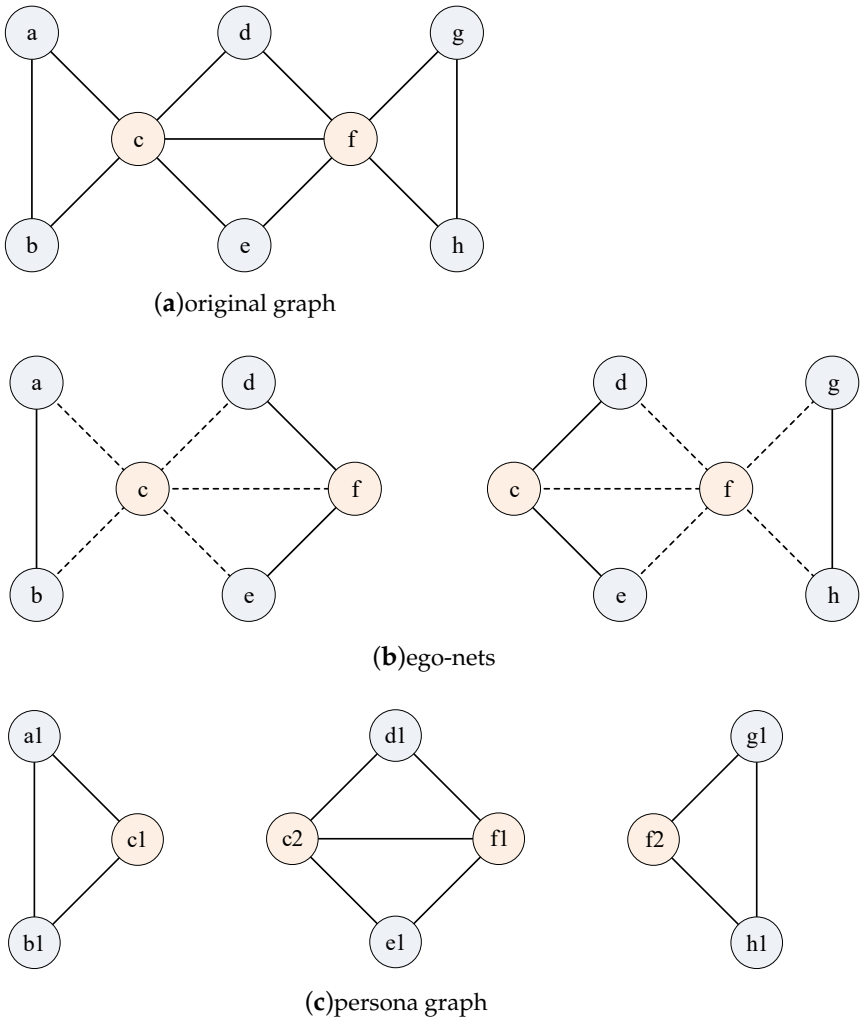
(iii) Nonnegative matrix factorization for network clustering is conducted using a simple initialization and an iterative multiplicative updating rule.

3.2. Ego-Splitting Partitioning

The main idea behind ego-splitting partitioning is to leverage the guidance of connectivity neighborhood structure for reducing the overlapping clustering issue to a non-overlapping partition problem along with the community membership calculation of overlapping vertices belonging to relevant clusters.

3.2.1. Persona Graph Construction

The ego-splitting procedure consists of two steps: local ego-net analysis and global network partitioning, as illustrated in Figure 2.



(**a**)original graph

(**b**)ego-nets

(**c**)persona graph

Figure 2. Graphical representation of ego-splitting on a simple network.

Firstly, ego-splitting constructs the ego-nets for each vertex u, which corresponds to the partitioning of the neighborhood of u through the connected component strategy. Taking the neighborhood of c in Figure 2b for instance, the two ego-nets of c are easily identified — they correspond exactly to the two connected sub-graphs $\{a, b\}$ and $\{d, e, f\}$.

Then, ego-splitting creates a new replica of u exactly for each ego-net in the partition, which is called a persona. The edges between vertices in original graph are duplicated between personas. Figure 2c represents the persona graph, which corresponds to three overlapping sub-networks $\{a_1, b_1, c_1\}$, $\{c_2, d_1, e_1, f_1\}$, and $\{f_2, g_1, h_1\}$.

The transformation from the original network to the graph of personas will expand the number of vertices in the graph but keeps the number of edges constant. Furthermore, the partitioning of the persona graph can be treated as a clustering of the edges, due to the one-to-one mapping of the edges between two graphs. For more information about ego-splitting, readers can refer to [14].

3.2.2. Community Membership Calculating

The overlapping vertex that correlates to more than one persona is jointly overlapped in multiple sub-graphs. Thus, the belonging factors are unequally distributed in these related sub-graphs but sum up to 1.

Assuming that $S_u^{(i)}$ denotes the ith sub-graph of vertex u, and $N_u^{(i)}$ indicates the neighborhood of u affiliated with $S_u^{(i)}$, the belonging factor of u in $S_u^{(i)}$, represented by $\beta_u^{(i)}$, is calculated [24] as

$$\beta_u^{(i)} = \sum_{v \in N_u^{(i)}} \frac{1}{o_u o_v} \cdot \frac{1}{2W} \left(w_{uv} - \frac{d_u d_v}{2W} \right) \tag{1}$$

where w_{uv} stands for the weight of edge between u and v, W indicates the sum of weights of all edges in global network, d_u expresses the sum of weights of edges incident on u, and o_u counts the number of sub-graphs associating with u.

Without loss of generality, the sum of belonging factors of each vertex in variant sub-graphs is then normalized to 1 (uniform scale).

After identifying the clustering structure in every sub-graph, we then extend the belonging factor of each vertex in the sub-graph to the membership degree of the vertex in the associated community analogically.

3.3. Priori Information Embedding

The clue that a dense sub-graph is very likely in a community can help extract useful priori information, which consists of local seed selection and priori information representation.

3.3.1. Seed Selection

The conductance $\phi(T)$ of a connectivity vertex set T in a global network [33] is defined as

$$\phi(T) = \frac{cut(T)}{min(vol(T), vol(\bar{T}))} \tag{2}$$

where $cut(T)$ indicates the sum of weights of edges connecting vertices in T to external ones, $vol(T)$ represents the sum of weights of edges incident on vertices in T, and $\bar{T}$ contains the complement set of T. In large-scale networks, we just need to calculate $vol(T)$ due to $vol(T) \ll vol(\bar{T})$.

The vertex u, of which the conductance of itself and its neighborhood reaches a local minimum, is selected as a seed. Specifically, the local minimum conductance here is restricted by

$$\phi(N'(u)) \leq \phi(N'(v)) \tag{3}$$

where $N'(u)$ contains u and the corresponding neighborhood $N(u)$, and $v \in N(u)$.

Under the condition that two or more seeds are adjacent to each other, which means that the equality in Equation (3) holds, only one needs to be marked and saved.

3.3.2. Priori Information Construction

Appropriately incorporating priori information into the clustering scheme not only increases the precision but also accelerates the speed. The procedure includes two main steps, namely, extracting high-quality sub-sub-groups and constructing partial information.

Firstly, a greedy algorithm based on seed-and-expand is leveraged to discover dense sub-sub-graphs. This strategy starts from a vertex (seed) and keeps expanding by adding a vertex repeatedly until the density of the sub-sub-graph is less than a predefined relative threshold α_t. Using α_f to represent the density of a full connectivity network with the average edge weight and α_c to represent the density of the current network, we fix $\alpha_t = \alpha_c + 0.8 \times \left(\alpha_f - \alpha_c\right)$ as default setting in this study [34].

The density sub-sub-graphs are indicated as $\left\{B_i^{[p]}\right\}_{i=1}^{g}$, where g denotes the number of sub-sub-graphs identified by the greedy strategy. A vector z_i is derived for each sub-sub-graph as

$$z_i(j) = \begin{cases} 1, & \text{if } u_j \in B_i^{[p]} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

which then deduces the partial matrix as

$$M^{[p]} = \bar{w} \sum_{i=1}^{g} z_i z_i^T \tag{5}$$

where $\bar{w}$ represents the average edge weight of network, and z_i^T is the transpose matrix of z_i. $M_{ij}^{[p]} > 0$ signifies that two correlative vertices are more likely to be clustered together. Finally, the priori information is incorporated into sub-graph S (symmetric matrix M) as

$$\hat{M} = M + rM^{[p]} \tag{6}$$

where r emerges as a tunable parameter balancing the effect of priori information. We set $r = 0.1$ as stated in [34]. Notice that there exist two restrictions: (i) each edge increases weight at most once; (ii) the upper limit of edge weight equals to 1.

3.4. Nonnegative Matrix Factorization

As an interpretable paradigm for dimensionality reduction, symmetric nonnegative matrix factorization [35] is leveraged for network clustering in this study since it uses the nonnegativity constraint to acquire parts-based representation.

3.4.1. Objective Function Leaning

The squared Euclid distance is a commonly used divergence that measures approximation error for community detection using nonnegative matrix factorization [34]. The objective function to be minimized can be formulated as

$$\min_{U \geq 0} \Gamma(U) = \left\| \hat{M} - UU^T \right\|_F^2 = \sum_{ij} \left[\hat{M}_{ij} - \left(UU^T\right)_{ij} \right]^2 \tag{7}$$

where $U \in \{0,1\}^{n \times c}$ denotes the cluster indicator matrix: if the ith vertex is allocated to the kth community, then $U_{ik} = 1$; otherwise $U_{ik} = 0$.

Unfortunately, directly optimizing over U leads to an NP-hard problem due to the discrete solution space. One of the popular solutions is combining nonnegativity with orthogonality to tolerate continuous relaxation. That is, U is replaced with $\hat{U}$ under constraints $\hat{U}_{ik} \geq 0$ and $\hat{U}^T\hat{U} = I$.

Minimizing $\left\|\hat{M}-\hat{U}\hat{U}^T\right\|_F^2$ over $\hat{U}$ subject to $\hat{U}^T\hat{U}=I$ equates with maximizing $Tr\left(U^T\hat{M}\hat{U}\right)$, where $Tr(A)=\sum_i A_{ii}$ is the trace of matrix A. To improve spectral clustering, the trace maximization is regularized by an additional penalty term on $\hat{U}$ as

$$\begin{aligned} &\min_{U\geq 0}\Gamma\left(\hat{U}\right)=-Tr\left(\hat{U}^T\hat{M}\hat{U}\right)+\lambda\sum_i\left(\sum_k\hat{U}_{ik}^2\right)^2 \\ &s.t.\ \hat{U}^T\hat{U}=I \end{aligned} \tag{8}$$

where $\lambda>0$ indicates the tradeoff parameter (it is fixed as $\lambda=1/2c$ in [36], here c denotes the number of clusters). The minimization emphasizes off-diagonal relevance in the trace because self-correlation usually gives little information for clustering vertices.

3.4.2. Multiplicative Update Optimization

Applying the Majorization-Minimization procedure in [37], the preliminary multiplicative update rule is employed, which can be used to solve the multiplier problem using the orthogonality constraint. Instead of multiplying directly in the preliminary update rule, an optimization strategy that iteratively executes the multiplicative update rule is acquired as

$$\hat{U}_{ik}^{new}=\hat{U}_{ik}\left[\frac{\left(\hat{M}\hat{U}+2\lambda\hat{U}\hat{U}^TV\hat{U}\right)_{ik}}{\left(2\lambda V\hat{U}+\hat{U}\hat{U}^T\hat{M}\hat{U}\right)_{ik}}\right]^{1/4} \tag{9}$$

where V denotes a diagonal matrix with $V_{ii}=\sum_l\hat{U}_{il}^2$.

The optimization procedure for establishment with respect to $\hat{U}$ is formally represented in Algorithm 1, where the number of clusters c is roughly set equal to the number of seeds (see Section 3.3.1).

Algorithm 1: Nonnegative matrix factorization with an iterative multiplicative update for network clustering.

Input: network adjacency matrix M;
Output: resulting community matrix U;

identify c number of seeds from M;
set the number of communities equal to c;
guess initial community matrix $\tilde{U}$;
construct the partial matrix $M^{[p]}$;
embed the priori information as $\hat{M}=M+rM^{[p]}$;
$\hat{U}=\tilde{U}$;
repeat
    update $\hat{U}$ by Equation (9);
    replace NaN with 0.0 if there exist;
until the number of iterations reaches 100 or $\hat{U}$ converges;
discrete $\hat{U}$ to cluster indicator matrix U (set the maximum value equal to 1 and the others 0 in each row);
return U;

3.4.3. Community Initialization

A clustering scheme should start from a relatively considerate initial community guess to attain a better local optimum. Furthermore, a cheap initialization method with high accuracy is an optimal choice for our clustering approach.

Assigning each seed (refer Section 3.3.1) with different community index, we then engage *Cellular Automaton* [38] to iteratively perform proper matching between the remaining vertices and the existing communities to obtain

$$G_c(u) = agr\max_{1\le g\le c} \sum_{v\in C_g} w_{uv} \tag{10}$$

where $G_c(u)$ represents the initial sequence number of the community attached to by vertex u, and C_g indicates the gth community consisting of associated vertices that have been assigned.

In one iteration, we check all the neighbors of each homeless vertex and attach it to the appropriate community if there exist clusters in the neighborhood. Theoretical analysis and experimental results demonstrate that several rounds of iterations certify the completeness of community initialization.

3.5. Computation Complexity

At first blush, a naive way for isolating all ego-nets would be prohibitively expensive, with a time complexity of $O(nm)$ for n vertices and m edges in the global network. Epasto et al. [39] apply a combinatorial bound to the number of triangles, demonstrating all ego-nets can be separated in time $O\left(m^{3/2}\right)$, which is a significant gain, especially for sparse graphs.

The seed selection for priori information involves the comparison of conductance between neighbor vertices, requiring $O(m')$ time, where m' denotes the number of edges in sub-network. The computational complexity of the greedy algorithm for searching high-density groups is $O\left(cn'^2\right)$, here n' indicates the number of vertices in a sub-network. The construction of priori information is implemented on each sub-network generated by ego-splitting, which leads to a time complexity of $O(m)$ for the global network.

As for nonnegative matrix factorization, the matrix multiplication of each iteration requires operations of $O\left(cn'^2\right)$, resulting in a computation complexity of $O\left(cn'^2p\right)$, where p stands for the number of iterations. Since the original graph is partitioned into many sub-networks, and we extract clusters in each sub-network separately, so the time complexity for the global network is derived as $O(pm\ln(n))$.

In conclusion, because $m^{1/2} > \ln(n)$ in large-scale connectivity network, the overall computational complexity of our proposed approach simplifies to $O\left(m^{3/2}\right)$.

4. Experiments and Result Evaluations

Experiments are carried out on a personal computer with a four-core 3.4-GHz processor, 16GB of RAM and Windows Server 2008 R2. The implementation is done using the programming languages Java-1.8 and Matlab-R2018a and the relational database MySQL-7.6.

4.1. Evaluation Criteria

To assess the clustering accuracy of the proposed approach, we utilize community modularity and normalized mutual information as the evaluation metrics.

4.1.1. Community Modularity

Community modularity [10], i.e., the divergence between the actual density of edges in clusters and the desired one of random graphs regardless of community structure, is calculated as

$$Q = \frac{1}{2W}\sum_{i=1}^{c}\sum_{u\in C_i, v\in C_i}\left(w_{uv} - \frac{d_u d_v}{2W}\right)\beta_u^{(i)}\beta_v^{(i)} \tag{11}$$

where the parameter specifications are stated in Equations (1) and (10).

Generally, the more accurate the community mining result, the higher the modularity value.

4.1.2. Normalized Mutual Information (NMI)

Given two covers, X and Y, which represent the set of true modules and a set of clusters discovered by an algorithm, respectively, we must quantify how similar or different they are to assess the accuracy of the algorithm.

An assessment index [40] extended from normalized mutual information has become popular for evaluating overlapping community algorithms, and is defined by

$$NMI_{\max} = \frac{I(X:Y)}{\max(H(X), H(Y))} \tag{12}$$

$I(X:Y)$ indicates the mutual information between X and Y, calculated as

$$I(X:Y) = \frac{1}{2}[H(X) - H(X|Y) + H(Y) - H(Y|X)] \tag{13}$$

where $H(X)$ denotes the entropy of X and $H(X|Y)$ signifies the variation of information between X and Y.

The more precise the resulting communities are, the greater the NMI value is.

4.2. Baseline Methods

In this research, two other existing state-of-the-art algorithms are engaged to verify the proposal by comparison, which are the ego-splitting-based and seed-expansion-based algorithms.

4.2.1. Connected Component of Ego-Splitting (Con-Com)

The *Ego-Splitting* [14] framework is designed to de-couple overlapping communities in complex networks by leveraging local clustering structure, which works in two steps: local ego-net analysis and global network partitioning.

Firstly, ego-splitting constructs the ego-nets for each vertex u and then partitions the neighborhood of u. For each ego-net, ego-splitting creates a new replica of u, called a persona, that is associated uniquely with an ego-net in the partition. Each edge between vertices in the original network is mapped onto an edge between personas to derive a new network called the persona graph.

Finally, ego-splitting runs the simple connected component algorithm on the resulting persona graph and acquires the communities.

The time complexity is $O\left(m^{3/2}\right)$, where m stands for the number of links in the network.

4.2.2. Low Conductance Cut with PageRank (Con-Cut)

The conductance of a set of vertices indicates the probability that a one-step random walk escapes out which begins from the set. The vertices, of which the conductance of itself and its neighborhood achieves a local minimum, are located as seeds (see Section 3.3.1). Given a seed, the procedures of Con-Cut [33] for finding the corresponding personalized PageRank community are described as follows.

(i) Set $\alpha = 0.99$, which specifies the own transition probability.

(ii) Initialize the community to contain the seed and the associated neighborhood.

(iii) The expected community size σ is assigned as the number of vertices in the initial community.

(iv) Execute the personalized PageRank random walk until the tolerance of translatable probability achieves $1/(10\sigma)$.

(v) Sweep over all cuts induced by the ordering of the degree-weighted probabilities, and choose the optimum vertex set as the resulting community.

The degree-weighted probability of vertex u is mathematically described as

$$p'(u) = \frac{p(u)}{d_u} \tag{14}$$

where $p(u)$ indicates the primitive probability of u, and d_u signifies the sum of weights of edges incident on u.

The computational complexity is $O(m)$, which means that the Con-Cut algorithm has a linear runtime complexity with the number of edges in the network.

4.3. Experiments on Synthetic Networks

In this section, two undirected and weighted graphs with heterogenous distributions of vertex degree and community size are first built as artificial datasets. Then, we display the graphical representation of the corresponding experimental results along with short descriptions.

4.3.1. Artificial Network Construction

In this study, the synthetic networks with overlapping communities are generated using the Lancichinetti-Fortunato benchmark [41] (LF model) to perform comparisons among the involved schemes.

Both the assignments of vertex degree and cluster size rely on two power laws with exponents τ_1 and τ_2, separately. The number of vertices and the mean degree are expressed as n and $\langle k \rangle$, respectively. The implementation of the LF model consists of the following 5 steps:

(i) The vertex degrees $\{k_i\}$ are specified by taking n random digits from a power law with exponent τ_1, where the extrema $k_{\max}$ and $k_{\min}$ are restricted to guarantee that the average degree equals $\langle k \rangle$. A topological mixing parameter μ_t is leveraged: $k_i^{(in)} = (1 - \mu_t)k_i$ indicates the internal degree of vertex i, i. e. the number of its links connected with neighbors in a common community.

(ii) The community sizes $\{y_j\}$ are sampled by drawing random numbers from another power law with exponent τ_2. Furthermore, the matching between vertices and clusters, which determines vertex assignment to a community, can be treated as a bipartite graph, where the two classes correspond to the n vertices and c communities separately.

(iii) To generate the whole network, c subgraphs, one for each community, are constructed. In practice, the configuration of community ε is nothing but a random subgraph of n_ε vertices with degrees $\left\{k_i^{(in)}(\varepsilon)\right\}$, which can be generated with a rewiring process to avoid multiple edges between any two vertices.

(iv) The links external to the communities are stochastically appended to the already-constructed network without altering the internal degree sequences, where $k_i^{(ext)} = \mu_t k_i$. To do this, a new network $\vartheta^{(ext)}$ of the same n vertices with degree sequences $\left\{k_i^{(ext)}\right\}$ is generated, and a rewiring procedure is executed under the case of an existing link between two vertices with a common community.

(v) To assign a positive real number to each link, two other parameters, φ and μ_w, need to be specified. The parameter φ is used to assign a strength t_i to each vertex i: $t_i = (k_i)^\varphi$; and the parameter μ_w is adopted to obtain the internal strength $t_i^{(in)}$: $t_i^{(in)} = (1 - \mu_w)t_i$.

We employ the fourth software package [41] that can be free downloaded for constructing directed networks and superpose the weights of two arcs between the same

vertices to determine the value of the corresponding edge in the undirected graph. Two synthetic graphs with variant scales are built through the software package, where the concrete parameters are described in detail below:

(1) A small and weighted network (LF-SW)

The parameter specifications for generating the LF-SW network are listed in Table 1.

Table 1. Parameter assignment of the LF-SW network.

Variable Description	Assigned Value
number of vertices	500
average degree	15
maximum degree	20
mixing parameter for the topology	0.28
mixing parameter for the weights	0.25
exponent for the weight distribution	1.5
exponent for the degree sequence	−1.8
exponent for the community size	−1.2
minimum community size	15
maximum community size	35
number of overlapping vertices	20
number of memberships (for each overlapping vertex)	2
time seed	75,413,212

(2) A large and weighted network (LF-LW)

Table 2 details the assignment of the parameters for constructing the LF-LW network.

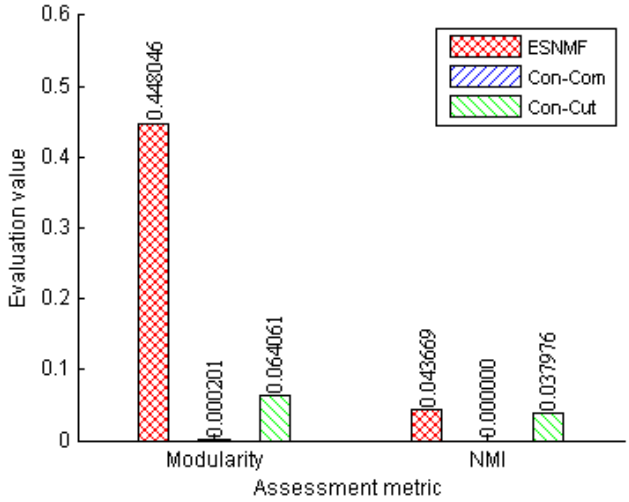
Table 2. Parameter specification of the LF-LW network.

Variable Indication	Specified Value
number of nodes	1000
average degree	16
maximum degree	25
mixing parameter for the topology	0.32
mixing parameter for the weights	0.28
exponent for the weight distribution	1.5
exponent for the degree sequence	−1.8
exponent for the community scale	−1.2
minimum community scale	20
maximum community scale	50
number of overlapping nodes	50
number of memberships (for each overlapping node)	2
time seed	78,452,144

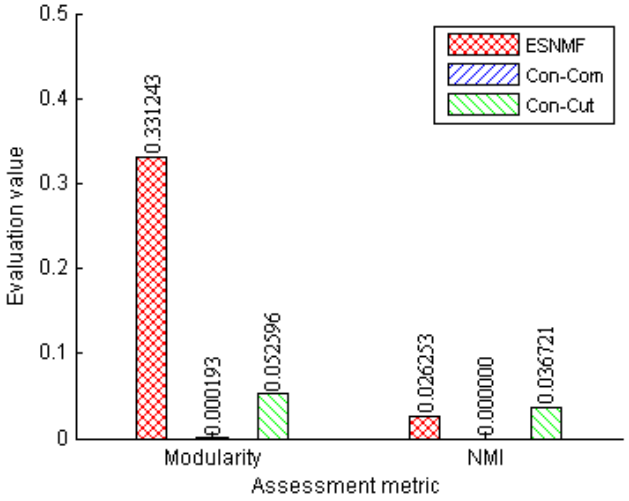
4.3.2. Results on Artificial Networks

We compare the assessment indicators of the involved methods on the LF-SW and LF-LW networks. As demonstrated in Figure 3, one can observe that our ESNMF approach achieves the maximum performance among all compared algorithms, with an obvious advantage on community modularity, even though it does not obtain the highest value on NMI on the LF-LW network. It is worth pointing out that the Con-Com method scores 0.0 on NMI on both the artificial networks due to a connectivity persona sub-graph that includes the whole original network. As for the reason why the ESNMF scheme shows a more outstanding effectiveness on community modularity but there are little differences on NMI between all involved methods, we argue that one of the most important causes is

the probabilistic dependency of the generating procedure for LF-SW and LF-LW networks, which can be avoid in actual networks.



(**a**)LF-SW



(**b**)LF-LW

Figure 3. Effectiveness comparison among different network clustering methods on the synthetic graphs.

As for the efficiency comparison, despite the fact that our framework consumes the longest time compared with the other schemes as illustrated in Figure 4, these differences are not very significant, diverging in the same order of magnitude, which means that the ESNMF method can satisfy the practical requirements. It should be pointed out that our approach can be implemented in parallel due to its design, which is the future direction of our work for further decreasing time consumption.

To clarify the time consumption of our ESNMF method in detail, Table 3 illustrates the burning time of each procedure, which highlights the direction for improving efficiency in future (conceiving a fast strategy for priori information embedding).

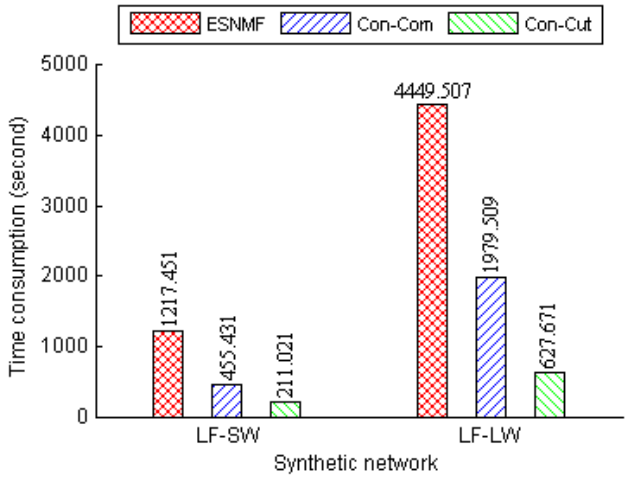


Figure 4. Time consumption of three community detection schemes running in serial on the artificial networks.

Table 3. Time consumption details of ESNMF on the synthetic networks.

Procedure Description	Execution Time on the LF-SW Network (Millisecond)	Execution Time on the LF-LW Network (Millisecond)	Sequence Number (Descending Sort)
building ego-nets for each vertex	33,012	197,384	7
persona graph construction	85,386	433,037	6
community seed selection	181,640	559,303	4(LF-SW)/5(LF-LW)
priori information construction	279,861	969,345	1
community initialization	254,154	760,662	2(LF-SW)/3(LF-LW)
nonnegative matrix factorization	159,059	662,376	5(LF-SW)/4(LF-LW)
community membership calculating	224,339	867,400	3(LF-SW)/2(LF-LW)

4.4. Experiments on Real-World Networks

In this section, we generate two real-world networks by downloading publicly available datasets and then carry out some contrastive experiments.

4.4.1. Real-World Network Generation

Stanford University has gathered a lot of network datasets and shares them to the public for free [42]. Given these conveniences, we pick out two ones to build the appropriate networks for performance qualification among compared methods.

(1) Email network from an institution (Email-Eu-Core)

The graph is built using interactive data from a large European research institution. There exists an edge (u, v) in the network if individual u sent to or received from v at least one email. The dataset also indicates the ground-truth of community membership for individuals, where everyone belongs to exactly one of 42 departments at the institute. The original network contains 1,005 vertices and 25,571 edges by statistics. In the data preprocessing, by assigning the constant of 0.5 to the weight value of each edge, two edges between the same vertices merge into one with weight of 1.0, which results in a concordant network including 16,706 edges.

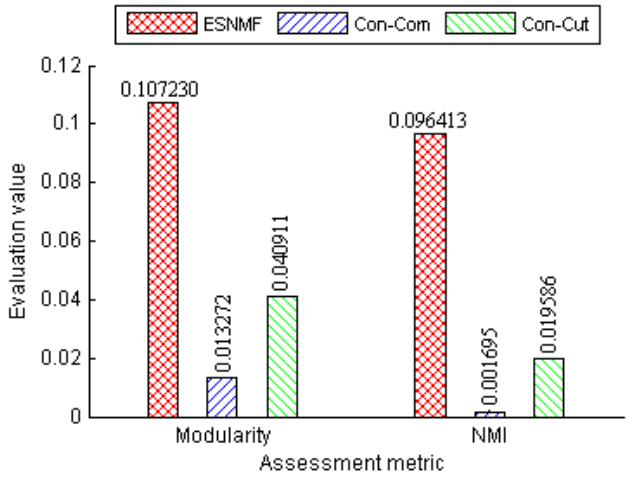
(2) Navigation network on Wikipedia (Wikispeedia)

The network is composed of human navigation paths on Wikipedia, acquired via the human-computation game called Wikispeedia. In Wikispeedia, individuals are required to navigate from a given article to a specified target source, by only clicking Wikipedia

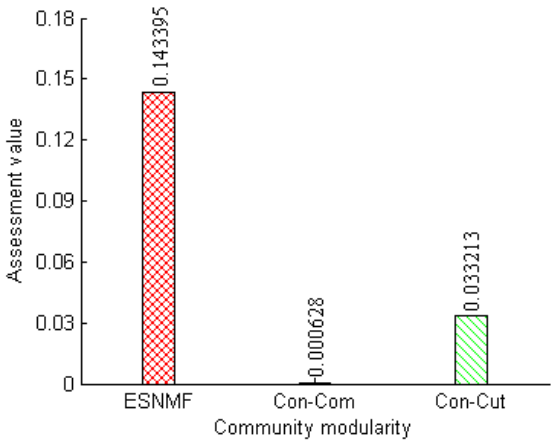
links. There are 4,604 vertices and 119,882 edges in the naive initial data. Using the same pre-processing method, we gain the concordant network with 106,647 edges in total.

4.4.2. Results on Real-World Networks

As displayed in Figure 5, our proposed approach has the best performance across all evaluation standards on these real-world networks. It needs to be stated here that, without knowing the ground truth of community structure, only the community modularity (assessment metric) can be adapted to evaluate the Wikispeedia network.



(**a**)Email-Eu-Core



(**b**)Wikispeedia

Figure 5. Performance comparison of three algorithms on the real-world networks for community detection.

The modularity values of all involved methods on the real-world networks are obviously lower than the ones on the synthetic networks, which can be attributed to the more obscure community structure.

Figure 6 enforces the efficiency results of all involved schemes on the real-world networks by serial processing. Despite some delay compared to the other two algorithms in time consumption, our proposal is still competitive since the distinctions are not substantial.

Table 4 demonstrates that the process of priori information construction costs a large amount of time, making it the primary target for improving the efficiency of our ESNMF algorithm.

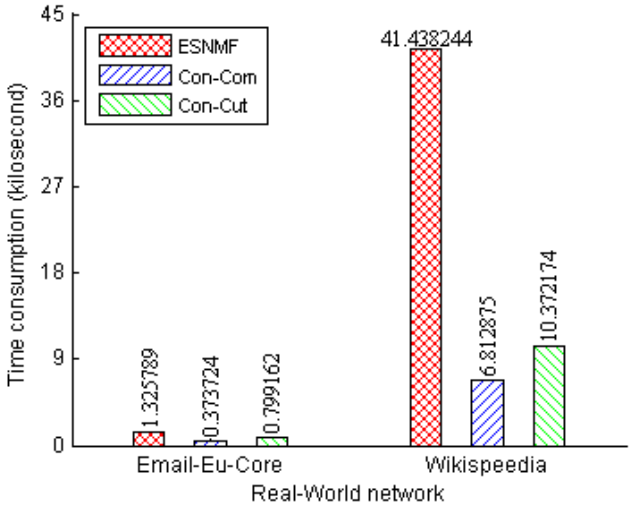


Figure 6. Serial execution time of three methods on the real-world networks for network clustering.

Table 4. Time spent of ESNMF in details on the real-world networks (*Email-Eu-Core* and *Wikispeedia* are abbreviated as Email and Wiki, respectively).

Step Description	Running Time on the Email Network (Millisecond)	Running Time on the Wiki Network (Millisecond)	Sequence Number (Descending Order)
building ego-nets for each node	38,194	730,991	7
persona graph construction	73,944	1,212,257	3(Email)/6(Wiki)
community seed selection	45,078	1,479,463	5
priori information construction	927,053	32,406,573	1
community initialization	46,731	1,632,527	4(Email)/3(Wiki)
nonnegative matrix factorization	39,720	1,614,154	6(Email)/4(Wiki)
community membership calculating	155,069	2,362,279	2

4.5. Discussion and Analysis

Puzzled by the above experimental outcomes, we further provide more details and discussion about the comparative results between our approach and the other two methods.

The procedure description revealing that our ESNMF algorithm is of particular relevance to the Con-Com and Con-Cut methods, but the diversity of performance exhibition between them is relatively prominent. We can investigate its reason from following aspects thinking. The Con-Com framework merges many clusters of fact into a community by terminating the further subdivision of resulting persona sub-graphs and regarding each connected component as a module, leading to ultra-low capability. The Con-Cut way absorbs extra vertices of other communities into the current cluster using PageRank random walk, inevitably resulting in low performance. The ESNMF algorithm continually discovers communities in persona sub-graphs after Ego-splitting of the original network, which achieves satisfactory success.

On the premise of preserving clustering structure, a large-scale network is fragmented into sub-graphs, which leads to a sharp decline of execution time of nonnegative matrix factorization. Therefore, the runtime of the ESNMF algorithm is in the same order of magnitude with the Con-Com and Con-Cut methods, which are two outstanding delegates in terms of efficiency for community detection.

5. Conclusions and Future Work

A novel proposal is detailed in this paper for overlapping community detection in large-scale networks using symmetric nonnegative matrix factorization, which first

divides the whole graph into many sub-networks while preserving clustering attributes, then supplements the adjacency matrix by extracting the priori information from well-connected sub-sub-graphs, and finally performs nonnegative matrix factorization on the reinforcement matrix by iterative multiplication. The theoretical analysis and results of comparison tests confirm that our scheme is superior to the other two state-of-the-art methods in effectiveness and efficiency.

The current design of our solution identifies overlapping communities in Ego-splitting networks through matrix analysis, which has several issues that must be tackled in future work. First, the number of communities in sub-networks should be precisely determined through heuristic algorithms, which is a challenging task in network clustering. Another improvement direction is to extend the priori information using other types of smoothing, such as diffusion kernels. Then, two communities should be merged if the corresponding overlap exceeds a well-chosen threshold that has been unsolved in this study. Finally, we intend to further assess the performance of our approach on more large-scale complicated networks with various verification criteria.

Author Contributions: Conceptualization, Q.J. and Q.Q.; methodology, M.H., Q.J. and Q.Q.; software, M.H. and A.R.; validation, Q.J., Q.Q. and A.R.; formal analysis, A.R.; investigation, M.H.; resources, Q.J. and Q.Q.; data curation, M.H.; writing—original draft preparation, M.H.; writing—review and editing, A.R.; visualization, M.H.; supervision, Q.J. and Q.Q.; project administration, Q.J. and Q.Q.; funding acquisition, Q.J., Q.Q. and M.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially sponsored by the Key-Area Research and Development Program of Guangdong Province under Grant No. 2019B010137002, the National Natural Science Foundation of China under Grant No. 61902385, and the China Postdoctoral Science Foundation under Grant No. 2020M672892.

Data Availability Statement: All the experimental data leveraged to construct the undirected and weighted graphs (emerging as symmetric matrices) for performance verification in this study can be publicly available. (**a**) Software package of LF model (for generating synthetic networks) [41]: http://santo.fortunato.googlepages.com/inthepress2 (accessed on 28 April 2021). (**b**) Public datasets archived by Stanford University (for building real-world networks) [42]: http://snap.stanford.edu/data/ (accessed on 28 April 2021).

Acknowledgments: The authors would like to thank all the anonymous reviewers for their insightful comments and constructive suggestions that have obviously upgraded the quality of this manuscript.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal circumstances that could have appeared to influence the work reported in this manuscript.

Abbreviations

The following abbreviations are used in this paper:

ESNMF	An Overlapping Community Detection Approach in Ego-splitting Networks Using Symmetric Nonnegative Matrix Factorization
Con-Com	Connected Component of Ego-Splitting
Con-Cut	Low Conductance Cut with PageRank
LF	Lancichinetti-Fortunato
LF-SW	Small and Weighted Network of LF Model
LF-LW	Large and Weighted Network of LF Model
Email-Eu-Core	Email Network from an Institution
Wikispeedia	Navigation Network on Wikipedia
NMI	Normalized Mutual Information
NaN	Not a Number
NP-hard	Non-deterministic Polynomial Hard

References

1. Bello-Orgaz, G.; Jung, J.J.; Camacho, D. Social big data: Recent achievements and new challenges. *Inf. Fusion* **2016**, *28*, 45–59. [CrossRef] [PubMed]
2. Villar-Rodriguez, E.; Ser, J.D.; Gil-Lopez, S.; Bilbao, M.N.; Salcedo-Sanz, S. A meta-heuristic learning approach for the non-intrusive detection of impersonation attacks in social networks. *Int. J. Bio-Inspired Comput.* **2017**, *10*, 109–118. [CrossRef]
3. Ferrara, E. Contagion dynamics of extremist propaganda in social networks. *Inf. Sci.* **2017**, *418*, 1–12. [CrossRef]
4. Westlake, B.G.; Bouchard, M. Liking and hyperlinking: Community detection in online child sexual exploitation networks. *Soc. Sci. Res.* **2016**, *59*, 23–36. [CrossRef]
5. Martinet, L.-E.; Kramer, M.A.; Viles, W.; Perkins, L.N.; Spencer, E.; Chu, C.J.; Cash, S.S.; Kolaczyk, E.D. Robust dynamic community detection with applications to human brain functional networks. *Nat. Commun.* **2020**, *11*, 1–13.
6. Huang, M.; Zou, G.; Zhang, B.; Gan, Y.; Jiang, S.; Jiang, K. Identifying influential individuals in microblogging networks using graph partitioning. *Expert Syst. Appl.* **2018**, *102*, 70–82. [CrossRef]
7. Javed, S.; Mahmood, A.; Fraz, M.M.; Koohbanani, N.A.; Benes, K.; Tsang, Y.-W.; Hewitt, K.; Epstein, D.; Snead, D.; Rajpoot, N. Cellular community detection for tissue phenotyping in colorectal cancer histology images. *Med. Image Anal.* **2020**, *63*, 101696. [CrossRef]
8. Liu, C.; Guo, C. STCCD: Semantic trajectory clustering based on community detection in networks. *Expert Syst. Appl.* **2020**, *162*, 113689. [CrossRef]
9. Newman, M.E.J.; Cantwell, G.T.; Young, J.G. Improved mutual information measure for clustering, classification, and community detection. *Phys. Rev. E* **2020**, *101*, 042304. [CrossRef]
10. Newman, M.E.J. Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA* **2006**, *103*, 8577–8582. [CrossRef]
11. Ma, X.; Wang, B.; Yu, L. Semi-supervised spectral algorithms for community detection in complex networks based on equivalence of clustering methods. *Phys. A Stat. Mech. Its Appl.* **2018**, *490*, 786–802. [CrossRef]
12. Li, H.-J.; Wang, L.; Zhang, Y.; Perc, M. Optimization of identifiability for efficient community detection. *New J. Phys.* **2020**, *22*, 063035. [CrossRef]
13. Javed, M.A.; Younis, M.S.; Latif, S.; Qadir, J.; Baig, A. Community detection in networks: A multidisciplinary review. *J. Netw. Comput. Appl.* **2018**, *108*, 87–111. [CrossRef]
14. Epasto, A.; Lattanzi, S.; Leme, R.P. Ego-splitting framework: From non-overlapping to overlapping clusters. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 145–154.
15. Li, Y.; Sha, C.; Huang, X.; Zhang, Y. Community detection in attributed graphs: An embedding approach. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 338–345.
16. Osaba, E.; Ser, J.D.; Camacho, D.; Bilbao, M.N.; Yang, X.-S. Community detection in networks using bio-inspired optimization: Latest developments, new results and perspectives with a selection of recent meta-heuristics. *Appl. Soft Comput.* **2020**, *87*, 106010. [CrossRef]
17. Chunaev, P. Community detection in node-attributed social networks: A survey. *Comput. Sci. Rev.* **2020**, *37*, 100286. [CrossRef]
18. Fani, H.; Jiang, E.; Bagheri, E.; Al-Obeidat, F.; Du, W.; Kargar, M. User community detection via embedding of social network structure and temporal content. *Inf. Process. Manag.* **2020**, *57*, 102056. [CrossRef]
19. Ma, T.; Shao, W.; Hao, Y.; Cao, J. Graph classification based on graph set reconstruction and graph kernel feature reduction. *Neurocomputing* **2018**, *296*, 33–45. [CrossRef]
20. Qin, G.; Gao, L. Spectral clustering for detecting protein complexes in protein–protein interaction (PPI) networks. *Math. Comput. Model.* **2010**, *52*, 2066–2074. [CrossRef]
21. Mercorio, F.; Mezzanzanica, M.; Moscato, V.; Picariello, A.; Sperlí, G. DICO: A graph-db framework for community detection on big scholarly data. *IEEE Trans. Emerg. Top. Comput.* **2019**, doi:10.1109/TETC.2019.2952765.
22. Palla, G.; Derényi, I.; Farkas, I.; Vicsek, T. Uncovering the overlapping community structure of complex networks in nature and society. *Nature* **2005**, *435*, 814–818. [CrossRef]
23. Jia, J.; Wang, B.; Cao, X.; Gong, N.Z. Certified robustness of community detection against adversarial structural perturbation via randomized smoothing. In Proceedings of the 2020 International Conference on World Wide Web, Taipei, Taiwan, 20–24 April 2020; pp. 2718–2724.
24. Huang, M.; Zou, G.; Zhang, B.; Liu, Y.; Gu, Y.; Jiang, K. Overlapping community detection in heterogeneous social networks via the user model. *Inf. Sci.* **2018**, *432*, 164–184. [CrossRef]
25. Li, Y.; He, J.; Wu, Y.; Lv, R. Overlapping Community Discovery Method Based on Two Expansions of Seeds. *Symmetry* **2021**, *13*, 18. [CrossRef]
26. Whang, J.J.; Gleich, D.F.; Dhillon, I.S. Overlapping community detection using neighborhood-inflated seed expansion. *IEEE Trans. Knowl. Data Eng.* **2016**, *28*, 1272–1284. [CrossRef]
27. Ma, T.; Liu, Q.; Cao, J.; Tian, Y.; Al-Dhelaan, A.; Al-Rodhaan, M. LGIEM: Global and local node influence based community detection. *Future Gener. Comput. Syst.* **2020**, *105*, 533–546. [CrossRef]
28. Liu, F.; Xue, S.; Wu, J.; Zhou, C.; Hu, W.; Paris, C.; Nepal, S.; Yang, J.; Yu, P.S. Deep learning for community detection: Progress, challenges and opportunities. *arXiv* **2020**, arXiv:2005.08225.

29. Yang, L.; Cao, X.; He, D.; Wang, C.; Wang, X.; Zhang, W. Modularity Based Community Detection with Deep Learning. In Proceedings of the 25th International Joint Conference on Artificial Intelligence, New York, NY, USA, 9–15 July 2016; Volume 16, pp. 2252–2258.
30. Wu, L.; Zhang, Q.; Chen, C.-H.; Guo, K.; Wang, D. Deep learning techniques for community detection in social networks. *IEEE Access* **2020**, *8*, 96016–96026. [CrossRef]
31. Moscato, V.; Picariello, A.; Sperlí, G. Community detection based on game theory. *Eng. Appl. Artif. Intell.* **2019**, *85*, 773–782. [CrossRef]
32. Geng, X.; Lu, H.; Sun, J. Network structural transformation-based community detection with autoencoder. *Symmetry* **2020**, *12*, 944. [CrossRef]
33. Gleich, D.F.; Seshadhri, C. Vertex Neighborhoods, Low Conductance Cuts, and Good Seeds for Local Community Methods. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing, China, 12–16 August 2012; Volume 16, pp. 597–605.
34. Ma, X.; Dong, D.; Wang, Q. Community detection in multi-layer networks using joint non-negative matrix factorization. *IEEE Trans. Knowl. Data Eng.* **2018**, *31*,273–286. [CrossRef]
35. Wang, P.; He, Z.; Lu, J.; Tan, B.; Bai, Y.; Tan, J.; Liu, T.; Lin, Z. An Accelerated Symmetric Nonnegative Matrix Factorization Algorithm Using Extrapolation. *Symmetry* **2020**, *12*, 1187. [CrossRef]
36. Yang, Z.; Hao, T.; Dikmen, O.; Chen, X.; Oja, E. Clustering by non-negative matrix factorization using graph random walk. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–8 December 2012; pp. 1079–1087.
37. Yang, Z.; Oja, E. Quadratic non-negative matrix factorization. *Pattern Recognit.* **2012**, *45*, 1500–1510. [CrossRef]
38. Manukyan, L.; Montandon, S.A.; Fofonjka, A.; Smirnov, S.; Milinkovitch, M.C. A living mesoscopic cellular automaton made of skin scales. *Nature* **2017**, *544*, 173–179. [CrossRef] [PubMed]
39. Epasto, A.; Lattanzi, S.; Mirrokni, V.; Sebe, I.O.; Taei, A.; Verma, S. Ego-net community mining applied to friend suggestion. In Proceedings of the International Conference on Very Large Data Bases, Kohala Coast, HI, USA, 31 August–4 September 2015; Volume 9, No.4, pp. 324–335.
40. McDaid, A. F.; Greene, D.; Hurley, N. Normalized mutual information to evaluate overlapping community finding algorithms. *arXiv* **2013**, arXiv:1110.2515v2.
41. Lancichinetti, A.; Fortunato, S. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys. Rev. E* **2009**, *80*, 016118. [CrossRef] [PubMed]
42. Leskovec, J.; Krevl, A. SNAP Datasets: Stanford Large Network Dataset Collection. 2014. Available online: http://snap.stanford.edu/data (accessed on 28 April 2021).