Received July 28, 2021, accepted August 11, 2021, date of publication August 18, 2021, date of current version September 1, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3105692

Robust Community Detection in Graphs

ESRAA M. AL-SHAROA[1], (Member, IEEE), BARA' M. ABABNEH[1,2], AND MAHMOOD A. ALKHASSAWENEH[2,3], (Senior Member, IEEE)

[1]Electrical Engineering Department, Jordan University of Science and Technology, Irbid 22110, Jordan
[2]Computer Engineering Department, Yarmouk University, Irbid 21163, Jordan
[3]Department of Engineering, Computing and Mathematical Sciences, Lewis University, Romeoville, IL 60446, USA

Corresponding author: Esraa M. Al-Sharoa (emalsharoa@just.edu.jo)

This work was supported in part by Lewis University, and in part by the Jordan University of Science and Technology.

ABSTRACT Community detection in network-type data provides a powerful tool in analyzing and understanding real-world systems. In fact, community detection approaches aim to reduce the network's dimensionality and partition it into a set of disjoint clusters or communities. However, real networks are usually corrupted with noise or outliers which affect the detected community structure quality. In this paper, a new robust community detection algorithm that is capable of recovering a clean or a smoothed version of the corrupted graph and detecting the correct community structure is introduced. The proposed approach combines robust principal component analysis (RPCA) and symmetric nonnegative matrix factorization (SymNMF) in a single optimization problem. The proposed problem is solved under the framework of alternating direction methods of multipliers (ADMM). In particular, the corrupted adjacency matrix is decomposed into a low-rank and sparse components using RPCA and the community structure is detected by applying SymNMF to the extracted low-rank component. Extensive experiments that have been conducted on real and simulated binary and weighted networks show that the proposed approach significantly outperforms existing algorithms in detecting the correct community structure even in grossly corrupted networks.

INDEX TERMS Community detection, graph theory, robust principal component analysis, symmetric nonnegative matrix factorization.

I. INTRODUCTION

In recent years, graph or network theory has become one of the most popular tools in modeling and analyzing relational data. Networks are currently used in many disciplines to describe the relationship between entities, such as biological [1], [2], social [3], [4] and communication networks [5], to name a few. In particular, objects in the system and the interactions between them can be modeled as the nodes and edges of the network, respectively. One of the most popular approaches used in investigating and analyzing networks is community detection [2], [6], [7]. Community detection methods reflect the organization of the nodes into clusters or communities where the nodes in each community share common properties. Moreover, it provides a significant tool in network's dimensionality reduction.

Various community detection or clustering approaches in networks have been proposed over the past few years [8], including partitional algorithms [9], [10], hierarchical clustering [11], and Newman-Girvan algorithm [12], [13]. In partitional algorithms, multiple approaches have been proposed to cluster objects into multiple clusters such as k-means clustering, spectral clustering and symmetric nonnegative matrix factorization (SymNMF) for graph clustering. In k-means [14], a cost function that minimizes the intra-cluster distances and maximizes the inter-clusters distances is adopted. In spectral clustering [9], an efficient solution to the relaxed versions of the different cut problems is provided. In particular, the cut cost depends on the spectral properties of the graph. In SymNMF [10], [15], a symmetric nonnegative lower rank approximation is computed for the input nonnegative similarity matrix. This low-rank approximation provides the nodes clustering assignment of the network.On the other hand, authors in [16] suggested dropping the symmetry for fast SymNMF. This will transfer the SymNMF problem to a nonsymmetric one and then the idea from the state-of-the-art algorithms for nonsymmetric NMF is adopted for the solution. Another approach that is built upon NMF is introduced in [17], where the authors propose to learn the affinity matrix adaptively (A^2NMF). In particular, A^2NMF embeds each node into a low dimentional space through a transformation matrix that preserves the community structure as a first stage. Then, the affinity matrix is learned in this low-dimensional space and utilized

The associate editor coordinating the review of this manuscript and approving it for publication was Hocine Cherifi.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

to guide the learning of the community membership matrix via manifold regularization. Under hierarchical clustering approaches, a hierarchical structure is constructed using a distance matrix. This structure is visualized through a dendrogram. More precisely, hierarchical clustering can be implemented by agglomerative or divisive algorithms [18], [19]. Under modularity optimization methods, in [13], the modularity is defined as the deviation of the within community edges from the expected value of the same quantity in a network with the same community partitions but random connections between the nodes. Specifically, the community structure is found by maximizing the modularity thorough a greedy algorithm namely agglomerative hierarchical clustering. In [20], the authors proposed a different greedy approach to maximize the modularity, known as Louvain modularity, to detect the community structure in large weighted networks. Although the aforementioned methods proved their significance in detecting the community structure in networks, their performance degrades when the networks are corrupted with noise.

More recently, community detection methods using label propagation algorithms have been proposed. In [21], a node's label influence policy for label propagation algorithm (LP-LPA) is proposed. The purpose of the LP-LPA algorithm is to improve the initial node selections and tie-break technique. In particular, it computes link strength value for links and nodes' label influence value for nodes in a new label propagation strategy with preference on link strength and for initial nodes selection and avoid random behavior in tie-break states to efficiently update order and rule update. In [22], a new version of the LPA algorithm for attributed graphs is introduced, namely structure-attribute similarities label propagation (SAS-LP). The purpose of SAS-LP is to detect the communities that solve the problems related to instability, low quality and to possessing structural cohesiveness and attribute homogeneity. Another local approach that depends on detecting and expansion of core nodes is proposed in [23]. This approach used local information and identify the different functions of the nodes to detect all the communities in the network. A detailed review of the various community detection algorithms in networks, problems and challenges can be found in [8], [24] and [25].

In this paper, a robust community detection algorithm in graphs is proposed. The proposed approach presents an improved formulation for graph clustering even when the network is corrupted with noise or outliers. In particular, the proposed approach aims to recover a clean version of the corrupted graph and use it for community detection. The contributions of the proposed algorithm are four fold. First, the proposed approach can detect the community structure in both binary and weighted networks under the same optimization problem. Second, the recovery of the low-rank component gets rid of outliers in the graph which leads to a better community detection results. Third, the proposed approach does not assume the number of clusters or any model for the underlying network unlike many other existing methods. Finally, the network's community structure is detected through nonnegative embedding.

This paper is organized as follows: Section II provides a background description about graph theory, nonnegative matrix factorization and robust principal component analysis. In Section III, the proposed algorithm along with the proposed solution are presented with a detailed solution of the problem in Appendix A- D. Experiments and results are presented in Section IV. Finally, the conclusions are summarized in Section V.

Notation: List of notation used in this paper is summarized in Table 1.

II. BACKGROUND

A. GRAPH THEORY

An undirected weighted graph can be defined as $\mathcal{G} = \{V, E, \mathbf{A}\}$ where $V = \{v_1, \ldots, v_n\}$ defines the set of nodes that models the objects in the network, and E defines the set of edges that models the pairwise similarities between the objects [6]. $|V|$ and $|E|$ are the number of nodes and edges in the network, respectively. The adjacency matrix, $\mathbf{A} \in \mathbb{R}^{n\times n}$, is symmetric and its elements represent the similarities between each pair of nodes. In this paper, $\mathbf{A}$ can be either weighted or binary, where $A^{ij} \in [0, 1]$ in the former and $A^{ij} \in \{0, 1\}$ in the latter.

TABLE 1. List of notation.

Symbol	Description
$\mathbf{A} \in \mathbb{R}^{n\times n}$	Adjacency matrix
$\mathbf{L} \in \mathbb{R}^{n\times n}$	Low-rank component
$\mathbf{S} \in \mathbb{R}^{n\times n}$	Sparse component
$\mathbf{H} \in \mathbb{R}^{n\times n}$	Nonnegative factors matrix
$\mathbf{M} \in \mathbb{R}^{n\times n}$	Auxiliary variable
$\mathbf{Z}_1, \mathbf{Z}_2 \in \mathbb{R}^{n\times n}$	Lagrange multipliers
k	Number of clusters
λ_1, λ_2	Regularization parameters
γ_1, γ_2	Lagrangian penalty parameters
l	iteration index
$\|.\|_F$	Frobenious norm
$\|.\|_1$	l-1 norm
$\|.\|_*$	Nuclear norm
$\top$	Transpose of a matrix
$s.t$	subject to

B. NONNEGATIVE MATRIX FACTORIZATION (NMF)

let $\mathbf{X} = [x_1, x_2, \ldots, x_m] \in \mathbb{R}^{n\times m}$ be a nonnegative data matrix with m samples and $x_i \in \mathbb{R}^n$ is the vector representation of the i-th sample. In order to reduce the dimensionality of the input data matrix, NMF factorizes $\mathbf{X}$ as follows:

$$\min_{\mathbf{U}\in\mathbb{R}^{n\times k}, \mathbf{H}\in\mathbb{R}^{m\times k}} \|\mathbf{X} - \mathbf{U}\mathbf{H}^\top\|_F^2 \quad s.t\ \mathbf{U} \geq 0, \mathbf{H} \geq 0 \tag{1}$$

where $\mathbf{U}$ and $\mathbf{V}$ are the basis matrix and the low-dimensional representation, respectively. $\|.\|_F$ is the Frobenious norm and $\top$ is the transpose operator. Multiple approaches have been proposed to solve the NMF problem [26], [27].

A special variant of this factorization is when the input matrix is symmetric. This variant is known as symmetric nonnegative matrix factorization (SymNMF) and plays an important role in network-type data clustering [10]. The input to SymNMF is the adjacency matrix where $\mathbf{X} = \mathbf{X}^\top = \mathbf{A}$ and its optimization problem is defined as follows:

$$\min_{\mathbf{H}\in\mathbb{R}^{n\times k}} \|\mathbf{A} - \mathbf{H}\mathbf{H}^\top\|_F^2 \quad s.t\ \mathbf{H} \geq 0, \tag{2}$$

where k represents the number of communities or clusters in the network. The solution of Eq. 2, $\mathbf{H} = [h_1, h_2, \ldots, h_k] \in \mathbb{R}^{n\times k}$, can be obtained by a Newton-like algorithm or an alternating nonnegative least squares (ANLS) algorithm as suggested in [10]. The clustering membership of the i-th node can be obtained as the location of the largest value in the i-th row of $\mathbf{H}$.

C. ROBUST PRINCIPAL COMPONENT ANALYSIS

Various methods have been proposed in literature to obtain a low-rank approximation for noisy matrices such as Principal Component Analysis (PCA) [28]–[30]. However, PCA performance in recovering the low-rank component of a noisy matrix decays when the noise is non-Gaussian. An alternative approach that is proposed to overcome this problem is the Robust Principal Component Analysis (RPCA) [31]. In RPCA, the noisy matrix, $\mathbf{A} \in \mathbb{R}^{n_1\times n_2}$, is decomposed into a low-rank component, $\mathbf{L} \in \mathbb{R}^{n_1\times n_2}$, and a sparse component, $\mathbf{S} \in \mathbb{R}^{n_1\times n_2}$. In particular, RPCA solves the following optimization problem:

$$\min_{\mathbf{L},\mathbf{S}} \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1 \quad s.t\ \mathbf{L} + \mathbf{S} = \mathbf{A}, \tag{3}$$

where $\|.\|_*$ and $\|.\|_1$ are the nuclear norm and the l_1-norm of a matrix, respectively. $\lambda > 0$ is the regularization parameter that penalizes the sparse term. The problem in Eq. 3 can be rewritten as an unconstrained problem as follows:

$$\min_{\mathbf{L},\mathbf{S}} \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1 + \frac{\gamma}{2}\|\mathbf{A} - \mathbf{L} - \mathbf{S}\|_F^2, \tag{4}$$

where the solution of the problem can be found using iterative singular value decomposition (SVD) soft-thresholding algorithm efficiently [31].

III. ROBUST COMMUNITY DETECTION IN GRAPHS (RCDG)

A. PROBLEM FORMULATION

Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be a noise-corrupted adjacency matrix of a low-rank network, $\mathbf{L} \in \mathbb{R}^{n\times n}$. In order to detect the true community structure of the underlying network, it is important to extract a clean version of the aforementioned corrupted adjacency matrix. In this paper, we decompose the adjacency matrix into a low-rank and sparse components, $\mathbf{L} \in \mathbb{R}^{n\times n}$ and $\mathbf{S} \in \mathbb{R}^{n\times n}$, respectively. Moreover, the recovered low-rank component, $\mathbf{L}$, is used to detect the network's community structure through learning nonnegative embedding. Our proposed problem is formulated as follows:

$$\begin{aligned}&\min_{\mathbf{L},\mathbf{S},\mathbf{H}} \|\mathbf{L}\|_* + \lambda_1\|\mathbf{S}\|_1 + \lambda_2\|\mathbf{L} - \mathbf{H}\mathbf{H}^\top\|_F^2\\ &s.t\ \mathbf{L} + \mathbf{S} = \mathbf{A}, \mathbf{L} = \mathbf{L}^\top, \quad \mathbf{L} \geq 0, \mathbf{H} \geq 0.\end{aligned} \tag{5}$$

The terms included in the optimization problem in Eq. 5 are considered to achieve the following objectives:

- The terms $\|\mathbf{L}\|_*$ and $\|\mathbf{S}\|$ with the constraint, $\mathbf{L}+\mathbf{S} = \mathbf{A}$, represent the RPCA problem. The additional constraints, $\mathbf{L} = \mathbf{L}^\top$ and $\mathbf{L} \geq 0$ are considered to guarantee the symmetry and nonnegativity of the recovered low-rank network.
- The term $\|\mathbf{L}-\mathbf{H}\mathbf{H}^\top\|_F^2$ with the constraint $\mathbf{H} \geq 0$ defines the SymNMF problem for graph clustering.
- $\lambda_1 > 0$ and $\lambda_2 > 0$ are regularization parameters.

B. PROBLEM SOLUTION

To solve the proposed problem in Eq. 5, an alternating iterative method is proposed. The solution starts with introducing an auxiliary variable $\mathbf{M} \in \mathbb{R}^{n\times n}$ in order to separate the variable $\mathbf{H}$ from $\mathbf{L}$ as follows:

$$\begin{aligned}&\min_{\mathbf{L},\mathbf{S},\mathbf{H},\mathbf{M}} \|\mathbf{L}\|_* + \lambda_1\|\mathbf{S}\| + \lambda_2\|\mathbf{M} - \mathbf{H}\mathbf{H}^\top\|_F^2\\ &s.t\ \mathbf{L} + \mathbf{S} = \mathbf{A}, \mathbf{M} = \mathbf{L}, \mathbf{M} = \mathbf{M}^\top, \quad \mathbf{M} \geq 0, \mathbf{H} \geq 0.\end{aligned} \tag{6}$$

The problem proposed in Eq. 4 is a nonconvex problem since the included SymNMF formulation term is a nonconvex optimization problem. To tackle this issue, an iteratively alternating approach is adopted where each variable is obtained by fixing the other variables until convergence. More precisely, assuming that $\mathbf{H}$ is estimated, the rest of the problem consists of three convex but non-smooth functions. Consequently, we propose to solve for $\mathbf{L}$, $\mathbf{S}$ and $\mathbf{M}$ using a combination of alternating direction methods of multipliers (ADMM) [32], [33] and proximal algorithms [34], [35]. In a similar fashion, in each iteration, after $\mathbf{L}$, $\mathbf{S}$ and $\mathbf{M}$ are obtained, the estimate of the variable $\mathbf{H}$ can be updated using SymNMF as proposed in [10].

The augmented Lagrange multiplier function with respect to the primal variables $\mathbf{L}$, $\mathbf{S}$ and $\mathbf{M}$ of Eq. 6 can be formulated by adding the Lagrange multipliers or dual variables, $\mathbf{Z}_1$ and $\mathbf{Z}_2$, as:

$$\begin{aligned}\mathcal{L}(\mathbf{L},\mathbf{S},\mathbf{W},\mathbf{H}) = &\min_{\mathbf{L},\mathbf{S},\mathbf{W},\mathbf{H}} \|\mathbf{L}\|_* + \lambda_1\|\mathbf{S}\|_1\\ &+\lambda_2\|\mathbf{M} - \mathbf{H}\mathbf{H}^\top\|_F^2 + \langle \mathbf{Z}_1^l, \mathbf{A} - \mathbf{L} - \mathbf{S}\rangle\\ &+\frac{\gamma_1}{2}\|\mathbf{A} - \mathbf{L} - \mathbf{S}\|_F^2 + \langle \mathbf{Z}_2^l, \mathbf{M} - \mathbf{L}\rangle\\ &+\frac{\gamma_2}{2}\|\mathbf{M} - \mathbf{L}\|_F^2,\\ &s.t\ \mathbf{M} = \mathbf{M}^\top, \quad \mathbf{M} \geq 0,\end{aligned} \tag{7}$$

where $\mathbf{Z}_1$ and $\mathbf{Z}_2$ can be computed at the l-th iteration as:

$$\mathbf{Z}_1^{l+1} = \mathbf{Z}_1^l + \gamma_1(\mathbf{A} - \mathbf{L}^{l+1} - \mathbf{S}^{l+1}), \tag{8}$$

and

$$\mathbf{Z}_2^{l+1} = \mathbf{Z}_2^l + \gamma_2(\mathbf{M}^{l+1} - \mathbf{L}^{l+1}). \tag{9}$$

ADMM along with proximal algorithms are adopted to solve for the primal and dual variables in Eq. 7. The proposed solution is summarized in Algorithm 1 and a detailed explanation can be found in Appendix A-D. The update rules of the primal variables can be defined as follows:

$$\mathbf{L}^{l+1} = prox_{\frac{1}{\gamma_1+\gamma_2}\|\mathbf{L}\|_*}\left(\frac{\gamma_1\mathbf{W}_1^l + \gamma_2\mathbf{W}_2^l}{\gamma_1+\gamma_2}\right), \tag{10}$$

where $\mathbf{W}_1^l = \mathbf{A} - \mathbf{S}^l + \frac{\mathbf{Z}_1^l}{\gamma_1}$, $\mathbf{W}_2^l = \mathbf{M}^k + \frac{\mathbf{Z}_2^l}{\gamma_2}$ and $prox_f(\mathbf{W}) = \text{argmin}_{\mathbf{L}\in\mathbb{R}^{n\times n}} f(\mathbf{L}) + \frac{1}{2} \parallel \mathbf{L} - \mathbf{W} \parallel_F^2$ is the proximity operator of the convex function f where $f(\mathbf{L})$ is defined as $f(\mathbf{L}) = \|\mathbf{L}\|_*$. Details are included in Appendix A.

Also,

$$\mathbf{S}^{l+1} = prox_{\frac{\lambda_1}{\gamma_1}\|\mathbf{S}^{(t)}\|_1}\left(\mathbf{A} - \mathbf{L}^{l+1} + \frac{\mathbf{Z}_1^l}{\gamma_1}\right). \tag{11}$$

as explained in Appendix B.

Next, $\mathbf{M}$ is updated, as explained in Appendix C, by computing the following closed form solution:

$$\mathbf{M}^{l+1} = \frac{2\lambda_2\mathbf{H}^l{\mathbf{H}^l}^\top + \gamma_2\mathbf{L}^l - \mathbf{Z}_2^l}{2\lambda_2 + \gamma_2} \tag{12}$$

Finally, the nonnegative factor $\mathbf{H}$ is computed using SymNMF [10] approach. In particular, SymNMF updates $\mathbf{H}$ by solving:

$$\min_{\mathbf{H}\in\mathbb{R}^{n\times k}} \|\mathbf{M}^{l+1} - \mathbf{H}\mathbf{H}^\top\|_F^2 \qquad s.t\ \mathbf{H} \geq 0, \tag{13}$$

where the solution suggested in [10] is based on a multistart global optimization algorithm which combines random sampling with a local search procedure.

C. COMPUTATIONAL COMPLEXITY OF RCDG APPROACH

In the proposed approach, the variables **L**, **S**, and **M** in the optimization problem are solved by ADMM and proximal algorithms assuming **H** is fixed until convergence. In each iteration, the computation of the nuclear norm proximal has a complexity of $\mathcal{O}(2n^3)$ and the computation of **H** using projected gradient descent (PGD) by alternating nonnegative least square (ANLS) requires $\mathcal{O}(kn^2)$, where n and k are the number of nodes and number of clusters in the network, respectively. Moreover, the number of clusters is determined during each iteration using AS metric by computing it over a range of clusters, K. The computational complexity of AS is $K\mathcal{O}(e)$ where e is the number of edges in the adjacency matrix. By Considering the computational complexity for the different terms during each iteration, the effective computational cost is due to the computation of the nuclear proximal operator. Consequently, assuming the total number of iterations needed for the algorithm to converge is l_f, the computational complexity of the proposed algorithm is $l_f\mathcal{O}(n^3)$.

Algorithm 1 RCDG

Input: $\mathbf{A} \in \mathbb{R}^{n\times n}$, λ_1, λ_2, ϵ.
Output: **L**, **S**, **H**, Clustering labels.

```
1:  $l \leftarrow 0$
2:  Initialize $\mathbf{L} \leftarrow \mathbf{A}$, $\mathbf{S} \leftarrow zeros(n, n)$, $\mathbf{M} \leftarrow \mathbf{L}$.
3:  $\gamma_1 \leftarrow 1$, $\gamma_2 \leftarrow 1$.
4:  $\mathbf{Z}_1^l \leftarrow \mathbf{A} - \mathbf{L}^l - \mathbf{S}^l$, $\mathbf{Z}_2^l \leftarrow \mathbf{M}^l - \mathbf{L}^l$.
    %Dual variables definition
5:  $\mathbf{P}_1^k = \|\mathbf{L}^l\|_*$, $\mathbf{P}_2^l = \lambda_1\|\mathbf{S}^l\|_1$, $\mathbf{P}_3^l = \lambda_2\|\mathbf{M}^l - \mathbf{H}\mathbf{H}^\top\|_F^2$.
    %Primal objectives definition
6:  while $\frac{\|\mathbf{P}_1^{l+1} - \mathbf{P}_1^l\|_F^2}{\|\mathbf{P}_1^l\|_F^2} > \epsilon$ and $\frac{\|\mathbf{P}_2^{l+1} - \mathbf{P}_2^l\|_F^2}{\|\mathbf{P}_2^l\|_F^2} > \epsilon$ and
    $\frac{\|\mathbf{P}_3^{l+1} - \mathbf{P}_3^l\|_F^2}{\|\mathbf{P}_3^l\|_F^2} > \epsilon$ do
7:      Estimate $\mathbf{H}^{k+1}$ by solving Eq. 13.
8:      if $l = 2$ then
9:          Determine the number of clusters using asymptotical surprise (AS) metric [36].
10:     end if
11:     Update $\mathbf{L}^{k+1}$ using Eq. (17).
        %Updating primal and dual variables
12:     Update $\mathbf{S}^{k+1}$ using Eq. (19).
13:     Update $\mathbf{M}^{k+1}$ using Eq. (22).
14:     Update $\mathbf{Z}_1^{l+1}$ using Eq. (8).
15:     Update $\mathbf{Z}_2^{l+1}$ using Eq. (9).
16:     Update $\mathbf{P}_1^{k+1}$, $\mathbf{P}_2^{k+1}$ and $\mathbf{P}_3^{k+1}$ using Step 5.
17:     $l \leftarrow l + 1$.
18: end while
19: Obtain clustering labels.
```

IV. RESULTS

A set of undirected weighted and binary simulated networks are generated to evaluate the performance and robustness of the proposed approach in extracting a clean version of the adjacency matrix and detecting the correct community structure in graphs. The experiments are performed using MATLAB R2020b on a desktop with the specifications (Intel(R) Core(TM) i7-9700 CPU @ 3.00GHz 3.00 GHz and RAM of 16GB). The proposed method is compared to other existing methods including, spectral clustering (SC) [9], modularity-Louvain[1] [20], symmetric nonnegative matrix factorization using alternating nonnegative least square[2] (SymNMF-ANLS) [10], [15], symmetric nonnegative matrix factorization using Newton-like algorithm[3] (SymNMF-Newton) [10], [15], fast symmetric nonnegative matrix factorization using hierarchical alternating least square[4] (SymNMF-HALS) [16] and adaptive affinity learning nonnegative matrix factorization[5] (A^2NMF) [17]. The number of the clusters is estimated for each network as the number that maximizes the asymptotical surprise[6] (AS) metric [36] for all the algorithms except for modularity-Louvain since the number of clusters is estimated by the algorithm

[1] https://sites.google.com/site/bctnet/
[2] https://github.com/hiroyuki-kasai/NMFLibrary
[3] https://github.com/hiroyuki-kasai/NMFLibrary
[4] https://github.com/hiroyuki-kasai/NMFLibrary
[5] https://github.com/smartyfh/AANMF
[6] https://github.com/CarloNicolini/communityalg

itself. The comparison is conducted using multiple accuracy measures including normalized mutual information (NMI), normalized variation of information (VI), F-value, precision, recall, purity and the detected number of clusters (DNOC) where all the measures are averaged over 50 simulations. In addition to these validation measures, an *Error Rate* (ER) measure is adopted form [37] to evaluate the performance of the different algorithms. The error rate is defined as:

$$Error\ Rate = \|C_L C_L^T - G_t G_t^T\|_F^2, \tag{14}$$

where $G_t \in \mathbb{R}^{n \times k}$ is the indicator matrix that is built for the network's community ground truth and $C_L \in \mathbb{R}^{n \times k}$ is the indicator matrix that is storing the clustering labels obtained by the algorithm. error rate are employed. In other words, the error rate quantifies the distance between the community structures represented by C_L and G_t. A better clustering results should achieve a higher NMI, F-value, precision, recall and purity values and a lower VI and ER values. The values of all metrics values are normalized between [0, 1] except for ER.

A. SIMULATED WEIGHTED NETWORKS

1) EXPERIMENT 1: SIMULATED NETWORKS WITH DIFFERENT COMMUNITY STRUCTURES

In this experiment, the generated simulated networks consist of 100 nodes and sparse noise $SN = 20\%$, with different community structures. The variation in the community structure is achieved by changing the number and the sizes of the clusters. The number of the clusters is estimated for each network as the number that maximizes the asymptotical surprise (AS) metric [36]. The intra- and inter-cluster edges are randomly generated from a truncated Gaussian distribution in the range of [0, 1] with $\mu_{intra} = 0.5$, $\sigma_{intra} = 0.2$, $\mu_{inter} = 0.1$, $\sigma_{inter} = 0.1$. The parameters are selected empirically as $\lambda_1 = 0.3$ and $\lambda_2 = 1$. The ground truth and the networks' statistics used in this experiment are presented in Table 2 and Table 3, respectively.

TABLE 2. The ground truth of the generated networks for experiment 1.

Network	Ground truth (Nodes in each cluster C)
Network 1	$C_1(1-60)$, $C_2(61-80)$, $C_3(81-100)$
Network 2	$C_1(1-30)$, $C_2(31-60)$, $C_3(61-80)$, $C_4(81-100)$
Network 3	$C_1(1-30)$, $C_2(31-45)$,$C_3(46-60)$, $C_4(61-80)$,$C_5(81-100)$
Network 4	$C_1(1-15)$, $C_2(16-30)$, $C_3(31-45)$, $C_4(46-60)$, $C_5(61-80)$,$C_6(81-100)$
Network 5	$C_1(1-15)$, $C_2(16-30)$, $C_3(31-45)$, $C_4(46-60)$, $C_5(61-70)$, $C_6(71-80)$, $C_17(81-90)$, $C_8(91-100)$

A comparison of the performance between the proposed algorithm and existing algorithms is presented in Table 4. The comparison is conducted by normalized mutual information (NMI), variation of information (VI), *Error Rate* (ER), F-value, precision, recall, purity and detected number of clusters (DNOC) where all the measures are averaged over 50 simulations.

As it can be seen from Table 4, the proposed algorithm performs better than the other methods in terms of the different validation measures. In this experiment, the effect of increasing the number of clusters while decreasing their size is studied. Usually, it is hard to detect small communities in networks and this is one of the challenges that faces community detection algorithms. From the results shown in Table 4, it is clear that RCDG can detect the correct community structure of the underlying network efficiently. Moreover, RCDG is capable of detecting small communities even in noisy networks, in addition to its ability of detecting the correct number of clusters. On the other hand, the performance of the other existing algorithms decays rapidly as the number of clusters increases and their sizes decrease. As it can be noticed from the results, many of these algorithms succeed in detecting the correct number of clusters, however, they fail in detecting the correct clustering labels.

2) EXPERIMENT 2: SIMULATED NETWORKS WITH DIFFERENT NOISE LEVELS

The purpose of this experiment is to evaluate the performance and robustness of the proposed algorithm in networks that is affected with different noise levels. In this experiment, the simulated networks consist of 100 nodes, 4 clusters and different noise levels $SN\%$. The ground truth of the nodes' community membership is $C_1(1-30)$, $C_2(31-60)$, $C_3(61-80)$, $C_4(81-100)$ and the networks' statistics are given in Table 3. The number of the clusters is estimated for each network by asymptotic surprise. Intra- and inter-cluster edges are randomly generated from a truncated Gaussian distribution in the range of [0, 1] with $\mu_{intra} = 0.5$, $\sigma_{intra} = 0.1$, $\mu_{inter} = 0.1$, $\sigma_{inter} = 0.1$. The parameters are selected as $\lambda_1 = 0.3$ and $\lambda_2 = 1$.

A comparison of the performance between RCDG and other algorithms is presented in Table 5. The comparison is done using normalized NMI, VI, ER, F-value, precision, recall, purity and DNOC where all the measures are averaged over 50 simulations.

As it can be seen in Table 5, The noise levels are set to $SN\% = \{5\%, 10\%, 15\%, 20\%, 30\%\}$. The results in the table show that RCDG outperforms the other methods in terms of the different measures. From this experiment, we can notice that the other algorithms perform well as long as the noise levels are low. However, as the noise level increases and the network become grossly corrupted, their performance decays significantly. On the other hand, the proposed RCDG shows robustness to higher levels of sparse noise where it can detect the correct community structure of the network and the correct number of clusters too. For instance, with $SN\% = 30\%$, all the other algorithms failed to detect the correct community structure while RCDG performed very well and achieved very high accuracy in term of the different measures.

TABLE 3. Networks 1-10 statistics averaged over 50 simulations, including: Number of nodes ($|V|$), Number of edges ($|E|$), average degree (D_{avg}), node betweenness centrality (NBC), edge betweenness centrality (EBC), density, clustering coefficient ($\mathfrak{C}$) and assortativity coefficient (r).

Network	Type	$\|V\|$	$\|E\|$	D_{avg}	NBC	EBC	Density	$\mathfrak{C}$	r
Network1	Weighted	100	4950	37.6308	65.3590	1.6436	1	0.2940	−0.0101
Network2	Weighted	100	4950	33.6967	72.7680	1.7177	1	0.2435	−0.0101
Network3	Weighted	100	4950	32.6114	74.5500	1.7355	1	0.2318	−0.0101
Network4	Weighted	100	4950	31.8496	75.9600	1.7496	1	0.2226	−0.0101
Network5	Weighted	100	4950	30.9284	77.5140	1.7651	1	0.2128	−0.0101
Network6	Weighted	100	4950	25.1605	83.4070	1.8241	1	0.1923	−0.0101
Network7	Weighted	100	4950	28.1740	77.2800	1.7628	1	0.2104	−0.0101
Network8	Weighted	100	4950	30.9970	74.5070	1.7351	1	0.2289	−0.0101
Network9	Weighted	100	4950	33.5975	72.6030	1.71603	1	0.2467	−0.0101
Network10	Weighted	100	4950	38.7608	67.3520	1.6635	1	0.2852	−0.0101

TABLE 4. Performance comparison between the proposed method (RCDG), spectral clustering, modularity, SymNMF-ANLS, SymNMF-Newton, SymNMF-HALS and AANMF in terms of average NMI, VI, ER, F-value, Precision, Recall, Purity and DNOC. Networks are constructed with 100 nodes, *SN* = 20% and variable number of clusters (NOC).

Network	Method	NMI	VI	ER	F-value	Precision	Recall	Purity	DNOC
Network1 3 clusters	Spectral clustering	0.87591	0.05202	385.15	0.87591	0.86725	0.88519	0.9685	**3**
	Modularity	0.58121	0.15548	1915.1	0.58121	0.68775	0.50517	0.7455	2
	SymNMF-ANLS	0.78646	0.09000	659.2	0.78646	0.77426	0.79968	0.936	**3**
	SymNMF-Newton	0.70271	0.12073	1349	0.70271	0.73872	0.68282	0.853	**3**
	SymNMF-HALS	0.75416	0.09199	961.3	0.75416	0.83717	0.69621	0.846	2
	A^2NMF	0.61654	0.14251	1935.25	0.58465	0.69149	0.51212	0.7815	4
	RCDG	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**3**
Network2 4 clusters	Spectral clustering	0.81692	0.10814	648.7	0.81692	0.82117	0.81367	0.9175	**4**
	Modularity	0.5524	0.2729	1719.8	0.5524	0.5575	0.5499	0.7380	**4**
	SymNMF-ANLS	0.75302	0.14883	892.15	0.75033	0.749964	0.75218	0.887	**4**
	SymNMF-Newton	0.67657	0.19668	1362.3	0.66426	0.66189	0.67142	0.843	**4**
	SymNMF-HALS	0.60325	0.19752	1912.9	0.60325	0.71309	0.53308	0.7195	3
	A^2NMF	0.44107	0.27389	2926.35	0.41854	0.5478	0.34949	0.612	**4**
	RCDG	**1**	**0**	**60**	**1**	**1**	**1**	**1**	**4**
Network3 5 clusters	Spectral clustering	0.6244	0.25563	1296.85	0.6244	0.63508	0.61652	0.78	**5**
	Modularity	0.3327	0.4576	2316.1	0.3318	0.3475	0.3200	0.5540	**5**
	SymNMF-ANLS	0.56131	0.30381	1501.7	0.56131	0.56087	0.56455	0.739	**5**
	SymNMF-Newton	0.47906	0.37907	1806.9	0.45376	0.44971	0.45965	0.687	6
	SymNMF-HALS	0.37119	0.32477	3236.9	0.37119	0.55431	0.2848	0.527	2
	A^2NMF	0.28361	0.368	3885.1	0.2742	0.42777	0.20608	0.458	3
	RCDG	**1**	**0**	**20**	**1**	**1**	**1**	**1**	**5**
Network4 6 clusters	Spectral clustering	0.39346	0.49187	1840	0.38069	0.37184	0.39021	0.5755	7
	Modularity	0.1981	0.5985	2551.4	0.1981	0.2070	0.1907	0.4030	**6**
	SymNMF-ANLS	0.37168	0.50344	1869.05	0.35978	0.35346	0.36697	0.568	74
	SymNMF-Newton	0.33326	0.54449	1920.35	0.3173	0.31098	0.32445	0.529	7
	SymNMF-HALS	0.10968	0.49727	4379.2	0.10968	0.17759	0.081366	0.295	2
	A^2NMF	0.20974	0.46667	3923.6	0.20974	0.31292	0.16174	0.354	4
	RCDG	**0.99896**	**0.00081**	**45.4**	**0.99896**	**0.99897**	**0.99895**	**0.9995**	7
Network5 8 clusters	Spectral clustering	0.31038	0.63487	1708.8	0.30104	0.29669	0.30571	0.417	9
	Modularity	0.18511	0.66402	2420.7	0.18511	0.20554	0.16858	0.31300	6
	SymNMF-ANLS	0.29582	0.63419	1816.55	0.29161	0.29149	0.29209	0.4065	**8**
	SymNMF-Newton	0.29639	0.63745	1798.15	0.29233	0.2912	0.29414	0.4055	9
	SymNMF-HALS	0.06765	0.57974	4511.3	0.06765	0.12104	0.048295	0.216	2
	A^2NMF	0.16241	0.54994	4199.9	0.1612	0.25653	0.1218	0.274	5
	RCDG	**0.95647**	**0.038779**	**157.6**	**0.95275**	**0.9567**	**0.94923**	**0.9655**	**8**

3) EXPERIMENT 3: SIMULATED NETWORKS WITH DIFFERENT SIZES

In this experiment, the simulated networks consist of 4 clusters, $SN = 20\%$ and variable size or number of nodes from 32 to 2048 on a logarithmic scale. In particular, the objective of this experiment is to test the scalability of the algorithm. The networks' Intra- and inter-cluster edges are randomly generated from a truncated Gaussian distribution in the range of [0, 1] with: $\mu_{intra} = 0.7$, $\sigma_{intra} = 0.2$, $\mu_{inter} = 0.2$, $\sigma_{inter} = 0.2$. A comparison of the run time between the proposed RCDG algorithm and the other algorithms is presented in Fig. 1.

As it can be seen from Fig. 1, the proposed algorithm takes longer time compared to the other methods except for

TABLE 5. Performance comparison between the proposed method (RCDG), spectral clustering, modularity, SymNMF-ANLS, SymNMF-Newton, SymNMF-HALS and AANMF in terms of average NMI, VI, ER, F-value, Precision, Recall, Purity and DNOC. Networks are constructed with 100 nodes, 4 clusters and variable sparse noise levels *SN*%.

Sparse Noise	Method	NMI	VI	ER	F-value	Precision	Recall	Purity	DNOC
Network6 5%	Spectral clustering	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**4**
	Modularity	0.98871	0.006	80	0.98871	**1**	0.97971	0.98	**4**
	SymNMF-ANLS	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**4**
	SymNMF-Newton	0.99854	0.00087	3.9	0.99854	0.99855	0.99853	0.9995	**4**
	SymNMF-HALS	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**4**
	A^2NMF	0.92422	0.04478	872.45	0.86684	0.90907	0.83459	0.962	6
	RCDG	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**4**
Network7 10%	Spectral clustering	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**4**
	Modularity	0.99646	0.00226	**0**	**1**	**1**	**1**	0.9925	**4**
	SymNMF-ANLS	0.99846	0.00091	4.9	0.99846	0.9984	0.99853	0.9995	**4**
	SymNMF-Newton	0.9737	0.016073	123.65	0.97724	0.97437	0.98053	0.985	**4**
	SymNMF-HALS	0.99646	0.00226	**0**	**1**	**1**	**1**	0.9925	**4**
	A^2NMF	0.83617	0.09558	836.1	0.83802	0.85203	0.82993	0.887	**4**
	RCDG	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**4**
Network8 15%	Spectral clustering	0.98618	0.0082109	44.1	0.98618	0.98575	0.98661	0.9955	**4**
	Modularity	0.94582	0.036199	167.05	0.94868	0.94837	0.94905	0.9625	**4**
	SymNMF-ANLS	0.97455	0.015114	81	0.97455	0.9739	0.9752	0.9915	**4**
	SymNMF-Newton	0.89959	0.061442	456.7	0.90069	0.89498	0.90875	0.951	**4**
	SymNMF-HALS	0.9584	0.023866	161.9	0.96182	0.971624	0.95409	0.9645	**4**
	A^2NMF	0.66594	0.18807	1726.65	0.66309	0.71683	0.62824	0.766	**4**
	RCDG	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**4**
Network9 20%	Spectral clustering	0.87023	0.07770	436	0.870234	0.86722	0.8736	0.9525	**4**
	Modularity	0.71328	0.17497	1134.25	0.71421	0.73603	0.6975	0.811	**4**
	SymNMF-ANLS	0.80515	0.11669	685.15	0.80515	0.80121	0.80945	0.92	**4**
	SymNMF-Newton	0.71269	0.17068	1217.4	0.70438	0.72141	0.69225	0.8405	**4**
	SymNMF-HALS	0.63764	0.19201	1663.6	0.63992	0.71022	0.58984	0.753	3
	A^2NMF	0.35436	0.28864	3799.4	0.35553	0.51283	0.29494	0.5425	3
	RCDG	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**4**
Network10 30%	Spectral clustering	0.23581	0.51063	2783.4	0.21067	0.19823	0.22489	0.558	6
	Modularity	0.08257	0.60085	3131.25	0.0755	0.07169	0.07989	0.409	5
	SymNMF-ANLS	0.21993	0.51821	2722.7	0.20858	0.19625	0.22275	0.553	6
	SymNMF-Newton	0.2069	0.52693	2826.05	0.19657	0.18613	0.20852	0.533	6
	SymNMF-HALS	0.10834	0.45516	4055.85	0.1074	0.15265	0.085017	0.411	3
	A^2NMF	0.0187	0.36273	6255.55	0.01718	0.08691	0.01228	0.317	3
	RCDG	**0.96989**	**0.01369**	**160**	**0.96989**	**0.98024**	**0.96468**	**0.9795**	**4**

the A^2NMF algorithm, especially as the size of the network increases. However, the other methods fall behind the proposed algorithm in extracting a clean version of the adjacency matrix and detecting the correct community structure. In fact, it can be said that the performance of the proposed algorithm is a trade-off between complexity and accuracy.

B. PERFORMANCE SENSITIVITY TO THE REGULARIZATION PARAMETERS

In the proposed algorithm, there are two regularization parameters; λ_1 and λ_2. λ_1 controls the l_1-norm of the sparse component while λ_2 controls the symmetric nonnegative matrix factorization term. To study the effect of these parameters, the performance of the proposed algorithm in terms of different quality metrics is explored for a range of the parameters. In particular, the effect of the variation of each parameter on the clustering results is investigated by fixing the other one. Fig. 2(a)- Fig.2(n) show the variation of the regularization parameters impact on the performance of RCDG in terms of average NMI, VI, ER, Recall, Precision, F-value and Purity. As it can be seen from the figures, RCDG performs well under a variety of parameter values. For instance, RCDG performs well under the different values of λ_2, e.g. $\lambda_2 \in [0.3, 1]$. In terms of λ_1, the proposed method performs efficiently for different values of λ_1. In RPCA problem, λ_1 can be selected as $\lambda_1 = \frac{1}{\sqrt{n}}$ as suggested in [31] and this value can be modified depending on the application. In the proposed method, RCDG, we suggest a value of $\lambda_1 = \frac{b}{\sqrt{n}}$ where $b \propto \frac{1}{\|\mathbf{A}\|_1}$. In other words, the parameter λ_1 can be set depending on the network's sparsity and its value decreases as the sparsity increases.

C. SIMULATED BINARY NETWORKS

In order to evaluate the performance of the proposed RCDG algorithm in detecting the community structure in binary networks, two network benchmarks are adopted. First, the classical Girvan-Newman benchmark[7] introduced in [13], where the network is divided into k equal sized clusters and each

[7] https://github.com/mmitalidis/ComDetTB

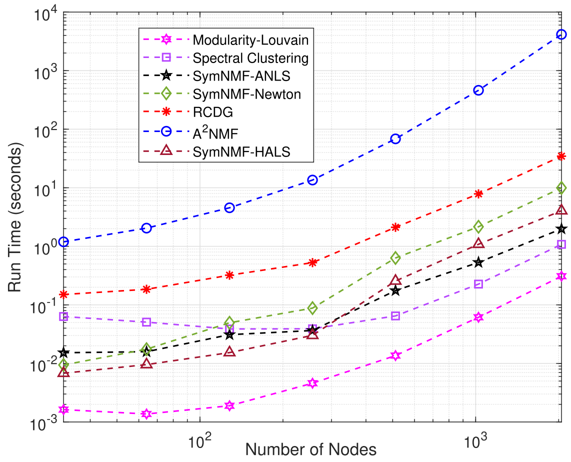


FIGURE 1. **Network size versus run time comparison between proposed method (RCDG), spectral clustering (SC), modularity-Louvain, SymNMF-ANLS, SymNMF-Newton, SymNMF-HALS and A^2NMF. Networks are constructed with 4 clusters and variable number of nodes.**

node has a fixed number of internal and external edges. As the number of internal edges, z_i, increases the density of the network increases and as the number of the external edges, z_e, decreases the clusters become more distinct. Second, the planted partitions benchmark[7] introduced in [38]. In the latter benchmark, the networks are created with unequal-sized partitions or clusters with internal edge probability, p_i and external edge probability, p_e. As p_i increases the clusters become more dense and as p_e decreases the clusters become more distinct.

Multiple networks with different specifications are generated to evaluate the performance of RCDG. As it can be seen in Table 8, six different experiments are conducted by generating six different networks. The first three networks are form Girvan-Newman benchmark (GNB), where the clusters in each of these networks are equally sized and the specifications are given in the table. The last three networks are generated from the planted partitions benchmark (PPB) where the clusters in each network are unequally sized. For each network, the ground truth clusters the networks' statistics are reported in Table 6 and Table 7, respectively.

The performance of RCDG is compared to the other existing algorithms using NMI, VI, ER, F-value, precision, recall, purity and DNOC. Each experiment is repeated 50 times and the average values of the validation measures are reported in Table 8. As it can be noticed from the table, the existing algorithms perform well in detecting the community structure and the number of clusters in the network when the clusters sizes are equal and the network size is relatively small, such as in GNB1-GNB2. However, their performance declines the network's size grows or when the clusters are unequally-sized, such as in PPB1-PPB3. On the contrary, the proposed RCDG proved that it is not affected by these factors and can detect the correct community structure in small and large networks with either equal or unequal-sized clusters. This is due to the fact that the proposed RCDG extracts a clean or a smoothed version of the corrupted adjacency matrix within the algorithm and uses it to detect the community structure of the underlying network.

D. RECOVERING A CLEAN VERSION OF THE ADJACENCY MATRIX

In this paper, we are proposing a robust community detection algorithm that decomposes the corrupted graph adjacency matrix, $\mathbf{A}$, into low-rank, $\mathbf{L}$, and sparse, $\mathbf{S}$, components. The low-rank property of the adjacency matrix is anticipated to strengthen the intra-clusters edges and diminish the inter-cluster edges [39], [40], [41]. Ideally, the rank of the adjacency matrix is equal to the number of communities in the network and the extracted low-rank component is considered as a clean or smoothed version of the input adjacency matrix. Furthermore, we solve for the matrix $\mathbf{M}$ which is constrained to equal $\mathbf{L}$ with additional constraints to satisfy the properties of the adjacency matrix.

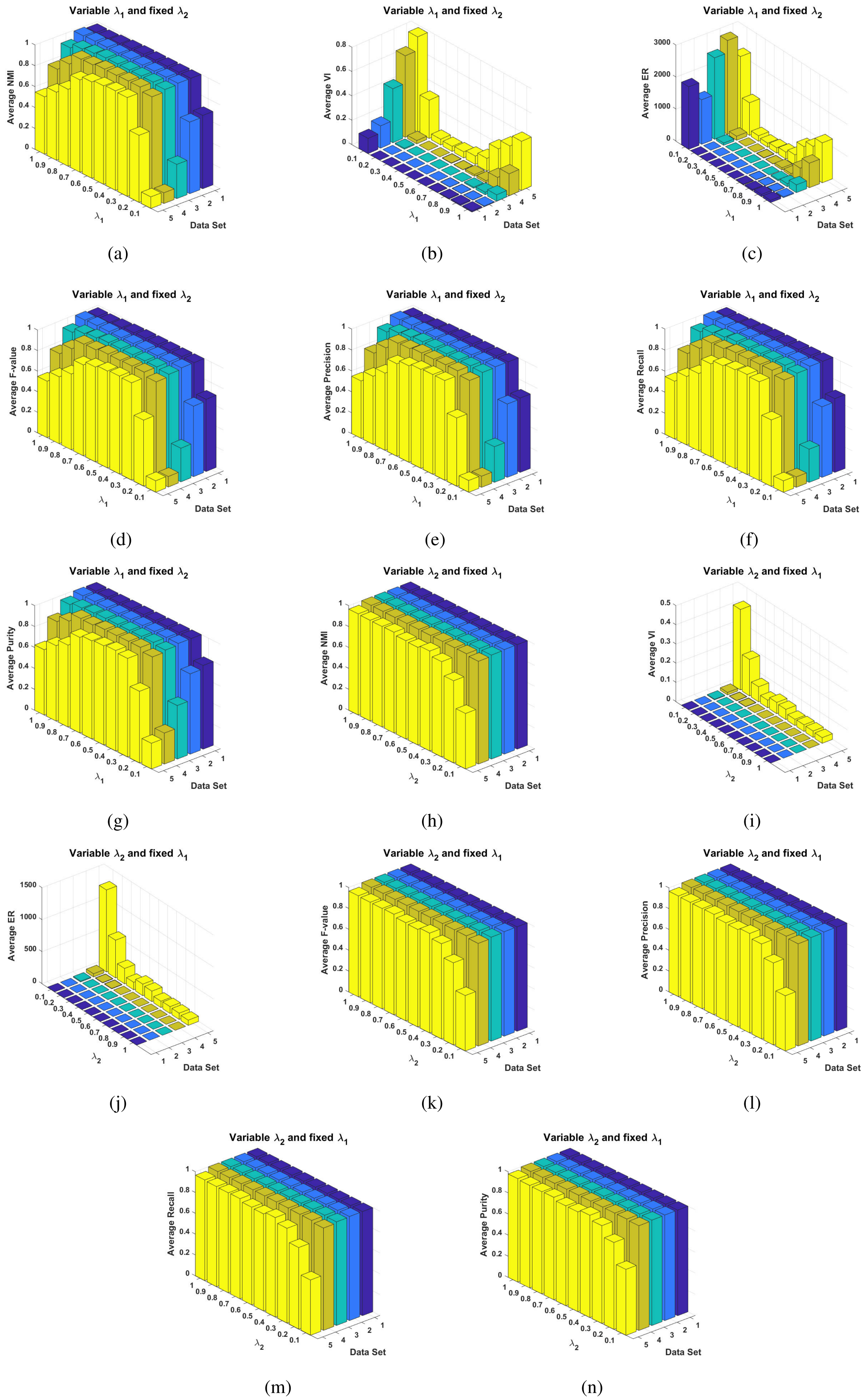
Figures from Fig. 3-Fig. 7 show some examples of the input and output of the proposed algorithm. In particular, they show the input corrupted adjacency matrices from GNB and PPB, the extracted low-rank component, the sparse component, and the final output smoothed adjacency matrix. As it can be seen in the figures, the low-rank components extracted by the proposed RCDG reinforces the intra-cluster edges and reduces the inter-cluster edges which creates more distinct clusters. Consequently, this leads to more accurate community detection results even when the input matrix is too noisy as in Fig. 5-Fig. 7.

In order to measure the accuracy of the resultant community structure, an *Error Rate* (ER) validation measure is adopted. This measure quantifies the distance between a clean graph that is built from the predetermined ground truth, G_t, and the resultant clustering assignment, C_t, by the algorithm. The results are reported in Table 4-Table 8. As it can be noticed from the ER values, the proposed algorithm achieved the lowest scores among all the algorithms and 0 ER in multiple experiments.

E. REAL NETWORKS

1) PRIMARY SCHOOL NETWORK

The data set comprises weighted network of face-to-face proximity between students and teachers in a primary school [42]. The school consists of 10 grades and 10 teachers. In the constructed network, the nodes represent the individuals and the edges represent the face-to-face interactions. Each node has two attributes: class name which represents the school class and the grade of the associated individual. Edges weights represent the duration measured in seconds. Duration is the total time for the face-to-face time proximity over the study period and recorded every 20 seconds. The network's statistics are: $|V| = 242$, $|E| = 5901$, $D_{avg} = 1.0747$, NBC $= 1017.165$, EBC $= 5.1502$, density $= 0.2024$, $\mathfrak{C} = 0.0077$ and $r = 0.1877$. Table 9 shows the grades' labels and number of students in each grade.



Variable λ_1 and fixed λ_2

(a)

Variable λ_1 and fixed λ_2

(b)

Variable λ_1 and fixed λ_2

(c)

Variable λ_1 and fixed λ_2

(d)

Variable λ_1 and fixed λ_2

(e)

Variable λ_1 and fixed λ_2

(f)

Variable λ_1 and fixed λ_2

(g)

Variable λ_2 and fixed λ_1

(h)

Variable λ_2 and fixed λ_1

(i)

Variable λ_2 and fixed λ_1

(j)

Variable λ_2 and fixed λ_1

(k)

Variable λ_2 and fixed λ_1

(l)

Variable λ_2 and fixed λ_1

(m)

Variable λ_2 and fixed λ_1

(n)

FIGURE 2. Performance sensitivity of RCDG w.r.t. the different values of the regularization parameters λ_1 and λ_2 for the simulated networks in Experiment 1 in terms of average: (a) NMI, variable λ_1; (b) VI, variable λ_1; (c) ER, variable λ_1; (d) F-value, variable λ_1; (e) Precision, variable λ_1; (f) Recall, variable λ_1; (g) Purity, variable λ_1; (h) NMI, variable λ_2; (i) VI, variable λ_2; (j) ER, variable λ_2; (k) F-value, variable λ_2; (l) Precision, variable λ_2; (m) Recall, variable λ_2;and (n) Purity, variable λ_2.

TABLE 6. The ground truth of the generated networks from Girvan-Newman benchmark (GNB) and planted-partitions benchmark (PPB).

Network	n	NOC	Ground truth (Nodes in each cluster C)
GNB1	50	2	$C_1(1-25)$, $C_2(26-50)$
GNB2	100	4	$C_1(1-25)$, $C_2(26-50)$, $C_3(51-75)$, $C_4(76-100)$
GNB3	200	8	$C_1(1-25)$, $C_2(26-50)$, $C_3(51-75)$, $C_4(76-100)$, $C_5(101-125)$, $C_6(126-150)$, $C_7(151-175)$, $C_8(176-200)$
PPB1	300	10	$C_1(1-25)$, $C_2(26-50)$,$C_3(51-80)$, $C_4(81-110)$, $C_5(111-140)$, $C_6(141-180)$, $C_7(181-210)$, $C_8(211-240)$,$C_9(241-270)$,$C_{10}(271-300)$
PPB2	600	15	$C_1(1-25)$, $C_2(26-50)$,$C_3(51-80)$, $C_4(81-110)$, $C_5(111-140)$,$C_6(141-180)$, $C_7(181-210)$, $C_8(211-240)$,$C_9(241-270)$,$C_{10}(271-300)$, $C_{11}(301-350)$, $C_{12}(351-400)$, $C_{13}(401-440)$,$C_{14}(441-500)$, $C_{15}(501-600)$
PPB3	1000	20	$C_1(1-50)$, $C_2(51-80)$, $C_3(81-110)$, $C_4(111-140)$, $C_5(141-180)$, $C_6(181-210)$, $C_7(211-240)$, $C_8(241-270)$, $C_9(271-300)$, $C_{10}(301-350)$, $C_{11}(351-400)$, $C_{12}(401-440)$, $C_{13}(441-500)$, $C_{14}(501-600)$, $C_{15}(600-650)$, $C_{16}(651-700)$, $C_{17}(701-740)$, $C_{18}(741-790)$, $C_{19}(791-830)$, $C_{20}(831-1000)$

TABLE 7. GNB and PPB Networks' statistics averaged over 50 simulations, including: Number of nodes (|*V*|), Number of edges (|*E*|), average degree (D_{avg}), node betweenness centrality (NBC), edge betweenness centrality (EBC), density, clustering coefficient ($\mathfrak{C}$) and assortativity coefficient (*r*).

Network	Type	$\|V\|$	$\|E\|$	D_{avg}	NBC	EBC	Density	$\mathfrak{C}$	r
GNB1	Binary	50	500	19.9670	29.0670	1.5610	0.4070	0.4630	−0.0413
GNB2	Binary	100	1255	25.117	74.263	1.7326	0.2537	0.3352	−0.0170
GNB3	Binary	200	4000	40.0105	159.164	1.7908	0.2011	0.3209	−0.0006
PPB1	Binary	300	15169	101.1267	197.8733	1.6562	0.3382	0.3426	−0.0095
PPB2	Binary	600	59677	198.925	400.075	1.6651	0.3321	0.3374	0.05350
PPB3	Binary	1000	163433	326.866	672.134	1.6711	0.3272	0.3321	0.09890

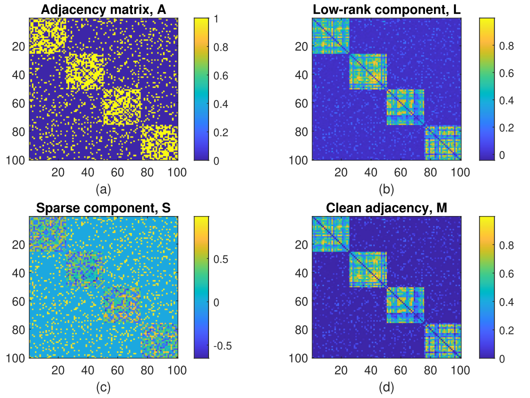


FIGURE 3. (a) Example of the corrupted adjacency matrix, A, from GNB2 (b) and (c) Low-rank, L and sparse, S decomposition, respectively, by RCDG, and (d) the clean version of the adjacency matrix.

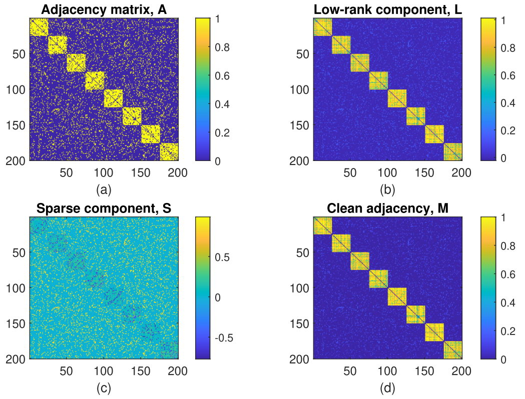


FIGURE 4. (a) Example of the corrupted adjacency matrix, A, from GNB3 (b) and (c) Low-rank, L and sparse, S decomposition, respectively, by RCDG, and (d) the clean version of the adjacency matrix.

The proposed algorithm is applied to the primary school data set to detect the community structure. The number of clusters is set to 10 which refers to the number of grades in the school. The regularization parameters are selected as $\lambda_1 = 0.2$ and $\lambda_2 = 0.1$. The detected clusters are shown in Fig. 8. The black frames represent the ground truth of the different grades in the school and the teachers are presented by the red frame. Whereas the colored rectangles represent the clusters detected by the proposed algorithm. As it can be seen in the figure, the detected clusters refer almost to the different grades including their teachers. This application of the proposed method to the primary school network shows its ability to detect the community structure in real-world networks. In fact, the extraction of the low-rank component or the clean version of the adjacency matrix plays an important role in improving the detected community structure quality. This due to the fact that it reinforces the intra-cluster edges and removes the sparse noise or outliers.

2) REALITY MINING NETWORK

This data set was collected in Massachusetts Institute of Technology (MIT) Reality Mining [43]. The collected data represents the recorded cell phone activity of 94 individuals at MIT over a year. Among the 94 individuals there was 68 who worked in the same building and 26 individuals were incoming students at the university's business school. The Media Access Control (MAC) addresses of nearby Bluetooth devices are recorded at five-minute intervals. The

TABLE 8. Performance comparison between the proposed method (RCDG), spectral clustering, modularity, SymNMF-ANLS, SymNMF-Newton, SymNMF-HALS and AANMF in terms of average NMI, VI, ER, F-value, Precision, Recall, Purity and DNOC. Networks are generated using GN-benchmark (GNB) and Planted Partitions-benchmark (PPB).

Network	Method	NMI	VI	ER	F-value	Precision	Recall	Purity	DNOC
GNB1	Spectral clustering	0.99391	0.00216	4.9	0.99391	0.99394	0.99388	0.999	**2**
$n = 50$	Modularity-Louvain	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**2**
NOC= 2	SymNMF-ANLS	0.99391	0.00216	4.9	0.99391	0.99394	0.99388	0.999	**2**
$z_i = 15$	SymNMF-Newton	0.99391	0.00216	4.9	0.99391	0.99394	0.99388	0.999	**2**
$z_e = 5$	SymNMF-HALS	0.99391	0.00216	4.9	0.99391	0.99394	0.99388	0.999	**2**
	A^{2}NMF	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**2**
	RCDG	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**2**
GNB2	Spectral clustering	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**4**
$n = 100$	Modularity-Louvain	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**4**
NOC= 4	SymNMF-ANLS	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**4**
$z_i = 15$	SymNMF-Newton	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**4**
$z_e = 10$	SymNMF-HALS	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**4**
	A^{2}NMF	0.82951	0.09983	1030.85	0.82244	0.8451	0.80819	0.8845	**4**
	RCDG	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**4**
GNB3	Spectral clustering	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**8**
$n = 200$	Modularity-Louvain	0.98913	0.00818	312.5	0.98913	**1**	0.97917	0.96875	**8**
NOC= 8	SymNMF-ANLS	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**8**
$z_i = 20$	SymNMF-Newton	0.99901	0.00079	46.25	0.99901	0.99806	**1**	**1**	**8**
$z_e = 20$	SymNMF-HALS	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**8**
	A^{2}NMF	0.78464	0.16098	3597.2	0.78246	0.82535	0.74837	0.73725	**8**
	RCDG	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**8**
PPB1	Spectral clustering	0.92122	0.063398	1455	0.92122	0.92109	0.92135	0.955	**10**
$n = 300$	Modularity-Louvain	0.7215	0.2011	8622	0.72154	0.81082	0.65049	0.65667	7
NOC= 10	SymNMF-ANLS	0.92241	0.062444	1379	0.92241	0.92224	0.92257	0.95833	**10**
$p_i = 0.7$	SymNMF-Newton	0.8565	0.11622	3574.5	0.8565	0.84911	0.86413	0.89833	11
$p_e = 0.3$	SymNMF-HALS	0.84086	0.12514	3604	0.84086	0.860874	0.82176	0.855	9
	A^{2}NMF	0.12169	0.60914	23353	0.12169	0.14068	0.10735	0.22	8
	RCDG	0.95693	**0.034656**	**775**	**0.95693**	**0.95686**	**0.957**	**0.97667**	**10**
PPB2	Spectral clustering	0.84795	0.12326	7855	0.84795	0.85611	0.83995	0.875	14
$n = 600$	Modularity-Louvain	0.59399	0.27858	44600	0.59399	0.73255	0.49979	0.56	7
NOC= 15	SymNMF-ANLS	0.82708	0.14171	8413	0.82708	0.82681	0.82752	0.8575	**15**
$p_i = 0.7$	SymNMF-Newton	0.69923	0.23783	17832	0.69923	0.72548	0.67509	0.70917	12
$p_e = 0.3$	SymNMF-HALS	0.72221	0.2193	15384	0.72221	0.7475	0.7003	0.74167	13
	A^{2}NMF	0.053055	0.66239	90444	0.053055	0.063914	0.045351	0.18167	13
	RCDG	**0.90705**	**0.076429**	**4674.5**	**0.90705**	**0.90404**	**0.9101**	**0.93083**	**15**
PPB3	Spectral clustering	0.81703	0.1508	31110	0.81703	0.81813	0.81592	0.847	18
$n = 1000$	Modularity-Louvain	0.60834	0.26809	105574	0.60834	0.76584	0.50457	0.53000	7
NOC= 20	SymNMF-ANLS	0.84585	0.12799	13791	0.84585	0.84074	0.85103	0.877	21
$p_i = 0.7$	SymNMF-Newton	0.58948	0.3386	62024	0.58948	0.58983	0.58912	0.625	18
$p_e = 0.3$	SymNMF-HALS	0.73664	0.21761	24216	0.73664	0.73575	0.73752	0.775	**20**
	A^{2}NMF	0.005421	0.83170	123436	0.005421	0.01065	0.00582	0.02817	16
	RCDG	**0.91914**	**0.066257**	**7364**	**0.91914**	**0.92576**	**0.91262**	**0.933**	21

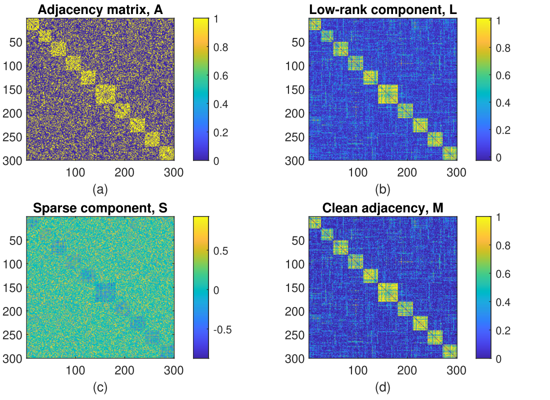


FIGURE 5. (a) Example of the corrupted adjacency matrix, A, from PPB1 (b) and (c) Low-rank, L and sparse, S decomposition, respectively, by RCDG, and (d) the clean version of the adjacency matrix.

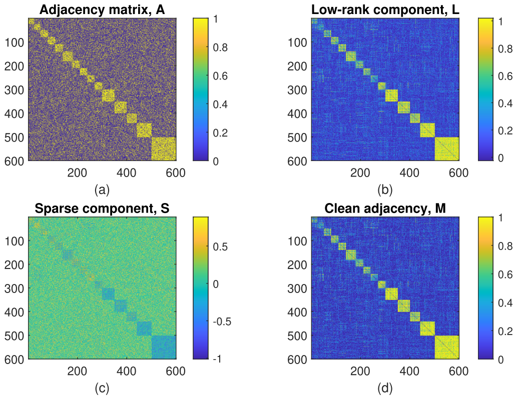


FIGURE 6. (a) Example of the corrupted adjacency matrix, A, from PPB2 (b) and (c) Low-rank, L and sparse, S decomposition, respectively, by RCDG, and (d) the clean version of the adjacency matrix.

similarity between two subjects refers to the number of times (intervals) in which there were physical proximity. The networks are constructed for multiple time steps where each time step represents a one week. In this paper, we obtained a static network by averaging the networks over the 46 time steps (The constructed networks between August 2004 and

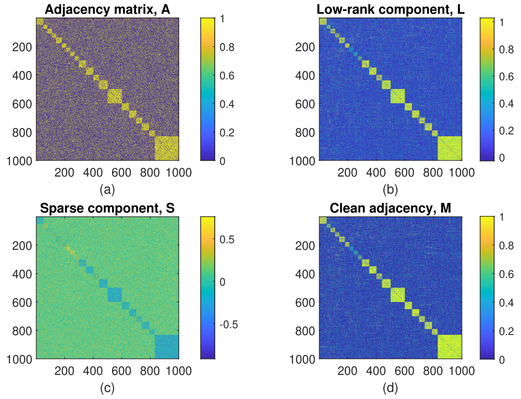


FIGURE 7. (a) Example of the corrupted adjacency matrix, A, from PPB3 (b) and (c) Low-rank, L and sparse, S decomposition, respectively, by RCDG, and (d) the clean version of the adjacency matrix.

TABLE 9. The grades labels and number of students in each grade.

Grade label	# of students
1A	23
1B	25
2A	23
2B	26
3A	23
3B	22
4A	21
4B	23
5A	22
5B	24
Teachers	10

Detected community structure by RCDG for the primary school cumulative network

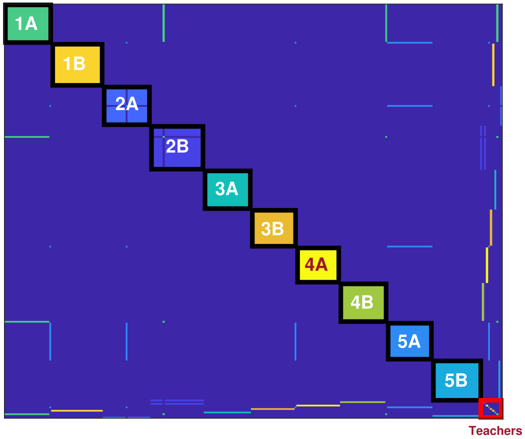


FIGURE 8. The detected community structure for the primary school network by the proposed algorithm.

June 2005). The network's statistics are: $|V| = 94$, $|E| = 3114$, $D_{avg} = 0.6025$, NBC $= 303.1064$, EBC $= 4.1928$, density $= 0.7124$, $\mathfrak{C} = 0.0026$ and $r = -0.0579$.

The proposed algorithm is applied to the reality mining network to detect its community structure.The number of clusters is set to 2. The regularization parameters are selected as $\lambda_1 = 0.3$ and $\lambda_2 = 1$. The detected clusters are shown in Fig. 9. The black frames represent the ground truth of the different students and the professor is presented by the red frame. Whereas the colored rectangles represent the clusters detected by the proposed algorithm. As it can be seen in the figure, the 2 clusters detected by the algorithm refer to the colleagues working in the same building on campus in addition to the professor while the second one refers to the incoming students.

Detected community structure by RCDG for the reality mining data set

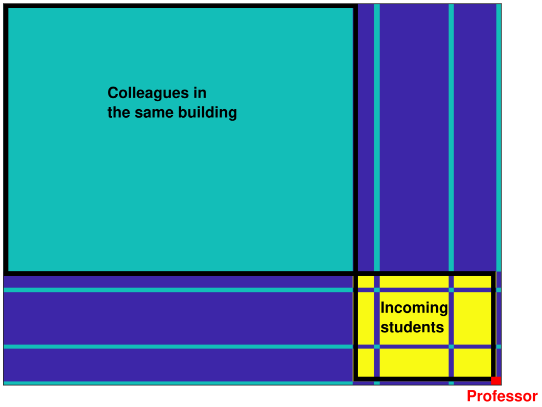


FIGURE 9. The detected community structure for reality mining network by the proposed algorithm.

V. CONCLUSION

In this paper, a new robust community detection algorithm in both binary and weighted graphs is proposed. The objective of the proposed approach is to detect the true community structure even in noisy networks. Particularly, the proposed approach decomposes the noisy adjacency matrix into low-rank and sparse components. The extracted low-rank component represents a clean version of the original noisy network. Moreover, the extracted low-rank component is used for clustering through nonnegative embedding. The robustness and accuracy of the proposed approach are tested and evaluated using multiple simulated and real-world networks. The proposed method shows high accuracy in detecting the network's community structure even with the presence of noise and outperforms other well-known algorithms.

APPENDIX A
UPDATE L

Updating each one of the variables is carried by iteratively alternating approach. In order to update each variable, we fix the other variables and solve for the variable of interest following the solution suggested in [4]. This procedure is repeated until convergence. A detailed derivation of the update rules is presented in this appendix and the following appendices. For the update of **L**, the terms with **L** only are kept:

$$\begin{aligned}\mathbf{L}^{l+1} &= \underset{\mathbf{L}\in\mathbb{R}^{n\times n}}{\text{argmin}} \|\mathbf{L}\|_* + \langle \mathbf{Z}_1^l, \mathbf{A}-\mathbf{L}-\mathbf{S}^l\rangle \\ &\quad + \frac{\gamma_1}{2}\|\mathbf{A}-\mathbf{L}-\mathbf{S}^l\|_F^2 + \langle \mathbf{Z}_2^l, \mathbf{M}^l-\mathbf{L}\rangle \\ &\quad + \frac{\gamma_2}{2}\|\mathbf{M}^l-\mathbf{L}\|_F^2, \qquad (15)\end{aligned}$$

by combining the quadratic terms in Eq. 15, it can be simplified to:

$$\begin{aligned}
&= \operatorname*{argmin}_{\mathbf{L}\in\mathbb{R}^{n\times n}} \|\mathbf{L}^{(t)}\|_* + \frac{\gamma_1}{2}\|\mathbf{L}-(\mathbf{A}-\mathbf{S}^l+\frac{\mathbf{Z}_1^l}{\gamma_1})\|_F^2 \\
&\quad +\frac{\gamma_2}{2}\|\mathbf{L}-(\mathbf{M}^l+\frac{\mathbf{Z}_2^l}{\gamma_2})\|_F^2 \\
&= \operatorname*{argmin}_{\mathbf{L}\in\mathbb{R}^{n\times n}} \|\mathbf{L}\|_* + \frac{\gamma_1+\gamma_2}{2}\|\mathbf{L}-\frac{\gamma_1\mathbf{W}_1^l+\gamma_2\mathbf{W}_2^l}{\gamma_1+\gamma_2}\|_F^2 \\
&= prox_{\frac{1}{\gamma_1+\gamma_2}\|\mathbf{L}\|_*}(\frac{\gamma_1\mathbf{W}_1^l+\gamma_2\mathbf{W}_2^l}{\gamma_1+\gamma_2}), \qquad (16)
\end{aligned}$$

where $\mathbf{W}_1^l = \mathbf{A}-\mathbf{S}^l+\frac{\mathbf{Z}_1^1}{\gamma_1}$, $\mathbf{W}_2^l = \mathbf{M}^l+\frac{\mathbf{Z}_2^l}{\gamma_2}$ and $prox_f(\mathbf{W}) = \operatorname{argmin}_{\mathbf{L}\in\mathbb{R}^{n\times n}} f(\mathbf{L}) + \frac{1}{2}\|\mathbf{L}-\mathbf{W}\|_F^2$ is the proximity operator of the convex function f [35]. Letting $\mathbf{W} = \frac{\gamma_1\mathbf{W}_1^l+\gamma_2\mathbf{W}_2^l}{\gamma_1+\gamma_2}$, $\gamma = \frac{\gamma_1+\gamma_2}{2}$ and $\mathbf{W} = \mathbf{Q}_{\mathbf{W}}\Sigma_{\mathbf{W}}\mathbf{F}_{\mathbf{W}}^\top$ be the SVD of the matrix $\mathbf{W}$, singular value soft thresholding is then used to update $\mathbf{L}^{l+1}$ as:

$$\mathbf{L}^{l+1} = \mathbf{Q}_{\mathbf{W}}\Omega_{\frac{1}{\gamma}}(\Sigma_{\mathbf{W}})\mathbf{F}_{\mathbf{W}}^\top, \qquad (17)$$

where Ω_τ is the element-wise thresholding operator defined as $\Omega_\tau(a) = sgn(a)max(|a|-\tau, 0)$.

APPENDIX B
UPDATE S

Update $\mathbf{S}$ by keeping only the terms with $\mathbf{S}$:

$$\begin{aligned}
\mathbf{S}^{l+1} &= \operatorname*{argmin}_{\mathbf{S}\in\mathbb{R}^{n\times n}} \|\mathbf{S}\|_1 + \langle \mathbf{Z}_1^l, \mathbf{A}-\mathbf{S}-\mathbf{L}^{l+1}\rangle \\
&\quad +\frac{\gamma_1}{2}\|\mathbf{A}-\mathbf{S}-\mathbf{L}^{l+1}\|_F^2 \\
&= \operatorname*{argmin}_{\mathbf{S}\in\mathbb{R}^{n\times n}} \|\mathbf{S}\|_1 + \frac{\gamma_1}{2}\|\mathbf{S}-(\mathbf{A}-\mathbf{L}^{l+1}+\frac{\mathbf{Z}_1^l}{\gamma_1})\|_F^2 \qquad (18)
\end{aligned}$$

Following the $\mathbf{L}^{l+1}$ update, $\mathbf{S}^{l+1}$ can be calculated as:

$$\begin{aligned}
\mathbf{S}^{l+1} &= prox_{\frac{\lambda_1}{\gamma_1}\|\mathbf{S}\|_1}\big(\mathbf{A}-\mathbf{L}^{l+1}+\frac{\mathbf{Z}_1^1}{\gamma_1}\big) \\
&= \Omega_{\frac{\lambda_1}{\gamma_1}}\big(\mathbf{A}-\mathbf{L}^{l+1}+\frac{\mathbf{Z}_1^l}{\gamma_1}\big) \qquad (19)
\end{aligned}$$

APPENDIX C
UPDATE M

In a similar fashion to $\mathbf{L}$ and $\mathbf{S}$, $\mathbf{M}$ is updated by keeping only the terms with $\mathbf{M}$ which reduces the main problem to:

$$\begin{aligned}
\mathbf{M}^{l+1} &= \operatorname*{argmin}_{\mathbf{M}\in\mathbb{R}^{n\times n}} \lambda_2\|\mathbf{M}-\mathbf{H}^l\mathbf{H}^{l^\top}\|_F^2 + \langle \mathbf{Z}_2^l, \mathbf{M}-\mathbf{L}^{l+1}\rangle \\
&\quad +\frac{\gamma_2}{2}\|\mathbf{M}-\mathbf{L}^{l+1}\|_F^2 \quad s.t\ \mathbf{M}=\mathbf{M}^\top, \mathbf{M}\geq 0 \\
&= \operatorname*{argmin}_{\mathbf{M}\in\mathbb{R}^{n\times n}} \lambda_2\|\mathbf{M}-\mathbf{H}^l\mathbf{H}^{l^\top}\|_F^2 \\
&\quad +\frac{\gamma_2}{2}\|\mathbf{M}-(\mathbf{L}^{l+1}-\frac{\mathbf{Z}_2^l}{\gamma_2})\|_F^2, \\
&\quad \mathbf{M}=\mathbf{M}^\top, \mathbf{M}\geq 0 \\
&= \operatorname*{argmin}_{\mathbf{M}\in\mathbb{R}^{n\times n}} \mathcal{F}(\mathbf{M}), \quad \mathbf{M}=\mathbf{M}^\top, \mathbf{M}\geq 0. \qquad (20)
\end{aligned}$$

By expanding the Frobenius norm terms as trace functions, the gradient of the function $\mathcal{F}(\mathbf{M})$ can be formulated as:

$$\nabla_{\mathbf{M}} f(\mathbf{M}) = \lambda_2(2\mathbf{M}-2\mathbf{H}^l\mathbf{H}^{l^\top}) + \gamma_2\big(\mathbf{M}-\mathbf{L}^{l+1}+\frac{\mathbf{Z}_2^l}{\gamma_2}\big). \qquad (21)$$

Then, a closed form solution can be computed for $\mathbf{M}^{l+1}$ as:

$$\mathbf{M}^{l+1} = \frac{2\lambda_2\mathbf{H}^l\mathbf{H}^{l^\top}+\gamma_2\mathbf{L}^l-\mathbf{Z}_2^l}{2\lambda_2+\gamma_2}, \qquad (22)$$

where a symmetric version of $\mathbf{M}$ can be calculated as $\frac{\mathbf{M}+\mathbf{M}^\top}{2}$ and $M_{ij} = 0\ if\ M_{ij} < 0$.

APPENDIX D
UPDATE H

Finally, updating $\mathbf{H}$ is performed using SymNMF approach proposed in [10]. In [10], the authors proposed two algorithms to solve the SymNMF problem, namely Newton-like algorithm and an ANLS-based algorithm. Both algorithms are guaranteed to converge to stationary point solutions. However, as suggested by the authors, it is preferred to consider practical considerations about the data of interest to achieve better clustering results. In particular, Newton-like algorithm results in higher accuracy but is more suitable for small networks, e.g. $n < 3000$. Whereas, the ANLS algorithm suits sparse networks and performs well with large networks, e.g. $n = 10^6$. In the proposed approach, the symmetric nonnegative matrix factorization problem is solved using the ANLS algorithm due to its low complexity compared to Newton-like algorithm. Furthermore, since our proposed approach extracts a clean version of the adjacency matrix and uses it as an input for the SymNMF, this improves the final clustering results using ANLS.

REFERENCES

[1] M. Rubinov and O. Sporns, “Complex network measures of brain connectivity: Uses and interpretations,” *NeuroImage*, vol. 52, no. 3, pp. 1059–1069, 2010.

[2] E. Al-Sharoa, M. Al-Khassaweneh, and S. Aviyente, “Tensor based temporal and multilayer community detection for studying brain dynamics during resting state fMRI,” *IEEE Trans. Biomed. Eng.*, vol. 66, no. 3, pp. 695–709, Mar. 2019.

[3] S. P. Borgatti, A. Mehra, D. J. Brass, and G. Labianca, “Network analysis in the social sciences,” *Science*, vol. 323, no. 5916, pp. 892–895, 2009.

[4] E. Al-Sharoa, M. A. Al-Khassaweneh, and S. Aviyente, “Detecting and tracking community structure in temporal networks: A low-rank + sparse estimation based evolutionary clustering approach,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, no. 4, pp. 723–738, Dec. 2019.

[5] B. Malin, “Data and collocation surveillance through location access patterns,” in *Proc. NAACSOS Conf.*, 2004.

[6] S. Fortunato, “Community detection in graphs,” *Phys. Rep.*, vol. 486, nos. 3–5, pp. 75–174, 2010.

[7] E. Al-Sharoa, M. Al-Khassaweneh, and S. Aviyente, “Low-rank estimation based evolutionary clustering for community detection in temporal networks,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 5381–5385.

[8] S. Fortunato and D. Hric, “Community detection in networks: A user guide,” *Phys. Rep.*, vol. 659, pp. 1–44, Nov. 2016.

[9] U. von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, 2007. [Online]. Available: http://dblp.uni-trier.de/db/journals/sac/sac17.html#Luxburg07

[10] D. Kuang, S. Yun, and H. Park, "SymNMF: Nonnegative low-rank approximation of a similarity matrix for graph clustering," *J. Global Optim.*, vol. 62, no. 3, pp. 545–574, Jul. 2015.

[11] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.

[12] M. E. Newman, "Modularity and community structure in networks," *Proc. Nat. Acad. Sci. USA*, vol. 103, no. 23, pp. 8577–8582, 2006.

[13] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 2, 2004, Art. no. 026113.

[14] A. Schenker, H. Bunke, M. Last, and A. Kandel, *Graph-Theoretic Techniques for Web Content Mining*, vol. 62. Singapore: World Scientific, 2005.

[15] D. Kuang, C. Ding, and H. Park, "Symmetric nonnegative matrix factorization for graph clustering," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2012, pp. 106–117.

[16] Z. Zhu, X. Li, K. Liu, and Q. Li, "Dropping symmetry for fast symmetric nonnegative matrix factorization," 2018, *arXiv:1811.05642*. [Online]. Available: http://arxiv.org/abs/1811.05642

[17] F. Ye, S. Li, Z. Lin, C. Chen, and Z. Zheng, "Adaptive affinity learning for accurate community detection," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2018, pp. 1374–1379.

[18] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sep. 1999.

[19] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. London, U.K.: Pearson, 2016.

[20] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech., Theory Exp.*, vol. 2008, no. 10, Oct. 2008, Art. no. P10008.

[21] K. Berahmand and A. Bouyer, "LP-LPA: A link influence-based label propagation algorithm for discovering community structures in networks," *Int. J. Mod. Phys. B*, vol. 32, no. 6, Mar. 2018, Art. no. 1850062.

[22] K. Berahmand, S. Haghani, M. Rostami, and Y. Li, "A new attributed graph clustering by using label propagation in complex networks," *J. King Saud Univ.-Comput. Inf. Sci.*, Sep. 2020.

[23] K. Berahmand, A. Bouyer, and M. Vasighi, "Community detection in complex networks by detecting and expanding core nodes through extended local similarity of nodes," *IEEE Trans. Comput. Social Syst.*, vol. 5, no. 4, pp. 1021–1033, Dec. 2018.

[24] H. Cherifi, G. Palla, B. K. Szymanski, and X. Lu, "On community structure in complex networks: Challenges and opportunities," *Appl. Netw. Sci.*, vol. 4, no. 1, pp. 1–35, Dec. 2019.

[25] G. Rossetti and R. Cazabet, "Community discovery in dynamic networks: A survey," *ACM Comput. Surv.*, vol. 51, no. 2, pp. 1–37, Jun. 2018.

[26] Y.-X. Wang and Y.-J. Zhang, "Nonnegative matrix factorization: A comprehensive review," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 6, pp. 1336–1353, Jun. 2013.

[27] N. Gillis, *Nonnegative Matrix Factorization*. Philadelphia, PA, USA: SIAM, 2020.

[28] N. K. Kumar and J. Schneider, "Literature survey on low rank approximation of matrices," *Linear Multilinear Algebra*, vol. 65, no. 11, pp. 2212–2244, 2017.

[29] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, Sep. 1936.

[30] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *J. Educ. Psychol.*, vol. 24, no. 6, p. 417, 1933.

[31] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 1, pp. 1–37, 2009.

[32] S. Boyd, N. Parikh, and E. Chu, *Distributed Optimization and Statistical Learning Via the Alternating Direction Method of Multipliers*. Boston, MA, USA: Now, 2011.

[33] S. Kontogiorgis and R. R. Meyer, "A variable-penalty alternating directions method for convex optimization," *Math. Program.*, vol. 83, no. 1, pp. 29–53, 1998.

[34] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 127–239, Jan. 2014.

[35] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. New York, NY, USA: Springer, 2011, pp. 185–212.

[36] V. A. Traag, R. Aldecoa, and J.-C. Delvenne, "Detecting communities using asymptotical surprise," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 92, no. 2, Aug. 2015, Art. no. 022816.

[37] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng, "Analyzing communities and their evolutions in dynamic social networks," *ACM Trans. Knowl. Discovery Data*, vol. 3, no. 2, pp. 1–31, 2009.

[38] A. Condon and R. M. Karp, "Algorithms for graph partitioning on the planted partition model," *Random Struct. Algorithms*, vol. 18, no. 2, pp. 116–140, Mar. 2001.

[39] I. Gutman and B. Borovicanin, "Nullity of graphs: An updated survey," *Zbornik Radova*, vol. 14, no. 22, pp. 137–154, 2011.

[40] B. Savas and I. S. Dhillon, "Clustered low rank approximation of graphs in information science applications," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2011, pp. 164–175.

[41] W. Bao and G. Michailidis, "Core community structure recovery and phase transition detection in temporally evolving networks," *Sci. Rep.*, vol. 8, no. 1, pp. 1–16, Dec. 2018.

[42] J. Stehlé, N. Voirin, A. Barrat, C. Cattuto, L. Isella, J.-F. Pinton, M. Quaggiotto, W. Van den Broeck, C. Régis, B. Lina, and P. Vanhems, "High-resolution measurements of face-to-face contact patterns in a primary school," *PLoS ONE*, vol. 6, no. 8, Aug. 2011, Art. no. e23176, doi: 10.1371/journal.pone.0023176.

[43] N. Eagle, "The reality mining data," Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep., 2010. [Online]. Available: https://www.media.mit.edu/ventures/EPROM/data/RealityMining_ReadMe.pdf