Improving the accuracy of community detection in social network through a hybrid method

Mahsa Nooribakhsh[1], Marta Fernández-Diego[2], Fernando González-Ladrón-De-Guevara[3], Mahdi Mollamotalebi[4]

[1,2,3] Instituto Universitario Mixto de Tecnología Informática, Universitat Politècnica de València, Camino de Vera, s/n, 46022 Valencia, Spain

[4]Department of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran

Abstract. The inherent complexity of social networks in terms of topological properties requires sophisticated methodologies to detect communities or clusters. Community detection in social networks is essential for understanding organizational structures and patterns in complex interconnected systems. Traditional methods face challenges in handling the scale and complexity of modern social networks, such as local optima trapping and slow convergence. This paper proposes a hybrid method to improve the accuracy of community detection, leveraging stacked auto-encoder (SAE) for dimensionality reduction and the Shuffled Frog Leaping (SFLA) as memetic algorithm for enhanced optimization alongside k-means clustering. The proposed method constructs a hybrid similarity matrix combining structural information and community-related features, followed by SAE to reduce dimensionality and facilitate efficient processing of high-dimensional data. SFLA optimizes the k-means clustering process, introducing adaptability and diversity to exploration of the solution space. Experimental results indicated its superior performance in terms of normalized mutual information (NMI) and modularity compared to existing approaches.

Keywords: community detection, social networks, stacked auto-encoder, shuffled frog leaping algorithm

1 Introduction

Community detection in social networks represents an area of research within the broader domain of network science that aims to reveal the organizational structures and patterns that characterize complex systems of interconnected nodes [1]. A social network can be conceptualized as a graph, where nodes correspond to individual entities (e.g. people, organizations, or web pages) and edges represent relations or interactions between these entities [2]. Traditional methods for community detection have relied on various algorithms, such as graph theory, modularity optimization, spectral clustering, and label propagation. However, the ever-increasing scale and complexity of social networks have prompted the exploration of innovative approaches, including the integration of deep learning techniques like auto-encoders. A classification offers an organized framework for understanding and exploring various approaches to community detection in social networks including graph partitioning algorithms, density-based algorithms, hierarchical clustering algorithms, spectral clustering, optimization algorithms, game-theoretic algorithms, and deep learning algorithms [3],[4].

By using iterative improvement processes and balancing exploration and exploitation, memetic algorithms can identify exact groups of individuals within social networks, considering connectivity patterns and community densities. These algorithms provide robustness to noise, and flexibility in customization which makes them suitable to discover meaningful community structures in social network. Examples of memetic algorithms include the Shuffled Frog Leaping Algorithm (SFLA), genetic algorithms, and, etc. [5].

Deep learning-based methods such as auto-encoders, are increasingly adopted for community detection due to their ability to discern intricate patterns and representations from network data [3]. A stack auto-encoder is a neural network architecture employed for unsupervised learning tasks such as community detection. It acts by compressing input data into a lower-dimensional representation through an encoder and then reconstructing the original input using a decoder [6],[7]. Recently, the hybrid methods have combined the capabilities of auto-encoders with meta heuristic algorithms such as memetic algorithms to detect communities on social networks. This paper provides a hybrid method using a stacked auto-encoder and SLFA as a memetic algorithm for detecting the communities accurately in social networks. The remainder of this paper is organized as follows: Section 2 includes the related works. The proposed method is described in Section 3. We present the simulation results and discussion in Section 4. Finally, Section 5 concludes the research and presents future works.

2 Related work

Community detection in social networks handles distinguishing the structures inside an organization that is more thickly associated inside than with the rest of the arrangement [4]. In the following, the recent works related to community detection on social networks is reviewed concisely. Xu et al. [8] have proposed community detection method using ensemble clustering with a stack auto-encoder for low-dimensional feature representation. It integrates multiple clustering results via nonnegative matrix factorization for reliability. However, its performance depends on similarity representation and initial clustering. Pierezan et al. [9] have proposed the Coyote Optimization Algorithm (COA) as a metaheuristic inspired by the social behavior of coyotes. It features a simple parameter set involving the number of packs and coyotes per pack. Despite its potential, COA is still in its early stages and has limitations such as the need for further refinement, potential scalability issues, and the absence of adaptive mechanisms.

Wang et al. [10] have introduced a proximity-based group formation game model for detecting communities. It formulates community formation as a two-step non-cooperative game and introduces a community interaction probability matrix to improve detection performance. However, the model's current application is limited to specific social networks, and its effectiveness in attributed social networks. Zhang et al. [11] have proposed a community detection algorithm based on core nodes and layer-by-layer label propagation, extended to detect overlapping communities. Layered label propagation starting from core nodes improves detection accuracy, and node labels are calibrated to reduce early misclassification. However, the algorithm's evaluation relies heavily on modularity.

Aslan et al. [12] presented a modified Coot bird model named MCOOT to detect the community in social networks which is inspired by the collective manners of coots on water surfaces. The update process improves the ratio between search and exploitation capabilities, leading to better detection. On the other hand, it suffers from the weakness of sensitiveness such that the performance of the MCOOT depends on parameter settings. Recent approaches to community detection highlight the use of learning methods to enhance clustering accuracy. While these methods show promise, challenges remain such as noise sensitivity, interpretability, etc. Research gaps enclose the need for better optimization techniques and using hybrid methods like stack auto-encoder and memetic algorithm is a promising way to improve these gaps and attain higher accuracy.

3 Proposed method

In this section, our proposed hybrid method is presented in detail, and it is evaluated in the next section. Our method is designed using a stacked auto-encoder along with SFLA as a memetic algorithm to detect the communities more accurately. The SFLA as a memetic algorithm can be used for community detection and leverages its population-based cooperative search. In this algorithm, a virtual population of frogs is separated into memeplexes, each defining a cultural unit of evolution known as a meme. Also, the frogs are occasionally shuffled among memeplexes [5]. The components of the proposed method are shown in Fig.1:

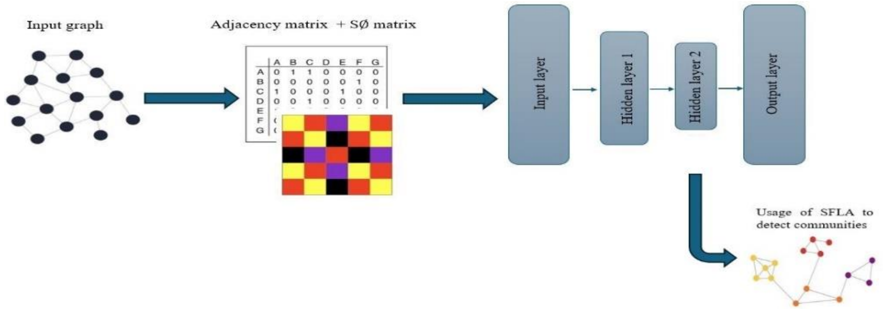

Fig. 1. The components of proposed method

According to Fig.1 the proposed method comprises three steps. The steps are presented in the following:

Step1: Matrix construction

Graphs can be effectively represented through the utilization of an adjacency matrix, The adjacency matrix comprehensively captures direct connections between nodes, which can be expressed mathematically as Equation (1):

$$\mathrm{Aij} = \begin{cases} 1 \text{ if } \mathrm{v_{ij}} \in \mathrm{E} \\ 0 \text{ otherwise} \end{cases} \qquad (1)$$

On the other hand, the use of a similarity matrix, known as SØ (SØ-similarity), offers a unique approach to comprehending the relationships between nodes [13]. Equation (2) presents the SØ matrix:

$$S\emptyset = \frac{2\mathrm{comNeig}(\mathrm{v_i},\mathrm{v_j})}{\mathrm{d}(\mathrm{v_i}) + \mathrm{d}(\mathrm{v_j})}\; S\emptyset \in R^{n*n} \qquad (2)$$

where $comNeig(v_i, v_j)$ is the number of common neighbors between two vertices $\mathrm{v_i}$ and $\mathrm{v_j}$, and $\mathrm{d(v_i)+d(v_j)}$ denotes the degrees of vertices vi and vj respectively.

In order to overcome the limitations of individual matrices, the enriched hybrid matrix H represents a combined relationship measure between nodes $\mathrm{V_i}$ (i-th node in the graph) and $\mathrm{V_j}$, (j-th node in the graph) aiming to balance the direct connectivity captured by adjacency and the community-based relationships highlighted by SØ. This combination is achieved through a weighted sum of the two matrices, resulting in a synthesized representation that enhances the accuracy of community detection in social networks. Equation (3) shows the structure of H:

$$H = \alpha . A + (1-\alpha). S\emptyset \qquad (3)$$

where α is a weighting parameter (between 0 and 1) that balances the influence of A and SØ.

Step 2: Dimensionality reduction with stacked auto-encoder

This step aims at reducing dimensionality through the utilization of a stacked auto-encoder (SAE) as follows: The SAE process commences with the input layer. The encoding process $\mathrm{E_i}$ refers to the operation of encoding in the i-th layer of the SAE. The encoding process can be mathematically articulated as follows:

$$\mathrm{Z}i = Ei(\mathrm{Z}i - 1),\ \mathrm{for}\ i = 1,2, \ldots, NumberOfLayers \qquad (4)$$

where $\mathrm{Z_0}$ =X, and $\mathrm{Z_{NumberOfLayers}}$ represents the encoded low-dimensional representation, $\mathrm{Z_0}$ is considered as the input matrix X.

In a simple linear transformation, Equation (5) represents a SAE:

$$Z_{ij} = \sigma(W_{ij}.(Z_{i-1}) + b_{ij}) \qquad (5)$$

where $\mathrm{Z_{ij}}$ is the encoded representation of the j-th node in layer i. $\mathrm{W_{ij}}$ is weight matrix associated with the connection between the j-th node in layer i-1 and the j-th node in layer i and $\mathrm{b_{ij}}$ is the bias term associated with the j-th node in layer i. $\mathrm{Z_{i-1}}$ is output from the previous layer (i-1). Moreover, σ is the activation function (commonly a nonlinear function like the sigmoid or ReLU). We employed the He initialization function as described in equation (6):

$$W\sim Normal(0,\sqrt{\frac{2}{numbr\ of\ input\ units}}) \qquad (6)$$

Simultaneously, for the sigmoid activation function in the output layer, we chose the Xavier initialization as shown in equation (7):

$$W\sim Normal(0,\sqrt{\frac{2}{numbr\ of\ input\ units+number\ of\ output\ units}}) \quad (7)$$

where the *number of input units* refers to the total number of input nodes or neurons in the layer for which you are initializing the weights. It represents the dimensionality of the input data that is fed into that layer.

After encoding, the SAE follows with decoding D_i. The decoding is presented as:

$$X_{reconstructed}=D_i(Z_i),\ \text{for } i=num_{layers},\ldots,1 \qquad (8)$$

where $X_{Reconstructed}$ is the reconstructed input.

In the training stage of our stacked auto-encoder, we used Adam optimization to update the weights, biases, or descent loss functions, depth, and number of layers (as described in section 4). The output of one of the encoding layers (Z_{final}) acts as a low-dimensional representation of the input hybrid matrix (H). The low-dimensional representation (Z_{final}) integrates in to with a community detection algorithm to provide an improved feature space for higher accuracy and efficiency.

Step 3: Using a memetic metaheuristic algorithm for community detection

In order to select the most efficient centroid point and increase the accuracy of detection, we employed the SFLA [5]. Unlike the pure SFLA, where frogs typically represent solutions to optimization problems, in our proposed adaptation, frogs represent potential centroids for the K-means algorithm. The presented process incorporates evolutionary operators along with a local search function to refine the positions of frogs. The fitness function evaluates the quality of centroids based on the alignment with the underlying community structure. Such adaptation enhances the efficiency of centroid initialization for community detection on social networks distinguishing it from the pure SFLA. Algorithm1 presents the customized version of SFLA used in the proposed method.

Algorithm 1. Customized version of SFLA used in proposed method	**Algorithm 2. Community detection via a stack auto-encoder & SFLA**
Input P: population size (number of frogs/centroids), N: number of dimensions, Max Iterations, Crossover Rate: Rate of crossover for evolutionary operators, Mutation Rate: Rate of mutation for evolutionary operators	**Input:** social network as graph representation, adjacency matrix A=[a_ij]∈R^(N*N) , Parameters of SAE, Parameters of SFLA **Output:** NMI, Modularity, Clustering result

```
Output: Objective function (centroid fitness)
For i = 1 to P
    Frogs[i] = Initialize random centroid(N)
End
For iteration = 1 to Max Iterations
    Evaluate fitness of each frog (centroid)
    For i = 1 to P
        Frogs[i]. Fitness = Fitness function (Frogs[i])
    End
    Sort Frogs by Fitness frog
    Memeplexes = Divide frog to memeplexes
    Shuffle memeplexes for diversity
    For each memeplex in Shuffled memeplexes
        For i = 1 to size(memeplex)
            memeplex[i] = Local search(memeplex[i])
        End
    End
    Frogs = Combine Memeplexes (Shuffled memeplexes)
    For i = 1 to P
        if Random () < Crossover Rate
            j = Random Integer (1, P)
            Offspring = Crossover (Frogs[i], Frogs[j])
            Frogs[i] = Offspring
        End
        if Random () < Mutation Rate
            Frogs[i] = Mutation (Frogs[i])
        End
    End
    Frogs = Shuffle (Frogs)
Return centroid fitness
```

```
Create hybrid matrix H: Construction of similarity matrix by Equation (3) based on adjacency A
Initialize the weight matrix and bias by Equation (6), (7)
Calculate the activation of each layer by Equation (5), z_ (final )
Repeat
    Update iteratively the weight, bias
Until maximum number of iterations
Initialize the Shuffled Frog Leaping (SFLA)
Apply evolutionary operators (crossover and mutation) and local search to refine the positions of the centroids
Repeat
    Update iteratively the variables
Until convergence
Aquire the result of clustering, NMI, Modularity
Return clustering result, NMI, Modularity
```

The process is terminated after a specified number of iterations, ensuring adaptation to the evolving characteristics of the network while preventing excessive computational costs. An objective function connected to k-means performance is used in fitness evaluation. SFLA dynamically evolves candidate centroids through iterative loops and finishes according to predetermined criteria. The optimal set f centroids for the k-means algorithm are determined by the final solution, which uses the frog with the highest fitness. Algorithm 2 presents the steps of community detection via combination of stack auto-encoder and SFLA.

4 Experimental results

To evaluate the performance of the proposed method, we used three real-world datasets including Football, Dolphin, and Political books (Polbooks). The experiments were carried out in the Google Collaboratory platform (GoogleColab). The details of the datasets and layer setting are shown in Table 1:

Table 1: The features of the dataset and layer setting of autoencoder on the selected dataset

Datasets	Network Format	Number of Nodes/Edges	Number of Communities	Layer Setting
Football[1]	undirected	115/613	12	N, 64,32
Dolphin[2]	undirected	62/159	2	N ,32,16
Polbooks[3]	undirected	105/441	3	N 64,32

The learning rate for each auto-encoder was 0.001 trained up to 1000 epochs.

Each dataset has a known number of communities, as listed in Table 1. When we run our proposed algorithm on a given dataset, our proposed algorithm can effectively identify the same number of communities as specified in Table 1 for each dataset. The quality of the detected communities is then assessed using Normalized Mutual Information (NMI) and Modularity, demonstrating the accuracy and reliability of our method in community detection. Two sets of partitions are compared for similarity using the NMI which is determined using the Equation (9), and its value can range from 0 to 1[5]:

$$NMI\ (Y,C) = \frac{2\times I(Y;C)}{[H(Y)+H(C)]} \qquad (9)$$

where Y refers to class labels, C refers to cluster labels, H is entropy, and I (Y, C) denote the mutual information between Y and C.

The analysis for Fig. 2 to 4 emphasizes the importance of both the configuration and the number of basic communities to have optimal performance based on NMI in community detection schemes. Fig. 2 to Fig.4 illustrate the results of NMI using selected layer settings of SAEs.

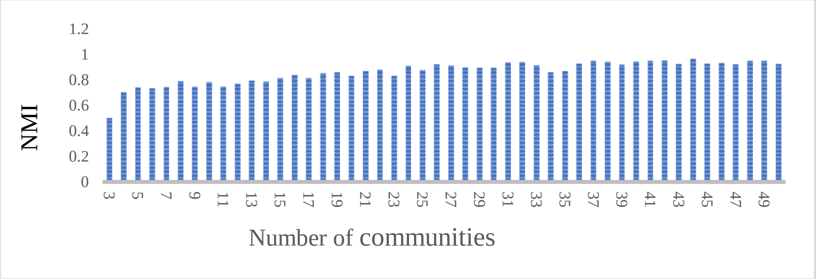

Fig. 2. The results of NMI using 62,32,16 stacked auto-encoders with different numbers of communities on the Dolphin dataset

[1] http://konect.cc/networks/dimacs10-football/

[2] https://networkrepository.com/

[3] https://networkrepository.com/

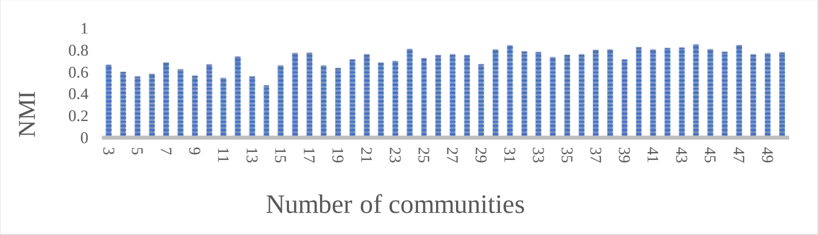

Fig. 3. The results of NMI using 105,64,32 stacked auto-encoders with different numbers of communities on the pollbook dataset

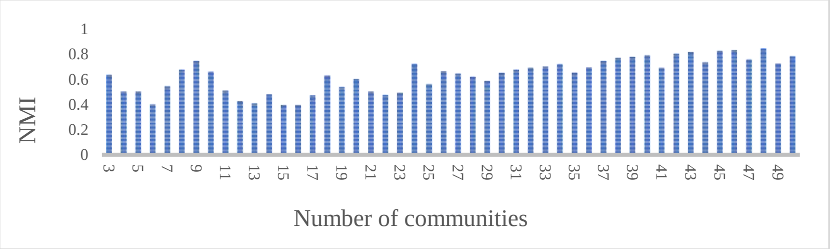

Fig 4. The results of NMI using 115,64,32 stacked auto-encoders with different numbers of communities on the football dataset

Fig. 2 to 4 present the effect of the designed stacked auto-encoder on 50 communities using three different datasets to measure average values of NMI. The results indicate that different configurations of stacked auto-encoders can affect the performance with regard to NMI. Further, it highlights the number of basic communities required to achieve the best NMI in each configuration. In order to evaluate the performance of the proposed method, its results were compared with recent methods SAECF [8], AECD-COA [9], PFGM [10] , and CNLLP [11] in terms of the NMI measures. In this paper, the experimental results are reported by the average NMI for datasets of real communities. Table 2 shows the NMI for all algorithms evaluated:

Table 2. NMI values for all the algorithms evaluated according to the selected datasets

Method	Football	Dolphin	Pollbooks
SAECF	0.67	0.49	0.39
AECD-COA	0.89	0.63	0.41
PFGM	0.72	0.7	0.34
CNLLP	0.27	0.60	0.48
Proposed method	0.84	0.96	0.85

Using SFLA to improve the k-means function in our method resulted in more diversity in population selection and avoided trapping in a local optimum. The algorithms like SFLA are able to adapt to environment changes; therefore, they can act appropriately in complex conditions with high dynamicity over time.

Another advantage of the proposed method is that it acts efficiently to search/explore within local and global scopes of the network structure. This is because the proposed method converges to optimal solutions quickly. Notably, our findings indicate that while the results of our proposed method using Football dataset for 50 clusters is comparatively lower than other methods, scaling up to 100 clusters results better outcomes for our method. It is noteworthy that if we consider very large number of clusters and use small datasets, the probability of trapping in local optimum increases. Modularity is known as a measure for quantifying the strength of the division of a network into communities. The modularity Q is computed as:

$$Q = \frac{1}{2m} \sum_{ij}(Aij - \frac{kikj}{2m})\, \delta(ci, cj) \qquad (10)$$

where A_{ij} is the adjacency matrix denoting the presence or absence of an edge between nodes i and j, k_i and k_j are the degrees of nodes i and j, respectively located in the same community c_i and c_j are the community assignments of nodes i and j will be equal to 1. Table 3 show the modularity value:

Table 3. Modularity values for all the algorithms evaluated according to the selected datasets

Method	Football	Dolphin	Pollbooks
SAECF	0.57	0.48	0.59
AECD-COA	0.59	0.30	0.43
PFGM	0.52	0.40	0.54
CNLLP	0.56	0.52	0.45
Proposed method	0.58	0.57	0.61

Comparing our proposed algorithm to other community detection algorithms on all of the selected datasets, the above results demonstrate that it performs more optimally in terms of modularity score. As opposed to other algorithms, the shuffled frog leaping algorithm clusters the nodes in the networks much more effectively, and this is responsible for the optimized performance. It is also paired with stacked auto-encoders. The complexity analysis of the proposed method involves considerations of both computational complexity and iterative processes. Constructing the hybrid has a time complexity of $O(N^2)$. If the SAE has L layers and each layer has d_i nodes, the time complexity for forward propagation is $O(L.N.d^2)$, where d is the maximum dimensionality across all layers. Initializing the population and memeplexes has a time complexity of O (P.N), where P is the population size.

5 Conclusion

The proposed method addressed the challenges posed by the complex topological properties, heterogeneous node properties, and dynamic evolution mechanisms of social networks. Using stacked auto-encoders for dimensionality reduction can reduce the complexity of high-dimensional data while preserving important features. Moreover, using SFLA provides adaptability and diversity to the optimization process, thus enabling effective exploration of the solution space. The proposed method exhibited robustness in dynamic environments, making it suitable for real-world applications where community detection is pivotal for understanding network structures and patterns. The evaluation of the proposed method showed its superiority in terms of the average NMI factor and modularity compared to recent works. In future works, we intend to study the scalability and consider large networks.

References

1. Newman, M.E., *Modularity and community structure in networks.* Proceedings of the national academy of sciences, 103(23): p. 8577-8582, (2006).
2. Fortunato, S., *Community detection in graphs.* Physics reports, 486(3-5): p. 75-174,(2010).
3. Souravlas, S., et al., *A classification of community detection methods in social networks: a survey.* International Journal of General Systems, 50(1): p. 63-91,(2021).
4. Su, X., et al., *A comprehensive survey on community detection with deep learning.* IEEE Transactions on Neural Networks and Learning Systems, 35(4): p. 4682 - 4702,(2022).
5. Maaroof, B.B., et al., *Current studies and applications of shuffled frog leaping algorithm: a review.* Archives of Computational Methods in Engineering, 29(5): p. 3459-3474,(2022).
6. Souravlas, S., S. Anastasiadou, and S. Katsavounis, *A survey on the recent advances of deep community detection.* Applied Sciences, 11(16): p. 7179,(2021).
7. Wu, L., et al., *Deep learning techniques for community detection in social networks.* IEEE Access, 8: p. 96016-96026,(2020).
8. Xu, R., et al., *Stacked autoencoder-based community detection method via an ensemble clustering framework.* Information sciences, 526: p. 151-165,(2020).
9. Pierezan, J. and L.D.S. Coelho. *Coyote optimization algorithm: a new metaheuristic for global optimization problems*. in *2018 IEEE congress on evolutionary computation (CEC)*. IEEE,(2018).
10. Wang, Y., et al., *Proximity-based group formation game model for community detection in social network.* Knowledge-Based Systems, 214: p. 106670,(2021).
11. Zhang, W., R. Shang, and L. Jiao, *Large-scale community detection based on core node and layer-by-layer label propagation.* Information Sciences, 632: p. 1-18,(2023).
12. Aslan, M. and İ. Koç, *Modified Coot bird optimization algorithm for solving community detection problem in social networks.* Neural Computing and Applications, 36(10): p. 5595-5619,(2024).
13. Jin, D., et al., *A survey of community detection approaches: From statistical modeling to deep learning.* IEEE Transactions on Knowledge and Data Engineering, 35(2): p. 1149-1170,(2021).