



# Community Detection Clustering via Gumbel Softmax

Deepak Bhaskar Acharya<sup>1</sup> · Huaming Zhang<sup>1</sup>

Received: 16 May 2020 / Accepted: 21 July 2020 / Published online: 11 August 2020  
© Springer Nature Singapore Pte Ltd 2020

## Abstract

Recently, in many systems such as speech recognition and visual processing, deep learning has been widely implemented. In this research, we are exploring the possibility of using deep learning in community detection among the graph datasets. Graphs have gained growing traction in different fields, including social networks, information graphs, the recommender system, and also life sciences. In this paper, we propose a method of community detection clustering the nodes of various graph datasets. We cluster different category datasets that belong to affiliation networks, animal networks, human contact networks, human social networks, miscellaneous networks. The deep learning role in modeling the interaction between nodes in a network allows a revolution in the field of science relevant to graph network analysis. In this paper, we extend the gumbel softmax approach to graph network clustering. The experimental findings on specific graph datasets reveal that the new approach outperforms traditional clustering significantly, which strongly shows the efficacy of deep learning in graph community detection clustering. We do a series of experiments on our graph clustering algorithm, using various graph datasets: Zachary's karate club, Highland tribes, Train bombing, American Revolution, Dolphins, Zebra, Windsurfers, Les Misérables, Political books.

**Keywords** Community detection · Gumbel softmax · Graph node clustering · Machine learning · Deep learning

## Introduction

Deep learning has become a hot topic in machine learning and artificial intelligence fields. In specific tasks such as speech recognition, natural language processing, image processing, and large-scale training frameworks for deep learning have been developed and widely implemented. The use of deep learning in community detection clustering has not yet been thoroughly studied, to our knowledge. The goal of this research is to carry out some preliminary research along that path.

Communication can be a human social networks, affiliation networks, animal networks, or human contact networks. Moreover, it is a subject of great interest in its economic and marketing implications to discover and analyze the community structure. For example, the advertisement can

be enhanced by recognizing and targeting the most active users of any community, taking advantage of effects such as word of mouth, and spreading information within the group. Likewise, it may be efficient to take advantage of user affiliations with communities to provide useful recommendations based on shared interests with their friends. Particularly in the expansion of information growth, a vast field of research has emerged and has seen significant attention to the available data. Identifying the circles that more frequently interact with each other enables us to understand how influence, knowledge, and even happiness, flow through a network. Finding these groups has been known as community detection in the network analysis. Many community detection methods have been proposed [3, 10, 12, 15–17]; However, most of them follow the topological structure or resemblances of the attributes. Graph clustering identifies the clusters of densely linked nodes when the graph is given as an input, taking into account the edge structure of the graph in such a way that there are several edges within each cluster and very few between clusters. Graph clustering intends to partition the nodes in the graph into disjoint groups. Graph clustering has been a long-standing subject of research. The main objective of graph clustering is to find strongly linked

---

✉ Deepak Bhaskar Acharya  
da0023@uah.edu

Huaming Zhang  
zhangh2@uah.edu

<sup>1</sup> Computer Science Department, The University of Alabama in Huntsville, Huntsville, AL 35806, USA

subgraphs in a graph, to synthesize the two closest related ones. (1) Within the same graph, the nodes of a graph will be very cohesive but sparsely connected with other sub-graphs. (2) Nodes exhibiting homogenous characteristics should be in the same group and heterogeneous characteristics in the other group. The number of clusters in the graph can be automatically identified by certain graph clustering methods, such as the heuristic method [4, 6]. However, in a few cases, we still need the number of clusters as an input parameter. In comparison, graphic clustering methods either take topological or homogenous node properties into account. So far, few algorithms have treated them synthetically.

In this research, we propose the new method for community detection and clustering by extending the gumbel softmax approach that [1] uses to select and extract the features for the Graph Neural Networks(GNNs) in the graph datasets using deep learning-based approach. GNNs are information-processing models that capture the graph dependence through passing the message between the nodes of the graphs. In contrast to standard neural networks, GNNs remember the state that can represent information from the neighboring nodes with the arbitrary depth. In this research, graph clustering is nothing but community detection and clustering. It is not to be mistaken with the clustering of graph sets based on structural similarity.

We conduct a series of experiments using a variety of datasets such as Zachary's karate club, Highland tribes, Train bombing, American Revolution, Dolphins, Zebra, Windsurfers, Les Misérables, Political books. The result seems to be very impressive with high modularity measure, which is one of the measures of the structure of graphs that we use to measure the strength of the communities or clusters. The proposed method outperforms various recently proposed graph community detection methods.

We organize the remaining of the paper as follows. In “[Background and Related Work](#)”, we introduce the background and related works in more detail. In “[Proposed Method](#)”, we introduce our main method. In “[Experimental Results](#)”, we present the experimental results. In “[Conclusion and Future Enhancements](#)”, we conclude our work.

## Background and Related Work

### Community Detection Algorithms

Community detection is widely researched, and so far, various algorithms have been proposed. Community detection is a method for identifying similar groups and can be a complicated process based on the graph network nature and scale.

Scientists have categorized community detection algorithms in many ways. Based on the scale of their research [8], has researched the community detection algorithms

thoroughly by accepting most of the methods addressed in other surveys. The algorithm proposed in [10] constructs a motif-based hypergraph, and modules are created in the hypergraph by partitioning the highest K-connecting components. Subsequently, to find the clique from each module, the connectivity structure is strengthened by constructing an edge set. From the edge set created in the previous step, the new rewired network includes both the lower-order structure and the higher-order structure constructed from the original network. Then, by partitioning the rewired network, the higher-order community structure can be found. Label propagation is another well-known identity identification strategy that identifies communities by spreading the labels iteratively across the network. Raghavan, Albert, and Kumara [13] proposed an algorithm where each node selects a label that has the highest frequency in its 1-neighborhood. These labels are permitted to propagate synchronously and asynchronously around the network until near-equilibrium within the network is reached. In addition, to assign vertices to clusters, Raghavan employs a local technique based on the majority law. In [15], they detect the community structure cluster, and through iterations, the similar nodes that exhibit similar properties are embedded together. The method deep autoencoder-like non-negative matrix factorization for community detection [17] uses hierarchical non-negative matrix factorization to create community-aware node embeddings. Here, the autoencoder concept is used to find the community in the network, i.e., first, the encoder component transforms the original network into low-dimensional hidden attributes captured in the hidden layers. Then, the decoder component is used to reconstruct the network with the community network findings from the encoder. The algorithm community preserving network embedding in [16] preserves both the microscopic structure (pairwise node similarity) and mesoscopic structure (community) for network embedding. For the mesoscopic structure, community preserving network embedding incorporates the first and second-order proximities to learn the representations using matrix factorization, and then the community is detected by a modularity constraint.

### Gumbel Softmax Approach on Feature Selection

Deepak and Huaming [1] selected Graph Neural Network(GNN) features in the paper feature selection and extraction for Graph Neural Networks, with the citation network datasets. (1) They apply the feature selection algorithm to GNNs using gumbel softmax and conduct a series of tests using various comparison datasets: Cora, Pubmed, and Citeseer. (2) They develop a method for ranking the features picked. To show the usefulness of algorithms for both feature selection and ranking of features. Deepak and Huaming demo an illustrative example for the Cora dataset, where they pick

225 features out of 1433 features and rank them according to prominent features. The proposed deep learning model works well with reduced features, which is around 80–85% decrease in the number of features that the dataset had initially. Results of the experiment reveal that the accuracy slowly decreases by using the selected features falling within range 1–50, 51–100, 100–150, and 151–200 for classification.

Consider the graph of ‘ $n$ ’ nodes and ‘ $f$ ’ features. Applying the principle of feature selection also takes down the number of features from  $f$  to  $k$ , where  $k$  reflects the number of features picked. First, to train the data collection and characteristics, using the selection matrix of the gumbel function applied (the matrix that has the features chosen while implementing the gumbel softmax algorithm).

In general, let  $X_{n \times f}$  be the matrix of the input features where ‘ $n$ ’ represents the total number of nodes, and ‘ $f$ ’ represents the total number of features in the graph data set for each node. Consider  $M_{f \times k}$  where ‘ $f$ ’ represents the total number of features in the dataset of the graph, and ‘ $k$ ’ denotes the features that we select from ‘ $f$ ’ features.

Using the gumbel softmax function and the method proposed, Deepak and Huaming select the features in a graph citation dataset. Gumbel softmax distribution is [7], “a continuous distribution over the simplex which can approximate samples from a categorical distribution”. A categorical distribution, by defining the highest probability to one, and all the other probability to zero is a one-hot vector.

Let  $z$  be a categorical, random variable with probabilities of the form  $\pi_1, \pi_2, \dots, \pi_k$ . Julius [5] uses gumbel max trick which provides an easy and efficient way to draw samples  $z$  from a categorical distribution with the stated class probabilities  $\pi$ :

$$z = \text{one\_hot}(\arg \max_i [g_i + \log \pi_i]). \tag{1}$$

Training a neural network by gradient descent requires the differentiation of each network operation. Remember that in Eq. (1), where  $i = 1, 2, \dots, k$ , the argmax function and the stochastic sampling process  $g_i$  are not differentiable. Next, an optimal solution for having argmax differentiable is to approximate it by a softmax function. One can also use a temperature of  $\tau$  to control the argmax approximation standard as follows:

$$y_i = \frac{\exp((\log(\pi_i) + g_i)/\tau)}{\sum_{j=0}^k \exp((\log(\pi_j) + g_j)/\tau)} \text{ for } i = 1, 2, 3, \dots, k. \tag{2}$$

The two layer Graph Convolution Network (GCN) used in the experiment is defined as

$$\text{GCN}(X, A) = \text{Softmax}(A(\text{ReLu}(AXW_G W_1))W_2). \tag{3}$$

To verify the selected features and calculate the accuracy for classification they use the following two layer Graph Convolution Network as defined below

$$\text{GCN}(X, A) = \text{Softmax}(A(\text{ReLu}(AXW'_G))W_2), \tag{4}$$

A adjacency matrix of the undirected graph  $G$ ,  $X$  input feature matrix,  $W_G$  gumbel softmax feature selection/feature extraction matrix.  $W'_G$  feature selection/feature extraction matrix obtained from the result of Eq. (3).  $W_1, W_2$  layer-specific trainable weight matrix, ReLu Activation function  $\text{ReLu}(\cdot) = \max(0, \cdot)$ .

### Proposed Method

The method proposed in the paper Feature Selection and Extraction for Graph Neural Networks (FSEGNN) [1] helps to select the exact features instead of a linear combination of the contribution of the features. Once the size of the features set is reduced, and the model is trained, it still performs well with the graph node classification problem. The machine learning model is trained with the labeled dataset, i.e., the algorithm learns on the labeled dataset, and the network’s loss is calculated against the target class each node of the graph belongs, and weights in the network are backpropagated to minimize the loss. The method FSEGNN works well with the labeled dataset (supervised or semi-supervised learning). The method does not provide any insight into the unlabeled dataset (unsupervised learning). The world is growing with data every day, and most of the datasets available in the outside world are not labeled. As an extension to FSEGNN, our method, Community Detection Clustering via Gumbel Softmax (CDCGS), gives an idea on how to work with the unlabelled graph datasets, and find the clusters for the community detection.

Our method CDCGS considers the graph of ‘ $n$ ’ nodes and the adjacency matrix ‘ $A$ ’. An adjacency matrix is a square matrix used to describe a finite graph. Matrix elements signify whether or not the pairs of vertices in the graph are adjacent. Then, applying our idea, we cluster the graph into  $k$  clusters.

Whenever we have a stochastic neural network with discrete variables, we can use gumbel softmax distributions to approximate the sampling process of discrete data. The network can then be trained using backpropagation, where the performance of the network will depend on the choice of the temperature parameter.

The method used in the experiment is defined as below:

$$\text{Graph – Cluster}(A) = \text{Softmax}(W'_C A W_C). \tag{5}$$

In the Eq. (5), ‘A’ indicates the Adjacency matrix of the undirected graph G and ‘ $W_C$ ’ indicates the gumbel cluster weight matrix.

In general, let  $A_{n \times n}$  be the adjacency matrix where ‘n’ represents the total number of nodes in the graph dataset. The size of the matrix  $W_C$  is  $n \times k$ , where  $k$  indicates the number of clusters. When we perform  $W_C^t A W_C$  operation, we obtain a matrix of the size  $k \times k$  and let us call this matrix as  $R_{k \times k}$ . The resultant  $R_{k \times k}$  matrix shows the strength of the cluster where the primary diagonal shows the strength of data points within cluster groups, and other elements of the matrix  $R$  provide details on the strength of data points between different cluster groups. Then, we apply softmax function on the obtained matrix  $R$  to express our inputs as a discrete probability distribution. Mathematically, this is defined as follows:

$$\text{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_{j=0}^m \exp(x_j)} \text{ for } i = 1, 2, 3, \dots, m. \quad (6)$$

Now, consider the trained gumbel cluster weight matrix ( $W_C$ ), which is of the size  $n \times k$ . Here, each row is a graph node and will sum up to 1, and the index of the maximum row value is the cluster to which the row node belongs. For example, let us consider  $k = 2$ , i.e., we are trying to cluster the dataset into 2 cluster groups. Here, we assume the first row data as [0.9 0.1], where 0.9 is at the 0th index, and 0.1 is at the 1st index. Looking at the index of the maximum value in the row vector, one can easily say that the graph node belongs to cluster 0. If the second-row assumed data is [0.29 0.71], where 0.29 is at the 0th index, and 0.71 is at the 1st index. Looking at the index of the maximum value in the row vector, one can easily say that the graph node belongs to cluster 1. Likewise, we look into all the rows and then cluster all the nodes of the graph dataset into the cluster group. The row data values indicate the influence of the graph node towards the cluster group.

The result obtained from the Eq. (5) is compared with the loss function. The loss function for our experiment used is the identity matrix ( $I_{k \times k}$ ) where each diagonal element represents the cluster group.

Using the gumbel softmax function and the method proposed, we find the community cluster of the graph dataset nodes.

## Experimental Results

We cluster different category dataset that belongs to affiliation networks, animal networks, human contact networks, human social networks, miscellaneous networks. The datasets are downloaded from Konect [9].

## Datasets

### Human Social Networks

Human social networks are real-world social networks among human beings. The links are offline, and not from a social network.

*Zachary’s Karate Club* (Figs. 1, 2, 3): The network data were collected from members of the University Karate Club by Wayne Zachary in 1977. Every node represents a member of the club, and an edge represents a link between the two members of the club. The network has 34 nodes, 78 edges and is undirected. The often-discussed problem of using this dataset is finding the group the student joins after an argument between two teachers. The network

*Highland tribes* (Figs. 4, 5): The network is the Gahuku-Gama alliance system of the Eastern Central Highlands of New Guinea, signed social network of tribes from Kenneth Read (1954). The network comprises seven-teen tribes connected by friendship (“Rova”) and enmity (“Hina”).

*Train bombing* (Figs. 6, 7): The dataset includes communications between alleged terrorists involved in the Madrid train bombing on 11 March 2004. A node represents a terrorist, and an edge between two terrorists shows that the two terrorists had a connection. The weights on edge indicate how ‘strong’ relation was. The relationship includes friendship and co-participating in training camps or previous attacks.

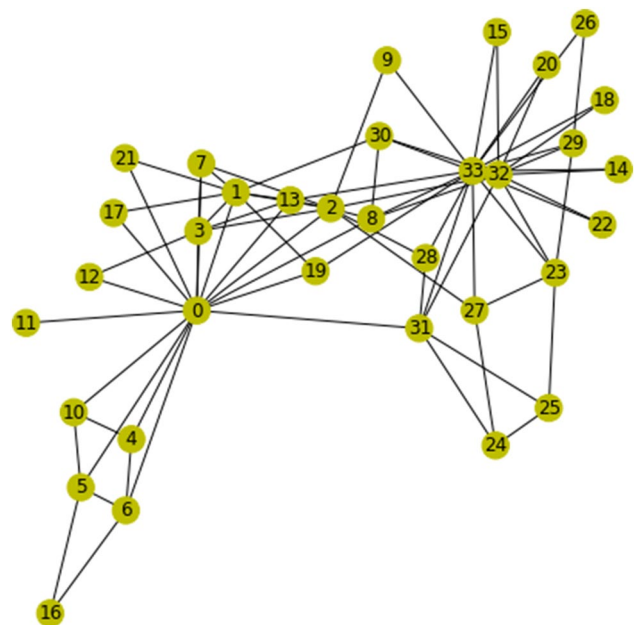


Fig. 1 Zachary’s karate club network

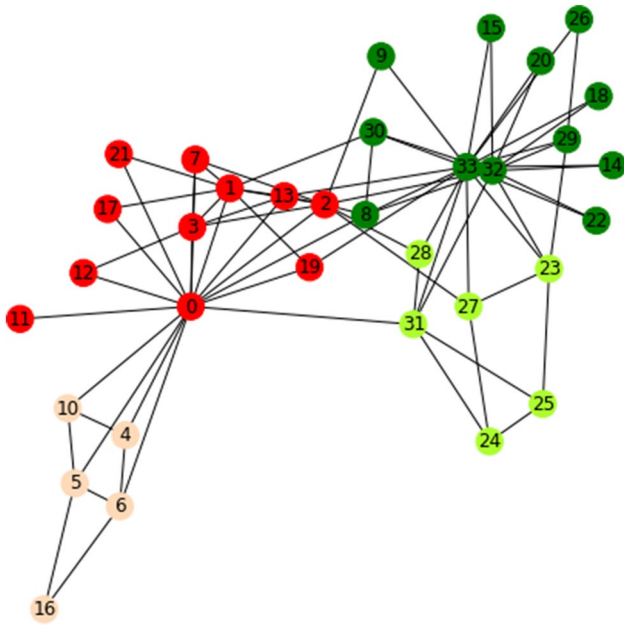


Fig. 2 Zachary's karate club network with 4-clusters

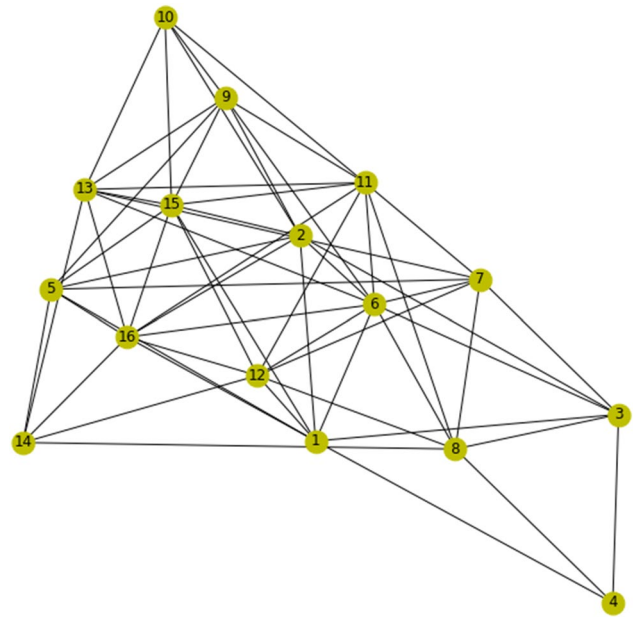


Fig. 4 Highland tribes network

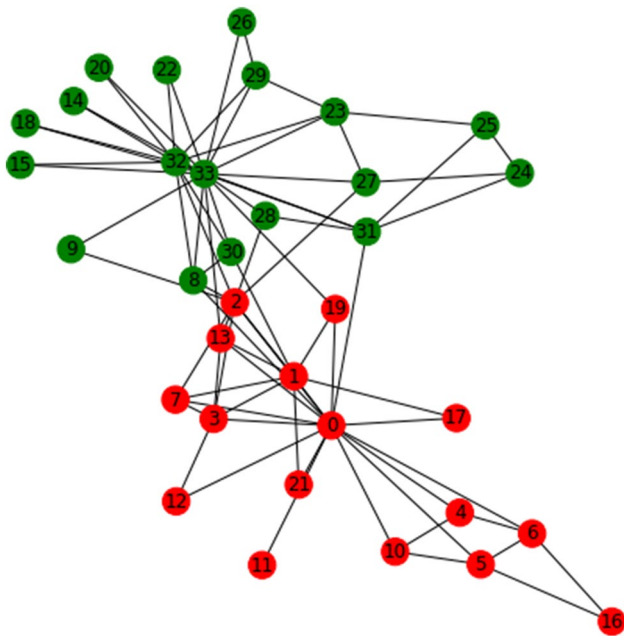


Fig. 3 Zachary's karate club network with 2-clusters

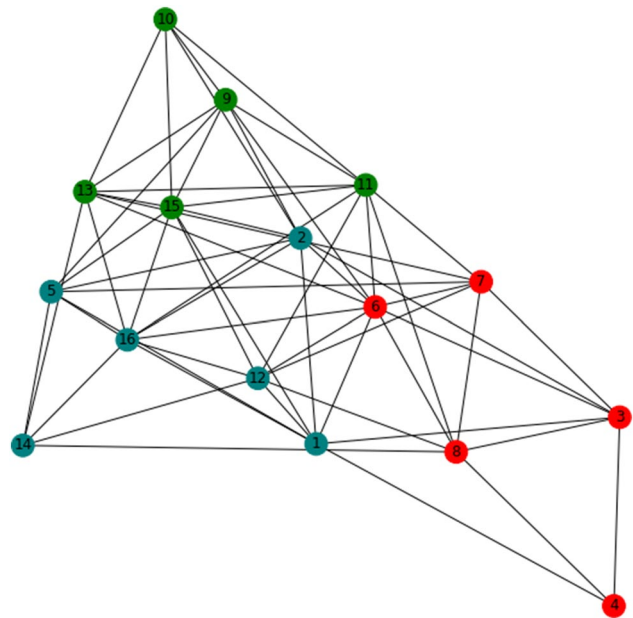


Fig. 5 Highland tribes network with 3-clusters

**Affiliation Networks**

Affiliation networks are the networks denoting the membership of actors in groups.

*American Revolution* (Figs. 8, 9): The network includes membership records of 136 people in 5 organizations from the pre-American Revolution era. The graph contains

well-known persons such as US activist Paul Revere. An edge between an individual and an agency suggests the individual was an organization member.

**Animal Networks**

Animal networks are networks of animal communications. They are the animal equivalent to human social networks.

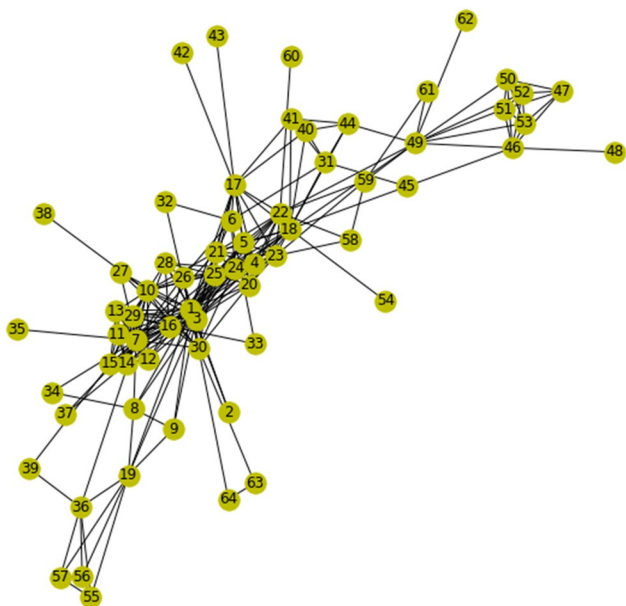


Fig. 6 2004 Madrid commuter train bombing terrorist network

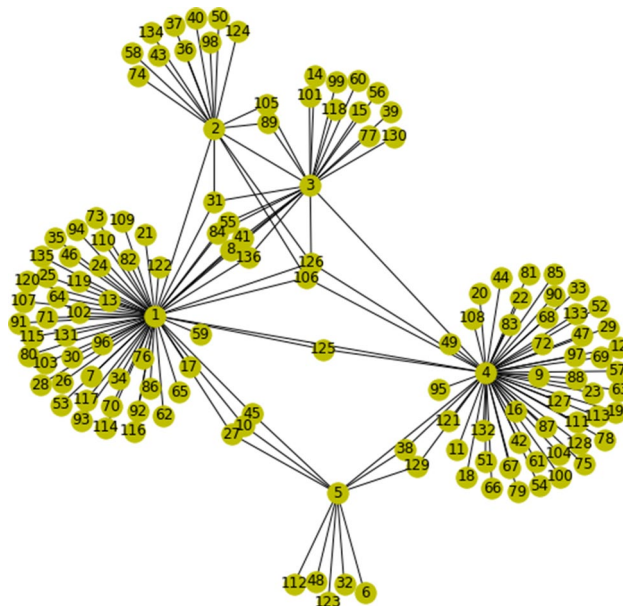


Fig. 8 American Revolution network

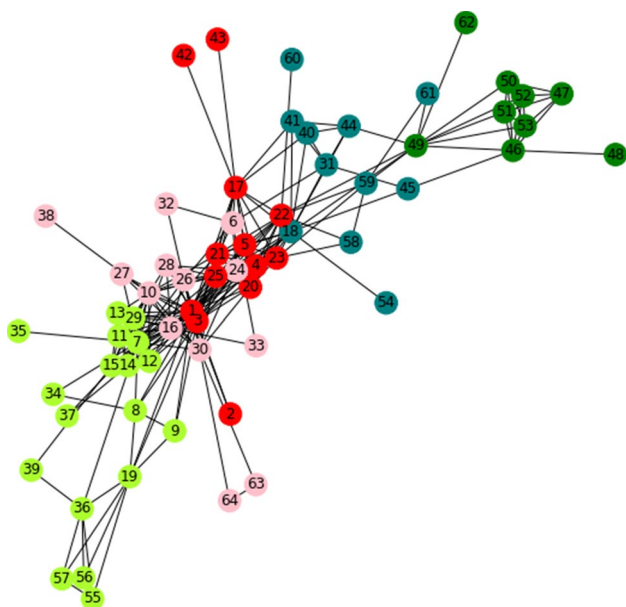


Fig. 7 2004 Madrid commuter train bombing terrorist network with 5-clusters

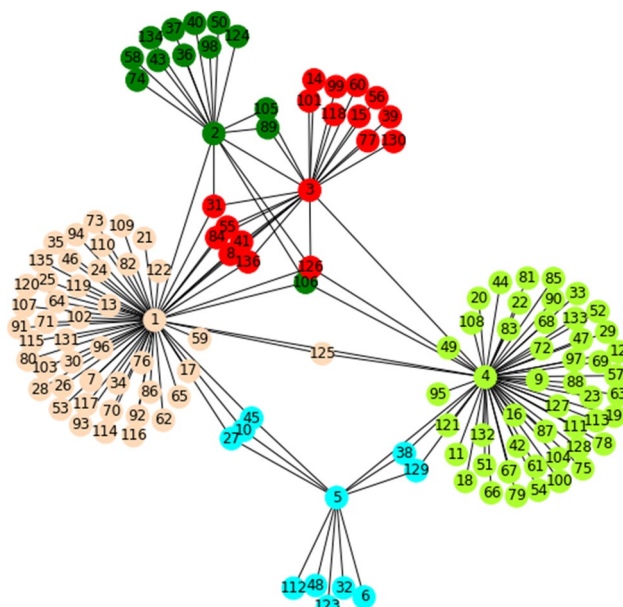


Fig. 9 American Revolution network with 5-clusters

*Dolphins* (Figs. 10, 11): The network includes a bottleneck dolphins social network. The nodes are the bottlenose dolphins (genus tursiops) of a group of bottlenose dolphins live off doubtful sound, a New Zealand fjord (spelled fiord in New Zealand). An edge suggests the interaction is regular. The dolphins were observed from 1994 through 2001.

*Zebra* (Figs. 12, 13): The network involves networks of animals from the group that communicates to each other.

They are the animal equivalent to human social networks. Note that website datasets such as Dogster are excluded here in the category of Social networks because humans generate the networks.

**Human Contact Networks**

Human communication networks are real networks of interaction between people, i.e., talking to each other, spending

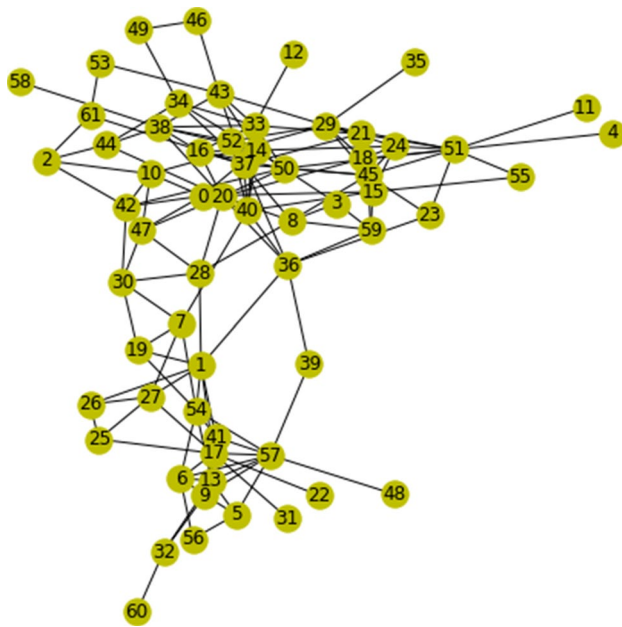


Fig. 10 Dolphins communication network

time together, or at least being physically close. More often than not, by giving out RFID labels to individuals with chips that monitor whether other individuals are nearby, these datasets are collected..

*Windsurfers* (Figs. 14, 15): This undirected network includes interpersonal communications during the Fall of 1986 between windsurfers in southern California. A node represents a windsurfer, and an edge between two windsurfers indicates an interpersonal interaction occurred.

### Miscellaneous Networks

Miscellaneous networks are any networks that do not fit into one of the other categories.

*Political books* (Figs. 16, 17): The network is of books published around the time of the 2004 presidential election about US politics and distributed through online bookseller Amazon.com. Edges between books reflect repeated co-purchases by the same purchasers of books.

*Les Misérables* (Figs. 18, 19): This undirected network includes character co-occurrences in the novel ‘Les Misérables’ by Victor Hugo. A node represents a character, and an edge between two nodes suggests that these two characters existed in the book’s same chapter. The weight of each relation reveals how much this kind of co-appearance happened.

### Results and Analysis

To evaluate the effectiveness of Community Detection Clustering via Gumbel Softmax, we have carried out numerous

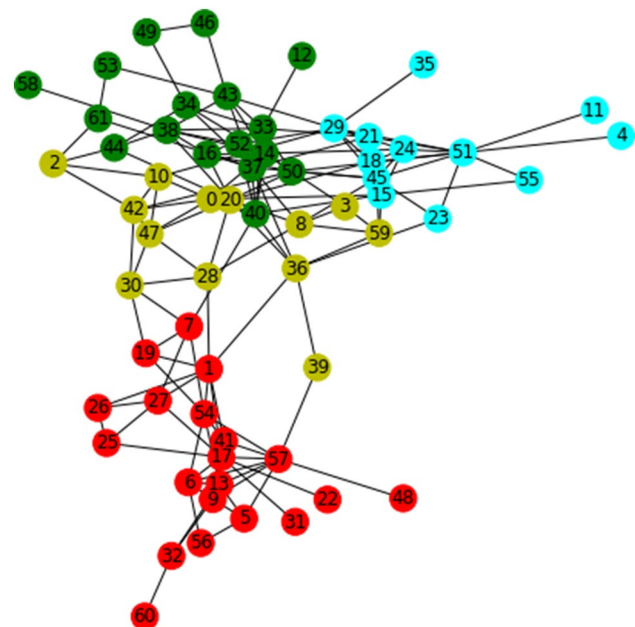


Fig. 11 Dolphins communication network with 4-clusters

tests on different types of network benchmarks. The same cluster nodes share the same color. On some of the datasets, we run the experiment multiple times to obtain the best modularity.

The main aim of our experiments is to find community and cluster unsupervised datasets. As the datasets used in our experiments are unlabelled to verify our algorithm’s efficiency, we use Zachary’s karate club network as one of the widely used benchmark datasets to analyze community detection algorithms. Initially, we treat the Zachary’s karate club network as an unlabelled dataset and run our method to obtain the clusters. Then, we compare our proposed method results with the results of algorithms such as Edge Motif Clustering (EDMOT) [10], Graph Embedding with Self Clustering (GEMSEC) [15], Scalable Community Detection (SCD) [12], Propagating Labels (PL) [13], Community Preserving Network Embedding (MNMF) [16], and Deep Autoencoder-like Nonnegative Matrix Factorization for Community Detection (DANMF) [17]. To complete the analysis, we also computed metrics such as Adjusted Randomized Index (ARI) [2], Normalized Mutual Information (NMI), Homogeneity (HOMO), Completeness (COMP), and V-measure (V-MES) [14]. To evaluate the identification of metadata groups by various algorithms, we took the best scores of the Zachary’s karate club network and different algorithms. Our method CDCGS outperforms EDMOT, GEMSEC, SCD, PL, MNMF, and DANMF algorithms with the metric results, and we have presented them in Table 1.

We do not know the specific communities with any of the other graph dataset networks as they are unlabelled, and

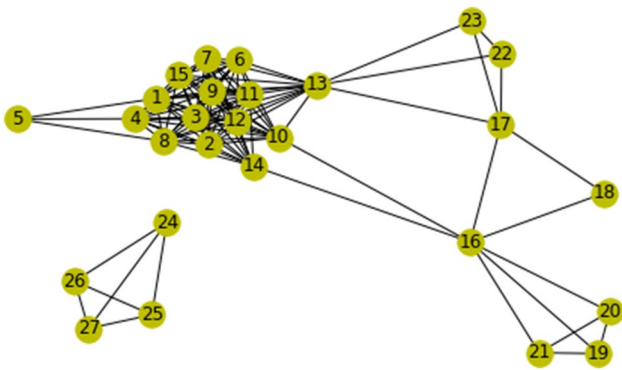


Fig. 12 Zebra communication network

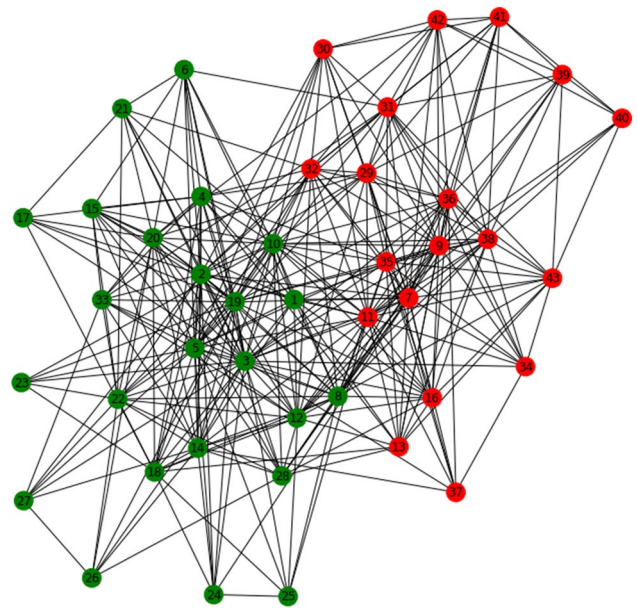


Fig. 15 Windsurfers network with 2-clusters

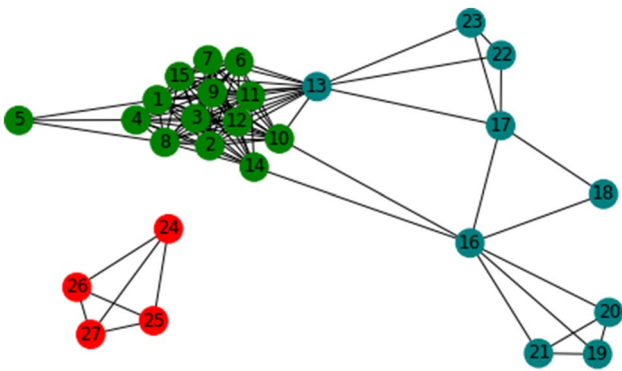


Fig. 13 Zebra communication network with 3-clusters

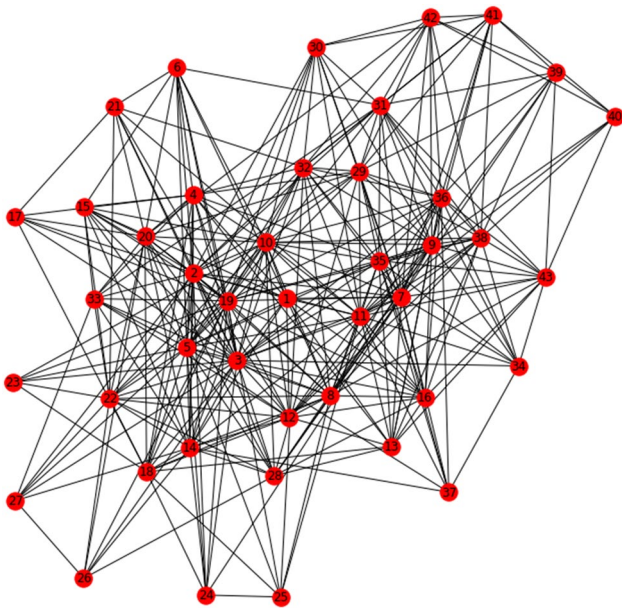


Fig. 14 Windsurfers network

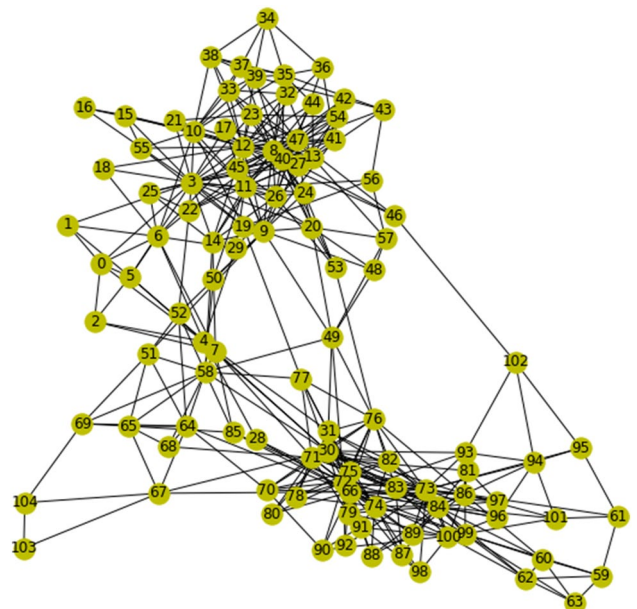


Fig. 16 Political books network

we cannot use the metric measures as they are not effective. Instead, we use the modularity metric [11], which does not require the actual group assignment and is based only on network structure. The modularity measure uses the fraction of edges that link vertices within the same group. Table 2 displays test results for the algorithms EDMOT, GEMSEC, SCD, PL, MNMF, DANMF, including CDCGS, on all of the dataset networks used. CDCGS outperforms algorithms that perform best on Zachary’s karate club, Dolphins, Train



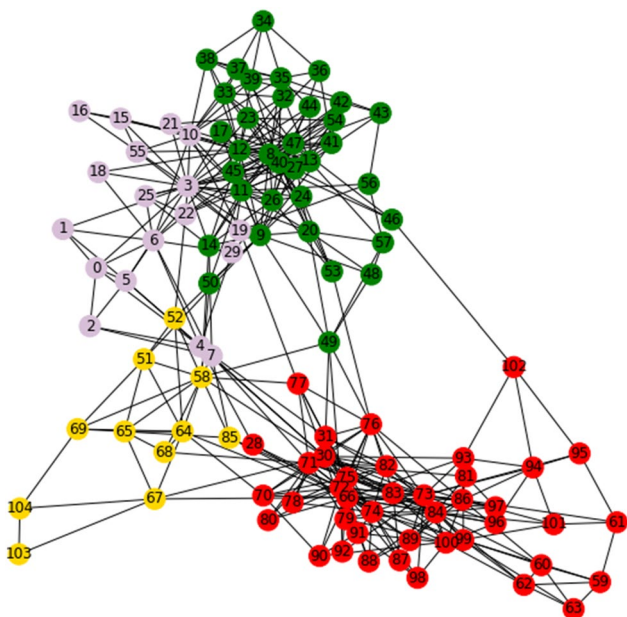


Fig. 17 Political books network with 4-clusters

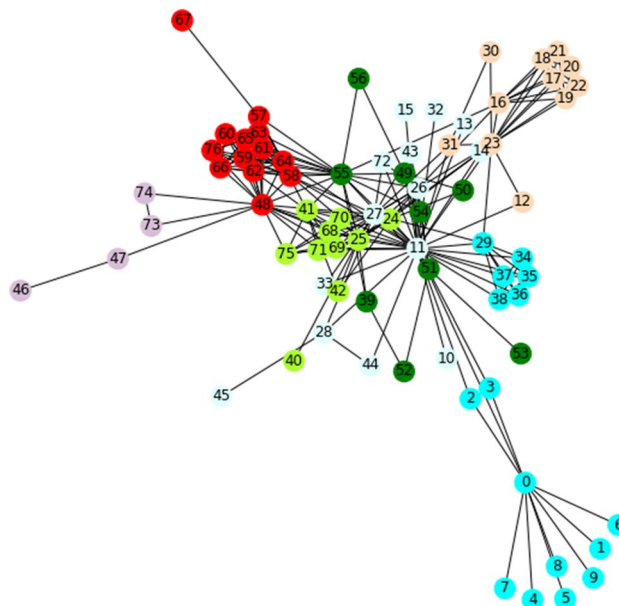


Fig. 19 Les Misérables network with 7-clusters

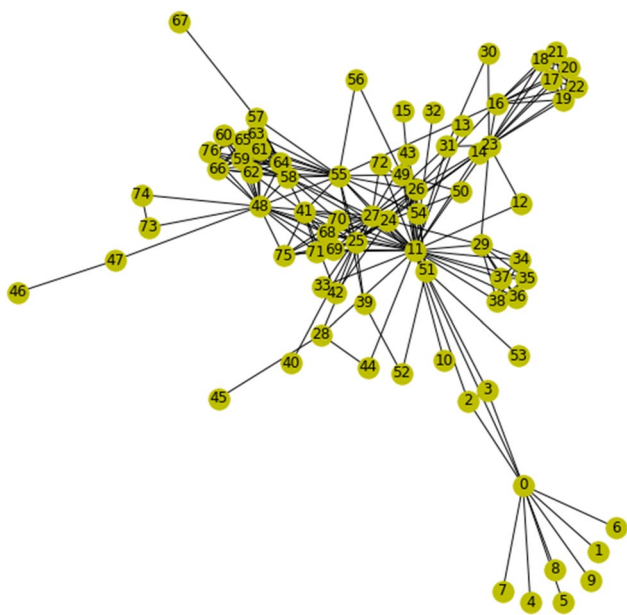


Fig. 18 Les Misérables network

bombing, American Revolution, and Windsurfers networks, with the best modularity being 0.4197, 0.5246, 0.4376, 0.5812, 0.2581 respectively. For Les Misérables, Highland tribes, Zebra, and Political books dataset networks, our method CDCGS stand second from the best results achieved algorithms. CDCGS performs almost near to the highest modularity achieved with the other dataset networks.

The best network that we can consider to analyze our method is the Zachary karate club network. As the Zachary network is labeled, we already know that graph node 0 and node 33 are the dominant two teachers nodes. Any other nodes in the network will join either node 0 teacher community or node 33 teacher community. The initial network before clustering is shown in Fig. 1. First, we run our approach to cluster the network into 2-clusters as ground truth community structure of the Zachary karate club has 2-clusters. When we cluster Zachary network into 2-clusters using CDCGS the resultant metric measures are ARI = 1, NMI = 1, HOMO = 1, COMP = 1, V-MES = 1 and the modularity measure = 0.371. The value of ARI, NMI, HOMO, COMP, and V-MES shows that the proposed algorithm clusters all the nodes to a correct teacher group as per the target label. Figure 3 represents Zachary’s karate club network with 2-clusters resulted from our proposed method.

Higher modularity value implies a better assignment to the cluster. To obtain the higher modularity, we identified 4-clusters Zachary karate club network that gives us the modularity measure of 0.4197 (Fig. 2). Here, assuming the students joining node 0 as cluster group 1 and node 33 as a cluster group 2, we still see that node 0 and node 33 are associated with more nodes and nodes such as 4, 5, 6, 10, 16 (cluster group 3) and 23, 24, 25, 27, 28, 31 (cluster group 4) shows that the students are not interested in joining the cluster group 1 or group 2. With the most number of the students associated with the cluster group 1 or cluster group 2 and the higher modularity compared to other recently proposed methods such as EDMOT, GEMSEC, SCD, MNMF, and DANMF shows that our method outperforms and shows the

**Table 1** Zachary karate club network metrics for the different algorithms and our method (CDCGS)

Metric	EDMOT	GEMSEC	SCD	PL	MNMF	DANMF	CDCGS
ARI	0.5131	0.5068	0.2466	0.4714	0.3887	0.4782	<b>0.5414</b>
NMI	0.6021	0.4958	0.4760	0.5282	0.6765	0.5477	<b>0.6873</b>
HOMO	0.8535	0.6474	1	0.6905	0.7496	0.7435	<b>1</b>
COMP	0.4468	0.4651	0.3123	0.4277	0.3762	0.4335	<b>0.5235</b>
V-MES	0.6021	0.4441	0.4760	0.5282	0.5010	0.5477	<b>0.6872</b>

Bold values indicate better results than other clustering algorithms

ARI Adjusted Randomized Index, NMI Normalized Mutual Information, HOMO Homogeneity, COMP-Completeness, V-MES V-measure

**Table 2** Best modularity results obtained by our method (CDCGS) and other algorithms for the real world networks

Dataset	EDMOT	GEMSEC	SCD	PL	MNMF	DANMF	CDCGS
ZACHARY KARATE	0.4188	0.3495	0.3016	0.3990	0.3717	0.3714	<b>0.4197</b>
DOLPHINS	0.5195	0.2362	0.4142	0.4512	0.4171	0.4977	<b>0.5246</b>
LES MISÉRABLES	<b>0.5582</b>	0.5299	0.4485	0.5367	0.4762	0.4391	0.5389
TRAIN BOMBING	0.3047	0.3607	0.2499	0.4295	0.4140	0.3505	<b>0.4376</b>
HIGHLAND TRIBES	<b>0.1689</b>	0.1562	0.0677	0	0.1432	0.1588	0.1612
AMERICAN REVOLN	0.5788	0.5567	0.5461	0.5170	0.3454	0.1894	<b>0.5812</b>
ZEBRA	0.2160	0.2564	0.1963	0.2606	0.2655	<b>0.2722</b>	0.2702
WINDSURFERS	0.1443	0.2545	0.1519	0.2581	0.2545	0.2468	<b>0.2581</b>
POLITICAL BOOKS	<b>0.5267</b>	0.4938	0.4294	0.4873	0.4519	0.4777	0.5208

Bold values indicate better results than other clustering algorithms

EDMOT Edge Motif Clustering, GEMSEC Graph Embedding with Self Clustering, SCD Scalable Community Detection, PL Propagating Labels, MNMF Community Preserving Network Embedding, DANMF Deep Autoencoder-like Nonnegative Matrix Factorization for Community Detection, CDCGS Community Detection Clustering via Gumbel Softmax

new way to cluster the community datasets. The proposed method works similarly with the other graph datasets.

Figures 1, 4, 6, 8, 10, 12, 14, 16 and 18 shows the dataset graph before clustering. Figures 2, 3, 5, 7, 9, 11, 13, 15, 17 and 19 shows the beauty of the proposed method after community detection clustering.

## Conclusion and Future Enhancements

We introduced the community detection clustering via the gumbel softmax strategy for the different types of graph network datasets. The experimental findings demonstrate the effectiveness by choosing appropriate parameter values and evaluating the consistency of the resultant clustering. The method currently available is just as diverse as the graph clustering applications. Thus, the suggested algorithm can be regarded as a feasible and effective algorithm for seeking optimal community detection problem solutions. The research completed, however, is on a dataset of unweighted and undirected graphs. We are currently experimenting with applying the principle of clustering for Graphs on a weighted and directed graph dataset.

**Source Code Available** The code for this research is available on github at: [https://github.com/deepakacharyab/community\\_detection\\_gumbel\\_softmax](https://github.com/deepakacharyab/community_detection_gumbel_softmax)

## Compliance with Ethical Standards

**Conflict of interest** On behalf of all authors, the corresponding author states that there is no conflict of interest.

## References

1. Acharya DB, Zhang H. Feature selection and extraction for graph neural networks. In: Proceedings of the 2020 ACM southeast conference, ACM SE '20, page 252–255, New York, NY, USA. Association for Computing Machinery; 2020.
2. Buitinck L, Louppe G, Blondel M, Pedregosa F, Mueller A, Grisel O, Niculae V, Prettenhofer P, Gramfort A, Grobler J, Layton R, VanderPlas J, Joly A, Holt B, Varoquaux G. API design for machine learning software: experiences from the scikit-learn project. In: ECML PKDD workshop: languages for data mining and machine learning; 2013. p. 108–122.
3. Cuijuan W, Wenzhong T, Bo S, Jing F, Yanyang W. Review on community detection algorithms in social networks. In: 2015 IEEE International conference on progress in informatics and computing (PIC); 2015. p. 551–555.
4. Girvan M, Newman ME. Community structure in social and biological networks. Proc Natl Acad Sci. 2002;99(12):7821–6.

5. Gumbel EJ. Statistical theory of extreme values and some practical applications. NBS Appl Math Ser. 1954;33.
6. Hastings MB. Community detection as an inference problem. *Phys. Rev. E*. 2006;74(3):035102.
7. Jang E, Gu S, Poole B. Categorical reparameterization with Gumbel-softmax. *ICLR: Toulon*; 2017.
8. Javed MA, Younis MS, Latif S, Qadir J, Baig A. Community detection in networks: a multidisciplinary review. *J Netw Comput Appl*. 2018;108:87–111.
9. Kunegis Jérôme. KONECT—The Koblenz Network Collection. In: *Proceedings of the international conference on world wide web companion*; 2013. p. 1343–1350.
10. Li P-Z, Huang L, Wang C-D, Lai J-H. Edmot: An edge enhancement approach for motif-aware community detection. In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining*; 2019. p. 479–487.
11. Newman MEJ, Girvan M. Finding and evaluating community structure in networks. *Phys Rev E*. 2004;69:026113.
12. Prat-Pérez A, Dominguez-Sal D, Larriba-Pey J-L. High quality, scalable and parallel community detection for large real graphs. In: *Proceedings of the 23rd international conference on world wide web, WWW '14*, page 225–236, New York, NY, USA. Association for Computing Machinery; 2014.
13. Raghavan UN, Albert R, Kumara S. Near linear time algorithm to detect community structures in large-scale networks. *Phys Rev E*. 2007;76(3):036106.
14. Rosenberg A, Hirschberg J. V-measure: A conditional entropy-based external cluster evaluation measure. In: *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*; 2007. p. 410–420.
15. Rozemberczki B, Davies R, Sarkar R, Sutton C. Gemsec: Graph embedding with self clustering. In: *Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining*; 2019. p. 65–72.
16. Wang X, Cui P, Wang J, Pei J, Zhu W, Yang S. Community preserving network embedding. In: *Thirty-first AAAI conference on artificial intelligence*. 2017.
17. Ye F, Chen C, Zheng Z. Deep autoencoder-like nonnegative matrix factorization for community detection. In: *Proceedings of the 27th ACM international conference on information and knowledge management*; 2018. p. 1393–1402.

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