

COMMUNITY PRESERVING NODE EMBEDDING

Anonymous authors

Paper under double-blind review

ABSTRACT

Detecting communities or the modular structure of real-life networks (e.g. a social network or a product purchase network) is an important task because the way a network functions is often determined by its communities.

The traditional approaches to community detection involve modularity-based approaches, which generally speaking, construct partitions based on heuristics that seek to maximize the ratio of the edges within the partitions to those between them. Node embedding approaches, which represent each node in a graph as a real-valued vector, transform the problem of community detection in a graph to that of clustering a set of vectors. Existing node embedding approaches are primarily based on first initiating uniform random walks from each node to construct a context of a node and then seeks to make the vector representation of the node close to its context. However, standard node embedding approaches do not directly take into account the community structure of a network while constructing the context around each node. To alleviate this, we explore two different threads of work. First, we investigate the use of biased random walks (specifically, maximum entropy based walks) to obtain more centrality preserving embedding of nodes, which we hypothesize may lead to more effective clusters in the embedded space. Second, we propose a community structure aware node embedding approach where we incorporate modularity-based partitioning heuristics into the objective function of node embedding. We demonstrate that our proposed approach for community detection outperforms a number of modularity-based baselines as well as K-means on a standard node-embedded vector space (specifically, node2vec) on a wide range of real-life networks of different sizes and densities.

INTRODUCTION

Partitioning a network (graph) into communities usually leads to better analyzing the functionality of the network and is of immense practical interest for real-world networks, because such communities potentially represent organizational units in social networks, scientific disciplines in authorship-citation academic publications networks, or functional units in biological networks (e.g. protein-protein interactions) (Girvan & Newman, 2002; Newman & Girvan, 2004; Waltman & Van Eck, 2013). A *network community* represents a set of nodes with a relatively dense set of connections between its members and relatively sparse connections between its member nodes and the ones outside the community.

Traditional approaches of community detection incrementally construct a community (set of nodes) by employ an objective function that seeks to maximize its internal connectivity and minimize the number of external edges (Newman & Girvan, 2004; Newman, 2006; Blondel et al., 2008; Prat-Pérez et al., 2014). Graph representation learning approaches such as (Perozzi et al., 2014; Grover & Leskovec, 2016) represent each node of a graph as a real-valued vector seeking to preserve the correlation between the topological properties of the discrete graph with the distance measures in the embedded metric space. For example, the vectors corresponding to a pair of nodes in the embedded space is usually close (low distance or high inner product similarity) if it is likely to visit a node of the pair with a random walk started at the other one.

However, a major limitation of the random walk based node representation learning approach is that a random walk may span across the community from which it started with, which eventually could lead to representing nodes from different communities in close proximity in the embedding

space. This in turn can may not result in effective community detection on application of a standard clustering algorithm, e.g. K-means, in the space of embedded node vectors.

Ideally speaking, for effective community detection with a clustering algorithm operating on the embedded space of node vectors, a node embedding algorithm should preserve the community structure from the discrete space of the sets of nodes to the continuous space of real-valued vectors as perceived with the conventional definitions of the distance metric (e.g. l_2 distance) and the inner product between pairs of vectors denoting the similarity between them. In other words, a central (hub) node of a community in the discrete graph representation should be transformed in the embedded space in such a way so that it contains other vectors, corresponding to the nodes of the other members in the community, in its close neighborhood. In our study, we investigate two methods to achieve such a transformation.

Our Contributions First, in contrast to the uniform random walk (URW) based contextualization of nodes in standard node embedding approaches, such as node2vec (Grover & Leskovec, 2016) and DeepWalk (Perozzi et al., 2014), we investigate a maximum-entropy based biased random walk (MERW) Sinatra et al. (2011), where in contrast to URW, the transition probabilities are non-local, i.e., they depend on the structure of the entire graph.

Alternately, in our second proposed approach, we investigate if traditional approaches to community detection that operate on a discrete graph (adjacency matrix), e.g. modularity-heuristic (Clauset et al., 2004) or InfoMap (Rosvall & Bergstrom, 2008), can be useful to contextualize a node for the purpose of obtaining its embedded representation. In other words, while training a classifier for a node vector that learns to predict its context, we favour those cases where the context nodes are a part of the same community as that of the current node, as predicted by a modularity-based heuristic).

We also investigate a combination of the two different community aware embedding approaches, i.e. employing MERW to first contextualize the nodes and then using the weighted training based on the modularity heuristic.

The rest of the paper is organized as follows. We first review the literature on community detection and node embedding. We then describe the details about the MERW-based node embedding and community-structure aware node embedding. Next, we describe the setup of our experiments, which is followed by a presentation and analysis of the results. Finally, we conclude the paper with directions for future work.

BACKGROUND AND RELATED WORK

In this section, we provide a brief overview of the community detection and node representation learning approaches.

COMBINATORIAL APPROACHES TO COMMUNITY DETECTION

In this section, we review a number of combinatorial approaches to community detection. Each combinatorial approach has the common underlying principle of first constructing an initial partition of an input graph into a set of sub-graphs (communities) and then refining the partition at every iterative step. Among a number of possible ways to modify a current partition, the one that maximizes a global objective function is chosen. The global objective, in turn, is computed by aggregating the local objectives over and across the constituent sub-graphs.

Clauset et al. (2004) defines *modularity* as an intrinsic measure of how effectively, with respect to its topology, a graph (network) is partitioned into a given set of communities. More formally, given a partition of a graph $G = (V, E)$ into p communities, i.e. given an assigned community (label) $c_v \in \{1, \dots, p\}$ for each node $v \in V$, the modularity, Q is defined as the expected ratio of the number of intra-community edges to the total number of edges, the expectation being computed with respect to the random case of assigning the nodes to arbitrary communities. More specifically,

$$Q = \frac{1}{2|E|} \sum_{vw} (A_{vw} - \frac{k_v k_w}{2|E|} \mathbb{I}(c_v = c_w)), \quad (1)$$

where A_{vw} denotes the adjacency relation between nodes v and w , i.e. $A_{vw} = 1$ if $(v, w) \in E$; k_v denotes the number of edges incident on a node v ; $\mathbb{I}(c_v, c_w)$ indicates if nodes v and w are a part of the same community. A high value of Q in Equation 1 represents a substantial deviation of the fraction of intra-community edges to the total number of edges from what one would expect for a randomized network, and Clauset et al. (2004) suggests that a value above 0.3 is often a good indicator of significant community structure in a network.

The ‘CNM’ (Clauset Newman Moore) algorithm (Newman & Girvan, 2004) proposes a greedy approach that seeks to optimise the modularity score (Equation 1). Concretely speaking, it starts with an initial state of node being assigned to a distinct singleton community, seeking to refine the current assignment at every iteration by merging a pair of communities that yields the maximum improvement of the modularity score. The algorithm proceeds until it is impossible to find a pair of communities which if merged yields an improvement in the modularity score.

The ‘Louvain’ or the ‘Multilevel’ algorithm (Blondel et al., 2008) involves first greedily assigning nodes to communities, favoring local optimizations of modularity, and then repeating the algorithm on a coarser network constructed from the communities found in the first step. These two steps are repeated until no further modularity increasing reassignments are found.

‘SCDA’ (Scalable Community Detection Algorithm) (Prat-Pérez et al., 2014) detects disjoint communities in networks by maximizing WCC, a recently proposed community metric Prat-Pérez et al. (2012) based on triangle structures within a community. SCD implements a two-phase procedure that combines different strategies. In the first phase, SCD uses the clustering coefficient as an heuristic to obtain a preliminary partition of the graph. In the second phase, SCD refines the initial partition by moving vertices between communities as long as the WCC of the communities increase.

Jiang & Singh (2010) proposed a scalable algorithm - ‘SPICi’ (‘Speed and Performance In Clustering’ and pronounced as ‘spicy’), which constructs communities of nodes by first greedily starting from local seed sets of nodes with high degrees, and then adding those nodes to a cluster that maximize a two-fold objective of the density and the adjacency of nodes within the cluster. The underlying principle of SPICi is similar to that of ‘DPPlus’ (Altaf-Ul-Amin et al., 2006), the key differences being SPICi exploits a simpler cluster expansion approach, uses a different seed selection criterion and incorporates interaction confidences.

Newman (2006) proposed ‘LEADE’ (Leading Eigenvector) applies a spectral decomposition of the modularity matrix M , defined as

$$M_{vw} = A_{vw} - \frac{k_v k_w}{2|E|}. \quad (2)$$

The leading eigenvector of the modularity matrix is used to split the graph into two sub-graphs so as to maximize modularity improvement. The process is then recursively applied on each sub-graph until the modularity value cannot be improved further.

‘LPA’ (Label Propagation Algorithm) (Raghavan et al., 2007) relies on the assumption that each node of a network is assigned to the same community as the majority of its neighbours. The algorithm starts with initialising a distinct label (community) for each node in the network. Each node, visited in a random order, then takes the label of the majority of its neighbours. The iteration stops when the label assignments cannot be changed further.

Rosvall & Bergstrom (2008) proposed the ‘InfoMap’ algorithm, which relies on finding the optimal encoding of a network based on maximizing the information needed to compress the movement of a random walker across communities on the one hand, whereas minimizing the code length to represent this information. The algorithm makes uses of the core idea that random walks initiated from a node which is central to a community is less likely to visit a node of a different community. Huffman encoding of such nodes, hence, are likely to be shorter.

The ‘WalkTrap’ algorithm (Pons & Latapy, 2005) is a hierarchical agglomerating clustering (HAC) algorithm using an idea similar to InfoMap that short length random walks tend to visit only the nodes within a single community. The distance metric that the algorithm uses for the purpose of HAC between two sets of nodes is the distance between the probability distributions of nodes visited by random walks initiated from member nodes of the two sets.

Different from the existing work in combinatorial approaches to community detection, in our work, we propose a framework to integrate a combinatorial approach within the framework of an embedding approach (specifically, node2vec).

EMBEDDING APPROACHES

In contrast to the combinatorial approaches which directly work on the discrete space (vertices and edges) of a graph, $G = (V, E)$, an embedding approach transforms each node of a graph, u , into a real-valued vector, \mathbf{u} , seeking to preserve the topological structure of the nodes. Formally,

$$\theta : u \mapsto \mathbf{u} \in \mathbb{R}^d, \forall u \in V. \quad (3)$$

The transformation function θ is learned with the help of noise contrastive estimation, i.e., the objective is to make the similarity (inner product) between vectors for nodes u and v higher if v lies in the neighborhood of u , and to be of a value small if v does not belong to the neighborhood of u (e.g. v being a randomly sampled node from the graph). Formally,

$$J(\theta) = \sum_u \sum_{v \in \mathcal{N}(u)} P(y = 1 | \mathbf{u}, \mathbf{v}) - \sum_u \sum_{\bar{v} \notin \mathcal{N}(u)} P(y = 1 | \mathbf{u}, \bar{\mathbf{v}}), \quad (4)$$

where y denotes a binary response variable to train the likelihood function, where $\mathcal{N}(u)$ denotes the neighborhood of node u , and the negative component in the likelihood function refers to the randomly sampled noise.

Popular approaches to learn the transformation function of Equation 3 includes node2vec (Grover & Leskovec, 2016) and DeepWalk (Perozzi et al., 2014), which differ in the way the neighborhood function, $\mathcal{N}(u)$, is defined. While DeepWalk uses a uniform random walk to constitute the neighborhood or context of a node, node2vec uses a biased random walk (with a relative importance to depth-first or breadth-first traversals).

A transformation of the nodes as real-valued vectors then allows the application of relatively simple (but effective) clustering approaches, such as K-means, to partition the embedding space of nodes into distinct clusters. This is because in contrast to the discrete space, the vector space is equipped with a metric function which allows to compute distance (or equivalently similarity) between *any* pair of nodes (as opposed to the discrete case).

Cavallari et al. (2017) proposed an expectation-maximization (EM) based approach to iteratively refine a current community assignment (initialized randomly) using node embeddings. The objective was to ensure that the embedded vectors of each community fits a Gaussian mixture model, or in other words, the embedded space results in relatively disjoint convex clusters. In contrast to (Cavallari et al., 2017), our method does not involve a feedback-based EM step.

Wang et al. (2016) proposed to include an additional term in the objective of the transformation function (Equation 3) corresponding to the *second order similarity* between the neighborhoods of two nodes. Different to (Wang et al., 2016), which seeks to obtain a general purpose embedding of graphs, we rather focus only on the community detection problem.

MAXIMAL-ENTROPY BIASED RANDOM WALK

Let $P \in \mathbb{R}^{|V| \times |V|}$ denote the stochastic transition matrix of a graph $G = (V, E)$, where P_{uv} denotes the probability of visiting node v in sequence after visiting node u . In a standard uniform random walk (URW), this probability is given by

$$P_{uv} = \frac{A_{uv}}{k_u}, \quad k_u = |\{w : (u, w) \in E\}|, \quad (5)$$

where k_u denotes the degree of node u . In other words, Equation 5 indicates that there is a equal likelihood of choosing a node v as the next node in sequence from the neighbors of node u .

Maximal-entropy random walk (MERW) is characterized by a stochastic matrix that maximises entropy of a set of paths (node sequences) with a given length and end-points (Ochab & Burda, 2013). It leads to the following stochastic matrix.

$$P_{uv} = \frac{A_{uv} \psi_v}{\lambda \psi_u}, \quad (6)$$

where λ denotes the largest eigenvalue of the adjacency matrix A , and ψ_v and ψ_u refer to the v^{th} and the u^{th} components of the corresponding eigenvector. Parry (1964) applied the Frobenius-Perron theorem to prove that the probability of visiting a node u_n after n time steps starting from node u_1 depends only on the number of steps and the two ending points, but is independent of the intermediate nodes, i.e.

$$P(u_1, \dots, u_n) = \prod_{i=1}^{n-1} P_{u_i, u_{i+1}} = \frac{1}{\lambda^n} \frac{\psi_{u_1}}{\psi_{u_n}}. \quad (7)$$

Consequently, the choice of the next node to visit in MERW is based on uniformly selecting the node from alternative paths of a given length and end-points.

Delvenne & Libert (2011) shows that the stationary distribution attained by MERW better preserves centrality than URW, thus resulting in random walks that tend to be more local as shown in (Burda et al., 2009). In the context of our problem, MERW based random walk initiated from a node of a community is more likely to remain within the confinements of the same community, as compared to URW.

Standard node embedding approaches, such as node2vec, uses URW to construct the set of contexts for a node for the purpose of learning its representation. We hypothesize that replacing the URW based neighborhood function to a MERW one results in less likelihood of including a node v in the neighborhood of u , i.e. $\mathcal{N}(u)$. This results in a low likelihood of including the term $P(y = 1 | \mathbf{u}, \mathbf{v})$ of Equation 4, which corresponds to associating nodes across two different communities, as a positive example while training node representations.

MODULARITY BASED NODE EMBEDDING

In this section, we describe a two-step approach to node embedding that is likely to preserve the community structure of the discrete space of an input graph in the output embedded space as well. The first step involves applying a combinatorial community detection algorithm that operates on the discrete input space to obtain an optimal partition, as per the objective function of the combinatorial approach, e.g. modularity (Clauset et al., 2004) or InfoMap (Rosvall & Bergstrom, 2008). Formally,

$$\mathcal{C} : G = (V, E) \mapsto \{V_i\}_{i=1}^p, \text{ s.t. } \cup_{i=1}^p V_i = V \quad (8)$$

EXPERIMENT SETUP

DATASETS

Real-world networks with ground-truth communities The experiments are performed over three small scale standard networks, viz., Zacharys karate club network, bottlenose dolphin network, and American college football network. Along with We have also tested the experiments on three real world networks viz. Amazon, Youtube and DBLP Yang & Leskovec (2015); Harenberg et al. (2014); Leskovec & Krevl (2014). These networks are undirected and unweighted and they are selected from different application domains. The overview of these networks are presented Table 1.

Amazon¹ is an online commercial network for purchasing products. Here nodes represent products and an edge exists between two products, if they are frequently purchased together. Each product (i.e. node) belongs to one or more product categories. Each ground-truth community is defined using hierarchically nested product categories that share a common function Yang & Leskovec (2015).

Youtube is a website to share videos and considered as a social network. Each user in the Youtube network is considered as a node and the friendship between two users is denoted as edge. Moreover, an user can create a group where other Youtube users can be a member through their friendship. These user created groups are considered as ground-truth communities Yang & Leskovec (2015).

DBLP is a bibliographic network of Computer Science publications. Here nodes represent authors and an edge between two nodes represent co-authorship. Ground-truth communities are defined as sets of authors who published in the same journal or conference Yang & Leskovec (2015).

¹www.amazon.com

Table 1: Overview of the Real World Networks

Data Sets	NV ¹	NE	MID	NCM	MaxCS	MinCS	MaxDeg	AvgDeg	AvgCS
KARATE	34	78	0.2288	2	18	16	17	4.588	17
DOLPHIN	62	159	0.1278	2	42	20	12	5.129	31
FOOTBALL	115	613	0.1101	12	13	5	12	10.66	9.583
AMAZON	16716	48739	0.0126	5000	328	3	51	5.831	13.49
YOUTUBE	39481	224235	0.0036	5000	2217	2	1575	11.26	14.59
DBLP	93432	335520	0.0011	5000	7556	6	213	7.182	22.45

²NV: number of vertices, NE: number of edges, NCM: number of communities, MID: Minimum Internal Density, MaxCS: Maximum Community Size, MinCS: Minimum Community Size, MaxDeg: Maximum Degree, AvgDeg: Average Degree, AvgCS: Average Community Size.

Table 2: Overview of the Artificial Networks

Name	n^3	m	NCM	k_{avg}	k_{max}	c_{min}	c_{max}	AvgCS
LFR2000	2000	3893	198	4	15	5	20	10.11
LFR4000	4000	7712	380	4	15	5	20	10.526
LFR6000	6000	12391	360	4	20	10	40	16.666
LFR8000	8000	16750	504	4	20	10	40	15.873
LFR10000	10000	17457	274	4	50	20	80	36.496
LFR12000	12000	18983	339	4	50	20	80	35.398

³ n : number of vertices, m : number of edges, k_{avg} : average degree, k_{max} : maximum degree, c_{min} : minimum community size, c_{max} : maximum community size, NCM: number of communities, AvgCS: Average Community Size.

The networks described above have several connected components and each connected component consisting of more than 3 nodes are considered as a separate ground-truth community. Leskovec, et al. observed that the average goodness metric of the top k communities first remain flat with increasing k , but then after approximately 5000 communities, degrades rapidly Yang & Leskovec (2015). Therefore they have implemented some community detection algorithms using different goodness metrics on the top 5000 communities of some of the networks described above. Eventually, they obtained nice results in terms of finding communities in those networks. Following the same idea we have used only the top 5000 ground-truth communities of each of these networks in the experimental evaluation.

Artificial networks with ground-truth communities Furthermore, we use the Lancichinetti-Fortunato-Radicchi (LFR) networks Lancichinetti et al. (2008) with ground-truth to study the behavior of a proposed community detection algorithm and to compare the performance across various competitive algorithms. The LFR model involve with a set of parameters which controls the network topology. In this model, degree distribution and community size distribution follow power laws with exponents γ and β , respectively. Furthermore, we can also specify the other parameters such as number of vertices n , average degree k_{avg} , maximum degree k_{max} , minimum community size c_{min} , maximum community size c_{max} , and mixing parameter μ . We vary these parameters depending on our experimental needs. The critical parameter is the mixing parameter μ , which indicates the proportion of relationships a node shares with other communities. Six artificial networks are produced for experimental evaluations using the following parameter setting, $\gamma = -2, \beta = -1, \mu = 0.01$ as mentioned by Lancichinetti et al. Lancichinetti et al. (2008). Table 2 provides the details of the other parameters to produce these artificial networks. The results presented on these networks are the average of 100 runs to reduce the effect of random assumptions.

EVALUATION METRICS

Omega Index. The Omega Index Collins & Dent (1988) is an Adjusted Rand Index (ARI) Hubert & Arabie (1985) generalization applicable to overlapping clusters. It is based on counting the number of pairs of elements occurring in exactly the same number of clusters as in the number of categories

and adjusted to the expected number of such pairs. Formally, given the ground-truth clustering C' consisting of categories $c'_i \in C'$ and formed clusters $c_i \in C$:

$$\Omega(C', C) = \frac{\text{Obs}(C', C) - \text{Exp}(C', C)}{1 - \text{Exp}(C', C)}$$

The observed agreement is:

$$\text{Obs}(C', C) = \sum_{j=0}^{\min(J', J)} \frac{A_j}{P},$$

where $J'(J)$ is the maximal number of categories (clusters) in which a pair of elements occurred, A_j is the number of pairs of elements occurring in exactly j categories and exactly j clusters, and $P = N(N-1)/2$ is the total number of pairs given a total of N elements (nodes of the network being clustered). The expected agreement is:

$$\text{Exp}(C', C) = \sum_{j=0}^{\min(J', J)} \frac{P'_j P_j}{P^2}$$

where $P'_j(P_j)$ is the total number of pairs of elements assigned to exactly j categories (clusters).

Mean F1 Score. The Average F1 score ($F1a$) is a commonly used metric to measure the accuracy of clustering algorithms Yang & Leskovec (2013; 2015); Prat-Pérez et al. (2014). $F1a$ is defined as the average of the weighted F1 scores of a) the best matching ground-truth clusters to the formed clusters and b) the best matching formed clusters to the ground-truth clusters. Formally, given the ground-truth clustering C' consisting of clusters $c'_i \in C'$ (called categories) and clusters $c_i \in C$ formed by the evaluating clustering algorithm:

$$F1a(C', C) = \frac{1}{2}(F_{C', C} + F_{C, C'}),$$

where

$$F_{X, Y} = \frac{1}{|X|} \sum_{x_i \in X} F1(x_i, g(x_i, Y))$$

and $g(x_i, Y) = \{\text{argmax}_y F1(x, y) | y \in Y\}$ in which $F1(x, y)$ is the F1 score of the respective clusters.

Normalized Mutual Information (NMI). Mutual Information (MI) is evaluated by taking all pairs of clusters from the formed and ground-truth clustering respectively and counts the number of common elements in each pair. Formally, given the ground-truth clustering C' consisting of clusters $c' \in C'$ and the formed clusters $c \in C$, mutual information is defined as:

$$I(C', C) = \sum_{c' \in C'} \sum_{c \in C} p(c', c) \log_2 \frac{p(c', c)}{p(c')p(c)},$$

where $p(c', c)$ is the normalized number of common elements in the pair of (category, cluster), $p(c')$ and $p(c)$ is the normalized number of elements in the categories and formed clusters respectively. The normalization is performed using the total number of elements in the clustering, i.e. the number of nodes in the input network. There is no upper bound for $I(C', C)$, so for easier interpretation and comparisons a normalized mutual information that ranges from 0 to 1 is desirable. There are two ways in which normalization is normally done Strehl & Ghosh (2002); Esquivel & Rosvall (2012). In the first one, the mutual information is divided by the average of the entropies, while in the second one, it is divided by the maximum of the entropies. These are defined as follows:

$$NMI_{sqr} = \frac{I(C', C)}{\sqrt{H(C')H(C)}} \quad NMI_{max} = \frac{I(C', C)}{\max\{H(C'), H(C)\}},$$

where $H(X) = -\sum_{x \in X} p(x) \log_2 p(x)$ is the entropy of the clustering X .

Modularity. The modularity of a graph compares the presence of each intra-cluster edge of the graph with the probability that that edge would exist in a random graph Newman & Girvan (2004); Blondel et al. (2008). Although modularity has been shown to have a resolution limit Fortunato &

Table 3: Comparison of Various Community Detection Methods on Karate Club Networks

Data Sets	Approaches	K		NMI_{max}		NMI_{sqr}		Ω Index		F1 Score	
		(No. of Clusters)		Modularity Function		Modularity Function		Modularity Function		Modularity Function	
		CNM	Louvain	CNM	Louvain	CNM	Louvain	CNM	Louvain	CNM	Louvain
KARATE	Modularity Partitioning	4	4	0.4518	0.4426	0.6231	0.6100	0.4909	0.4619	0.751785	0.750677
	KMeans-Node2Vec	4	4	0.5066	0.5278	0.6626	0.6859	0.5728	0.5982	0.799725	0.805801
	KMeans-Mod-Node2Vec	4	4	0.5689	0.5650	0.6972	0.6902	0.6492	0.6581	0.838412	0.854334
	KMeans-Node2Vec-Biased	4	4	0.5314	0.5279	0.6700	0.6786	0.6114	0.5956	0.820446	0.818937
	KMeans-Mod-Node2Vec-Biased	4	4	0.5671	0.5673	0.7041	0.7066	0.6444	0.6451	0.841761	0.846761
IDEAL CASE											
KARATE	KMeans-Node2Vec	2	2	0.689118	0.719544	0.695934	0.726091	0.744408	0.764389	0.933834	0.940061
	KMeans-Mod-Node2Vec	2	2	0.814953	0.822966	0.817697	0.825675	0.860193	0.871239	0.965374	0.968192
	KMeans-Node2Vec-Biased	2	2	0.742158	0.724971	0.744238	0.727342	0.807011	0.793847	0.950372	0.947802
	KMeans-Mod-Node2Vec-Biased	2	2	0.814703	0.819078	0.817368	0.822091	0.860198	0.871239	0.965374	0.968192

Barthelemy (2007), some of the most popular clustering algorithms use it as an objective function Waltman & Van Eck (2013); Blondel et al. (2008). Modularity is defined as $\sum_k (e_{kk} - a_k^2)$ where e_{kk} , the probability of intra-cluster edges in cluster C_k , and a_k , the probability of either an intra-cluster edge in cluster C_k or of an inter-cluster edge incident on cluster C_k .

The higher the values of the four performance indices, omega index, modularity, normalized mutual information and average F1 score, the better the quality of the detected communities.

EXPERIMENTAL RESULTS

REFERENCES

- Md Altaf-Ul-Amin, Yoko Shinbo, Kenji Mihara, Ken Kurokawa, and Shigehiko Kanaya. Development and implementation of an algorithm for detection of protein complexes in large interaction networks. *BMC bioinformatics*, 7(1):207, 2006.
- Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008 (10):P10008, 2008.
- Z. Burda, J. Duda, J. M. Luck, and B. Waclaw. Localization of the maximal entropy random walk. *Phys. Rev. Lett.*, 2009.
- Sandro Cavallari, Vincent W. Zheng, Hongyun Cai, Kevin Chen-Chuan Chang, and Erik Cambria. Learning community embedding with community detection and node embedding on graphs. In *Proc. of CIKM '17*, pp. 377–386, 2017.
- Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.
- Linda M Collins and Clyde W Dent. Omega: A general formulation of the rand index of cluster recovery suitable for non-disjoint solutions. *Multivariate Behavioral Research*, 23(2):231–242, 1988.
- Jean-Charles Delvenne and Anne-Sophie Libert. Centrality measures and thermodynamic formalism for complex networks. *Physical Review E*, 83(4), 2011.
- Alcides Viamontes Esquivel and Martin Rosvall. Comparing network covers using mutual information. *arXiv preprint arXiv:1202.0425*, 2012.
- Santo Fortunato and Marc Barthelemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2007.
- Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.
- Aditya Grover and Jure Leskovec. Node2vec: Scalable feature learning for networks. In *Proc. of KDD'16*, pp. 855–864, 2016.

- Steve Harenberg, Gonzalo Bello, L Gjeltema, Stephen Ranshous, Jitendra Harlalka, Ramona Seay, Kanchana Padmanabhan, and Nagiza Samatova. Community detection in large-scale networks: a survey and empirical evaluation. *Wiley Interdisciplinary Reviews: Computational Statistics*, 6(6): 426–439, 2014.
- Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.
- Peng Jiang and Mona Singh. Spici: a fast clustering algorithm for large biological networks. *Bioinformatics*, 26(8):1105–1111, 2010.
- Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical review E*, 78(4):046110, 2008.
- Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. <http://snap.stanford.edu/data>, June 2014.
- M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69:026113, 2004.
- Mark EJ Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104, 2006.
- J.K. Ochab and Z. Burda. Maximal entropy random walk in community detection. *The European Physical Journal Special Topics*, 216(1):73–81, Jan 2013.
- William Parry. Intrinsic markov chains. *Transactions of American Mathematical Society*, 112:55–66, 1964.
- Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proc. of KDD'14*, pp. 701–710, 2014.
- Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. In *Computer and Information Sciences-ISCIS 2005*, pp. 284–293. Springer, 2005.
- Arnau Prat-Pérez, David Dominguez-Sal, Josep M Brunat, and Josep-Lluís Larriba-Pey. Shaping communities out of triangles. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pp. 1677–1681. ACM, 2012.
- Arnau Prat-Pérez, David Dominguez-Sal, and Josep-Lluís Larriba-Pey. High quality, scalable and parallel community detection for large real graphs. In *Proceedings of the 23rd international conference on World wide web*, pp. 225–236. ACM, 2014.
- Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3):036106, 2007.
- Martin Rosvall and Carl T Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.
- Roberta Sinatra, Jess Gmez-Gardees, Renaud Lambiotte, Vincenzo Nicosia, and Vito Latora. Maximal-entropy random walks in complex networks with limited information. *Physical Review E.*, 83(3):030103–1–030103–4, 2011.
- Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617, 2002.
- Ludo Waltman and Nees Jan Van Eck. A smart local moving algorithm for large-scale modularity-based community detection. *The European Physical Journal B*, 86(11):471, 2013.
- Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *Proc. of KDD '16*, pp. 1225–1234, 2016.
- Jaewon Yang and Jure Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pp. 587–596. ACM, 2013.
- Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42(1):181–213, 2015.