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 stared 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

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) $c_v \in \{1, \ldots, p\}$ 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,

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

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 ‘DPClus’ (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

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

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,

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

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,

$$J(\theta) = \sum_{u} \sum_{v \in \mathcal{N}(u)} P(y = 1|\mathbf{u}, \mathbf{v}) + \sum_{u} \sum_{\bar{v} \in \bar{\mathcal{N}}(u)} P(y = 0|\mathbf{u}, \bar{\mathbf{v}}), \tag{4}$$

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 ($y = 0$) in the likelihood function refers to the randomly sampled noise (the number of negative samples is determined by the way the complement of the neighborhood, $\bar{\mathcal{N}}$, is defined).

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

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

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.

$$P_{uv} = \frac{A_{uv}}{\lambda} \frac{\psi_v}{\psi_u}, \tag{6}$$

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.

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

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,

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

i.e., a combinatorial algorithm partitions the vertex set, V, of a graph into p distinct communities.

In the second step, for obtaining the node embedding instead of providing as input the unpartitioned graph (as in standard approaches), we rather input the partitioned set of vertices obtained from Equation 8. Based on the supplied partition, we modify the objective function of node2vec (Equation 4) to address differently the two types of positive node association within a context, i.e., one, where node pairs belong to the same community (partition) as induced by the partition, and the other, where they belong to different communities. We put more emphasis on the first case than on the second one. Formally speaking,

$$J(\theta|\mathcal{C}) = \alpha \sum_{u \in V_i} \sum_{v \in \mathcal{N}(u) \cap V_i} P(y = 1|\mathbf{u}, \mathbf{v}) + (1 - \alpha) \sum_{u \in V_i} \sum_{v \in \mathcal{N}(u) - V_i} P(y = 1|\mathbf{u}, \mathbf{v}) + \sum_{u} \sum_{\bar{v} \in \bar{\mathcal{N}}(u)} P(y = 0|\mathbf{u}, \bar{\mathbf{v}}), \tag{9}$$

where the first component indicates those cases where u and v are predicted to be a part of the same community by a combinatorial algorithm $\mathcal{C}$, the second component indicates the ones where u and v are predicted to be a part of different communities as per $\mathcal{C}$, and $\alpha \in [0, 1]$ indicates a relative importance of the first component over the second (specifically for our experiments, we set $\alpha = 0.8$).

The intuition behind Equation 9 is to rely on two different sources of information, for determining the similarities between node pairs. The risk of only using the random walk based information is that a random walk initiated from the periphery of a community is likely to visit a peripheral node of a different community. Considering these cases as positive examples in the node2vec objective could result in falsely embedding two such nodes close to each other, in which case, it would be difficult for a downstream clustering algorithm, such as K-means, to assign them into two distinct clusters. However, using the additional information about the estimated communities is likely to identify these false cases and hence down-weight them in the embedding objective. Note that the contribution to the objective for node pairs belonging to different communities is still positive (i.e.

$y = 1$) as compared to the negative samples ($y = 0$) when a vertex is selected at random from outside the set of visited nodes.

Additionally, since MERW based neighborhood construction is likely to result in more local (with respect to a community) contexts (Burda et al., 2009), we also learn the embedding objective function of Equation 9 with MERW based neighborhood instead of an URW based one.

Experiment Setup

Datasets We conduct experiments on a range of different undirected and unweighted networks of varying sizes (number of nodes) and densities (relative number of edges with respect to a complete graph). All the graphs that we experimented with are associated with the ground-truth community information.

First, we perform experiments on three relatively small-scale standard benchmark networks for community detection. The first among these is the 'karate club'[1] graph, which comprises 34 nodes and 78 edges, where every node represents a member of a karate club at an American university. If two members are observed to have social interactions within or away from the karate club, they are connected by an edge. The next network that we experiment with is the 'dolphin network' comprising 62 nodes that represent bottlenose dolphins living in Doubtful Sound, New Zealand. The edges in this graph (159 in total) represent associations between dolphin pairs that were observed to be more frequent than the occasional expectation. The third network used in our experiments is the network of American football games between Division IA colleges during regular season of Fall 2000 Girvan & Newman (2002).

Along with the three relatively small-scale networks, we also conduct experiments on a set of large networks, namely the Amazon product purchase network and the Youtube user group network Leskovec & Krevl (2014).

We have also tested the experiments on three real world networks viz. Amazon 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.

In the Amazon product purchase network, 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).

Each user in the Youtube network is considered to be a node and the friendship between two users is denoted as edge. Moreover, an user can create a closed group by inviting his friends. Such groups are considered as ground-truth communities Mislove et al. (2007).

Yang & Leskovec (2015) quantified the quality (termed as *goodness*) of the community structure of a network as a function of how compact are the communities and how well are they connected internally while being relatively well-separated from the rest of the network. They also observed that the average goodness metric of the top k communities starts monotonically decreasing for values of k higher than 5000. The study (Yang & Leskovec, 2015) also shows that for the sake of comparing community detection approaches, it suffices to restrict the computation to the most representative communities of a large graph (typically the top 5000 communities). Following this observation, we in our experiments, also restrict the computation to the top 5000 communities for the DBLP and the Youtube networks. Table 1 summarizes the various statistics of these networks.

In addition to the real-life networks, we also conduct a set of experiments on synthetic networks. We use the standard LFR (Lancichinetti-Fortunato-Radicchi) mechanism to generate graphs with good community structures (i.e. relatively dense communities with sparse inter-community links) (Lancichinetti et al., 2008). An important parameter in the power law based LFR generative mechanism is the mixing parameter μ, which indicates the proportion of relationships a node shares with other communities. As prescribed by Lancichinetti et al. (2008), we used $\mu = 0.1$. Table 1 provides the details of the LFR parameters to generate the artificial networks for our experiments. To reduce

[1] `https://networkdata.ics.uci.edu/data.php?id=105`

Data Sets	#nodes	#edges	ρ	C_{num}	C_{max}	C_{min}	k_{max}	k_{avg}	C_{avg}
Karate	34	78	0.2288	2	18	16	17	4.588	17.00
Dolphin	62	159	0.1278	2	42	20	12	5.129	31.00
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
LFR500	500	1410	0.1408	38	45	2	49	5.64	13.16
LFR1000	1000	7105	0.1248	29	96	5	100	14.21	34.48

Table 1: Overview of a number of benchmark real-life networks used in our experiments. Acronyms: ρ (Minimum Internal Density), C_{num} (#communities), C_{max} (Maximum Community Size), C_{min} (Minimum Community Size), k_{max} (Maximum Degree), k_{avg} (Average Degree), C_{avg} (Average Community Size).

randomization effects of the artificially generated networks, we report the average results (over a set of 100 instances) obtained with each competing method.

Methods Investigated The objective of our experiments is to investigate if our proposed node embedding approaches (with the MERW and the modified objective function based on a combinatorial approach) is able to outperform standard embedding and combinatorial approaches for community detection. As our combinatorial baselines to community detection, we employ two methods that use the modularity score to greedily aggregate nodes into communities, and a random walk based method. Specifically, as the modularity score based approaches, we use CNM algorithm (Newman & Girvan, 2004), which operates on a graph as a whole, and the Louvian algorithm (Blondel et al., 2008) (denoted as 'LV' in our experiments), which successively coarsens a graph for community aggregation. As the random-walk based baseline, we employ the LPA algorithm. It is to be noted that these combinatorial baseline approaches automatically estimate the optimal number of clusters (communities) by making use of a global heuristic function representing the quality of the community structure.

As a node embedding based baseline for community detection, we employ a two-step method, the first step applying node2vec to obtain the embedded node representations of a graph, followed by conducting K-means on the node vectors to predict the communities (each cluster corresponding to a community). We denote this baseline as **n2v** in our experiments. In contrast to the combinatorial approaches, for K-means clustering, the number of communities needs to be provided as input. For each combinatorial community detection algorithm, as mentioned before, we employ the number of communities obtained by each as the value of K in the clustering based approach.

In addition to taking as input the number of clusters, K, our proposed approach of modifying the node2vec objective (which we denote as **CA-n2v** or community aware node2vec) also takes as input the partition induced by a combinatorial method, leading to a likely different output partitioning. Consequently, we report results on three different instances of CA-node2vec (one each for CNM, Louvian and LPA). In a similar manner, we report results with the three different cases (each corresponding to a combinatorial community detection approach) for the MERW-based node2vec (denoted as **MERW-n2v**) and community-aware MERW based node2vec (combination of both MERW based context construction and community partition driven modified node2vec objective), which we denote as **MERW-CA-n2v**.

Evaluation Measures The evaluation metrics for evaluating the effectiveness of community detection approaches typically measure the overlap between a set of predicted communities and the ground-truth ones. Indeed, the evaluation metrics for community detection are mostly borrowed from the clustering literature. The following is a brief description of the community detection evaluation metrics used in our experiments.

Omega-Index (OI): It is an extension of the standard 'Adjusted Rand Index' (ARI) (Hubert & Arabie, 1985) of clustering evaluation measure generalized to the case of evaluating the effectiveness of overlapping clusters (Collins & Dent, 1988). It measures the fraction of correct pairwise decisions, i.e. whether two nodes are correctly predicted to be in the same cluster or different clusters as per the ground-truth.

	#clusters			$\text{NMI}_{\max}$			NMI_{sqrt}			OI			F-score		
	(K)			Heuristic			Heuristic			Heuristic			Heuristic		
Approaches	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA
Combinatorial	4	4	3	0.4518	0.4426	0.5902	0.6231	0.6100	0.7058	0.4909	0.4619	0.7022	0.7518	0.7507	0.8677
n2v	4	4	3	0.5172	0.5172	0.6170	0.6626	0.6626	0.7162	0.5728	0.5982	0.7075	0.7997	0.7997	0.8796
MERW-n2v	4	4	3	0.5296	0.5296	0.6199	0.6700	0.6786	0.7193	0.6114	0.5956	0.6980	0.8204	0.8189	0.8688
CA-n2v	4	4	3	**0.5689**	0.5650	0.6427	0.6972	0.6902	**0.7449**	**0.6492**	**0.6581**	0.7403	0.8384	**0.8543**	0.8857
MERW-CA-n2v	4	4	3	0.5671	**0.5673**	**0.6539**	**0.7041**	**0.7066**	0.7323	0.6444	0.6451	**0.7542**	**0.8418**	0.8468	**0.9095**
Oracle Settings: Ground-truth #Communities															
	(K^*)			Heuristic			Heuristic			Heuristic			Heuristic		
Approaches	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA
n2v	2	2	2	0.7147	0.7147	0.7147	0.6959	0.6959	0.6959	0.7444	0.7444	0.7444	0.9338	0.9338	0.9338
MERW-n2v	2	2	2	0.7421	0.7249	0.7385	0.7443	0.7273	0.7421	0.8070	0.7938	0.7946	0.9504	0.9478	0.9480
CA-n2v	2	2	2	0.8149	**0.8229**	**0.8258**	**0.8177**	**0.8257**	**0.8279**	**0.8601**	**0.8712**	**0.8712**	**0.9654**	**0.9682**	**0.9682**
MERW-CA-n2v	2	2	2	**0.8147**	0.8191	0.8149	0.8174	0.8221	0.8177	**0.8601**	**0.8712**	0.8602	0.9653	**0.9682**	0.9674

Table 2: Results of community detection on the karate network. The outputs of n2v/MERW-n2v depend only on the value of K, whereas the outputs of CA-n2v/MERW-CA-n2v also depend on the partitions induced by a combinatorial algorithm (e.g. LPA). In the oracle settings, K is set to the number of ground-truth communities (K^*) to estimate the upper bounds of the algorithms.

	#clusters			$\text{NMI}_{\max}$			NMI_{sqrt}			OI			F-score		
	(K)			Heuristic			Heuristic			Heuristic			Heuristic		
Approaches	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA
Combinatorial	4	6	4	0.4225	0.3201	0.4960	0.5867	0.5312	0.7042	0.4509	0.2709	0.5090	0.7860	0.6333	0.8031
n2v	4	6	4	0.4801	0.3739	0.4832	0.6513	0.5753	0.6500	0.5475	0.3695	0.5234	0.8097	0.7101	0.8139
MERW-n2v	4	6	4	0.4868	0.3774	0.4867	0.6518	0.5813	0.6590	0.5380	0.3637	0.5314	0.8142	0.7064	0.8189
CA-n2v	4	6	4	0.5010	0.3872	0.5016	0.6589	0.5987	0.6643	0.5424	0.3752	**0.5760**	0.8176	0.7126	**0.8363**
MERW-CA-n2v	4	6	4	**0.5132**	**0.3996**	**0.5025**	**0.6627**	**0.6014**	**0.6720**	**0.5627**	**0.3904**	0.5342	**0.8270**	**0.7152**	0.8202
Oracle settings: Ground-truth #Communities															
	(K^*)			Heuristic			Heuristic			Heuristic			Heuristic		
Approaches	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA
n2v	2	2	2	0.8303	0.8324	0.8241	0.8390	0.8437	0.8345	0.8970	0.8914	0.8908	0.9744	0.9730	0.9729
MERW-n2v	2	2	2	0.8319	0.8394	0.8219	0.8433	0.8504	0.8342	0.8972	0.9035	0.8909	0.9746	0.9761	0.9735
CA-n2v	2	2	2	0.8580	0.8582	0.8642	0.8675	0.8677	0.8733	0.9160	0.9160	0.9223	0.9793	0.9793	0.9809
MERW-CA-n2v	2	2	2	0.8588	0.8483	0.8836	0.8682	0.8586	0.8914	0.9160	0.9097	0.9348	0.9793	0.9777	0.9841

Table 3: Results of community detection on the Dolphin network.

Mean F-score (F1): This metric is also based on pairwise decisions; the difference with OI is that correctly predicting if two nodes belong to different communities do not contribute positively to the metric value. It rather relies only on the precision and the recall computed over pairs of nodes in the same community as per the ground-truth.

Normalized Mutual Information (NMI): NMI is a measure which corresponds to the homogeneity of a predicted community, i.e., this metric yields a high value if the nodes within a predicted community indeed belong to the same community as per the ground-truth. The NMI metric is typically normalized in two ways (Strehl & Ghosh, 2002; Esquivel & Rosvall, 2012), the first involving a square-root (NMI_{sqrt}), and the second with a maximum ($\text{NMI}_{\max}$) operation of the entropy values of the predicted and the ground-truth communities, respectively.

RESULTS

We now present the comparisons between the effectiveness of the different approaches investigated. Tables 2 to 5 show the results on real-life networks, whereas Tables 6 and 7 report results for the LFR-based synthetic networks.

	#clusters			$\mathrm{NMI}_{\mathrm{max}}$			$\mathrm{NMI}_{\mathrm{sqrt}}$			OI			F-score		
	(K)			Heuristic			Heuristic			Heuristic			Heuristic		
Approaches	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA
Combinatorial	1122	1123	1617	0.9690	0.9697	0.9360	0.9813	0.9814	0.9577	0.6633	0.5135	0.4008	0.9540	0.9502	0.8139
n2v	1122	1123	1617	0.9248	0.9316	0.9490	0.9482	0.9341	0.9580	0.4556	0.3700	0.3766	0.8137	0.8154	0.8399
MERW-n2v	1122	1123	1617	0.9308	0.9321	0.9511	0.9518	0.9401	0.9567	0.4736	0.3802	0.3731	0.8122	0.8192	0.8415
CA-n2v	1122	1123	1617	**0.9338**	0.9413	0.9526	0.9547	**0.9523**	**0.9600**	**0.4749**	0.3808	0.3730	0.8166	0.8245	**0.8500**
MERW-CA-n2v	1122	1123	1617	0.9334	**0.9444**	**0.9546**	**0.9568**	0.9512	0.9578	0.4743	**0.3832**	0.3778	**0.8176**	**0.8276**	0.8496
Oracle settings: Ground-truth #Communities															
	(K)			Heuristic			Heuristic			Heuristic			Heuristic		
Approaches	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA
n2v	1517	1517	1517	0.9471	0.9461	0.9457	0.9577	0.9563	0.9576	0.4770	0.3727	0.4839	0.8304	0.8357	0.8313
MERW-n2v	1517	1517	1517	0.9454	0.9468	0.9452	0.9578	0.9586	0.9562	0.4825	0.3771	0.4781	0.8324	0.8332	0.8311
CA-n2v	1517	1517	1517	0.9501	0.9483	0.9482	0.9583	0.9581	0.9579	0.4801	0.3758	0.4737	0.8348	0.8372	0.8316
MERW-CA-n2v	1517	1517	1517	0.9516	0.9481	0.9497	0.9577	0.9598	0.9576	0.4813	0.3765	0.4832	0.8367	0.8401	0.8333

Table 4: Results of community detection on the Amazon product purchase network.

	#clusters			$\mathrm{NMI}_{\mathrm{max}}$			$\mathrm{NMI}_{\mathrm{sqrt}}$			OI			F-score		
	(K)			Heuristic			Heuristic			Heuristic			Heuristic		
Approaches	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA
Combinatorial	1271	890	2695	0.3680	0.3875	0.5065	0.5817	0.5898	0.6770	0.0666	0.0848	0.0893	0.2773	0.2809	0.4109
n2v	1271	890	2695	0.7295	0.7037	0.7549	0.7726	0.7597	0.7717	0.1713	0.1799	0.1172	0.4174	0.4036	0.3954
MERW-n2v	1271	890	2695	0.7307	0.7034	0.7819	0.7730	0.7592	0.7933	0.1685	0.1769	0.1341	0.4167	0.4018	0.4317
CA-n2v	1271	890	2695	0.7361	0.7104	0.7887	0.7760	0.7621	0.7903	0.1665	0.1828	0.1214	0.4210	0.4060	0.4316
MERW-CA-n2v	1271	890	2695	0.7365	0.7104	0.7874	0.7754	0.7641	0.7960	0.1574	0.1882	0.1285	0.4196	0.4075	0.4346
Oracle settings: Ground-truth #Communities															
	(K)			Heuristic			Heuristic			Heuristic			Heuristic		
Approaches	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA
n2v	4771	4771	4771	0.7975	0.7982	0.7986	0.8100	0.8101	0.8107	0.1082	0.1153	0.1063	0.4446	0.4447	0.4439
MERW-n2v	4771	4771	4771	0.7986	0.7973	0.7987	0.8102	0.8097	0.8112	0.1144	0.1072	0.1105	0.4461	0.4439	0.4471
CA-n2v	4771	4771	4771	0.8003	0.8012	0.8023	0.8131	0.8138	0.8140	0.1143	0.1229	0.1183	0.4511	0.4537	0.4525
MERW-CA-n2v	4771	4771	4771	0.7989	0.7994	0.7990	0.8132	0.8134	0.8136	0.1143	0.1151	0.1206	0.4510	0.4526	0.4531

Table 5: Results of community detection on the Youtube network.

In general, the following trends could be observed from the results. First, K-means on embedded node vectors mostly outperforms purely combinatorial approaches, e.g. CNM and LPA, more so for the large networks (e.g. the increase of $\mathrm{NMI}_{\mathrm{max}}$ from 0.5056 with LPA to 0.7549 with n2v). Second, we observe that the use of MERW for node embedding mostly improves community detection ef-

	#clusters			$\mathrm{NMI}_{\mathrm{max}}$			$\mathrm{NMI}_{\mathrm{sqrt}}$			OI			F-score		
	(K)			Heuristic			Heuristic			Heuristic			Heuristic		
Approaches	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA
Combinatorial	27	30	50	0.8789	0.9088	0.9142	0.9278	0.9234	0.9302	0.7905	0.8237	0.8620	0.8939	0.9113	0.9071
n2v	27	30	50	0.8934	0.9110	0.9131	0.9302	0.9321	0.9335	0.8526	0.8562	0.8447	0.8921	0.9027	0.8888
MERW-n2v	27	30	50	0.9007	0.9111	0.9206	0.9379	0.9280	0.9418	0.8716	0.8374	0.8443	0.9032	0.8923	0.8953
CA-n2v	27	30	50	0.9118	0.9193	0.9215	0.9464	0.9469	0.9394	0.8915	0.8905	0.8410	0.9236	0.9169	0.8879
MERW-CA-n2v	27	30	50	0.9129	0.9345	0.9154	0.9433	0.9433	0.9328	0.8889	0.8829	0.8491	0.9156	0.9112	0.8988
Oracle settings: Ground-truth #Communities															
	(K)			Heuristic			Heuristic			Heuristic			Heuristic		
Approaches	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA
n2v	38	38	38	0.9174	0.9331	0.9232	0.9196	0.9371	0.9384	0.8038	0.8572	0.8319	0.8739	0.9096	0.8948
MERW-n2v	38	38	38	0.9291	0.9252	0.9273	0.9359	0.9405	0.9293	0.8099	0.8981	0.8320	0.8823	0.9145	0.8809
CA-n2v	38	38	38	0.9273	0.9334	0.9262	0.9282	0.9390	0.9436	0.8202	0.8606	0.8541	0.8671	0.9029	0.9042
MERW-CA-n2v	38	38	38	0.9415	0.9346	0.9324	0.9420	0.9423	0.9421	0.8647	0.8538	0.8689	0.9152	0.9052	0.9074

Table 6: Results of community detection on a synthetic network (LFR-500).

	#clusters			NMI_{max}			NMI_{sqrt}			OI			F-score		
	(K)			Heuristic			Heuristic			Heuristic			Heuristic		
Approaches	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA
Combinatorial	18	22	28	0.8197	0.9279	0.9600	0.8754	0.9387	0.9714	0.7720	0.9186	0.9387	0.8654	0.9244	0.9644
n2v	18	22	28	0.8729	0.9462	0.9597	0.9261	0.9599	0.9640	0.8331	0.9166	0.9292	0.8804	0.9284	0.9477
MERW-n2v	18	22	28	0.8890	0.9446	0.9592	0.9371	0.9629	0.9682	0.8806	0.9424	0.9585	0.9089	0.9474	0.9593
CA-n2v	18	22	28	0.8821	0.9519	0.9577	0.9328	0.9723	0.9674	0.8483	0.9511	0.9405	0.8945	0.9578	0.9586
MERW-CA-n2v	18	22	28	0.8922	0.9579	0.9612	0.9329	0.9749	0.9707	0.8666	0.9696	0.9471	0.8892	0.9576	0.9568
Oracle settings: Ground-truth #Communities															
	(K)			Heuristic			Heuristic			Heuristic			Heuristic		
Approaches	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA	CNM	LV	LPA
n2v	29	29	29	0.9406	0.9269	0.9600	0.9474	0.9379	0.9685	0.9025	0.8608	0.9470	0.9206	0.9039	0.9616
MERW-n2v	29	29	29	0.9477	0.9482	0.9436	0.9514	0.9547	0.9549	0.9129	0.8948	0.9379	0.9227	0.9308	0.9471
CA-n2v	29	29	29	0.9472	0.9455	0.9639	0.9599	0.9550	0.9656	0.9083	0.9119	0.9542	0.9301	0.9360	0.9540
MERW-CA-n2v	29	29	29	0.9548	0.9519	0.9642	0.9632	0.9562	0.9713	0.9301	0.9178	0.9486	0.9529	0.9253	0.9499

Table 7: Results of community detection on a synthetic network (LFR-1000).

fectiveness, e.g. NMI_{max} of MERW-n2v (0.9308) is higher than that of n2v (0.9248). This confirms our hypothesis that maximum entropy based random walk is likely to include nodes of the same community in the contexts that are used to train the embedding.

Third, we observe that incorporating the partition information within the objective of node embedding results (i.e. CA-n2v) substantially improves the results in comparison to n2v and MER-n2v (both do not use the partition information), e.g., as seen from most of the bold-faced metric values in Tables 2 to 7. Finally, a combination of both partition awareness and MERW for constructing node contexts while training node vectors is seen to improve community detection effectiveness further (particularly for the LFR networks).

Conclusions and Future work

In this paper, we proposed a mechanism to incorporate the information about the optimal partitions of a network obtained with combinatorial approaches into the objective function for learning node representations. This seeks to addresses the problem of a random-walk based context construction for node embedding, as the random walk may eventually lead to including nodes from different communities in the context of a node. Further, we investigated a maximal entropy based random walk (which is known to preserve locality), and its combination with the partition augmented embedding objective. The results of our experiments demonstrate that including the partitional information helps improve community detection effectiveness.

Future work may investigate how node embedding in combination with combinatorial approaches for graph partition could be used to detect communities in dynamically evolving networks.

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.

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.

Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and Analysis of Online Social Networks. In *Proceedings of the 5th ACM/Usenix Internet Measurement Conference (IMC'07)*, San Diego, CA, October 2007.

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-Lluis 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-Lluis 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. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42(1):181–213, 2015.