Community-enhanced Network Representation Learning for Network Analysis

Cunchao Tu[1]*, Hao Wang[1]*, Xiangkai Zeng[2], Zhiyuan Liu[1]†, Maosong Sun[1]

[1]Department of Computer Science and Technology, State Key Lab on Intelligent Technology and Systems,
National Lab for Information Science and Technology, Tsinghua University, Beijing, China

[2]Beihang University, Beijing, China

Abstract

Network representation learning (NRL) aims to build low-dimensional vectors for vertices in a network. Most existing NRL methods focus on learning representations from local context of vertices (such as their neighbors). Nevertheless, vertices in many complex networks also exhibit significant global patterns widely known as communities. It's a common sense that vertices in the same community tend to connect densely, and usually share common attributes. These patterns are expected to improve NRL and benefit relevant evaluation tasks, such as link prediction and vertex classification. In this work, we propose a novel NRL model by introducing community information of vertices to learn more discriminative network representations, named as Community-enhanced Network Representation Learning (CNRL). CNRL simultaneously detects community distribution of each vertex and learns embeddings of both vertices and communities. In this way, we can obtain more informative representation of a vertex accompanying with its community information. In experiments, we evaluate the proposed CNRL model on vertex classification, link prediction, and community detection using several real-world datasets. The results demonstrate that CNRL significantly and consistently outperforms other state-of-the-art methods. Meanwhile, the detected meaningful communities verify our assumptions on the correlations among vertices, sequences, and communities.

Introduction

Network data is constantly growing along with the development of online social networks such as Facebook and Twitter. How to represent network data is critical when applying machine learning algorithms to network analysis tasks, such as vertex classification, personalized recommendation, anomaly detection, and link prediction [Shepitsen *et al.*, 2008; Heard *et al.*, 2010; Liben-Nowell and Kleinberg, 2007]. Traditional graph-based representation regards each vertex as a discrete symbol. Nevertheless, this representation scheme does not consider the relations between vertices and usually suffers from the sparsity problem.

Recently, network representation learning (NRL) has been widely adopted to network analysis, which aims to build low-dimensional vectors for vertices according to their structural roles in networks. NRL enables us to measure the semantic relations between vertices, and also alleviates the sparsity issue in conventional graph-based representation.

Most NRL methods learn vertex representations according to their *local context* information. For example, DeepWalk [Perozzi *et al.*, 2014] performs random walks over the network topology and learns vertex representations by maximizing the likelihood of predicting their local contextual vertices in walk sequences; LINE [Tang *et al.*, 2015] learns vertex representations by maximizing the likelihood of predicting their neighbor vertices. Both *contextual vertices* in DeepWalk and *neighbor vertices* in LINE are local context.

In a typical complex network, vertices usually group into multiple communities with each community densely connected inside [Newman, 2006]. Vertices in a community usually share certain common attributes. For example, Facebook users with the same education-based attributes ("School name" or "Major") tend to form communities [Yang *et al.*, 2013]. Hence, the community structure is an important *global pattern* of vertices, which is expected to benefit network analysis tasks. Inspired by this, we propose a novel NRL framework by taking community information into consideration, named as Community-enhanced NRL (CNRL).

The basic idea of CNRL is demonstrated in Fig. 1. We consider each vertex is grouped into multiple communities, and these communities are overlapping. In conventional NRL methods, the vertex embedding is typically learnt from local context vertices. In contrast, CNRL will learn the vertex embedding from both local context and global community information.

How to determine which community each vertex in a sequence belongs to is crucial in CNRL. As the analogy between words in text and vertices in walk sequences has been verified by [Perozzi *et al.*, 2014], we assume that there are correlations between word preference on topics and vertex preference on communities as well. Following the idea in topic models, each vertex in a specific sequence is assigned

*Indicates equal contribution

†Corresponding Author: Zhiyuan Liu (liuzy@tsinghua.edu.cn)

with a specific community, according to both the community distributions of the vertex and the sequence. Afterwards, each vertex and its assigned community are applied to predict its context vertices in the walk sequence. Therefore, representations of both vertices and communities are learnt by maximizing the prediction likelihood. Note that community distributions of vertices are also updated iteratively in the learning process.

The community representations learnt in CNRL will serve to enhance vertex representations in network analysis tasks, such as vertex classification and link prediction. Community representations are expected to be of great help for those long-tail vertices with less local context information.

We conduct experiments on several real-world network datasets, and compare CNRL with other baselines using the tasks of vertex classification, link prediction and community detection. Experimental results show that, CNRL can significantly improve the performance of all the tasks, and the superiority is consistent with respect to various datasets and training ratios. It demonstrates the effectiveness of considering global community information for network representation learning.

Related Work

NRL is becoming an important technique for network analysis in recent years. Current methods [Tang and Liu, 2009; Perozzi *et al.*, 2014; Tang *et al.*, 2015] embed each vertex into a real-valued vector space based on modeling local information and take the representations as features for tasks like vertex clustering, classification and link prediction. There are also a variety of studies that attempt to incorporate heterogenous information into NRL. Text-associated DeepWalk (TADW) [Yang *et al.*, 2015] extends DeepWalk to take advantage of text information of a network. By utilzing labeling information of vertices, max-margin DeepWalk (MMDW) [Tu *et al.*, 2016] employs max-margin principle to learn discriminative network representations. [Chang *et al.*, 2015] designed a deep embedding architecture for capturing complex heterogenous data in network. [Chen *et al.*, 2016] incorporate group information into NRL.

Community Detection methods focus on clustering or partitioning the vertices into different groups [Newman, 2006; Fortunato, 2010]. The major drawback of these traditional methods is that they cannot detect overlapping communities. To address this problem, sequential clique percolation (SCP) [Palla *et al.*, 2005] was proposed to generate overlapping communities by merging overlapping k-cliques. [Ahn *et al.*, 2010] proposed link clustering for overlapping community detection by partitioning links instead of vertices. In recent years, there have been many non-negative matrix factorization (NMF) methods for community detection [Wang *et al.*, 2011; Yang and Leskovec, 2013; Yang and Leskovec, 2012], which approximate adjacency matrix of a network by vertex-community affinity matrix.

Most of existing NRL methods only consider the local neighbors of vertices, and ignore the global patterns of networks, such as community structure. To the best of our knowledge, our model is the first attempt to jointly model local neighborhood information and global community structure in NRL.

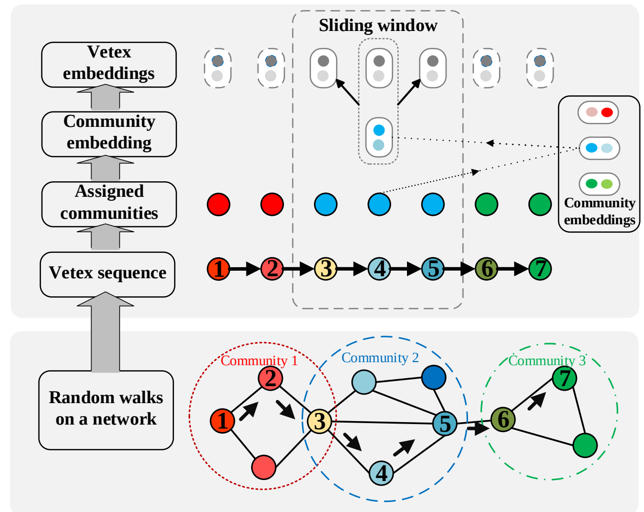

Figure 1: An illustration of CNRL.

Community-enhanced NRL

We start with discussing the necessary notations and formalizations of network representation learning.

Formalizations

We denote a network as $G = (V, E)$, where V is the set of vertices and $E \subseteq (V \times V)$ is the set of edges, with $(v_i, v_j) \in E$ indicating there is an edge between v_i and v_j. For each vertex v, NRL aims to learn a low-dimensional vector with the corresponding bold face $\mathbf{v} \in \mathbb{R}^d$. Here d denotes the dimension of representation space.

The vertices in G can be grouped into K communities $C = \{c_1, \ldots, c_K\}$. The communities are usually overlapping. That is, one vertex may be the member of multiple communities in different degrees. Hence, we record the membership degree of a vertex v to a community c as the probability $\Pr(c|v)$, and the role of the vertex in c as the probability $\Pr(v|c)$. In this work, we will also learn representations of each community c, denoted as $\mathbf{c}$.

In the following part, we first give a brief introduction to DeepWalk. Afterwards, we implement the idea of CNRL by extending DeepWalk to Community-enhance DeepWalk.

DeepWalk

DeepWalk [Perozzi *et al.*, 2014] performs random walks over the given network G firstly, and forms a set of walk sequences $S = \{s_1, \ldots, s_N\}$, where each sequence can be denoted as $s = \{v_1, \ldots, v_{|s|}\}$.

DeepWalk treats each walk sequence s as a word sequence by regarding vertices as words. By introducing Skip-Gram, a widely-used word representation learning algorithm, DeepWalk is able to learn vertex representations from the sequence set S. More specifically, given a vertex sequence $s = \{v_1, \ldots, v_{|s|}\}$, each vertex v_i has $\{v_{i-t}, \ldots, v_{i+t}\} \setminus \{v_i\}$

as its local context vertices. Following Skip-Gram, DeepWalk learns vertex representations by maximizing the average log probability of predicting context vertices:

$$\mathcal{L}(s) = \frac{1}{|s|} \sum_{i=1}^{|s|} \sum_{i-t \leq j \leq i+t, j \neq i} \log \Pr(v_j|v_i), \tag{1}$$

where v_{i+j} is the context vertex of the vertex v_i, and the probability $\Pr(v_j|v_i)$ is defined by softmax function:

$$\Pr(v_j|v_i) = \frac{\exp(\mathbf{v}'_j \cdot \mathbf{v}_i)}{\sum_{v \in V} \exp(\mathbf{v}' \cdot \mathbf{v}_i)}. \tag{2}$$

Here, $\mathbf{v}_i$ is the representation of the center vertex v_i and $\mathbf{v}'_j$ is the context representation of its context vertex v_j.

Community-enhanced DeepWalk

With random walk sequences, DeepWalk aims to maximize the local conditional probability of two vertices within a context window. That means, the co-occurrence of vertices in a sequence only relies on the affinity between vertices, while ignoring their global patterns. A critical global pattern of social networks is homophily, i.e., "birds of a feather flock together" [McPherson *et al.*, 2001]. That is, those similar vertices sharing the same "feather" may group into communities.

The community provides rich context information of a vertex. Hence, we take community information into consideration to broaden the context for modeling. We make two assumptions on the correlations among vertices, sequences and communities.

Assumption 1: Each vertex in a network may belong to multiple communities with different preferences, i.e., $\Pr(c|v)$, and each vertex sequence also owns its community distribution $\Pr(c|s)$.

With the above assumption, we make another assumption about the particular community of a vertex in a sequence.

Assumption 2: A vertex in a specific sequence belongs to a distinct community, and the community is determined by the sequence's distribution over communities $\Pr(c|s)$ and the community's distribution over vertices $\Pr(v|c)$.

With the above assumptions and generated random walk sequences, we conduct the following two steps iteratively to detect community structure and learn vertex representations: (1) **Community Assignment.** We assign a discrete community label for each vertex in a particular walk sequence, according to both local context and global community distribution. (2) **Representation Learning.** Given a vertex and its community label, we learn optimized representations to maximize the log probability of predicting context vertices.

The two steps are demonstrated in Fig. 1. As shown in Fig. 1, we aim to learn an embedding for each vertex and each community. Besides, we also want to learn the community distribution of each vertex. We introduce the two steps in detail as follows.

Community Assignment

For a vertex v in a walk sequence s, we compute the conditional probability of a community c as follows:

$$\Pr(c|v, s) = \frac{\Pr(c, v, s)}{\Pr(v, s)} \propto \Pr(c, v, s) \tag{3}$$

According to our assumptions, the joint distribution of (c, v, s) can be formalized as

$$\Pr(c, v, s) = \Pr(s) \Pr(c|s) \Pr(v|c) \tag{4}$$

where $\Pr(v|c)$ indicates the role of v in the community c, and $\Pr(c|s)$ indicates the local affinity of the sequence s with the community c.

From Eq. 3 and Eq. 4, we have

$$\Pr(c|v, s) \propto \Pr(v|c) \Pr(c|s), \tag{5}$$

In this work, we propose two strategies to implement $\Pr(c|v, s)$ as follows:

Statistic-based assignment. We employ Gibbs Sampling [Griffiths, 2002] to estimate the conditional distributions of $\Pr(v|c)$ and $\Pr(c|s)$ as follows:

$$\Pr(v|c) = \frac{N(v, c) + \beta}{\sum_{v' \in V} N(v', c) + |V|\beta}, \tag{6}$$

$$\Pr(c|s) = \frac{N(c, s) + \alpha}{\sum_{c' \in C} N(c', s) + |K|\alpha}. \tag{7}$$

Here $N(v, c)$ indicates how many times that the vertex v is assigned to the community c, and $N(c, s)$ indicates how many vertices in the sequence s are assigned to the community c. Both $N(v, c)$ and $N(c, s)$ will be updated dynamically as community assignments change. Moreover, β and α are smoothing factors following the idea of Latent Dirichlet Allocation (LDA) [Blei *et al.*, 2003].

Embedding-based assignment. As CNRL will gain the embeddings of vertices and communities, we can measure the conditional probabilities from an embedding view instead of global statistics. Therefore, $\Pr(c|s)$ can be formalized as follows:

$$\Pr(c|s) = \frac{\exp(\mathbf{c} \cdot \mathbf{s})}{\sum_{c' \in C} \exp(\mathbf{c}' \cdot \mathbf{s})}, \tag{8}$$

where $\mathbf{c}$ is the community vector learnt by CNRL, and $\mathbf{s}$ is the semantic vector of the sequence s, which is the average of the embeddings of all vertices in s.

In fact, we can also calculate $\Pr(v|c)$ in the similar way:

$$\Pr(v|c) = \frac{\exp(\mathbf{v} \cdot \mathbf{c})}{\sum_{v' \in V} \exp(\mathbf{v}' \cdot \mathbf{c})}. \tag{9}$$

However, the usage of Eq. (9) will badly degrade the performance. We suppose that the vertex embedding is not exclusively learnt for measuring community membership, hence Eq. (9) could not be so discriminative as compared to the statistic-based Eq. (6). Therefore, in the embedding-based method, we only calculate $\Pr(c|s)$ using embeddings and still use statistic-based $\Pr(v|c)$.

With estimated $\Pr(v|c)$ and $\Pr(c|s)$, we assign a discrete community label c for each vertex v in sequence s according to Eq. 5.

Representation Learning of Vertices and Communities

Given a certain vertex sequence $s = \{v_1, \ldots, v_{|s|}\}$, for each vertex v_i and its assigned community c_i, we will learn representations of both vertices and communities by maximizing the log probability of predicting context vertices using both v_i and c_i, which is formalized as follows:

$$\mathcal{L}(s) = \frac{1}{|s|}\sum_{i=1}^{|s|}\sum_{i-t\leq j\leq i+t, j\neq i} \log \Pr(v_j|v_i) + \log \Pr(v_j|c_i), \quad (10)$$

where $\Pr(v_j|v_i)$ is identical to Eq. (2), and $\Pr(v_j|c_i)$ is calculated similar to $\Pr(v_j|v_i)$ using a softmax function:

$$\Pr(v_j|c_i) = \frac{\exp(\mathbf{v}'_j \cdot \mathbf{c}_i)}{\sum_{v\in V}\exp(\mathbf{v}' \cdot \mathbf{c}_i)}. \quad (11)$$

Enhanced Vertex Representation

After learning CNRL from random walk sequences, we will obtain representations of both vertices and communities, as well as community information of vertices, including $\Pr(v|c)$ and $\Pr(c|v)$. We can apply these information to build enhanced vertex representations.

When dealing with a vertex v with no specific context, we can build its community representation as follows:

$$\mathbf{c}_v = \sum_{c_i\in C} \Pr(c_i|v)\mathbf{c}_i. \quad (12)$$

Afterwards, we can concatenate the original vertex vector $\mathbf{v}$ and its community vector $\mathbf{c}_v$ and build an enhanced vertex representation $\hat{\mathbf{v}} = \mathbf{v} \oplus \mathbf{c}_v$.

The enhanced representations encode both local and global context of vertices, which are expected to promote discriminability of network representation. In experiments, we will investigate the performance of the enhanced vertex representations on network analysis tasks.

Experiments

In experiments, we adopt the tasks of vertex classification and link prediction for evaluation. We also conduct community detection to prove the practical significance of the communities detected by our model.

Datasets

Table 1: Statistics of the real-world networks

Datasets	Cora	Citeseer	Wiki	BlogCatalog
# Vertices	2,708	3,312	2,405	10,312
# Edges	5,429	4,732	15,985	333,983
# Labels	7	6	19	47
Avg.Degree	**4.01**	**2.86**	**6.65**	**32.39**

We conduct our experiments on four widely adopted network datasets, including Cora[1], Citeseer, Wiki and BlogCatalog. Cora and Citeseer [McCallum *et al.*, 2000] are both research paper set. Here, the citation relationship between papers forms typical social networks. Wiki [Sen *et al.*, 2008] is a Web page collection from Wikipedia, and the hyperlinks among these pages compose a web network. BlogCatalog [Tang and Liu, 2009] is a social network among blogger authors. Detailed information is listed in Table 1.

[1]https://people.cs.umass.edu/ mccallum/data.html

Baseline Methods

We employ the following two typical network representation learning models, **DeepWalk** and **LINE** as baselines. Besides, we also employ four link prediction baseline methods[Lü and Zhou, 2011] which are mainly based on local topological properties:

Common Neighbors (CN) For vertex v_i and v_j, CN[Newman, 2001] simply counts the common neighbors of v_i and v_j as similarity score: $sim(v_i, v_j) = |N_i^+ \cap N_j^+|$

Salton Index For vertex v_i and v_j, Salton index[Salton and McGill, 1986] further considers the degree of v_i and v_j. The similarity score can be formulated as: $sim(v_i, v_j) = (|N_i^+ \cap N_j^+|)/(\sqrt{|N_i^+| \times |N_j^+|})$

Jaccard Index For vertex v_i and v_j, Jaccard index is defined as: $sim(v_i, v_j) = (|N_i^+ \cap N_j^+|)/(|N_i^+ \cup N_j^+|)$

Resource Allocation Index (RA) RA index[Zhou *et al.*, 2009] is the sum of resources received by v_j: $sim(v_i, v_j) = \sum_{v_k\in N_i^+} \frac{1}{|N_k^+|}$

Moreover, we also select three community detection baselines:

Sequential Clique Percolation (SCP) is a faster version of Clique Percolation [Palla *et al.*, 2005].

Link Clustering (LC) [Ahn *et al.*, 2010] aims to find link communities rather than nodes.

BigCLAM [Yang and Leskovec, 2012] is a typical non-negative matrix factorization based model.

Parameter Settings and Evaluation Metrics

For a fair comparison, we apply the same representation dimension as 128 in all methods. In LINE, as suggested in [Tang *et al.*, 2015], we set the number of negative samples to 5 and the learning rate to 0.025. We set the number of side samples to 1 billion for BlogCatalog and 10 million for other datasets. As both DeepWalk and CNRL generate walk sequences for training, we employ the default settings in DeepWalk, in which the walk length is 40, sequence number for each vertex is 10 and the window size is 5.

Note that, the representation vectors in CNRL consist of two parts, including the original vertex vectors and the corresponding community vectors. Thus, we set the dimension of both vectors to 64 and finally obtain a 128-dimensional vector for each vertex. Besides, the smoothing factor α is set to 2 and β is set to 0.5.

As each vertex in Cora, Citeseer and Wiki has only one label, we employ L2-regularized logistic regression (L2R-LR), the default setting of Liblinear [Fan *et al.*, 2008] to build classifiers. For multi-label classification in BlogCatalog, we train one-vs-rest logistic regression, and employ *micro-F1* and *macro-F1* for evaluation.

In the link prediction task, we employ a standard link prediction metric, **AUC** [Hanley and McNeil, 1982], to evaluate baselines and our method. Given the similarity of all vertex pairs, AUC is the probability that a random unobserved link has higher similarity than a random nonexistent link. Assume that we draw n independent comparisons, the AUC value is $(n_1 + 0.5n_2)/n$, where n_1 is the times that unobserved link

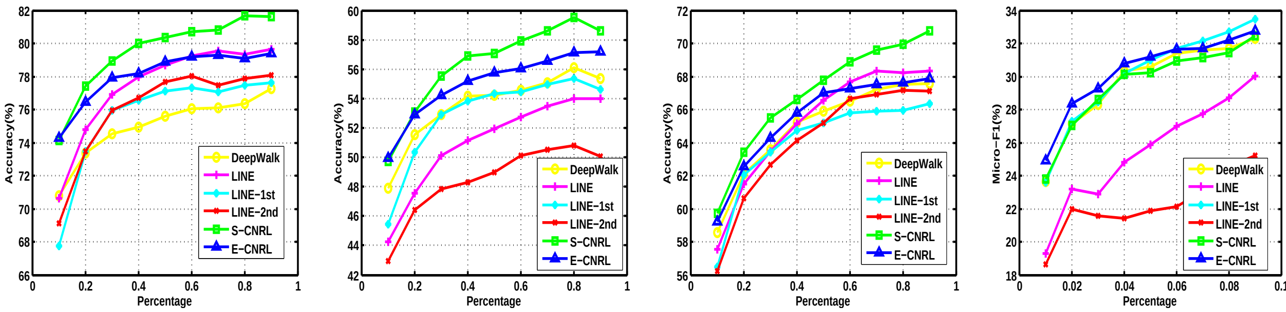

Figure 2: Vertex classification (from left to right: Cora, Citeseer, Wiki and BlogCatalog)

has higher score and n_2 is the times that they have equal score.

To measure the quality of detected communities, we employ **modified modularity** [Zhang *et al.*, 2015] for overlapping community detection task.

Vertex Classification

In Fig. 2, we show the classification accuracies under different training ratios and different datasets. We denote the two implements of CNRL as S-CNRL and E-CNRL, which represent the statistic-based CNRL and embedding-based CNRL. From these tables, we have the following observations:

(1) The proposed CNRL model consistently and significantly outperforms all the baseline methods on vertex classification task. It states the importance of incorporating community information and the flexibility of CNRL to various networks. Moreover, with the consideration of community structure, CNRL is able to learn more meaningful and discriminative network representations and the learnt representations are suitable to predictive tasks.

(2) With half less training data, CNRL still outperforms the baseline methods on different datasets. It demonstrates that CNRL can handle the sparse situation well. Furthermore, the incorporation of community structure can enhance the quality of representation vectors.

Link Prediction

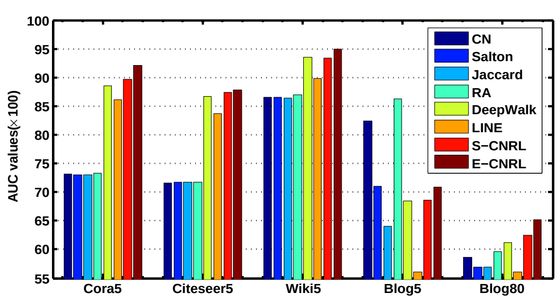

Figure 3: Link prediction (number behind each dataset indicates the portions of removed edges)

Community modeling methods should have the ability to correctly predict links. Therefore, we employ link prediction task to evaluate our proposed CNRL model.

In Fig. 3, we show the AUC values of link prediction on different datasets while removing different portions of edges. Note that, we show the results of LINE-1st on Blog5 and Blog80, as it outperforms LINE on BlogCatalog. From this figure, we observe that:

(1) In most cases, NRL methods outperform traditional hand-crafted link prediction methods. It proves that NRL methods are effective to encode network structure into real-valued representation vectors. Moreover, our proposed CNRL model consistently outperforms other NRL methods on different datasets, even when DeepWalk performs strongly in some situations. The results demonstrate the reasonability and effectiveness of considering community structure again.

(2) In a dense network like BlogCatalog, the average degree (i.e., 32.39) of vertices is much larger than other networks, which will benefit the simple statistic-based methods, such as CN and RA. Nevertheless, when the network turns sparse, the performance of these simple methods will badly decrease (25 percent around). On the contrary, CNRL only decreases about 5 percents. It indicates that CNRL is more suitable to deal with sparse issues.

Community Detection

Table 2: Community detection

Datasets	SCP	LC	BigCLAM	S-CNRL	E-CNRL
Cora	0.076	0.334	0.464	0.464	**1.440**
Citeseer	0.055	0.315	0.403	0.486	**1.861**
Wiki	0.063	0.322	0.286	0.291	**0.564**

We use modified modularity to evaluate the quality of detected communities. From Table 2, we observe that, S-CNRL is comparable with other state-of-the-art community detection methods, while E-CNRL significantly outperforms these baselines. It states that the communities detected by CNRL are meaningful under the measurement of community quality. Moreover, it conforms to our assumptions about the community assignment.

To summarize, all the results demonstrate the effectiveness and robustness of CNRL for incorporating community structure into vertex representations. It achieves consistent improvements comparing with other NRL methods on all the network analysis tasks.

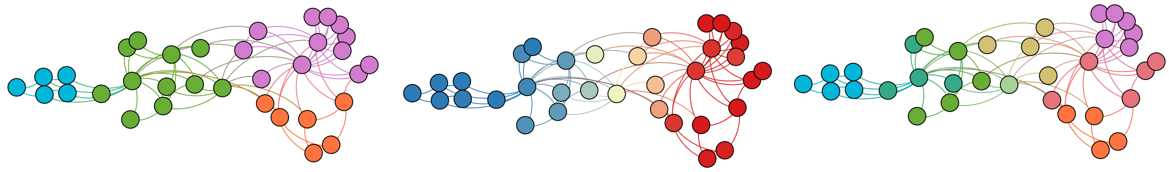

Figure 4: Detected communities on Karate (Fast Unfolding, 2 communities by CNRL, 4 communities by CNRL)

Visualizations of Detected Communities

For a more intuitive sense of detected communities, we visualize the detected overlapping communities by CNRL on a toy network named Zachary's Karate network [Zachary, 1977] in Fig. 4. For comparison, we also show the detected non-overlapping communities by a typical community detection algorithm, Fast Unfolding [Blondel *et al.*, 2008]. Note that, we mark different communities with different colors, and use gradient color to fill the vertices belonging to multiple communities. From Fig. 4, we observe that:

(1) CNRL is able to detect community structure with multiple scales, rather than clustering or partitioning vertices into fixed communities. Both the 2-community version and 4-community one are straightforward and reasonable according to the network structure.

(2) CNRL is well versed dealing with the overlapping issues in community detection. It can accurately identify vertices on community boundaries and balance the weights of the communities they belong to.

Note that, we present CNRL under the assumption that the vertex sequences reserve global community patterns. Here, we learn CNRL with vertex sequences and visualize the reproduced community issues. Results from Fig. 4 conform to our intuition and verify the assumption.

Case Study

Comparing with DeepWalk, CNRL can learn not only the community-enhanced vertex representations, but also the community assignments. To demonstrate the significance of assigned communities and give an intuitive experience on them, we conduct a case study on community assignments.

We provide a case in Cora and its community assignments in Table 3. The selected "paper" is titled as "Using a Case Base of Surfaces to Speed-Up Reinforcement Learning" and belongs to the research field of "Reinforcement Learning". As shown in Table 3, we employ S-CNRL to measure $\Pr(c|v)$ and obtain the representative communities of the example. For each community, we follow Eq. 6 to select representative vertices.

From this table, we observe that each community has its own characteristics. For example, community 1 is related to "Dynamic-programming", which is a sub-field in "Reinforcement-learning". Community 2 is relevant to "Cased-based" research, and community 3 concerns more about the learning and modeling methods in "Reinforcement learning". According to the title of the selected vertex, we find that it is involved in all the communities and the weights can reflect its relevance to the communities.

Table 3: Community assignments and representative vertices

Representative vertex	Label
Community 1 (weight = 0.56)	
Learning to Act using Real-Time Dynamic Programming	Reinforcement Learning
Generalized Markov Decision Processes: Dynamic-programming and Reinforcement-learning Algorithms	Reinforcement Learning
On the Convergence of Stochastic Iterative Dynamic Programming Algorithms	Reinforcement Learning
Community 2 (weight = 0.20)	
The Structure-Mapping Engine: Algorithm and Examples	Case Based
Case-based reasoning: Foundational issues, methodological variations, and system approaches	Case Based
Concept Learning and Heuristic Classification in Weak-Theory Domains	Case Based
Community 3 (weight = 0.12)	
Learning to Predict by the Methods of Temporal Differences	Reinforcement Learning
Generalization in Reinforcement Learning: Safely Approximating the Value Function	Reinforcement Learning
Exploration and Model Building in Mobile Robot Domains	Reinforcement Learning

Conclusion and Future Work

In this paper, we propose a novel NRL method to jointly embed local vertex characteristics and global community attributes into vertex representations. This is the first successful attempt to introduce community information into NRL. We simultaneously learn vertex representations and detect communities structure in our CNRL method. Moreover, we present two efficient community assignment strategies in CNRL, one of which is statistic-based and the other is embedding-based. The proposed CNRL model overcomes the drawbacks of previous works which only focus on local information, and achieves significant improvements comparing with existing state-of-the-art NRL algorithms when applying the learnt representations to network analysis tasks (e.g., vertex classification and link prediction). Besides, CNRL can effectively detect overlapping communities on multiple scales, which demonstrates the reasonability of our community assumptions in random walk sequences.

For future work, we may investigate the extensibility of our model on incorporating heterogenous information in social networks. As mentioned in related work part, vertices in real-world networks usually accompany with heterogenous information such as text contents and attributes, and many NRL works focus on encoding these additional information into vertex representations. It is a common sense that these heterogenous information can benefit network analysis tasks. Thus our method is expected to be flexible to incorporate these heterogenous information comprehensively. Another intriguing direction would be semi-supervised learning model of our algorithm. We can adapt the representation learning for specific tasks such as vertex classification. For example, we can take label information of training set into account to enhance the quality of vertex representations and predict the labels of test set simultaneously.

References

[Ahn *et al.*, 2010] Yong-Yeol Ahn, James P Bagrow, and Sune Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307):761–764, 2010.

[Blei *et al.*, 2003] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[Blondel *et al.*, 2008] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *JSTAT*, 2008(10):P10008, 2008.

[Chang *et al.*, 2015] Shiyu Chang, Wei Han, Jiliang Tang, Guo-Jun Qi, Charu C. Aggarwal, and Thomas S. Huang. Heterogeneous network embedding via deep architectures. In *Proceedings of SIGKDD*, pages 119–128, 2015.

[Chen *et al.*, 2016] Jifan Chen, Qi Zhang, and Xuanjing Huang. Incorporate group information to enhance network embedding. In *Proceedings of CIKM*, 2016.

[Fan *et al.*, 2008] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *JMLR*, 9:1871–1874, June 2008.

[Fortunato, 2010] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

[Griffiths, 2002] Tom Griffiths. Gibbs sampling in the generative model of latent dirichlet allocation. 2002.

[Hanley and McNeil, 1982] James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

[Heard *et al.*, 2010] Nicholas A Heard, David J Weston, Kiriaki Platanioti, David J Hand, et al. Bayesian anomaly detection methods for social networks. *AOAS*, 4(2):645–662, 2010.

[Liben-Nowell and Kleinberg, 2007] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *JASIST*, 58(7):1019–1031, 2007.

[Lü and Zhou, 2011] Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A*, 390(6):1150–1170, 2011.

[McCallum *et al.*, 2000] Andrew McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval Journal*, 3:127–163, 2000.

[McPherson *et al.*, 2001] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, pages 415–444, 2001.

[Newman, 2001] Mark EJ Newman. Clustering and preferential attachment in growing networks. *Physical Review E*, 64(2):025102, 2001.

[Newman, 2006] Mark EJ Newman. Modularity and community structure in networks. *PNAS*, 103(23):8577–8582, 2006.

[Palla *et al.*, 2005] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of SIGKDD*, pages 701–710, 2014.

[Salton and McGill, 1986] Gerard Salton and Michael J McGill. Introduction to modern information retrieval. 1986.

[Sen *et al.*, 2008] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93, 2008.

[Shepitsen *et al.*, 2008] Andriy Shepitsen, Jonathan Gemmell, Bamshad Mobasher, and Robin Burke. Personalized recommendation in social tagging systems using hierarchical clustering. In *Proceedings of RecSys*, pages 259–266, 2008.

[Tang and Liu, 2009] Lei Tang and Huan Liu. Relational learning via latent social dimensions. In *Proceedings of SIGKDD*, pages 817–826, 2009.

[Tang *et al.*, 2015] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of WWW*, pages 1067–1077, 2015.

[Tu *et al.*, 2016] Cunchao Tu, Weicheng Zhang, Zhiyuan Liu, and Maosong Sun. Max-margin deepwalk: Discriminative learning of network representation. In *Proceedings of IJCAI*, 2016.

[Wang *et al.*, 2011] Fei Wang, Tao Li, Xin Wang, Shenghuo Zhu, and Chris Ding. Community discovery using nonnegative matrix factorization. *Data Mining and Knowledge Discovery*, 22(3):493–521, 2011.

[Yang and Leskovec, 2012] Jaewon Yang and Jure Leskovec. Community-affiliation graph model for overlapping network community detection. In *Proceedings of ICDM*, pages 1170–1175, 2012.

[Yang and Leskovec, 2013] Jaewon Yang and Jure Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *Proceedings of WSDM*, pages 587–596, 2013.

[Yang *et al.*, 2013] Jaewon Yang, Julian McAuley, and Jure Leskovec. Community detection in networks with node attributes. In *Proceedings of ICDM*, pages 1151–1156, 2013.

[Yang *et al.*, 2015] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y Chang. Network representation learning with rich text information. In *Proceedings of IJCAI*, 2015.

[Zachary, 1977] Wayne W Zachary. An information flow model for conflict and fission in small groups. *JAR*, pages 452–473, 1977.

[Zhang *et al.*, 2015] Hongyi Zhang, Irwin King, and Michael R Lyu. Incorporating implicit link preference into overlapping community detection. In *Proceedings of AAAI*, pages 396–402, 2015.

[Zhou *et al.*, 2009] Tao Zhou, Linyuan Lü, and Yi-Cheng Zhang. Predicting missing links via local information. *EPJ B*, 71(4):623–630, 2009.