

Article

Structural Hierarchy-Enhanced Network Representation Learning

Cheng-Te Li ^{1,*}  and Hong-Yu Lin ²¹ Institute of Data Science, National Cheng Kung University, Tainan 70101, Taiwan² Department of Statistics, National Cheng Kung University, Tainan 70101, Taiwan; d0231469@gmail.com

* Correspondence: chengte@mail.ncku.edu.tw; Tel.: +886-6-2757575 (ext. 53628)

Received: 26 August 2020; Accepted: 6 October 2020; Published: 16 October 2020



Abstract: Network representation learning (NRL) is crucial in generating effective node features for downstream tasks, such as node classification (NC) and link prediction (LP). However, existing NRL methods neither properly identify neighbor nodes that should be pushed together and away in the embedding space, nor model coarse-grained community knowledge hidden behind the network topology. In this paper, we propose a novel NRL framework, Structural Hierarchy Enhancement (SHE), to deal with such two issues. The main idea is to construct a structural hierarchy from the network based on community detection, and to utilize such a hierarchy to perform level-wise NRL. In addition, lower-level node embeddings are passed to higher-level ones so that community knowledge can be aware of in NRL. Experiments conducted on benchmark network datasets show that SHE can significantly boost the performance of NRL in both tasks of NC and LP, compared to other hierarchical NRL methods.

Keywords: network representation learning; node embeddings; community detection; structural hierarchy; node classification; link prediction

1. Introduction

Network representation learning (NRL) is a crucial task in social and information network analysis. The idea of NRL is to learn a mapping function that converts each node into a low-dimensional embedding space while preserving the structural proximity between nodes in the given network. The derived node embedding vectors can be utilized for downstream tasks, including node classification, link prediction, and community detection. Typical NRL methods include DeepWalk [1], LINE [2], and node2vec [3], which consider structural neighborhood to depict every node. Metapath2vec [4] extends the skip-gram based NRL to heterogeneous information networks that contain multiple types of nodes and links. DANE [5] incorporates the similarity between node attributes into NRL. GCN [6] further learns node embeddings for semi-supervised node classification through a layer-wise propagation with graph convolution.

Regarding the typical NRL approaches [1–3] that preserve structural proximity in node embeddings, we think there are two major insufficiencies. First, every node is only aware of its few-hop neighbors by random walk sampling, which push them to be close in the embedding space. Nevertheless, as illustrated in the general node embedding space in Figure 1, non-neighbor nodes (v_5) being pushed away by negative sampling could still have some connections with the target node (v_1), e.g., belonging to the same network community (C1). In addition, neighbor nodes (v_8) sampled by random walks being pushed together could have weak connections to the target node (v_1), e.g., belonging to different communities (C1 and C3). In other words, community-level information cannot be incorporated into the learning of node embeddings. We think network communities can inform nodes to better recognize which nodes should better push together and away, as shown in the

community-aware embedding space in Figure 1. Second, the given graph depicts only the fine-grained interactions between nodes. As shown in Figure 1, the given graph is a collaboration network, and thus the links depict co-authorships. The coarse-grained semantics, including how authors involve in different research areas and belong to different institutes, together with their interactions, is not aware by typical NRL approaches. We think encoding coarse-grained semantics can improve the effectiveness of node embeddings.

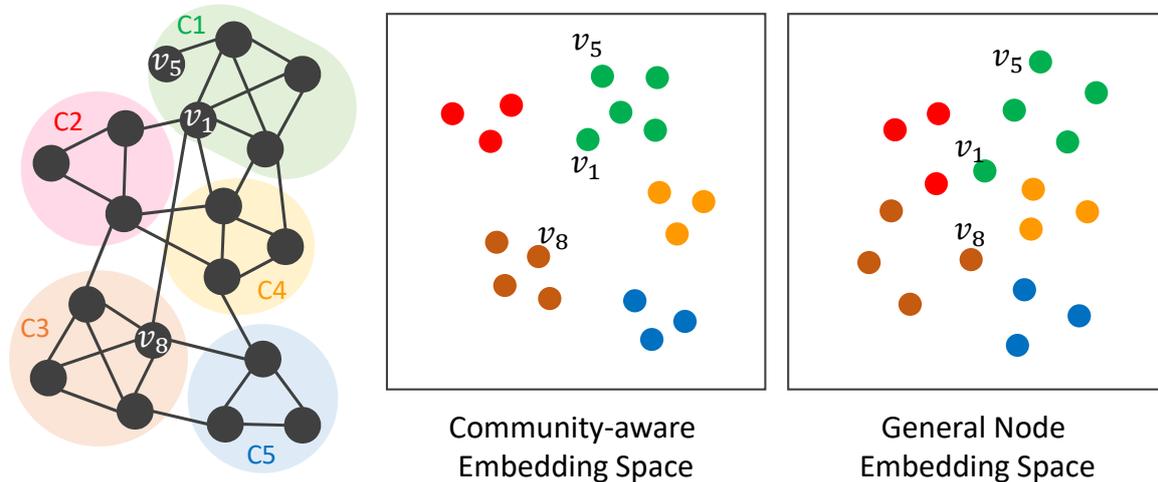


Figure 1. Given a collaboration network in the left, we illustrate and compare community-aware embedding space with general node embedding space, and expect that the community-aware version can better encode the structure of nodes in the embedding space.

In this paper, we propose a novel framework, Structural Hierarchy Enhancement (SHE), to enhance the effectiveness of network representation learning. The main idea is two-fold. First, we construct a structural hierarchy that depicts both fine-grained and coarse-grained semantics of nodes and their interactions. We consider that node semantics is depicted by communities (i.e., clusters of nodes), and thus utilize community detection techniques to produce the structural hierarchy. By learning the embeddings of nodes in different levels of the hierarchy, our model will be capable of encoding multiple different views of each node. Hence, the semantics of nodes can be enriched, and nodes can be better distinguished from one another in the embedding space. Second, we utilize such a hierarchy to enhance NRL. Our SHE can be seamlessly applied to enhance any of existing typical NRL models mentioned above. Besides, any existing and state-of-the-art hierarchical community detection algorithm can be applied to generate the hierarchy. In other words, our SHE model provides great flexibility to be compatible with different combinations of NRL methods and community detection techniques.

Related Work. The most relevant studies are HARP [7] and Marc [8], which are hierarchical NRL methods. HARP collapses nodes according to edge and star connections so that the hierarchy can be constructed for NRL. Marc iteratively consider 3-cliques as super nodes to construct the hierarchy. However, community knowledge in networks is not considered in both HARP and Marc. Besides, different-level's node embeddings are learned independently in HARP and Marc. That said, higher-level NRL cannot utilize node embeddings derived from lower-level NRL. We will compare the proposed SHE with HARP and Marc in the experiments. As for NRLs using various hierarchical information, NetHiex [9] assumes each node is associated with a category, and categories form a hierarchical taxonomy, which is used for NRL. HRE [10] uses the relational hierarchy that comes from edge attributes for heterogeneous NRL. MINES [11] models multi-dimensional relations between different node types, along with their hierarchical connections, into the embeddings of users and items for recommender systems. Poincare [12] specializes NRL for graphs whose nodes naturally form a hierarchical structure. DiffPool [13] classifies graphs by learning their embeddings based

on differentiable pooling applied to hierarchical groups of nodes. While these studies presume a variety of additional hierarchical information, i.e., category taxonomy, edge attributes, edge relations, hierarchical graph, and node groups, is accessible, our work does not rely on any of them.

2. Problem Statement

We first describe the notations for our problem. Let $G = (V, E)$ denote a network, in which V is the node set ($n = |V|$ is the number of nodes), and E is the edge set. We construct a structural hierarchy \mathcal{H} from a network G : $\mathcal{H} = \{H^0, H^1, \dots, H^{\tau-1}\}$, where τ is the number of levels in hierarchy \mathcal{H} , and H^τ is the level- τ graph. Level-0 graph is the original network, i.e., $H^0 = G$. Level- $(h + 1)$ graph H^{h+1} is constructed from level- h graph H^h . We present the list of all notations used in this paper in Table 1.

Table 1. Table of notations.

Notation	Description
$G = (V, E)$	the original graph with node set V and edge set E
k	the embedding dimension
n	the number of nodes in G
\mathcal{H}	the structural hierarchy
H^h	the level- h graph in \mathcal{H}
τ	the number of levels in \mathcal{H}
C_i^h	the i -th community node in H^h
e_{ij}^h	the edge that connects node C_i^h with node C_j^h in H^h
D^h	the set of edges that connect community nodes in H^h
ρ	the hyperparameter controlling the height of \mathcal{H}
n_h	the number of nodes in H^h
X^h	the embeddings of nodes in H^h
v_j^h	the j -th node in H^h
x_v^h	the embedding vector of node v in H^h
\mathbf{x}_v^h	the new-initialized embedding vector of node v in H^h

Structural Hierarchical-Enhanced NRL (SHE-NRL). Given a graph $G = (V, E)$, in which each node’s embedding vector $\mathbf{x}_v \in \mathbb{R}^{1 \times n}$ ($v \in V$) is initialized by a unit vector, along with its structural hierarchy \mathcal{H} , SHE-NRL is to learn a mapping function $f : V \rightarrow \mathbb{R}^k$ from nodes to low-dimensional embedding vectors so that nodes sharing similar connections in the graph, i.e., having a larger overlap on their neighbor sets, are projected as closely as possible in the embedding space. Here k is the embedding dimension, and f is a matrix of size $n \times k$, where $k \ll |V|$.

3. The Proposed SHE-NRL Model

The proposed SHE-NRL consists of four phases: (1) construction of structural hierarchy, (2) intra-level NRL, (3) level-wise pooling mechanism, and (4) generating final node embeddings. We first elaborate these four phases based on Figure 2. First, we construct the structural hierarchy by performing network community detection algorithm from lower- to higher-level graphs. Second, an existing or state-of-the-art NRL method is performed at the level- h graph (the original graph is level-0 graph) to generate level- h node embeddings. Third, a level-wise pooling mechanism is utilized to aggregate level- h node embeddings to not only initialize the level- $(h + 1)$ node embeddings for its NRL in a bottom-up manner, but also to initialize the level- $(h - 1)$ node embeddings for level- $(h - 1)$

NRL in a top-down manner. The second and third phases are performed iteratively until the highest level of the hierarchy is reached. Lastly, the final node embeddings can be produced by concatenating node embeddings at different levels based on their community memberships.

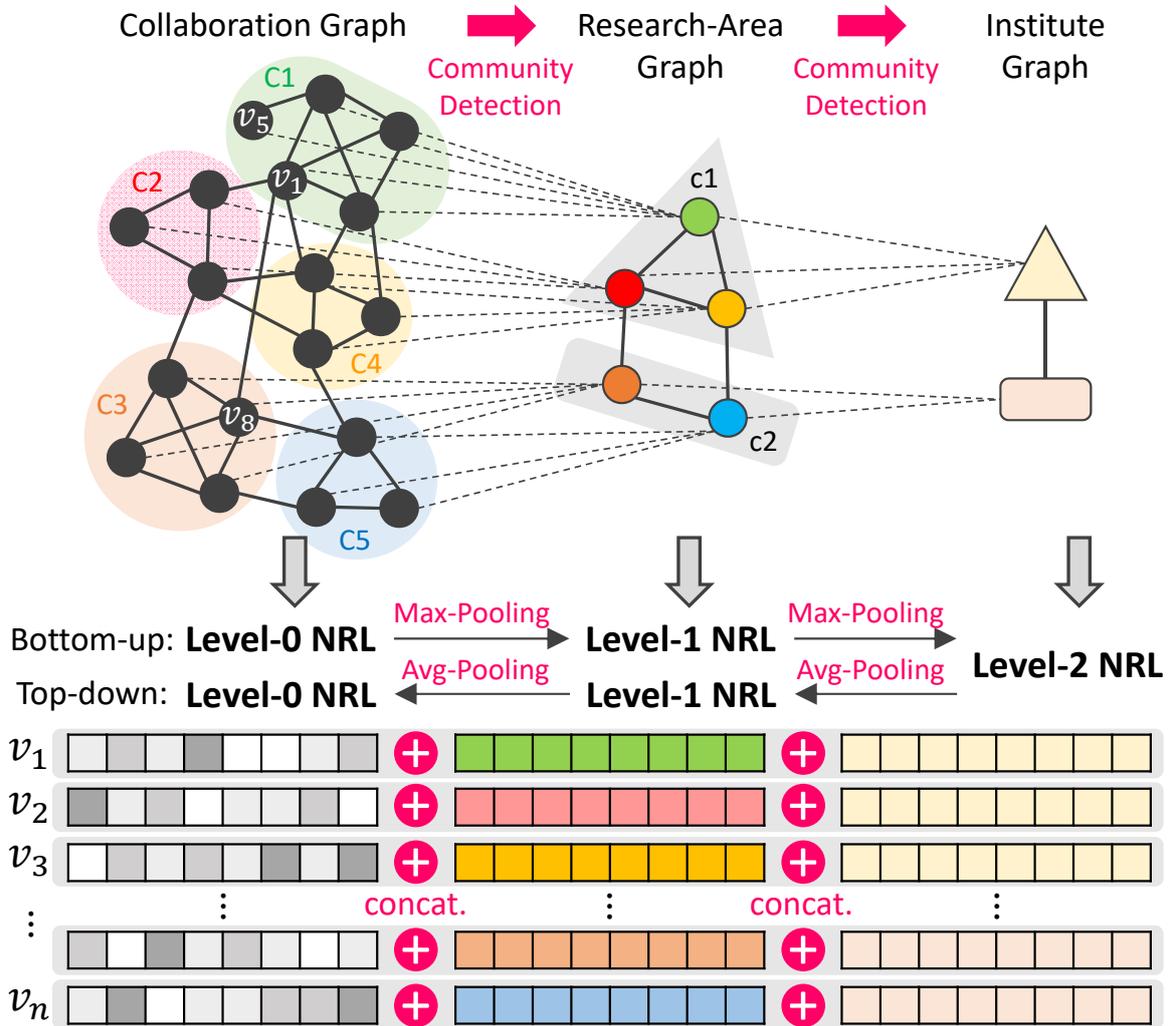


Figure 2. Overview of the proposed SHE framework.

Phase 1: Construction of Structural Hierarchy. The structural hierarchy \mathcal{H} is constructed from a network G . By applying a certain community detection algorithm to level- h graph H^h , we can obtain a set of communities (i.e., node sets) $\mathcal{C}^h = \{C_1^h, C_2^h, \dots, C_{n_h}^h\}$, where n_h is the number of communities in H^h , and C_i^h is the i -th community. These communities are treated as nodes at level- $(h + 1)$ graph H^{h+1} , given by:

$$H^{h+1} = (\mathcal{C}^h, \mathcal{D}^h), \tag{1}$$

where \mathcal{D}^h is the set of edges that connect communities. For every pair of communities C_i^h and C_j^h , we create an edge e_{ij}^{h+1} to connect them in H^{h+1} if there exists at least one edge between nodes in C_i^h and nodes in C_j^h in H^h . To adaptively determine the number of communities n_h for every graph H^h , we utilize Louvain [14] algorithm for community detection. We do not pre-define the number of levels τ in the hierarchy, but continue to produce H^{h+1} from H^h until $n_h < \rho$, where ρ is a hyperparameter controlling the height of the hierarchy. We set $\rho = 5$ by default. In other words, we will not generate level- $(h + 1)$ graph H^{h+1} if the number of nodes n_h in H^h is lower than ρ . In other words, the hyperparameter ρ is the minimum number of communities at the last (highest) level of the hierarchy, rather than the hierarchy height. Another hyperparameter τ , described in Phase 2, is the

hierarchy height. The τ is automatically determined by repeatedly generating new communities from the previous level's graph until $n_h < \rho$, where n_h is the number of communities in level- h 's graph H^h .

Phase 2: Intra-level NRL. Given level- h graph H^h , we perform intra-level NRL. We allow any of NRL methods that preserves structural proximity between nodes in our SHE-NRL framework. That said, typical NRL methods, such as DeepWalk [1], LINE [2], and node2vec [3], can be used for intra-level NRL. Let \mathbf{X}^h be the generated embedding matrix for all nodes in H^h . The intra-level NRL is iteratively performed in a one-round circular manner. The bottom-up way is first executed, followed by the top-down way. Specifically, the intra-level NRL is performed one after another from H^0, H^1 to H^τ , i.e., the bottom-up way. Then we again iteratively perform the intra-level NRL from $H^\tau, H^{\tau-1}$ to H^0 , i.e., the top-down way. The bottom-up way brings fine-grained level's interactions between nodes into the intra-level NRLs at higher levels in the hierarchy. The top-down way of message passing makes the intra-level NRLs at lower levels be aware of coarse-grained community knowledge. All intra-level NRLs at level- h graphs H^h ($0 \leq h \leq \tau - 1$) are executed twice while the level- τ NRL is executed only one time. In the next phase, we will discuss how \mathbf{X}^h can be used to initialize the node embeddings of graphs H^{h+1} and H^{h-1} . We randomly initialize the embeddings of nodes for NRL in the original graph $H^0 = G$.

Phase 3: Level-wise Pooling Mechanism. In order to bring fine-grained information of node interactions into higher-level NRLs, and make lower-level NRLs be aware of coarse-grained community knowledge at higher levels of the hierarchy, we propose the level-wise pooling mechanism. The pooling mechanism consists of bottom-up pooling and top-down pooling. The bottom-up pooling utilizes the node embeddings \mathbf{X}^h of H^h to initialize the node embeddings of H^{h+1} 's NRL. The max pooling is adopted as the bottom-up pooling. The max pooling is performed to initialize the embedding of node v_j^{h+1} at H^{h+1} from the corresponding i -th community C_i^h at H^h . We exploit the most significant learned node embedding in community C_i^h at H^h to be the initial node embedding of node v_j^{h+1} at H^{h+1} . On the other hand, the average pooling is adopted as the top-down pooling. The average pooling is performed to initialize the embeddings of all nodes in the i -th community C_i^h at H^h from the learned node embedding of node v_j^{h+1} at H^{h+1} . That said, given the new-learned node embedding of node u_j^{h+1} at H^{h+1} , denoted by \mathbf{x}_u^{h+1} , and the previously-generated embedding of corresponding lower-level node $v^h \in C_i^h$ at H^h , denoted by \mathbf{x}_v^h , we have the new initial embedding of node v^h , denoted by \mathbf{x}'_v^h , based on the equation:

$$\mathbf{x}'_v^h = avg(\mathbf{x}_v^h, \mathbf{x}_u^{h+1}). \tag{2}$$

Utilizing the average pooling for the top-down way of NRLs is able to distribute coarse-grained community knowledge back to the NRLs at lower-level graphs. Note that we utilize two different letters v and u here is to better distinguish nodes at different levels, i.e., v refers to a node at level h and u refers to a node at level $h + 1$.

Phase 2 and Phase 3 are iteratively adopted one after the other in first the bottom-up way, then the top-down way. In other words, in the bottom-up way, when node embeddings are generated by NRL in H^h (Phase 2), they are immediately brought to initialize the embeddings of nodes in H^{h+1} via max pooling (Phase 3). In the top-down way, when node embeddings are produced by NRL in H^{h+1} (Phase 2), they are instantaneously employed to initialize the embeddings of nodes in H^h via average pooling (Phase 3).

We utilize Figure 2 to better elaborate the interweaved process between Phase 2 and Phase 3 in an alternative view. In this paragraph, that said, we will describe the bottom-up process, followed by the top-down process. Given the structural hierarchy, consisting of three graphs at different levels, i.e., H^0, H^1 , and H^2 , in the bottom-up process, we first perform a typical NRL method on H^0 and obtain node embedding \mathbf{x}_v^0 for every $v \in H^0$. Then we apply max pooling on nodes, which belong to the same level-1 community, to initialize the embedding of each level-1 community node in H^1 . We again apply the same NRL method to H^1 to obtain node embedding for every level-1 community node, and use max pooling to initialize each level-2 community node in H^2 . Then NRL method is applied again on

H^2 . Next, we are doing the top-down process. We utilize average pooling in Equation (2) to initialize level-1 community node embedding from level-2 community node embedding. Then the NRL method is applied to H^1 , followed by performing average pooling to initialize node embeddings in H^0 . The last NRL is executed on H^0 to generate the embeddings of nodes in the original graph.

Phase 4: Generating Final Node Embeddings. Equipped with the derived node embedding matrix X^h at every level- h graph, we can generate the final embeddings for all nodes in the original network G (i.e., H^0). To let final node embeddings contain both fine-grained and coarse-grained information, the concatenation operation is adopted. We concatenate the embedding vector of node v in H^0 with all of its corresponding higher-level embedding vectors in H^h , where $h = 1, 2, \dots, \tau$. Suppose the dimension of each level's node embedding is equal and denoted as b . The dimension of final node embeddings will be $(\tau + 1)b$.

Note that it is apparent that there can be alternative approaches, such as the operators of average, Hadamard, and weighted-L2 [3], to fuse node embeddings at different levels of the hierarchy. We leave the design of better embedding aggregation for future investigation.

Remark. Two of the important ideas in the proposed SHE-NRL model are the bottom-up and top-down correlations on embedding vectors. If we first generate node embeddings for every level from H^0 to H^τ , then apply the top-down corrections, our model can encounter a critical issue—missing the interactions and connections between levels. Bottom-up and top-down correlations are to bring the fine-grained semantics encoded by lower-level node embeddings to higher-levels ones, and to deliver the coarse-grained semantics to lower-level nodes, respectively. Such a design follows the realistic intuition that, for an example in the university, a graduate student is depicted by her advisor, her college, and her school in order. We need to make the advisor recognize that student, and make the college see the advisor, and so on. By doing so, through the immediate correction of vectors in H^h level based on the H^{h+1} vectors, the node embeddings can better encode the semantics, and become more robust.

4. Experiments

We conduct experiments to answer three questions. (a) Can SHE-NRL improve the effectiveness of NRL for different NRL methods? (b) Is SHE-NRL able to outperform the state-of-the-art competing method? (c) Does a structural hierarchy with more levels lead to better performance?

4.1. Experimental Setup

Data and Settings. We use three benchmark network datasets for the experiments, including Cora, Citeseer, and PubMed (<https://lincs.soe.ucsc.edu/data>). Cora data contain 2708 nodes, 5429 edges, and 7 labels, Citeseer has 3312 nodes, 4715 edges, and 6 labels, and PubMed contains 19,717 nodes, 44,338 edges, and 3 labels. We evaluate SHE-NRL on two well-known NRL models, DeepWalk (DW) [1], node2vec (n2v) [3], and LINE [2]. Two competing methods are employed. They are the state-of-the-art hierarchical NRL methods, HARP [7] and Marc [8]. Both construct hierarchical structures for NRL. The embedding dimension of all methods is set $k = 128$. Note that since there are τ levels in a method, the embedding dimension of a level graph's NRL is $\frac{k}{\tau}$.

Evaluation Tasks. We evaluate the effectiveness of node embeddings on two downstream tasks, node classification (NC) and link prediction (LP). Given a certain fraction of nodes and all their labels, the goal of NC is to classify the labels for the remaining nodes. Node embeddings are treated as features. We utilize one-vs-rest logistic regression classifier with L2 regularization as the classifier. The default ratio of training and test is 80:20%. In our main experiment, we will vary the ratio of training and testing to see how different methods perform. On the other hand, LP is to predict the existence of links, given the existing network structure. We need to have the feature vectors of links and non-links. We follow node2vec [3] to employ Hadamard operator, i.e., element-wise product, to generate the feature vectors from the embeddings of node pairs. To obtain links, we remove 50% of edges chosen

randomly from the network while ensuring that the residual network is connected. To have non-links, we randomly sample an equal number of node pairs without edges connecting them.

Evaluation Metrics. For node classification, we consider Macro-F1 (MAF) and Micro-F1 (MIF) as the evaluation metrics. For link prediction, we utilize Area Under Curve (AUC) scores. Higher values indicate better performance in all metrics. Note that due to page limit, we report only MAF for node classification while MIF exhibits very similar results.

4.2. Experimental Results

Main Results. The main results are shown in Figures 3–5. We can have several observations. First, DeepWalk, node2vec, and LINE enhanced by the proposed SHE can get significant performance improvement (i.e., red vs. blue curves) in both tasks of NC and LP across datasets. The improvement margin is around 60% and 30% for NC and LP (these two improvement percentages are obtained by averaging the differences of MAF and AUC scores between red and blue curves over all training percentages on Citeseer and Cora datasets for node classification and link prediction, respectively), respectively. Second, SHE can further outperform the state-of-the-art methods HARP and Marc while both have already led to apparent improvement from the original DeepWalk and node2vec. We think the reason is that SHE not only leverages community knowledge, but also brings both fine-grained and coarse-grained information into the learning of different levels' NRLs. Besides, the superiority of SHE is more obvious in node classification than in link prediction. Third, when the training percentage increases, SHE is able to consistently outperform the other hierarchical NRL enhancement methods. Such results prove the effectiveness of SHE.

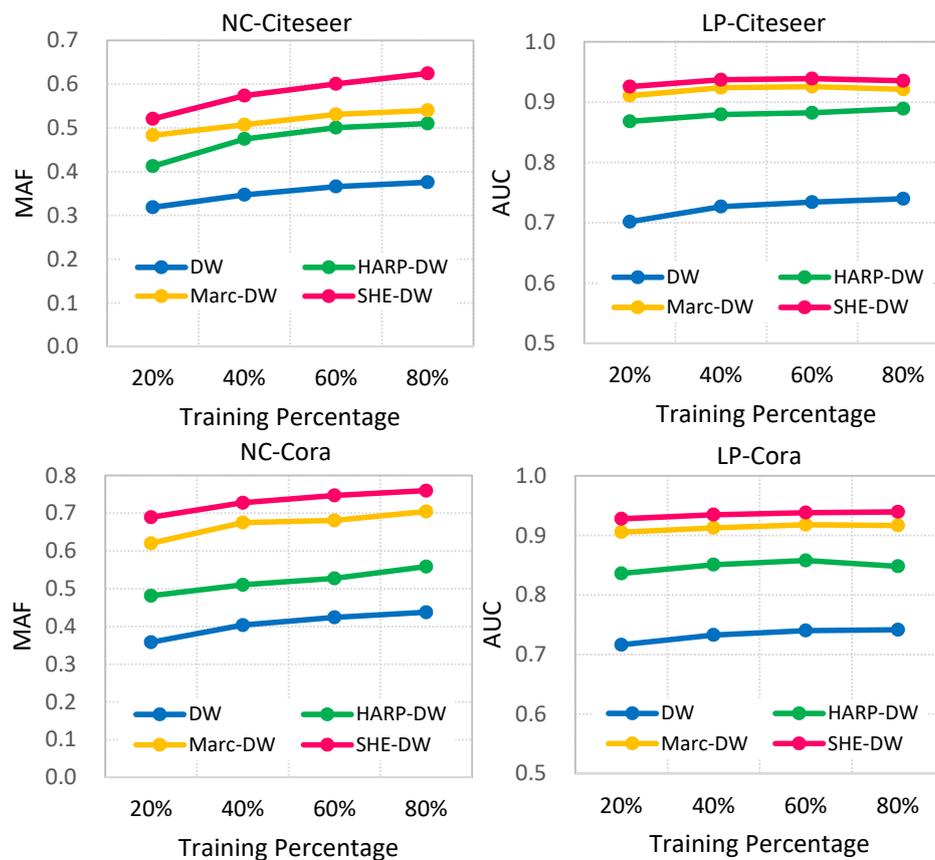


Figure 3. Results on DeepWalk (DW) by varying the training percentage. The left and right columns are on node classification and link prediction, respectively. The top and bottom rows are using Citeseer and Cora datasets, respectively.

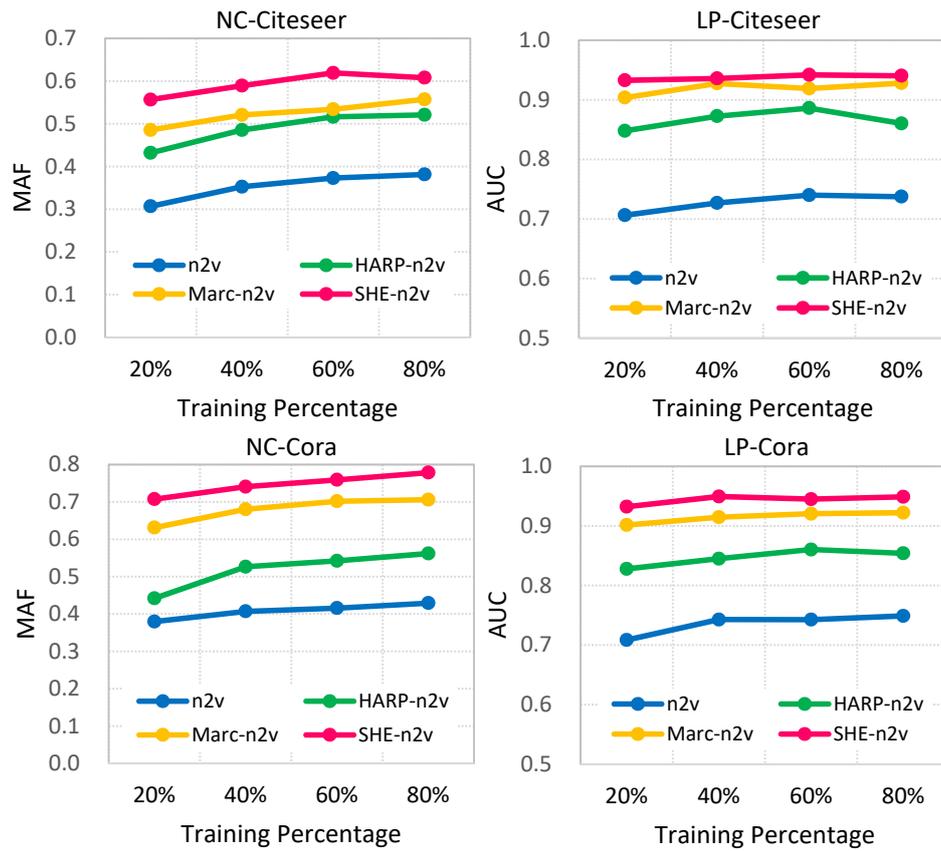


Figure 4. Results on node2vec (n2v) by varying the training percentage. The left and right columns are on node classification and link prediction, respectively. The top and bottom rows are using Citeseer and Cora datasets, respectively.

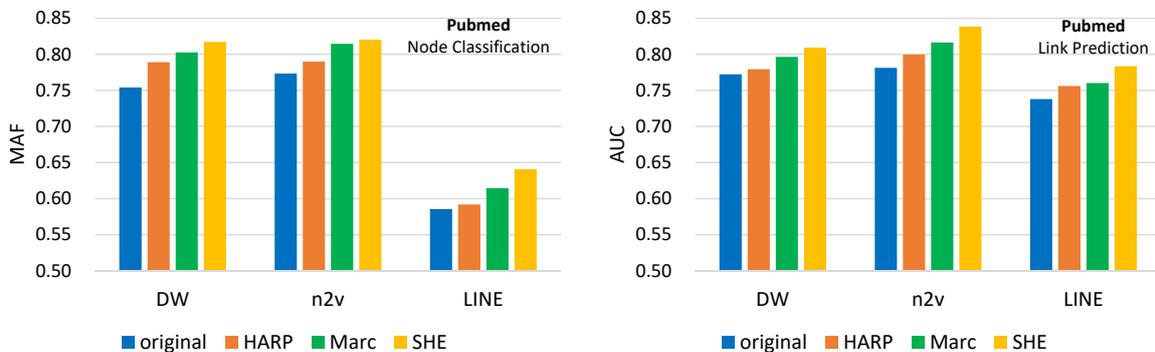


Figure 5. Performance comparison on Pubmed data. Left: node classification by MAF scores. Right: link prediction by AUC scores. Compared methods include the combinations of node embedding methods DeepWalk (DW), node2vec (n2v), and LINE with hierarchical NRL methods: original (original NRL method), HARP, Marc, and our proposed SHE.

The reason that the proposed SHE can outperform state-of-the-art methods is two-fold. On one hand, regarding constructing the hierarchy, HARP collapses nodes based on edge and star connections and Marc consider 3-cliques as super nodes. However, neither connection collapsing nor clique structure in both HARP and Marc can depict the semantic correlation between nodes, i.e., the community knowledge in networks. In our SHE, the obtained communities are used to form the hierarchy so that the fine-grained community information with respect to every node can be encoded into the learning of its embedding. On the other hand, different-level's node embeddings are learned independently in HARP and Marc. That said, higher-level NRL cannot utilize node

embeddings derived from lower-level NRL, and lower-level NRL cannot be enhanced by high-level coarse-grained semantics. Therefore, less informed node embeddings lead to worse performance, comparing to our SHE method that jointly learns and exploits node embeddings at different levels in the hierarchy.

Level Analysis. We aim to understand how the number of hierarchy levels affects the performance improvement of the proposed SHE. We vary the number of levels as 0, 0–1, and 0–2, which indicate no hierarchy (NRL on the original network), only one additional level in the hierarchy, and adopting a three-level hierarchy, respectively. The results of level analysis are exhibited in Figure 6. We can find that the hierarchy with only one additional level is enough to bring significant performance improvement. In addition, the performance of the hierarchy with two additional levels is nearly the same as that with one additional level. We can draw an insight from such results. Using only the community knowledge obtained from the original network is sufficient for SHE to boost the effectiveness of NRL.

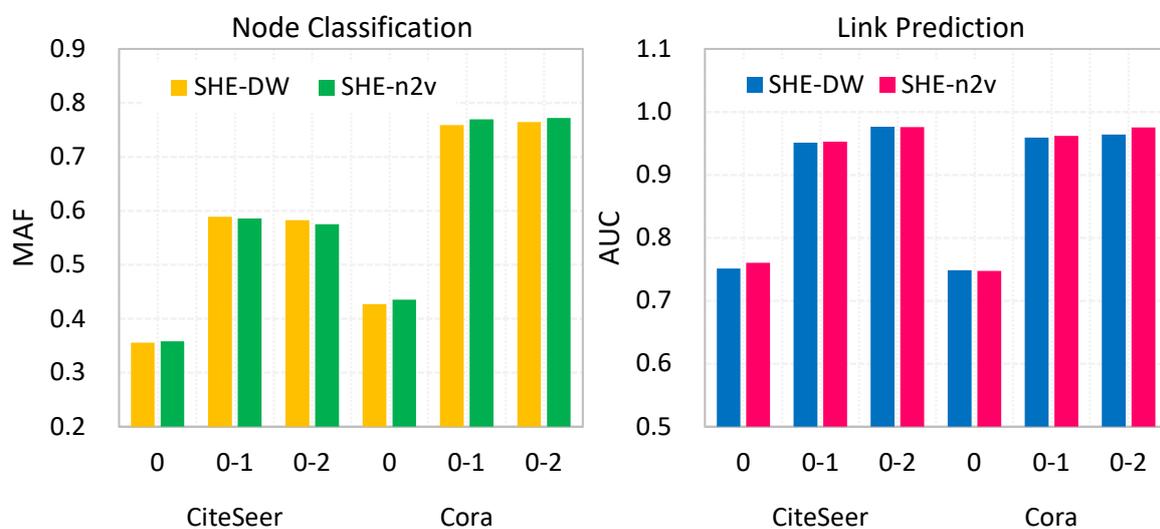


Figure 6. Results by changing the number of utilized levels in SHE based on DW and n2v, respectively.

5. Conclusions and Future Work

This paper proposes a structural hierarchy-enhanced network representation learning (SHE-NRL) framework to improve the effectiveness of the learned node embeddings. Our SHE can be incorporated into existing NRL methods, such as DeepWalk, node2vec, and LINE, so that their performance in downstream tasks can be boosted. Experiments conducted on real datasets for node classification and link prediction prove the effectiveness of SHE-NRL. An extensive empirical study also exhibits that even using one additional level hierarchy can significantly boost the effectiveness of NRL methods.

The promising results of SHE encourage us to create a three-fold extension. First, while graph neural networks (GNN) [15] are widely proven being effective for graph-based applications, we are developing a hierarchical message passing mechanism based on SHE so that both fine-grained and coarse-grained nodes can receive each other's information to improve the performance of semi-supervised node classification. Second, we will define the relational hierarchy in a heterogeneous network so that the learning of heterogeneous node embeddings, such as metapath2vec [4], can be aware of multi-typed community knowledge. Last, the current SHE treats the construction of structural hierarchy and the learning of node embeddings as two independent modules. An ongoing extension is to jointly optimize the hierarchy and node embeddings in an end-to-end manner.

Author Contributions: Conceptualization, C.-T.L.; Data curation, H.-Y.L.; Formal analysis, C.-T.L. and H.-Y.L.; Funding acquisition, C.-T.L.; Investigation, C.-T.L. and H.-Y.L.; Methodology, C.-T.L. and H.-Y.L.; Project administration, C.-T.L.; Resources, C.-T.L.; Supervision, C.-T.L.; Validation, H.-Y.L.; Visualization, H.-Y.L.; Writing—original draft, C.-T.L. and H.-Y.L.; Writing—review & editing, C.-T.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: This work is supported by Ministry of Science and Technology (MOST) of Taiwan under grants 109-2636-E-006-017 (MOST Young Scholar Fellowship) and 109-2221-E-006-173, and also by Academia Sinica under grant AS-TP-107-M05.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Perozzi, B.; Al-Rfou, R.; Skiena, S. DeepWalk: Online Learning of Social Representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'14), New York, NY, USA, 14–27 August 2014; ACM: New York, NY, USA, 2014; pp. 701–710.
2. Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; Mei, Q. LINE: Large-Scale Information Network Embedding. In Proceedings of the 24th International Conference on World Wide Web (WWW '15), Florence, Italy, 18–22 May 2015; ACM: New York, NY, USA, 2015; pp. 1067–1077.
3. Grover, A.; Leskovec, J. Node2vec: Scalable Feature Learning for Networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16), San Francisco, CA, USA, 13–17 August 2016; ACM: New York, NY, USA, 2016; pp. 855–864.
4. Dong, Y.; Chawla, N.V.; Swami, A. Metapath2vec: Scalable Representation Learning for Heterogeneous Networks. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '17), Halifax, NS, Canada, 13–17 August 2017; ACM: New York, NY, USA, 2017; pp. 135–144.
5. Gao, H.; Huang, H. Deep Attributed Network Embedding. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18), Stockholm, Sweden, 13–19 July 2018; pp. 3364–3370.
6. Kipf, T.N.; Welling, M. Semi-supervised Classification with Graph Convolutional Networks. In Proceedings of the International Conference on Learning Representations (ICLR '17), Toulon, France, 24–26 April 2017; pp. 873–876.
7. Chen, H.; Perozzi, B.; Hu, Y.; Skiena, S. HARP: Hierarchical Representation Learning for Networks. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18), New Orleans, LA, USA, 2–7 February 2018; pp. 2127–2134.
8. Xin, Z.; Chen, J.; Chen, G.; Zhao, S. Marc: Multi-Granular Representation Learning for Networks Based on the 3-Clique. *IEEE Access* **2019**, *7*, 141715–141727. [[CrossRef](#)]
9. Ma, J.; Cui, P.; Wang, X.; Zhu, W. Hierarchical Taxonomy Aware Network Embedding. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18), London, UK, 19–23 August 2018; ACM: New York, NY, USA, 2018; pp. 1920–1929.
10. Chen, M.; Quirk, C. Embedding Edge-Attributed Relational Hierarchies. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19), Paris, France, 21–25 July 2019; ACM: New York, NY, USA, 2019; pp. 873–876.
11. Ma, Y.; Ren, Z.; Jiang, Z.; Tang, J.; Yin, D. Multi-Dimensional Network Embedding with Hierarchical Structure. In Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM '18, Los Angeles, CA, USA, 5–9 February 2018; ACM: New York, NY, USA, 2018; pp. 387–395.
12. Nickel, M.; Kiela, D. Poincaré Embeddings for Learning Hierarchical Representations. In *Advances in Neural Information Processing Systems 30*; 2017; pp. 6338–6347. Available online: <https://papers.nips.cc/book/advances-in-neural-information-processing-systems-30-2017> (accessed on 10 October 2020)
13. Ying, R.; You, J.; Morris, C.; Ren, X.; Hamilton, W.L.; Leskovec, J. Hierarchical Graph Representation Learning with Differentiable Pooling. In Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS '18), Montreal, QC, Canada, 3–8 December 2018; Curran Associates Inc.: Red Hook, NY, USA, 2018; pp. 4805–4815.

14. Blondel, V.D.; Guillaume, J.L.; Lambiotte, R.; Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, *2008*, P10008. [[CrossRef](#)]
15. Cai, H.; Zheng, V.W.; Chang, K.C. A Comprehensive Survey of Graph Embedding: Problems, Techniques, and Applications. *IEEE Trans. Knowl. Data Eng.* **2018**, *30*, 1616–1637. [[CrossRef](#)]

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).