AI Open 3 (2022) 91–100

Contents lists available at ScienceDirect

AI Open

journal homepage: www.keaipublishing.com/en/journals/ai-open

Hierarchical label with imbalance and attributed network structure fusion for network embedding

Shu Zhao [a,b,c,*], Jialin Chen [a,b,c], Jie Chen [a,b,c], Yanping Zhang [a,b,c], Jie Tang [d]

[a] Key Laboratory of Intelligent Computing and Signal Processing, Ministry of Education, Hefei 230601, China
[b] School of Computer Science and Technology, Anhui University, Hefei 230601, China
[c] Information Materials and Intelligent Sensing Laboratory of Anhui Province, Hefei 230601, China
[d] Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Keywords:
Attributed network embedding
Hierarchical label
Representation learning

ABSTRACT

Network embedding (NE) aims to learn low-dimensional vectors for nodes while preserving the network's essential properties (e.g., attributes and structure). Previous methods have been proposed to learn node representations with encouraging achievements. Recent research has shown that the hierarchical label has potential value in seeking latent hierarchical structures and learning more effective classification information. Nevertheless, most existing network embedding methods either focus on the network without the hierarchical label, or the learning process of hierarchical structure for labels is separate from the network structure. Learning node embedding with the hierarchical label suffers from two challenges: (1) Fusing hierarchical labels and network is still an arduous task. (2) The data volume imbalance under different hierarchical labels is more noticeable than flat labels. This paper proposes a **H**ierarchical Label and **A**ttributed **N**etwork **S**tructure Fusion model(HANS), which realizes the fusion of hierarchical labels and nodes through attributes and the attention-based fusion module. Particularly, HANS designs a directed hierarchy structure encoder for modeling label dependencies in three directions (parent–child, child–parent, and sibling) to strengthen the co-occurrence information between labels of different frequencies and reduce the impact of the label imbalance. Experiments on real-world datasets demonstrate that the proposed method achieves significantly better performance than the state-of-the-art algorithms.

1. Introduction

Network embedding (NE) is a paradigm that automates learning low-dimensional representation for nodes while preserving essential properties of the network and the inter-node similarity reflected by the network structure. Therefore various network embedding methods have been proposed to preserve the kth-order proximity (Cao et al., 2015; Tang et al., 2015) and the neighborhood structure explored by random walks (Grover and Leskovec, 2016; Perozzi et al., 2014). Besides structure information, the attributes information (such as text, labels, scores, Etc.) attached to nodes is also used to enhance the effectiveness of network embedding. It has been shown that a myriad of network analysis tasks, including node clustering (Ding et al., 2001), node classification (Grover and Leskovec, 2016), link prediction (Liben-Nowell and Kleinberg, 2007), and network visualization (Rauber et al., 2016), can be significantly facilitated by the learned low-dimensional vector representation. With the rapid expansion of network scale, many complex real-world networks are often associated with abundant hierarchical label information. Based on the research of hierarchical classification trees (Zhang et al., 2018a; Shang et al., 2020a,b) and hierarchical text classification (Huang et al., 2019; Chen et al., 2020a; Pereira et al., 2021), the notion of hierarchical node classification, which assumes and seeks the latent hierarchical structure underlying the seemingly flatly connected nodes, has been explored by pioneering research (Ma et al., 2018; Yang et al., 2020).

The hierarchical tree formed by the hierarchical label is an independent topology separate from the node network. The hierarchical label and node embedding processes are separated into two different spaces when learning hierarchical labels and network structure. In order to realize the fusion of hierarchical label and network structure in the same space, it is necessary to construct the association between the two topologies.

* Corresponding author.
E-mail address: zhaoshuzs@ahu.edu.cn (S. Zhao).

[1] https://pubmed.ncbi.nlm.nih.gov
[2] https://meshb.nlm.nih.gov/search

https://doi.org/10.1016/j.aiopen.2022.07.002
Received 16 February 2022; Received in revised form 19 May 2022; Accepted 28 July 2022
Available online 2 August 2022
2666-6510/© 2022 The Authors. Publishing services by Elsevier B.V. on behalf of KeAi Communications Co. Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

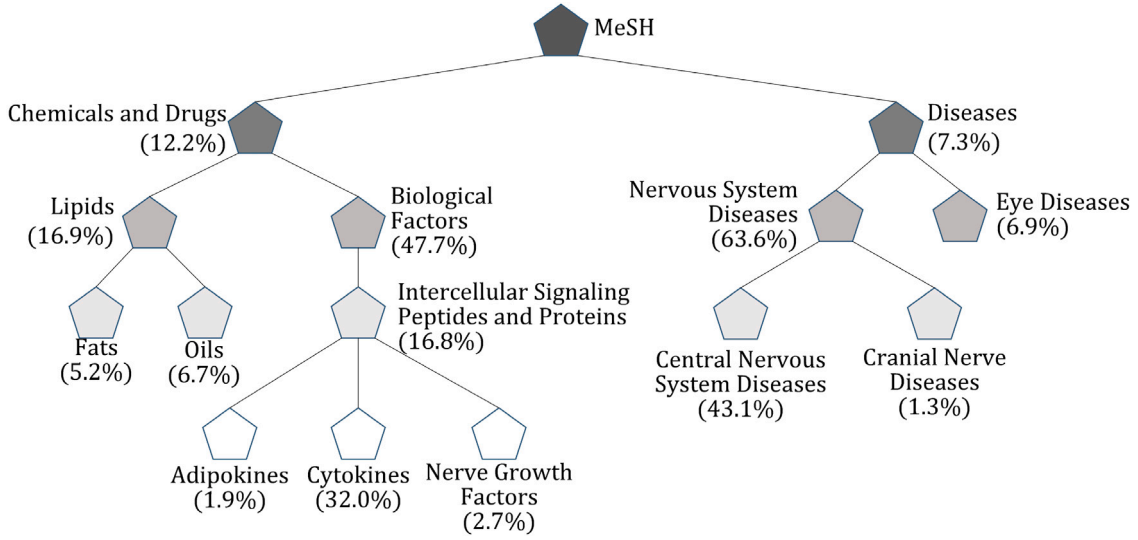


Fig. 1. Partial Medical Subject Headings Classification System with Hierarchical Labels: Percentage represents the frequency of partial child nodes under the parent node.

Nevertheless, with the increase of labels, the balance of label distribution is difficult to be guaranteed. There will have many labels that are difficult to predict by only using the node's attributes. There is data volume imbalance under different hierarchical labels, which the existing network embedding approaches have largely neglected. For example, PubMed[1] organizes papers through the hierarchical classification system Medical Subject Headings (MeSH)[2] Fig. 1 presents partial hierarchical labels of MeSH, and the frequency of different child nodes under the parent node is also marked. The difference between the number of papers under the "Neurological Disease" label and the number under the "Eye Disease" label is significant, as the "Neurological Disease" label occurs almost ten times more frequently under the "Disease" category than the "Eye Disease" label. Without considering data volume imbalance under different hierarchical labels, the label distribution information will be lost in network embedding.

On the whole, in this paper, we study the problem of fusing hierarchical labels and attributed network structures. We aim at answering the following questions. (1) How to construct the association between node attributes and label space to realize the fusion of hierarchical label and node structure. (2) How to fit the label distribution and reduce the influence of data volume imbalance under different hierarchical labels.

For the first challenge, the neighborhood structure of the network is usually closely related to an underlying hierarchical label (Ma et al., 2018), and the attributes of nodes are also correlated with its labels, illustrated via an example in Fig. 2. For text attributes, the text semantics are usually similar to the meaning of the label, just as the exact colored text and the label in Fig. 2 are semantically related. Typically, nodes are labeled by their attribute information. Thus, we can use attributes as a bridge to associate and fuse network structure and hierarchical labels.

For the second challenge, the sibling label is helpful in enhancing the low-frequency labels representation. Since some labels belonging to the same parent label are concurrent or have a sibling relationship, such as "Fats" and "Adipose Tissue". Although the two labels are attached to different parent labels, they often appear together and have similar meanings. More related information can be obtained from the co-occurrence sibling label of low-frequency labels to strengthen the learning of low-frequency label representation. The hierarchy and sibling relationship is formulated as a directed graph, and the prior probabilities of label dependencies can be used to aggregate label information. Moreover, the relations between two remotely connected labels (that are several hops away from each other) will be strengthened if the two labels share standard labels in the hierarchy.

We present the **H**ierarchical Label and **A**ttributed **N**etwork **S**tructure Fusion (HANS) model by investigating these questions, which simultaneously models and leverages the hierarchical label to generate network embedding. This model is composed of two phases. In the first phase, we propose a novel encoder for modeling the hierarchical label dependencies and introducing the sibling relationship to fitting the label distribution to reduce the impact of the data volume imbalance. In the second phase, an attention based fusion module is constructed to effectively learn the joint fusion between the node and label spaces and generate node embedding. Specifically, this module conducts multi-label attention with attributes for label-aware node embedding and updates the label embedding and node embedding across the strong representation power of deep neural networks to capture the complex correlations of the two information sources, which is composed of a neighbor enhancement autoencoder and skip-gram model.

The main contributions of our work are summarized as follows.

- Propose an attributed network embedding framework HANS, which learns a unified embedding representation by incorporating hierarchical labels into network embedding. To be more specific, HANS leverages the multi-label attention for label-aware node embedding and updates the label embedding and node embedding across a neighbor enhancement autoencoder and skip-gram model.
- We design a directed hierarchy structure encoder that models the label dependencies, considers the sibling relationship to fit the label distribution and generates the hierarchical label embedding.
- Experimental results validate the effectiveness of HANS on three hierarchical classified real-world datasets through the node classification and link prediction task. The results show that HANS improves the effectiveness of the baseline methods on the classification and link prediction task.

The remainder of this paper is organized as follows. Section 2 briefly reviews some related work. Section 3 introduces some concepts to formalize the problem. Section 4 discusses the specific details of the proposed model HANS. We report and analyze the experimental results of the proposed model in Section 5. Finally, we summarize our conclusions and outline directions for future work in Section 6.

2. Related work

In this section, we review related state-of-the-arts of network embedding from three aspects, including unsupervised network embedding methods, flat label guided network embedding methods and hierarchical label guided network embedding methods.

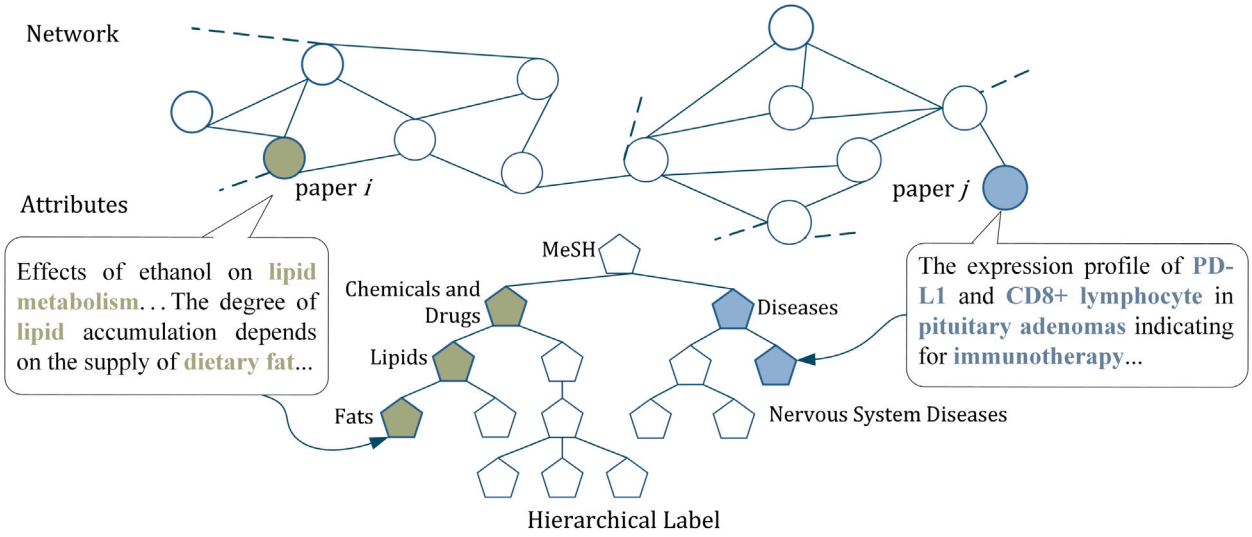


Fig. 2. An Instance of the Relationship between Attributes and Hierarchical Labels: Each network node is attached with text attributes. We mark words that are semantically similar to the hierarchical label. The same color labels in the figure correlate with the text.

Unsupervised Network Embedding Methods. Representative works for structure-only network embedding include DeepWalk (Perozzi et al., 2014), Node2vec (Grover and Leskovec, 2016), LINE (Tang et al., 2015), Etc., which generates corpus based on random walks or edge sampling to explore diverse neighborhoods efficiently. These methods utilize the Skip-Gram (Mikolov et al., 2013) for node embedding and can be implicitly approximated by matrix factorization. NECS (Li et al., 2019a) preserves the high order proximity and incorporates the community structure in node representation learning. DistNE (Lin, 2021) recursively partitions a graph into several small-sized subgraphs to capture the internal and external structural information of nodes, and obtain the embeddings of nodes in a linear cost by aggregating the outputs on all subgraphs.

Attributed network embedding aims to preserve the original network topological structure and node attribute proximity in the low-dimensional network embedding for nodes. ANRL (Zhang et al., 2018b) and DANE (Gao and Huang, 2018) propose their own customized deep neural network architectures to learn node embeddings while capturing the underlying high non-linearity in topological structure and attributes. PGE (Hou et al., 2019) uses node clustering to assign biases to differentiate neighbors of a node and leverages multiple data-driven matrices to aggregate the property information of neighbors sampled based on a biased strategy. GraphZoom (Deng et al., 2020) allows any existing embedding methods to be applied to the coarsened graph before progressively refining the embeddings obtained at the coarsest level to increasingly finer graphs. CFANE (Pan et al., 2021) seamlessly enjoys the advantages of both propagation-based methods and encoder–decoder methods.

In addition, there also has been some efforts exploring representation learning in heterogeneous information networks (Cen et al., 2019; Xie et al., 2021). SHINE (Wang et al., 2018) extracts users' latent representations from heterogeneous networks and utilizes multiple deep autoencoders to map each user into a low-dimension feature space while preserving the network structure. HERec (Shi et al., 2018) uses a meta-path based random walk strategy to generate meaningful node sequences to learn network embedding that is first transformed by a set of fusion functions and subsequently integrated into an extended matrix factorization (MF) model. Wang et al. (2021) propose a method that does not require meta-paths and uses very few non-sensitive parameters across all the domains and down-streaming tasks.

Flat Label Guided Network Embedding Methods. This kind of method uses flat label information to enrich the information contained in the network embedding. TriDNR (Pan et al., 2016) captures three kinds of relations in networks. It utilizes the Skip-Gram model to train node, content, and label co-occurrence. LANE (Huang et al., 2017) projects the attributed network and its associated labels into a unified embedding space while preserving their correlations. RECT (Wang et al., 2020) is a new class of graph neural networks that benefit from the completely imbalanced labels and explores the knowledge of class-semantic descriptions.

The graph neural network based methods are also supervised or semi-supervised, which learn the node feature representations with the information of labels (Chen et al., 2020b; Xue et al., 2020; Jin et al., 2021). GCN (Kipf and Welling, 2017) incorporates neighbors' feature representations into the node feature representation using convolutional operations. GCN-LASE (Li et al., 2019b) is a novel GCN model taking both node and link attributes as inputs. AdaGCN (Sun et al., 2021) introduces AdaBoost into network computing, which can effectively extract knowledge from the high-order neighbors of current nodes and then integrate knowledge from different hops of neighbors into the network. Nevertheless, these methods generally ignore the multi-level characteristics of data with hierarchical labels.

Hierarchical Label Guided Network Embedding Methods. This kind of method introduces the hierarchical label structure into the network embedding. NetHiex (Ma et al., 2018) is a network embedding model that captures the latent hierarchical taxonomy. It builds a taxonomy tree by the network structure but does not use the existing hierarchical label information. TaxoGAN (Yang et al., 2020) mainly models the two novel important properties of conditional node proximity and hierarchical label proximity, which considers the structure relation between the network and the hierarchical label but ignores the data volume imbalance.

3. Problem definition

We first give basic notations and definitions in this work. Let $G = (V, E, X)$ be an attributed network, where V denotes the set of $|V|$ nodes and E represents the set of $|E|$ edges. $X \in \mathbb{R}^{|V|\times q}$ is a matrix that encodes all node attributes information, where q denotes the number of attributes, and $\boldsymbol{x}_v \in \mathbb{R}^{1\times q}$ describes the attribute vector associated with node v.

Definition 1 (*Attributed Network Embedding*)**.** Given an attributed network $G = (V, E, X)$, we aim to represent each node $v \in V$ as a low-dimensional vector $\boldsymbol{z}_v$ by learning a mapping function $f : G \rightarrow Z \in \mathbb{R}^{|V|\times d}$, where $d \ll |V|$ and the mapping function f preserves the network's essential properties.

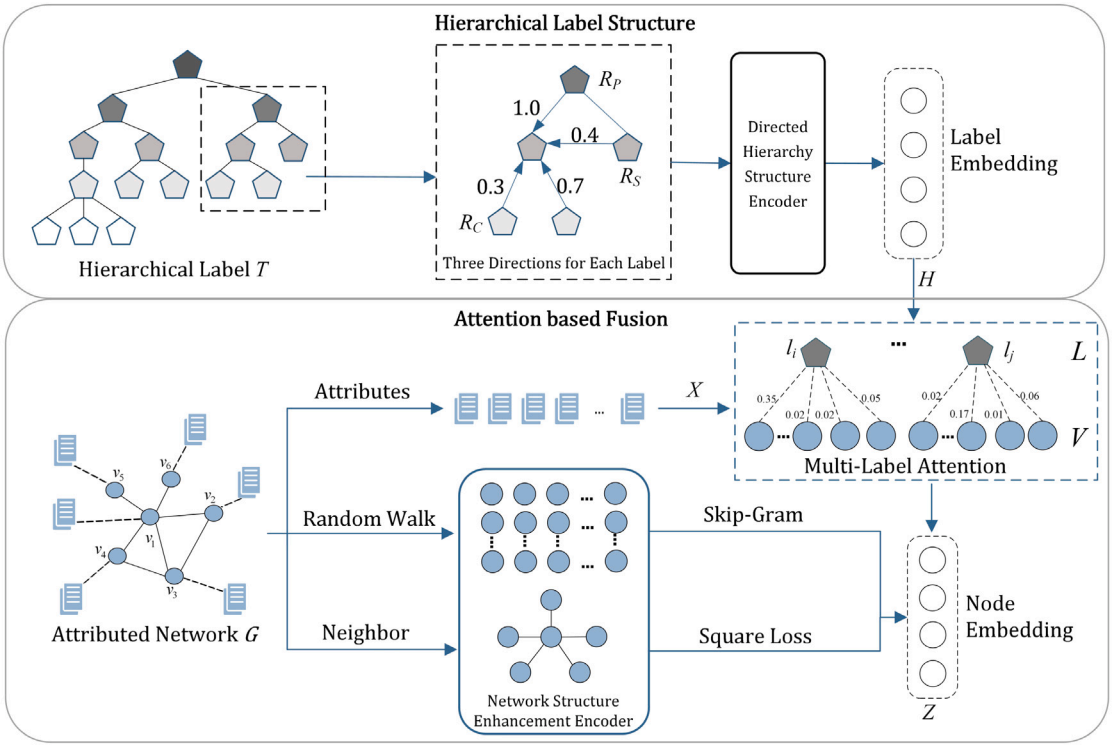


Fig. 3. The framework of HANS.

Definition 2 (*Flat Label Guided Network Embedding*)**.** Given a network associated with flat labels (G, L), it aims to represent each node $v \in V$ as a continuous low-dimensional vector representation $\boldsymbol{z}_v$ by learning a mapping $f : (G, L) \rightarrow Z \in \mathbb{R}^{|V|\times d}$ in a way that incorporates with the label information.

Definition 3 (*Hierarchical Labels*)**.** Let hierarchical labels be a directed tree defined as $T = (L, R_P, R_C, R_S)$, where L presents the set of flat labels $L = \{l_1, l_2, \ldots, l_{|L|}\}$. $R_P = \{(l_i, l_j)|l_i \in L, l_j \in child(l_i)\}$ is the top-down hierarchy path. $R_C = \{(l_j, l_i)|l_i \in L, l_j \in child(l_i)\}$ denotes the bottom-up hierarchy path. $R_S = \{(l_i, l_k)|l_i \in L, l_k \in sib(l_i)\}$ is the sibling relationship, where $sib(l_i)$ denotes the label that appears with label l_i and their parent label at the same level.

Problem formulation. Given an attributed network associated with hierarchical labels (G, T), we aim to represent each node $v \in V$ as a continuous low-dimensional vector representation $\boldsymbol{z}_v$ by learning a mapping $f : (G, T) \rightarrow Z \in \mathbb{R}^{|V|\times d}$ in a way that generates the hierarchical labels-aware node embedding and the hierarchy-aware label embedding $\boldsymbol{h}_l$.

4. The proposed method: HANS

This section presents an attributed network embedding framework that integrates hierarchical labels into network structure and nodes attributes for the node embedding and hierarchical label embedding named HANS. The overall framework of HANS is shown in Fig. 3. The directed hierarchy structure encoder models the hierarchical label as a three-directed tree (parent–child R_P, child–parent R_C, and sibling relationship R_S). This encoder aims to generate hierarchical label embedding by fitting the label distribution and reducing the impact of data imbalance. An attention based fusion module effectively learns the mutual interactions between the node space and the hierarchical label space. To illustrate the attention based fusion module's implementation in more detail, we explain how the label embedding and node embedding updates through a neighbor enhancement autoencoder and skip-gram model to capture the complex correlations of the label-aware attributes and network structure.

4.1. Hierarchical label structure

One of the essential properties for a network with a hierarchical label is the hierarchical relations among labels. For the labels linked with their parent label, we should ensure that their embeddings capture the bidirectional information between the parent–child labels. However, some sibling labels co-occurrence or have a causal relationship, enhancing the feature of low-frequency labels and fitting the distribution of labels. Thus, we build a directed hierarchy structure encoder to model label dependencies and the sibling relationship among the same level.

4.1.1. Sibling and hierarchical prior probability distributions

For the hierarchical label tree, the parent node directly affects the child nodes, and the occurrence frequency of different children under the parent is different. Given a hierarchical label tree T with $|L|$ nodes, we can represent the graph using an $|L| \times |L|$ adjacency matrix A. To capture the sibling correlations between labels, we add the sibling edges into the hierarchical label tree. If the label l_i appears with the label l_j at the same level, they have an edge. Based on Bayesian statistical inference, HANS introduces the prior knowledge of label correlations regarding the predefined hierarchy and corpus. We exploit the prior probability of label dependencies as prior hierarchy knowledge:

$$A_{i,j} = P(l_i|l_j) = \begin{cases} \frac{|V_{l_i}|}{|V_{l_j}|} & (l_i, l_j) \in R_C \\ 1 & (l_i, l_j) \in R_P \\ \frac{|V_{l_i} \bigcap V_{l_j}|}{|V_{l_j}|} & (l_i, l_j) \in R_S \end{cases} \tag{1}$$

The relation between label l_i and l_j can be the top-down, the bottom-up, and the co-occurrence labels restricted at the same level. $A_{i,j}$ equating $P(l_i|l_j)$ is the weights between label l_i and l_j, which is the conditional probability of l_i given that l_j occurs, which presents different directions information of label l_i. The $|V_{l_k}|$ denotes the number of nodes labeled with l_k. Intuitively, for edges in R_C or R_P, if l_j co-appears with l_i more often than l_k, the probabilities of $P(l_i|l_j)$ should have more substantial dependencies.

4.1.2. Directed hierarchy structure encoder

Since the hierarchical label tree is also a topological graph, we propose a directed hierarchy structure encoder to model three directions (parent–child, child–parent, and sibling) information introduced into the hierarchy tree and generate the hierarchical label embedding. Directed hierarchy structure encoder aggregates the hierarchical label information within the relations R_P, R_C, R_S, and self-loop edges. Those information should conduct node transformations with edge-wise linear transformations. We simplify this transformation with a weighted adjacent matrix A, divided into three parts $A_P \in \mathbb{R}^{|L|\times|L|}$, $A_C \in \mathbb{R}^{|L|\times|L|}$ and $A_S \in \mathbb{R}^{|L|\times|L|}$ to present edges concerning R_P, R_C and R_S. It is usually necessary to incorporate a self-loop to avoid ignoring the feature of the label itself. The self-loop edges are presented with I. Formally, this encoder encodes the hidden state of label l_i based on its associated neighborhood:

$$\begin{aligned}
U_C &= (A_C \bigotimes H^{k-1} + \boldsymbol{b}_{A_C}^{k-1}) \bigodot \sigma(H^{k-1} \bigotimes W_C^{k-1} + \boldsymbol{b}_{W_C}^{k-1}) \\
U_S &= (A_S \bigotimes H^{k-1} + \boldsymbol{b}_{A_S}^{k-1}) \bigodot \sigma(H^{k-1} \bigotimes W_S^{k-1} + \boldsymbol{b}_{W_S}^{k-1}) \\
U_P &= (A_P \bigotimes H^{k-1} + \boldsymbol{b}_{A_P}^{k-1}) \bigodot \sigma(H^{k-1} \bigotimes W_P^{k-1} + \boldsymbol{b}_{W_P}^{k-1}) \\
U_I &= (I \bigotimes H^{k-1} + \boldsymbol{b}_I^{k-1}) \bigodot \sigma(H^{k-1} \bigotimes W_I^{k-1}) \\
H^k &= ReLU(U_C + U_S + U_P + U_I)
\end{aligned} \tag{2}$$

where $W \in \mathbb{R}^d$, $\boldsymbol{b}_A \in \mathbb{R}^{|L|\times d}$, $\boldsymbol{b}_W \in \mathbb{R}^{|L|}$ and $\boldsymbol{b}_I \in \mathbb{R}^{|L|\times d}$. U_C, U_S, U_P and U_I are the label representations calculated from the adjacency matrix under different orientations (A_C, A_S, A_P, and I). H^k presents the hidden state of labels at the kth hidden layer of this encoder. Since we ignore the description text of labels, every label has no prior features and H^0 is randomly initialized. Finally, the output of the directed hierarchy structure encoder is $\boldsymbol{h}_l$ denoting its label embedding corresponding to the hierarchy structural information.

Based on the directed hierarchy structure encoder, HANS avoids the hierarchical label embedding learning level by level and captures the global structure across various levels. Label embeddings are enhanced with bidirectional hierarchical information and sibling relationship information. This encoder implements learning label embedding across different levels in a single model.

4.2. Attention based fusion

After the directed hierarchy structure encoder, we obtain the label embedding $\boldsymbol{h}_l$. It is necessary to capture these associations between node attributes and the hierarchical label structure. The multi-label attention is leveraged to integrate the network and label structural features. Then learn the hierarchical label-aware node embedding. A neighbor enhancement autoencoder and skip-gram model used as the objective function is proposed to capture the complex correlations of the two information sources.

4.2.1. Multi-label attention

Due to the fact that different labels may focus on different node attributes, we utilize multi-label attention to learn label-aware node embedding that fuses node attributes and hierarchical label embedding. First, the attributes are fed into an arbitrary attribute encoder to generate a node feature vector with the same dimensionality as the hierarchical label embedding. Then, we generate the attention weight via vector product:

$$\alpha_{jk} = \frac{\exp(\boldsymbol{x}_j \cdot \boldsymbol{h}'_{l_k})}{\sum_{j=1}^{|V|} \exp(\boldsymbol{x}_j \cdot \boldsymbol{h}'_{l_k})} \tag{3}$$

where α_{jk} indicates how informative the feature vector of node v_j is for the label l_k, $\boldsymbol{x_j}$ is the attribute information associated with node v_j, and $\boldsymbol{h}_{l_k}$ presents the label embedding associated with label l_k. The $\boldsymbol{h}'$ means the transpose of the vector.

Finally, we use the attribute representation $\boldsymbol{x}_j$ and the attention weight α_{jk} to generate the hierarchical label-aware node embedding:

$$\boldsymbol{z}_j = \sum_{k=1}^{|L|} \alpha_{jk} \cdot \boldsymbol{x}_j. \tag{4}$$

$\boldsymbol{z}_j$ is the embedding of node v_j. The information of hierarchical labels and the attributes are integrated into attributed network embedding, and the attention bias between different labels and node attributes is considered.

4.2.2. Network structure enhancement encoder

Now that the hierarchy structure and attributes have been compressed into the node embedding $\boldsymbol{z}$. To employ the network structure to optimize node embedding and realize the fusion of hierarchical labels, attributes, and network structure, we design a network structure enhancement encoder to facilitate the representation learning procedure. For the network node, we aim to reconstruct its neighbors instead of the node itself. For the node v_j with its embedding $\boldsymbol{z}_j$ and the neighbors function $nei(\cdot)$, our goal is to minimize the following autoencoder loss function:

$$\mathcal{L}_{ne} = \sum_{j=1}^{|V|} \|\boldsymbol{z}_j - \frac{1}{|nei(v_j)|} \sum_{k\in nei(v_j)} \boldsymbol{z}_k\|_2^2 \tag{5}$$

This autoencoder retains better proximity among nodes. The resulting representations are more effective to variations since it constrains closely located nodes to have similar representations by forcing them to refactor similar neighbors. Thus, the local network structure information is fused into node embedding.

Except for the neighbor constraints, a skip-gram model is employed to compute local community structure information in time sublinear to the size of the input graph. The skip-gram module assumes nodes with similar contexts should be similar in latent semantic space. Expressly, we set the following log probability of skip-gram model minimized as the objective function by giving current node v_j with its associated embedding $\boldsymbol{z}_j$ for all random walk contexts $s \in S_k$:

$$\mathcal{L}_{sg} = -\sum_{j=1}^{|V|} \sum_{s\in S_k} \sum_{-m\leqslant k\leqslant m} log(p(v_{j+k}|\boldsymbol{z}_j)) \tag{6}$$

where m is the window size of the skip-gram model and v_{j+k} is the node context in the generated sequence. $p(v_{j+k}|\boldsymbol{z}_j)$ is the likelihood of target context given the node embedding and defined as:

$$p(v_{j+k}|\boldsymbol{z}_j) = \frac{\exp(\boldsymbol{z}'_{j+k}\boldsymbol{z}_j)}{\sum_{v=1}^{|V|} \exp(\boldsymbol{z}'_v\boldsymbol{z}_j)} \tag{7}$$

Directly optimizing Eq. (7) requires the summation over the entire set of nodes. We adopt the sample softmax (Jean et al., 2015) instead of $p(v_{j+k}|\boldsymbol{z}_j)$ that allows us to compute the normalization constant during training using only a small subset of the target nodes, resulting in much lower computational complexity for each parameter update.

HANS can be a supervised or unsupervised model. If used as a supervised model, the node embedding is fed into a fully connected layer for prediction. For multi-label classification, HANS uses a binary cross-entropy loss function:

$$\mathcal{L}_c = -\sum_{j=1}^{|V|} \sum_{k=1}^{|L|} [y_{jk} log(\hat{y}_{jk}) + (1 - y_{jk}) log(1 - \hat{y}_{jk})] \tag{8}$$

where y_{jk} and $\hat{y}_{jk}$ are the ground truth and sigmoid score for the label l_k of the node v_j. Finally, the objective function of the supervised HANS model is formulated as the combination:

$$\mathcal{L} = \mathcal{L}_c + \mathcal{L}_{ne} + \mathcal{L}_{sg} \tag{9}$$

In this way, HANS preserves node attributes, network structure, and the hierarchical labels information in a unified framework. We summarize our algorithm in Algorithm 1. The algorithm follows exactly the three phases that we describe above.

Algorithm 1 The algorithm of HANS

Input: attribute network $G = (V, E, X)$, hierarchical labels tree T
Output: network embedding Z, hierarchical label embedding H

Generate the weighted adjacent matrix A with Eq.(1)
while not converged **do**
  //Directed Hierarchy Structure Encoder
  Learn hierarchical label embedding H with weighted adjacency matrix of three directions A_P, A_C, A_S via Eq.(2)
  //Attention based Fusion Module
  for each node $v_j \in V$ **do**
    //Multi − Label Attention
    for each label $l_k \in L$ **do**
      Generate the attention weight α_{jk} via Eq.(3)
      Generate the hierarchical label-aware node embedding z_j via Eq.(4)
    end for
    //Network Structure Enhancement Encoder
    Sample a mini-batch of nodes with its context
    Compute the gradient of $\bigtriangledown L_{ne}$ based on Eq.(5)
    Update autoencoder module parameters
    Compute the gradient of $\bigtriangledown L_{sg}$ based on Eq.(6)
    Compute the gradient of $\bigtriangledown L_c$ based on Eq.(8)
    Update all parameters
  end for
end while
return network embedding matrix Z, hierarchical label embedding matrix H

5. Experiments

We evaluate the effectiveness of the HANS method on the benchmark application, node classification and linking prediction, which is a commonly used task for many existing networks embedding methods evaluation.

5.1. Datasets

In our experiments, we employ three benchmark datasets: Patent, Cora and PubMed. We introduce these three datasets as follows:

The Patent is a cite network from USPTO. Each patent document is labeled as at least one Cooperative Patent Classification scheme (CPC) number, a hierarchical classification method. We use the first three CPC number levels as hierarchical labels because of the excessive number of leaves in the CPC classification tree. The Patent dataset includes 10,744 real-world granted US patents, and we use the title and abstract as attributes. In the first level of CPC, most patents are under the labels "physics" and "electricity", which together account for nearly 50%. We perform multi-level labels classification and linking prediction tasks on this dataset.

The Cora dataset is a cite network provided by Andrew McCallum.[3] We clean up the "Cora Research Paper Classification" file to extract the citation relation, title, abstract text, and hierarchical classification information. After collating the data, 20,402 papers remained, each with two levels of labels. In the first level, the number of papers labeled "Artificial Intelligence" was the largest, accounting for 39.58%. The number of papers labeled "Information Retrieval" was the least, which accounted for 2.40%. We perform multi-level labels classification and linking prediction tasks on this dataset.

PubMed is a dataset consisting of more than 32 million cites for biomedical literature from MEDLINE, life science journals, and online books. Each paper is labeled at least one label, and these labels are the MeSH classification tree nodes. The label levels for each node are not fixed, and we extract the data with a label level of three. Overall, we extracted 103,002 papers to construct a cite network, and each paper is associated with title and abstract content. In the first level of MeSH, the number of papers labeled as "organism" was the largest, while the number labeled as "humanities" was the least, a difference of more than 50 times. We perform multi-level labels classification and linking prediction tasks on this dataset.

The statistics of datasets are listed in Table 1.

5.2. Baselines

In order to demonstrate the effectiveness of HANS, we categorize our competitors into the following three groups:

Unsupervised Network Embedding Methods We compare with DeepWalk (Perozzi et al., 2014), Node2vec (Grover and Leskovec, 2016), ANRL (Zhang et al., 2018b), PGE (Liu et al., 2018), DANE (Gao and Huang, 2018), and GraphZoom (Deng et al., 2020). DeepWalk and Node2vec use truncated random walks to generate node sequences. Then the node sequences are feed into the skip-gram model to learn the latent node representations. ANRL models the node attribute information and an attribute-ware skip-gram model based on the attribute encoder to capture the network structure. PGE assigns biases to differentiate neighbors of a node and leverages multiple data-driven matrices. PGE can be used for supervised and unsupervised studies. Considering the label information, we set a supervised study for PGE. DANE captures the high nonlinearity and preserves various proximities in both topological structure and node attributes. GraphZoom is a framework that aims to improve both the performance and scalability of graph embedding techniques.

Flat Label Guided Network Embedding Methods We compare with RECT (Wang et al., 2020), GAT (Velickovic et al., 2018) and GCN (Kipf and Welling, 2017). RECT is a new class of graph neural network that benefits from the completely imbalanced labels and explores the knowledge of class-semantic descriptions.

GAT is a neural network architecture that operates on graph-structured data, leveraging masked self-attentional layers to address the shortcomings of prior methods based on graph convolutions or approximations. GCN presents a scalable approach to graph-structured data based on an efficient variant of convolutional neural networks.

Hierarchical Label Guided Network Embedding Methods We compare with NetHiex (Ma et al., 2018) and TaxoGAN (Yang et al., 2020). NetHiex is a network embedding model that captures the latent hierarchical taxonomy. It builds a taxonomy tree by the network structure but does not use the existing hierarchical classification information. TaxoGAN mainly models the two novel important properties of conditional node proximity and hierarchical label proximity.

Following a commonly adopted way, we validate the effectiveness of different learned embeddings on node classification tasks and linking prediction tasks. We repeat the experiments five times and report the averaged performance.

We set all baseline methods except with $d = 128$ dimension on Patent and Cora, and with $d = 256$ dimension on PubMed dataset. For the methods that need random walk (Grady, 2006) results, we set the walk numbers with 10 for all datasets, the walk length 10 for the Patent and Cora datasets, and 50 for the PubMed dataset. All baseline methods are implemented using source code provided by the papers, and we adjust the parameters in each method to obtain the best results.

5.3. Experimental results and analysis

5.3.1. Node classification task

In this section, we set up multiple groups of experiments to prove the effectiveness of HANS. Following existing work (Perozzi et al., 2014), we train a logistic regression classifier on the learned embeddings to predict the node labels. In all classification experiments, we predict labels at all levels, not just leaf labels. To measure the node

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

Table 1
The statistics of datasets.

Dataset	nodes	edges	attributes	labels	levels	level 1	level 2	level 3
Patent	10,744	19,678	30,960	653	3	9	125	519
Cora	20,402	71,238	21,754	80	2	10	70	/
PubMed	103,002	298,911	61,948	8,024	3	112	1,693	6,219

Table 2
Classification Results on Patent Dataset.

Algorithm	90%		70%		50%		30%		10%	
	$Micro\text{-}F1$	$Macro\text{-}F1$	$Micro\text{-}F1$	$Macro\text{-}F1$	$Micro\text{-}F1$	$Macro\text{-}F1$	$Micro\text{-}F1$	$Macro\text{-}F1$	$Micro\text{-}F1$	$Macro\text{-}F1$
DeepWalk	0.4778	0.2935	0.4596	0.3363	0.4473	0.3278	0.4218	0.2970	0.3591	0.2093
Node2vec	0.4817	0.3006	0.4795	0.3454	0.4705	0.3371	0.4435	0.2972	0.3726	0.1918
ANRL	0.5671	0.2347	0.5463	0.2419	0.5361	0.2342	0.5181	0.2054	0.4665	0.1341
DANE	0.5248	0.1127	0.5217	0.1082	0.5107	0.0943	0.4974	0.0740	0.4517	0.0426
GraphZoom	0.5123	0.2998	0.5047	0.3488	0.4910	0.3473	0.4645	0.3157	0.4073	0.2209
PGE	0.5809	0.2829	0.5777	0.1810	0.5630	0.2120	0.5524	0.1415	0.4588	0.0540
RECT	0.5536	0.1786	0.5412	0.1749	0.5358	0.1683	0.5189	0.1533	0.4746	0.1341
GAT	0.5130	0.1383	0.5123	0.1226	0.5089	0.1336	0.5033	0.1074	0.4183	0.1123
GCN	0.5392	0.1079	0.5278	0.1088	0.5079	0.1035	0.4972	0.0605	0.4257	0.0247
Nethiex	0.5803	0.3295	0.5620	0.3587	0.5633	0.3757	0.5364	0.3390	0.4293	0.2362
TaxoGAN	0.6050	0.2490	**0.6022**	0.2623	0.5727	0.2040	0.4439	0.1154	0.3391	0.0186
HANS	**0.6114**	**0.3348**	0.5982	**0.3782**	**0.5842**	**0.3726**	**0.5688**	**0.3368**	**0.4975**	**0.2482**

Table 3
Classification Results on PubMed and Cora Dataset.

Algorithm		DeepWalk	Node2Vec	ANRL	DANE	GraphZoom	PGE	RECT	GAT	GCN	Nethiex	TaxoGan	HANS
Cora	$Micro\text{-}F1$	0.6250	0.6419	0.4533	0.4272	0.6562	0.6431	0.6340	0.6233	0.6429	0.6646	0.5437	**0.6870**
	$Macro\text{-}F1$	0.5083	0.5144	0.1726	0.1277	0.3857	0.3927	0.3377	0.3925	0.4189	0.4877	0.4215	**0.5244**
PubMed	$Micro\text{-}F1$	0.4711	0.4624	0.4725	0.5092	0.4749	0.5206	0.4847	0.4012	0.3987	0.4572	0.5138	**0.5362**
	$Macro\text{-}F1$	0.1901	0.1724	0.1522	0.1249	0.1696	0.1098	0.0347	0.1019	0.0433	0.1170	0.1433	**0.1981**

classification results, we employ $Micro\text{-}F1$ and $Macro\text{-}F1$ as the metric, and the calculation formulas are:

$$Precision_{micro} = \frac{\sum_{l\in L} TP_l}{\sum_{l\in L} TP_l + \sum_{l\in L} FP_l} \qquad Recall_{micro} = \frac{\sum_{l\in L} TP_l}{\sum_{l\in L} TP_l + \sum_{l\in L} FN_l} \tag{10}$$

$$Micro - F1 = 2 \cdot \frac{Precision_{micro} \cdot Recall_{micro}}{Precision_{micro} + Recall_{micro}} \tag{11}$$

$Micro\text{-}F1$ first calculates the total $Precision$ and $Recall$ of all labels. Because it considers the number of various labels, it is more suitable for cases where the data distribution is imbalanced. In this case, labels with a large amount of data will significantly impact $Micro\text{-}F1$.

$$\begin{aligned} Precision_{macro} &= \sum_{l\in L} \frac{TP_l}{TP_l + FP_l} \;/\; |L| \\ Recall_{macro} &= \sum_{l\in L} \frac{TP_l}{TP_l + FN_l} \;/\; |L| \end{aligned} \tag{12}$$

$$Macro - F1 = 2 \cdot \frac{Precision_{macro} \cdot Recall_{macro}}{Precision_{macro} + Recall_{macro}} \tag{13}$$

$Macro\text{-}F1$ first calculates the average $Precision$ and $Recall$ for each label. The distribution of labels is not taken into account. In this case, labels with higher $Precision$ and $Recall$ will have a more significant impact on $Macro\text{-}F1$.

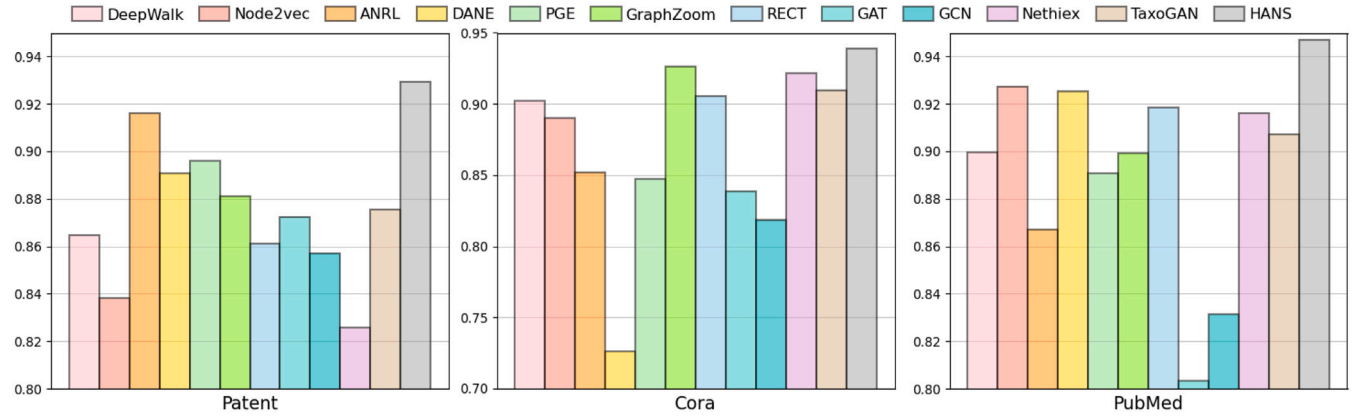
Table 2 shows the results with training rates from 90% to 10% for all baseline methods on the Patent dataset. In Table 2, our proposed HANS can achieve the best overall performance among all the methods, even when the labeling ratio is only 10% and 30%. HANS keeps the local community structure information and the neighbor constraints which can also help HANS learn the effective embedding results. Table 3 reports the results of all baseline methods when trained with 50% data on the Cora dataset and the PubMed dataset. HANS achieves higher $Micro\text{-}F1$ and $Macro\text{-}F1$ scores than all the other methods.

It is important to note that the Cora dataset has only 80 labels and only two levels of labels. Compared with the Cora dataset and the other two datasets, the method without hierarchical labels can achieve a similar effect as the method with the hierarchical label when the number of labels is reduced. Nevertheless, this highlights the importance of considering hierarchical label structures when the number of labels increases. Compared with Patent and Cora, the labels of the PubMed dataset have increased by ten times to hundred times, which leads to a much lower score for $Macro\text{-}F1$. Moreover, compared with the graph convolution methods in the PubMed dataset, the network embedding methods achieve better results while better preserving the network structure information.

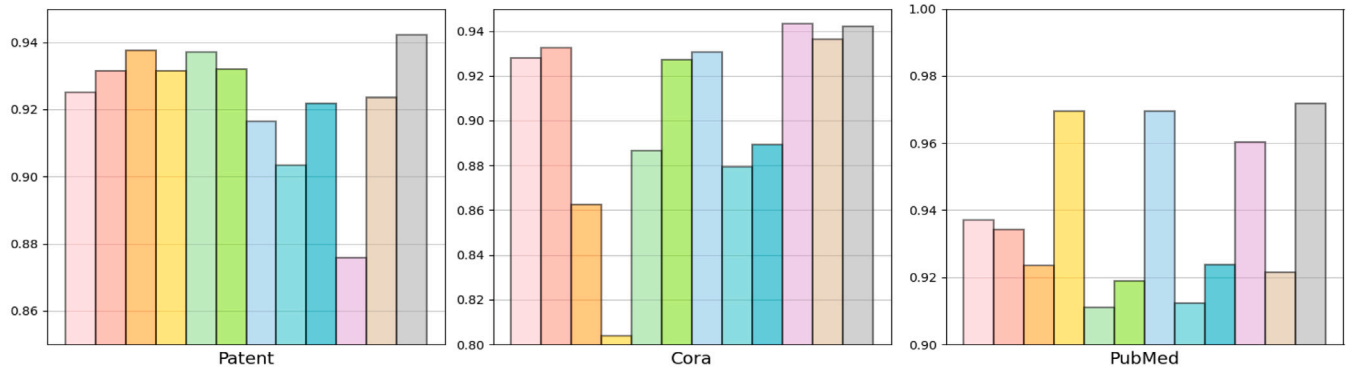
Combined with Tables 2 and 3, HANS outperforms all the competitors. Nethiex, which considers the potential hierarchical structure of labels, achieves better classification results on the Patent dataset than other structure-only network embedding methods. Nevertheless, it has poor performance on the PubMed dataset. Nethiex attempts to obtain and construct the potential hierarchical structure but does not use the existing hierarchical label information of the network. When the number of labels increases, the potential hierarchical structure is hard to construct. TaxoGAN and HANS all introduce the hierarchical label into network embedding, but TaxoGAN only fusion the network structure and hierarchical label and does not consider the relationship between sibling nodes. According to the results of the Cora dataset, TaxoGAN relies too much on the up-down structure of hierarchical labels, resulting in its performance degradation in datasets with fewer labels and fewer levels. HANS fused the three directions of the hierarchical label tree to learn the hierarchical label embedding and combined the hierarchical embedding to make the attribute network embedding. Therefore, the learned node embedding is essentially more discriminative.

5.3.2. Link prediction task

Next, we evaluate the quality of node embedding for link prediction. The link prediction task tests the ability of embedding methods to predict unseen network structures. Given two nodes' embeddings $\boldsymbol{z}_v$ and $\boldsymbol{z}_{v'}$, the model should predict whether there is a potential edge existing between them. We used the area under the curve($ROC\text{-}AUC$) and the



(a) *ROC-AUC*



(b) *PR-AUC*

Fig. 4. Link Prediction Results.

area under the *Precision-Recall*(PR-AUC) to evaluate link prediction performance, commonly used in related literature. We compared HANS with all baselines on three datasets. For three datasets, we sampled 30% edges for the test and set the epoch number to 5 to avoid the data insufficiency problem.

Fig. 4 reports the link prediction performance of our HANS and the baselines on the three datasets. On the whole, graph convolution is in the lower reaches of the link prediction task, and its structure retention is weaker than other network embedding methods. As shown, the overall effect of HANS consistently performs better than any of the baseline models, indicating that our inferred embeddings work well in the link prediction task. It is mainly because the network structure enhancement encoder strengthens the neighbor structure mining of the relationship between nodes.

5.3.3. Ablation study

Here, we conduct an ablation study by deleting some of the design encoders shown in Fig. 3. The classification results on Patent, Cora, and Pubmed datasets are shown in Table 4. The link prediction results are shown in Table 5.

In the table, HANS-Flat means substituting the hierarchical label in HANS with only the loss function of the flat label and shielding the directed hierarchy structure encoder in HANS. The result is similar to the supervised network embedding method. Compared with the result of HANS, it shows the effect of directed hierarchy structure encoder and the importance of considering the hierarchical label. Especially for the PubMed datasets with too many labels, the improvement of hierarchical labels is hugely significant.

HANS-Without-Structure means deleting the network structure enhancement encoder and only considering the hierarchical label and attributes to evaluate the effectiveness of this encoder. This structure is similar to the hierarchical text classification method, and its performance in the link prediction task is poor. HANS-Without-Neighbor means ignoring the information of neighbors and deleting the loss $\mathcal{L}_{ne}$. HANS-Unsupervised means learning without supervision and deleting the loss $\mathcal{L}_c$.

As we can see from the table, all four design choices effectively improve the embedding results for both node classification and link prediction.

5.3.4. Visualization

The hierarchical label embedding in the 2-dimensional latent space obtained by directed hierarchy structure encoder are demonstrated respectively in Fig. 5, reduced to 2-dim by standard TSNE. Gray dots are labels in the third level, while red and blue dots are labels in the first and second levels. It can be seen from the figure that the coarser the label granularity, the closer the embedding result is to the center. In contrast, the finest-grained label embedding spreads to the periphery. When we consider the parent–child relationship, we also introduce the sibling relationship, which binds the same level nodes together. HANS well preserves the hierarchical label structure representing the parent–child relations between labels. As expected, the results are highly interpretable and insightful, which provide knowledge about the relative distances among labels.

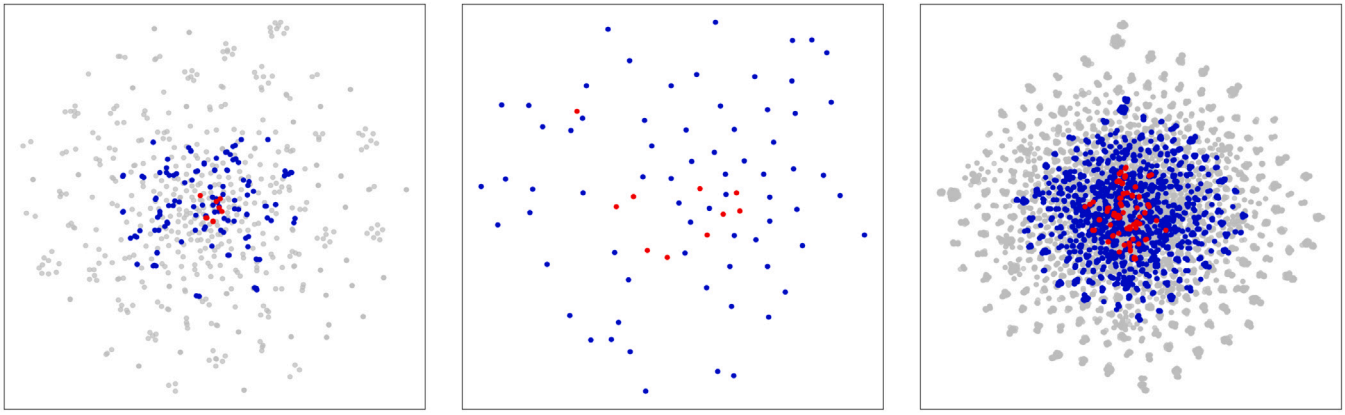
Moreover, due to the parent–child inclusion relationship of hierarchical labels, the embedding distribution of different hierarchical labels should be similar. This is also shown in Fig. 5, especially the second and third levels of labels in Fig. 5(c), which are shown more clearly because the PubMed dataset contains a large number of hierarchical labels. Fig. 5 verifies the fitting effect of HANS for label distribution.

Table 4
Ablation of Classification.

Algorithm	Patent		Cora		PubMed	
	Micro-F1	*Macro-F1*	*Micro-F1*	*Macro-F1*	*Micro-F1*	*Macro-F1*
HANS-Flat	0.5712	0.3569	0.6610	0.5147	0.4867	0.1867
HANS-Without-Structure	0.5662	0.3517	0.6571	0.4645	0.4704	0.1779
HANS-Without-Neighbor	0.5803	0.3629	0.6744	0.5209	0.5308	0.1920
HANS-Unsupervised	0.5411	0.2472	0.6539	0.5146	0.4925	0.1544
HANS	**0.5842**	**0.3726**	**0.6870**	**0.5244**	**0.5362**	**0.1981**

Table 5
Ablation of Link Prediction.

Algorithm	Patent		Cora		PubMed	
	ROC-AUC	*PR-AUC*	*ROC-AUC*	*PR-AUC*	*ROC-AUC*	*PR-AUC*
HANS-Flat	0.8835	0.9274	0.9158	0.9118	0.9331	0.9637
HANS-Without-Structure	0.8282	0.8998	0.7501	0.8639	0.8785	0.8549
HANS-Without-Neighbor	0.9257	0.9338	0.9015	0.9159	0.9212	0.9350
HANS-Unsupervised	0.9216	0.9384	0.9269	0.9376	0.9263	0.9422
HANS	**0.9294**	**0.9423**	**0.9387**	**0.9421**	**0.9471**	**0.9719**



(a) Patent (b) Cora (c) PubMed

Fig. 5. Visualization of Hierarchical Label Embedding.

6. Conclusion

In this paper, we generalize the problem of learning and updating the embedding of attributed networks with hierarchical labels. Then, we propose a novel attributed network embedding framework HANS with the directed hierarchy structure encoder and the attention based fusion module. The directed hierarchy structure encoder models label dependencies and the sibling relationship among the same level. The attention based fusion module introduces a network structure enhancement encoder for modeling network structure and multi-label attention for the attributes, network, and hierarchical labels fusing.

We test HANS with classification and link prediction tasks on three real-world datasets, and the experiment results show that HANS empirically achieves significant improvement on three real-world datasets. We also set the ablation study and the visualization of hierarchical label embedding, which proves the effectiveness of each module in HANS.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant Nos. 61876001, Nos. 61602003, Nos. 61673020 and Nos. 61702003), the National Key Research and Development Program of China (Grant Nos. 2017YFB1401903), the Major Program of the National Social Science Foundation of China (Grant Nos. 18ZDA032) and the State Education Ministry (Forty-ninth batch) and the Recruitment Project of Anhui University for Academic and Technology Leader. The authors acknowledge the High-performance Computing Platform of Anhui University for providing computing resources.

References

Cao, S., Lu, W., Xu, Q., 2015. Grarep: Learning graph representations with global structural information. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. ACM, pp. 891–900.

Cen, Y., Zou, X., Zhang, J., Yang, H., Zhou, J., Tang, J., 2019. Representation learning for attributed multiplex heterogeneous network. In: KDD, Anchorage. AK, USA, August 4-8, 2019, pp. 1358–1368. http://dx.doi.org/10.1145/3292500.3330964.

Chen, B., Huang, X., Xiao, L., Cai, Z., Jing, L., 2020a. Hyperbolic interaction model for hierarchical multi-label classification. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, the Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, the Tenth AAAI Symposium on Educational Advances in Artificial Intelligence. EAAI 2020, New York, NY, USA, February 7-12, 2020, AAAI Press, pp. 7496–7503, URL https://aaai.org/ojs/index.php/AAAI/article/view/6247.

Chen, M., Wei, Z., Huang, Z., Ding, B., Li, Y., 2020b. Simple and deep graph convolutional networks. In: Proceedings of the 37th International Conference on Machine Learning. ICML 2020, 13-18 July 2020, Virtual Event, In: Proceedings of Machine Learning Research, vol. 119, PMLR, pp. 1725–1735, URL http://proceedings.mlr.press/v119/chen20v.html.

Deng, C., Zhao, Z., Wang, Y., Zhang, Z., Feng, Z., 2020. GraphZoom: A multi-level spectral approach for accurate and scalable graph embedding. In: 8th International Conference on Learning Representations. ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net.

Ding, C.H.Q., He, X., Zha, H., Gu, M., Simon, H.D., 2001. A min-max cut algorithm for graph partitioning and data clustering. In: Cercone, N., Lin, T.Y., Wu, X. (Eds.), Proceedings of the 2001 IEEE International Conference on Data Mining. 29 November - 2 December 2001, San Jose, California, USA, IEEE Computer Society, pp. 107–114. http://dx.doi.org/10.1109/ICDM.2001.989507.

Gao, H., Huang, H., 2018. Deep attributed network embedding. In: IJCAI. July 13-19, 2018, Stockholm, Sweden, pp. 3364–3370. http://dx.doi.org/10.24963/ijcai.2018/467.

Grady, L., 2006. Random walks for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 28 (11), 1768–1783. http://dx.doi.org/10.1109/TPAMI.2006.233.

Grover, A., Leskovec, J., 2016. Node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 855–864.

Hou, Y., Chen, H., Li, C., Cheng, J., Yang, M.-C., 2019. A representation learning framework for property graphs. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 65–73.

Huang, W., Chen, E., Liu, Q., Chen, Y., Huang, Z., Liu, Y., Zhao, Z., Zhang, D., Wang, S., 2019. Hierarchical multi-label text classification: An attention-based recurrent network approach. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management. pp. 1051–1060.

Huang, X., Li, J., Hu, X., 2017. Label informed attributed network embedding. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. ACM, pp. 731–739.

Jean, S., Cho, K., Memisevic, R., Bengio, Y., 2015. On using very large target vocabulary for neural machine translation. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing. ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers, The Association for Computer Linguistics, pp. 1–10. http://dx.doi.org/10.3115/v1/p15-1001.

Jin, M., Chang, H., Zhu, W., Sojoudi, S., 2021. Power up! robust graph convolutional network via graph powering. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, the Eleventh Symposium on Educational Advances in Artificial Intelligence. EAAI 2021, Virtual Event, February 2-9, 2021, AAAI Press, pp. 8004–8012, URL https://ojs.aaai.org/index.php/AAAI/article/view/16976.

Kipf, T.N., Welling, M., 2017. Semi-supervised classification with graph convolutional networks. In: ICLR. Toulon, France, April 24-26, 2017, Conference Track Proceedings, URL https://openreview.net/forum?id=SJU4ayYgl.

Li, Y., Wang, Y., Zhang, T., Zhang, J., Chang, Y., 2019a. Learning network embedding with community structural information. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence. AAAI Press, pp. 2937–2943.

Li, Z., Zhang, L., Song, G., 2019b. GCN-LASE: towards adequately incorporating link attributes in graph convolutional networks. In: Kraus, S. (Ed.), Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence. IJCAI 2019, Macao, China, August 10-16, 2019, ijcai.org, pp. 2959–2965. http://dx.doi.org/10.24963/ijcai.2019/410.

Liben-Nowell, D., Kleinberg, J.M., 2007. The link-prediction problem for social networks. J. Assoc. Inf. Sci. Technol. 58 (7), 1019–1031. http://dx.doi.org/10.1002/asi.20591.

Lin, W., 2021. Large-scale network embedding in apache spark. In: Zhu, F., Ooi, B.C., Miao, C. (Eds.), KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. Virtual Event, Singapore, August 14-18, 2021, ACM, pp. 3271–3279. http://dx.doi.org/10.1145/3447548.3467136.

Liu, J., He, Z., Wei, L., Huang, Y., 2018. Content to node: Self-translation network embedding. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, pp. 1794–1802.

Ma, J., Cui, P., Wang, X., Zhu, W., 2018. Hierarchical taxonomy aware network embedding. In: Guo, Y., Farooq, F. (Eds.), Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. KDD 2018, London, UK, August 19-23, 2018, ACM, pp. 1920–1929. http://dx.doi.org/10.1145/3219819.3220062.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J., 2013. Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems. pp. 3111–3119.

Pan, S., Wu, J., Zhu, X., Zhang, C., Wang, Y., 2016. Tri-party deep network representation. In: IJCAI. New York, NY, USA, 9-15 July 2016, pp. 1895–1901, URL http://www.ijcai.org/Abstract/16/271.

Pan, G., Yao, Y., Tong, H., Xu, F., Lu, J., 2021. Unsupervised attributed network embedding via cross fusion. In: Lewin-Eytan, L., Carmel, D., Yom-Tov, E., Agichtein, E., Gabrilovich, E. (Eds.), WSDM '21, the Fourteenth ACM International Conference on Web Search and Data Mining. Virtual Event, Israel, March 8-12, 2021, ACM, pp. 797–805. http://dx.doi.org/10.1145/3437963.3441763.

Pereira, R.M., Costa, Y.M.G., Jr., C.N.S., 2021. Toward hierarchical classification of imbalanced data using random resampling algorithms. Inform. Sci. 578, 344–363. http://dx.doi.org/10.1016/j.ins.2021.07.033.

Perozzi, B., Al-Rfou, R., Skiena, S., 2014. Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 701–710.

Rauber, P.E., Falcão, A.X., Telea, A.C., 2016. Visualizing time-dependent data using dynamic t-SNE. In: Bertini, E., Elmqvist, N., Wischgoll, T. (Eds.), 18th Eurographics Conference on Visualization. EuroVis 2016 - Short Papers, Groningen, the Netherlands, June 6-10, 2016, Eurographics Association, pp. 73–77. http://dx.doi.org/10.2312/eurovisshort.20161164.

Shang, C., Dash, S., Chowdhury, M.F.M., Mihindukulasooriya, N., Gliozzo, A., 2020a. Taxonomy construction of unseen domains via graph-based cross-domain knowledge transfer. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J.R. (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. ACL 2020, Online, July 5-10, 2020, Association for Computational Linguistics, pp. 2198–2208. http://dx.doi.org/10.18653/v1/2020.acl-main.199.

Shang, J., Zhang, X., Liu, L., Li, S., Han, J., 2020b. NetTaxo: Automated topic taxonomy construction from text-rich network. In: Huang, Y., King, I., Liu, T., van Steen, M. (Eds.), WWW '20: The Web Conference 2020. Taipei, Taiwan, April 20-24, 2020, ACM / IW3C2, pp. 1908–1919. http://dx.doi.org/10.1145/3366423.3380259.

Shi, C., Hu, B., Zhao, W.X., Philip, S.Y., 2018. Heterogeneous information network embedding for recommendation. IEEE Trans. Knowl. Data Eng. 31 (2), 357–370.

Sun, K., Zhu, Z., Lin, Z., 2021. AdaGCN: Adaboosting graph convolutional networks into deep models. In: 9th International Conference on Learning Representations. ICLR 2021, Virtual Event, Austria, May 3-7, 2021, OpenReview.net, URL https://openreview.net/forum?id=QkRbdiiEjM.

Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q., 2015. Line: Large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, pp. 1067–1077.

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y., 2018. Graph attention networks. In: 6th International Conference on Learning Representations. ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, OpenReview.net, URL https://openreview.net/forum?id=rJXMpikCZ.

Wang, L., Gao, C., Huang, C., Liu, R., Ma, W., Vosoughi, S., 2021. Embedding heterogeneous networks into hyperbolic space without meta-path. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, the Eleventh Symposium on Educational Advances in Artificial Intelligence. EAAI 2021, Virtual Event, February 2-9, 2021, AAAI Press, pp. 10147–10155, URL https://ojs.aaai.org/index.php/AAAI/article/view/17217.

Wang, Z., Ye, X., Wang, C., Cui, J., Yu, P.S., 2020. Network embedding with completely-imbalanced labels. IEEE Trans. Knowl. Data Eng. http://dx.doi.org/10.1109/TKDE.2020.2971490.

Wang, H., Zhang, F., Hou, M., Xie, X., Guo, M., Liu, Q., 2018. SHINE: signed heterogeneous information network embedding for sentiment link prediction. In: Chang, Y., Zhai, C., Liu, Y., Maarek, Y. (Eds.), Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining. WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018, ACM, pp. 592–600. http://dx.doi.org/10.1145/3159652.3159666.

Xie, Y., Ou, Z., Chen, L., Liu, Y., Xu, K., Yang, C., Zheng, Z., 2021. Learning and updating node embedding on dynamic heterogeneous information network. In: Lewin-Eytan, L., Carmel, D., Yom-Tov, E., Agichtein, E., Gabrilovich, E. (Eds.), WSDM '21, the Fourteenth ACM International Conference on Web Search and Data Mining. Virtual Event, Israel, March 8-12, 2021, ACM, pp. 184–192. http://dx.doi.org/10.1145/3437963.3441745.

Xue, H., Sun, X., Sun, W., 2020. Multi-hop hierarchical graph neural networks. In: Lee, W., Chen, L., Moon, Y., Bourgeois, J., Bennis, M., Li, Y., Ha, Y., Kwon, H., Cuzzocrea, A. (Eds.), 2020 IEEE International Conference on Big Data and Smart Computing. BigComp 2020, Busan, Korea (South), February 19-22, 2020, IEEE, pp. 82–89. http://dx.doi.org/10.1109/BigComp48618.2020.00-95.

Yang, C., Zhang, J., Han, J., 2020. Co-embedding network nodes and hierarchical labels with taxonomy based generative adversarial networks. In: Plant, C., Wang, H., Cuzzocrea, A., Zaniolo, C., Wu, X. (Eds.), 20th IEEE International Conference on Data Mining. ICDM 2020, Sorrento, Italy, November 17-20, 2020, IEEE, pp. 721–730. http://dx.doi.org/10.1109/ICDM50108.2020.00081.

Zhang, C., Tao, F., Chen, X., Shen, J., Jiang, M., Sadler, B.M., Vanni, M., Han, J., 2018a. TaxoGen: Unsupervised topic taxonomy construction by adaptive term embedding and clustering. In: Guo, Y., Farooq, F. (Eds.), Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. KDD 2018, London, UK, August 19-23, 2018, ACM, pp. 2701–2709. http://dx.doi.org/10.1145/3219819.3220064.

Zhang, Z., Yang, H., Bu, J., Zhou, S., Yu, P., Zhang, J., Ester, M., Wang, C., 2018b. ANRL: attributed network representation learning via deep neural networks. In: IJCAI. July 13-19, 2018, Stockholm, Sweden, pp. 3155–3161. http://dx.doi.org/10.24963/ijcai.2018/438.