Iterative Deep Graph Learning with Local Feature Augmentation for Network Alignment

Jiuyang Tang, Zhen Tan(✉), Hao Guo, Xuqian Huang, Weixin Zeng, and Huang Peng

Laboratory for Big Data and Decision,
National University of Defense Technology, Changsha, China
{tanzhen08a,guo_hao,huangxuqian15,phmail}@nudt.edu.cn

Abstract. Networks are structures that naturally capture relations between entities in different data sources and information systems. To establish the connections among different networks, the task of network alignment is proposed and intensively studied in network-related research field. Most network alignment methods are based on the representation learning of network structure, which rely only on network topology and are susceptible to structural noise. In addition, such methods focus on the global features, while largely neglect local structure features and fail to take account of the data sparsity issue in real networks. To address these pivotal issues, we propose a novel network alignment method based on iterative deep network learning and local feature augmentation. We first design an iterative deep graph learning model to learn high-quality network structural representation and reduce the structural noise. Furthermore, we embed knowledge representation learning method into the alignment process, which helps to characterize better local structure and alleviate the data sparsity issue. Experiments on real-world network datasets demonstrate that our proposed model achieves state-of-the-art alignment results.

Keywords: Network alignment · Graph learning · Knowledge representation learning

1 Introduction

Networks are natural but powerful structure that capture the relationships between different entities in many domains, such as social networks, referral networks, and bioinformatics networks [12,27]. Network analysis, also known as network science, has received a lot of attention for decades and remains an attractive field [1]. While the analysis of individual network is critical for a variety of applications (e.g., link prediction; community detection), it cannot sufficiently

Supported by Ministry of Science and Technology of China under grants No. 2020AAA0108800, NSFC under grants Nos. 71971212 and 61902417.

B. Li et al. (Eds.): APWeb-WAIM 2022, LNCS 13421, pp. 511–526, 2023.
https://doi.org/10.1007/978-3-031-25158-0_41

address the tasks that require considering the relationships between graphs (e.g., graph clustering; graph alignment). As thus, a subfield of network science is put forward to analyze the relationships between different graphs. In this paper, we aim to solve one of the fundamental problems in comparative graph analysis - network alignment (NA).

The goal of network alignment is to identify the corresponding nodes in different networks. For example, there are a large number of users with accounts in different social networks [23], and network alignment can help identify the same users in different social networks, as shown in Fig. 1. The user correspondence established by network alignment can alleviate the sparsity issue of a single social network, benefiting applications such as link prediction [5] and cross-domain recommendation [14]. Besides, network alignment can also help build a more compact knowledge graph based on existing vertical or cross-language knowledge bases, and enable better knowledge inference. In addition, in bioinformatics, aligning protein-protein interaction networks from different species has also been extensively studied to identify common functional structures [7].

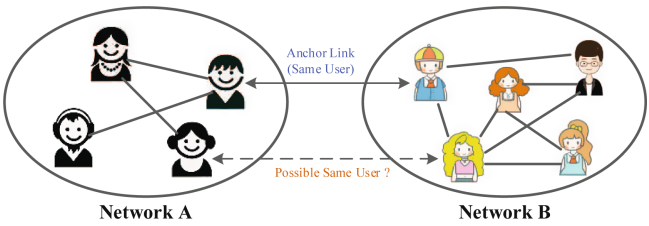

Fig. 1. An example of network alignment. The black lines between two networks are anchor links. And the dash lines are potential aligned nodes.

However, network alignment faces three primary challenges: network noise, data sparsity, and alignment efficiency.

Alignment Efficiency. Some NA work describes network alignment as the maximum matching problem of a binary graph, such as the largest common subgraph problem, but these are all NP-difficult problems [2]. Therefore, many methods employ matrix decomposition formulas such as IsoRank [18], FINAL [28], and REGAL [8]. But the approaches cannot handle very large networks because the computational effort required will grow rapidly as the size of the network increases.

Network Noise. Due to the inevitable error of data measurement or collection, real-world networks are generally noisy or even incomplete. The network noise originates from both the topology of the network and the feature matrix of the nodes.

Data Sparsity. Similar to other types of large-scale data, large-scale networks all obey the long-tail distribution and have severe data sparsity problems [26].

For long-tail nodes, only a few paths are associated with them, so their semantic or inference representation is extremely inaccurate.

To improve alignment efficiency, methods based on the network representing learning are proposed, such as PALE [15], and DeepLink [30]. These alignment techniques can take advantage of the scalability of graph embedding to handle large networks, but these methods rely only on topology information and are susceptible to structure noise, making the models lack generalization capabilities. To overcome the network structure noise, we generate better network structure representation basing on iterative deep graph learning framework.

Moreover, most network alignment methods pay more attention to the global features of the network without valuing the local structure features. Particularly in sparse networks, such methods lead to the poor performance because of the long-tail distribution. While knowledge representation learning methods, such as TransE [3], TransH [24], DistMult [25], ComplEx [21], and RotatE [20] models, can well characterize the local structure of networks. In view of that, to address the data sparsity, we use various graph representation learning methods for network alignment to obtain high-quality local features.

In this paper, we propose **I**terative **D**eep Graph Learning with **L**ocal **F**eature **A**ugmentation for Network Alignment (IDLFA). The model is composed of two parts: encoder module and decoder module. The encoder module learns node structure embedding by iterative deep graph learning model. The decoder module integrates the knowledge representation learning method into the alignment method to augment local feature. And in the training process, the bootstrapping algorithm is applied to add the newly generated alignment nodes to the training set to further alleviate the data sparsity issue. The contributions of this paper can be summarized as follows:

1) We propose a unified network alignment framework, which combines encoder module and decoder model for solving network structure noise and data sparsity.
2) At encoder module, we leverage Iterative Deep Graph Learning for Graph Neural Networks to obtain better node structural embedding and reduce networks noise.
3) At decoder module, for easing data sparsity, we integrate knowledge representation methods to augment local feature and apply bootstrapping algorithm to produce newly alignments for model training.
4) Experiments on real-world datasets demonstrate that the network alignment method based on iterative deep graph learning outperforms state-of-the-art models and is highly robust on alignment tasks.

2 Related Work

In our work, the network alignment techniques are divided into two categories, the **spectral method** and the **network representation learning method**. The goal of the spectral approach is to align the two networks based on the

adjacency matrix operation. While, the network representation learning method requires an intermediate step where the nodes in the network are represented as embedding.

2.1 Spectral Method

Many spectral methods [11,17], using matrix factorization, aim to directly calculate the alignment matrices. Assuming that the input graph is in the form of an adjacency matrix, the spectral alignment technique defines the model in the form of a loss function, which considers the adjacency matrix of the source network and the target network. The node features are constants and the alignment matrices are variables. During the alignment process, the alignment matrix is learned by optimizing the loss function based on the structure or property consistency assumptions.

IsoRank [18] is one of the popular and typical techniques in the category which utilizes only topological information. The main idea is that two nodes in two networks are similar if their neighbors are similar, but the technique is highly sensitive to structural noise. BigAlign [10] uses only the attribute information to align the nodes. FINAL [28] is different from previous methods that only used topological or attribute information, and chooses to use both of them to better capture the information of the network nodes. REGAL [8] models the alignment matrix by topology and feature similarity and then employs low-rank matrix approximation to speed up calculation.

2.2 Network Representation Learning Method

Network representation learning approaches [6,16,29] solve the network alignment problem by exploiting graph embedding. It includes two steps: embedding generation and alignment matrix generation. At first, a graph embedding technique is used to represent nodes and the two embedding matrices of graphs are obtained separately. Then, the alignment matrix is designed to map the source network's embeddings to the target network's embedding space.

PALE [15] involves a pre-process step where a priori mapping between two networks is used to populate the missing edges available in one map but not available in the other. DeepLink [30] has the same method as the PALE to construct graph embeddings, but its mapping function varies by considering the mapping direction. DeepLink adopts unbiased random walk to generate embeddings and uses a linked-dual learning process to improve its quality. IONE [13] uses the same mapping function as PALE, its embedding function is more complex because it considers the neighborhood of the node. IONE aims to meet two goals: close nodes in each graph should have similar node embedding and the nodes with close embedding are good candidates.

3 Preliminaries

In this section, we first formally define the task of network alignment, and then we briefly review the knowledge representation model.

3.1 Problem Formulation

Network alignment is the task of identifying corresponding nodes between two different networks. Given two networks: source network $G_s = (V_s, E_s))$and target network $G_t = (V_t, E_t))$, where V_s and V_t are sets of network nodes, E_s and E_t are sets of network edges. **Anchor links** represent a node pair(v, v') , where$v \in V_s$, $v' \in V_t$, and v and v' are aligned. The goal of network alignment is to predict all potential anchor links.

3.2 Knowledge Representation Model

In this section, we introduce TransE [3], TransH [24], DistMult [25], ComplEx [21], and RotatE [20] models, which are adapted to our network alignment framework. u and v denote the node embedding; e is the edge embedding.

TransE. The idea of TransE is that the embeddings of the nodes in the source network can be close enough to the corresponding embeddings in the target network through the edge embedding, so that the score function of the TransE model can be represented by the following formula:

$$f_{TransE}(u + e, v) = \|u + e - v\| . \tag{1}$$

TransH. To overcome the defects of TransE in edge modeling, the nodes have a distributed representation when different edges are involved. For an edge, the model positions an edge-specific translation vector d_e in the hyperplane w_e of a specific edge rather than in the space nodes embedded:

$$f_{TransH}(u, v) = \left\|(u - w_e^T u w_e) + d_e - (v - w_e^T v w_e)\right\|_2^2 . \tag{2}$$

DistMult employs bilinear encoding, and the embeddings of nodes and edges can be learned through a neural network. The first layer projects a pair of input nodes onto a low-dimensional vector, and the second layer combines the two vectors onto a scalar. With edge-specific parameters B_e, the score function is:

$$f_{DistMult}(u, v) = u^T B_e v. \tag{3}$$

ComplEx. The model introduces the complex vector space into the embedding, and its score function is:

$$f_{ComplEx}(e, u, v; \Theta) = Re(< e, u, \overline{v} >) = Re(\sum_{k=1}^{K} e_k u_k \overline{v}_k), \tag{4}$$

where $Re(\cdot)$ denotes imaginary; $\overline{v}_k$ and v_k are conjugate.

RotatE. Similar to complEx model, RotatE models the nodes and edges in the complex vector space. The difference is that the RotatE limits the modulus of the edge vector to 1, so that it becomes the rotation vector from the source network node to the corresponding node of the target network. Therefore, its score function is expressed as:

$$f_{RotatE}(u, v) = \|u \odot e - v\|, \tag{5}$$

where $\odot$ denotes Hadamard product; $\|e_i\| = 1$ denotes that the modulus of the edge vector are set to 1.

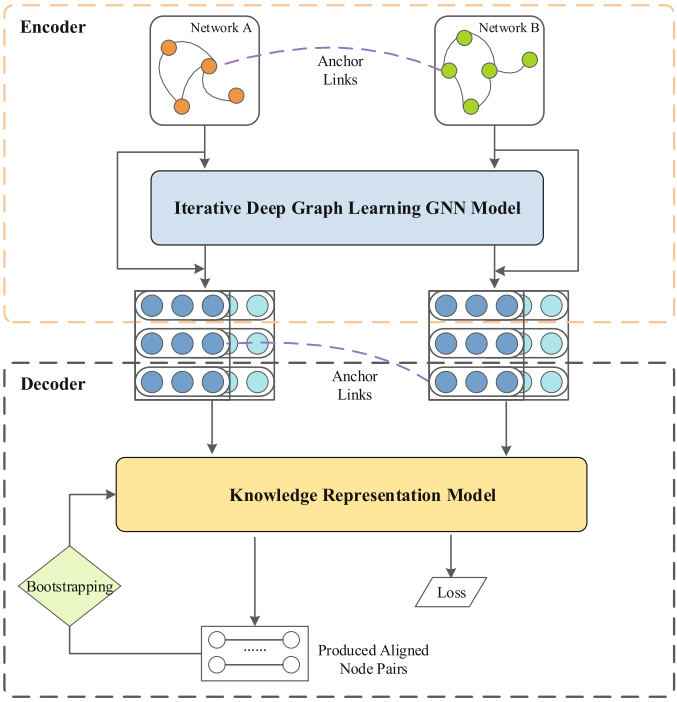

Fig. 2. Overview of IDLFA framework

4 Method

In this section, we present our approach IDLFA, which consists of encoder module and decoder module. The framework is shown in Fig. 2. In encoder module, the network structure is learned according to the iterative deep graph learning framework. In decoder module, we combine the knowledge representation model with the loss function of the IDGL model and adapt it to the network alignment, which helps to learn better local features. To further alleviate the data sparse problem, we use Bootstrapping algorithm [19] to add the predicted aligned node pairs to training data.

4.1 IDGL-Based Node Embedding

Iterative Deep Graph Learning model [4] (IDGL) is an end-to-end graph learning framework, which can jointly and iteratively learning the graph structure and graph embedding. In view of the advantage of obtaining better network representation, we transform it to alignment networks. Figure 3 is the overall architecture of IDGL framework. The brief introduction to the model is following.

Similarity Metric Learning. Without loss of generality, the IDGL model designs a weighted cosine similarity as metric function, $s_{ij}^{p} = cos(w \odot v_i, w \odot v_j)$, where $\odot$ denotes the Hadamard product, and w is a learnable weight vector which has the same dimension as the input vectors v_i and v_j. Note that the two input vectors could be either raw node features or computed node embeddings.

Graph Node Embeddings and Prediction. Both the learned graph structure A and the original graph topology $A^{(0)}$ are helpful to formulate an optimized graph for GNNs. IDGL combines the learned graph with the initial graph,

$$\tilde{A}^{(t)} = \lambda L^{(0)} + (1-\lambda)\{\eta f(A^{(t)}) + (1-\eta) f(A^{(1)})\}, \tag{6}$$

where $L^{(0)} = D^{(0)^{-1/2}} A^{(0)} D^{(0)^{-1/2}}$ is the normalized adjacency matrix of the initial graph. $A^{(t)}$ and $A^{(1)}$ are the two adjacency matrices computed at the t-th and 1-st iterations, respectively. $A^{(1)}$ is computed from the raw node features X, and $A^{(t)}$ is computed from the previously updated node embeddings $Z^{(t-1)}$ that is optimized toward the downstream prediction task. Hyperparameter η is used to combine the advantages of both; λ is used to balance the learned graph structure and the initial one.

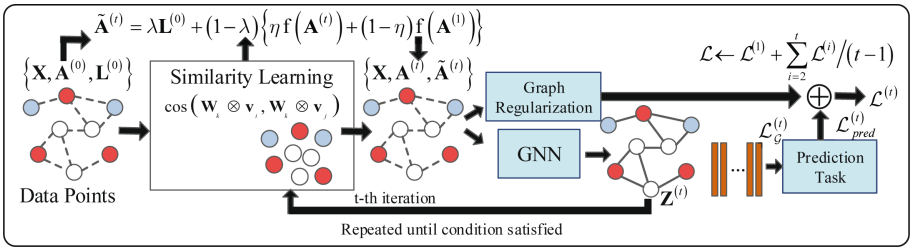

Fig. 3. Overall architecture of the proposed IDGL framework. Dashed lines (in data points on left) indicate the initial noisy graph topology A.

We follow the setting of IDGL model, and adopt a two-layered GCN [9] where the first layer (denoted as GNN_1) maps the raw node features $\mathbf{X}$ to the intermediate embedding space, and the second layer (denoted as GNN_2) further maps the intermediate node embeddings $\mathbf{Z}$ to the output space.

$$\mathbf{Z} = \mathrm{ReLU}\left(\mathbf{MP}(\mathbf{X}, \tilde{\mathbf{A}})\mathbf{W}_1\right), \quad \widehat{\mathbf{y}} = \sigma\left(\mathbf{MP}(\mathbf{Z}, \tilde{\mathbf{A}})\mathbf{W}_2\right), \quad \mathcal{L}_{pred} = \ell(\widehat{\mathbf{y}}, \mathbf{y}), \tag{7}$$

where $\sigma(\cdot)$ and $\mathcal{L}(\cdot)$ are task-dependent output function and loss function, respectively. $\mathbf{MP}(\cdot, \cdot)$ is a massage passing function.

Joint Learning with a Hybrid Loss. IDGL model proposes to jointly and iteratively learning the graph structure and the GNN parameters by minimizing a hybrid loss function combining both the task prediction loss and the graph regularization loss, namely:

$$\mathcal{L} = \mathcal{L}_{pred} + \mathcal{L}_{\mathcal{G}}, \tag{8}$$

where $\mathcal{L}_{\mathcal{G}}$ is the graph regularization loss. At each iteration, a hybrid loss is computed. After all iterations, the overall loss is back-propagated through all previous iterations to update the model parameters.

4.2 Model Training

$\mathcal{L}_{pred}$ is a task-dependent loss function. For network alignment, the object is to embed equivalent nodes as closely as possible in the vector space. So the model training process is performed by minimizing the following margin-based ranking loss function:

$$\mathcal{L}_{pred} = \sum_{(u,v)\in S} \sum_{(u',v')\in S'_{(u,v)}} \left[f\left(u, v\right) + \gamma - f\left(u', v'\right)\right]_+, \tag{9}$$

where $[x]_+ = \max\{0, x\}$; $(u, v) \in S$ represents the set of anchor links used to train the model; $S'_{(u,v)} \in S'_{(u,v)}$ denotes the set of negative instances constructed by corrupting (u, v), i.e., replacing u or v with a randomly chosen nodes in G_s or G_t; $\gamma > 0$ denotes the margin hyper-parameter separating positive and negative instances. The margin-based loss function requires that the distance between the entities in positive pairs should be small, and the distance between the entities in negative pairs should be large. Based on various knowledge representation methods at Sect. 3.3, we design correspond loss function. Following are the detailed instruction.

TransE. We merge it with margin-based ranking loss function:

$$\mathcal{L}_{\text{TransE}} = \sum_{(u,v)\in S} \sum_{(u',v')\in S'_{(u,v)}} \left[f_{TransE}\left(u + e, v\right) + \gamma_1 - f_{TransE}\left(u' + e, v'\right)\right]_+, \tag{10}$$

where $\gamma_1 > 0$ the specific boundary hyperparameter separating the positive and negative node alignment in the TransE model.

TransH. We merge it with margin-based ranking loss function:

$$\mathcal{L}_{\text{TransH}} = \sum_{(u,v)\in S} \sum_{(u',v')\in S'_{(u,v)}} \left[f_{TransH}\left(u,v\right) + \gamma_2 - f_{TransH}\left(u',v'\right)\right]_+, \quad (11)$$

where $\gamma_2 > 0$ is the specific boundary hyperparameter.

DistMult. Based on DistMult model, the loss function $\mathcal{L}_{pred}$ can be adjusted to:

$$\mathcal{L}_{\text{DistMult}} = \sum_{(u,v)\in S} \sum_{(u',v')\in S'_{(u,v)}} \left[f_{DistMult}\left(u,v\right) - f_{DistMult}\left(u',v'\right) + 1\right]_+. \quad (12)$$

ComplEx. We minimize the negative log-likelihood of the logical model, and train the model using small-batch stochastic gradient descent and AdaGrad. By regularizing the parameters of the considered model, we adjust the learning rates:

$$\mathcal{L}_{\text{ComplEx}} = \sum_{(u,v)\in S} \log\left(1 + \exp\left(-\mathbf{Y}_{euv} f_{\text{ComplEx}}\left(u,e,v;\Theta\right)\right)\right) + \lambda\|\Theta\|_2^2, \quad (13)$$

where $\mathbf{Y}_{euv} = 1$ when the node pairs are positive; and else $\mathbf{Y}_{euv} = -1$. λ is a weight parameter.

RotatE. Different from above models, RotatE adopts self-adversarial loss function basing on negative sampling for training:

$$\mathcal{L}_{\text{RotatE}} = -\log\sigma\left(\gamma_3 - f_{\text{RotatE}}\left(\mathbf{u},\mathbf{v}\right)\right) - \sum_{i=1}^{n} p\left(u'_i,e,v'_i\right)\log\sigma\left(f_{\text{RotatE}}\left(\mathbf{u}'_i,\mathbf{v}'_i\right) - \gamma_3\right), \quad (14)$$

where γ_3 is the specific boundary hyperparameter; σ is the sigmoid function; (u'_i, e, v'_i) is the i-th negative alignment nodes; $p_{\left(u'_i,e,v'_i\right)}$ can be defined as:

$$p\left(u'_j,e,v'_j \mid \{(u_i,e_i,v_i)\}\right) = \frac{\exp\alpha f_{\text{RotatE}}\left(\mathbf{u}'_j,\mathbf{v}'_j\right)}{\sum_i \exp\alpha f_{\text{RotatE}}\left(\mathbf{u}'_i,\mathbf{v}'_i\right)}, \quad (15)$$

where α denotes the sample weight.

The loss function of the above models will learn a better network structure representation. At the same time, the model further adds the new alignment nodes into the training set through the Bootstrapping algorithm, which helps to alleviate the data sparse problem and further improve the performance.

4.3 Alignment Prediction

We predict the alignment results based on the distance between learned nodes representations from two networks.

The Euclidean distance and Manhattan distance are commonly used distance measures in the Euclidean space. For entities u_i in G_s and v_j in G_t, the distance is defined as:

$$D\left(u_i, v_j\right) = \frac{f\left(\mathbf{u}_i, \mathbf{v}_j\right)}{d}, \tag{16}$$

where $f(x, y) = \|x - y\|_1$, $\|\cdot\|_1$ is the L_1 norm; d denotes the dimension of embedding. The distance is expected to be small for equivalent entities and large for non-equivalent ones. For a specific entity u_i in G_s, our approach computes the distances between u_i and all the entities in G_t, and returns a list of ranked entities as candidate alignments.

5 Experiment

In the section, we conduct extensive experiments to verify the effectiveness of the model. First, we introduce the experimental settings and then evaluate the performance of our method. For further analysis, ablation study and parameter analysis are performed.

5.1 Experiment Setup

Datasets. This section conducts experiments on 2 real-world datasets (4 real-world networks). The detailed information is shown in Table 1.

Flickr and Myspace datasets: The two subnetworks of Flickr and Myspace are collected in the paper [27] and then processed according to the method in the paper [28]. Flickr's subnet contains 6714 nodes, and Myspace's subnet contains 10,733 nodes. The gender of the user is used to represent node attributes, and only part of the ground truth is available for alignment.

Allmovie and Imdb datasets: The Allmovie network is constructed from Rotten Tomatoes website[1]. Two films have an edge connecting them if they have at least one common actors. Imdb network is constructed in a similar way from Imdb website[2]. The alignment output is constructed by the identity of the film, containing 5176 anchor links.

Evaluation Metrics. We use both *Success*@q [28] and MAP (Mean Average Precision) [15] to evaluate the effectiveness of our proposed model. *Success*@q denotes whether a true positive match appears in the previous q candidate. For ranking perspective, MAP is also known as Mean Reciprocal Rank under pairwise setting. Considering that the network alignment is a bidirectional task, we use the average value of $G_s \rightarrow G_t$ and $G_t \rightarrow G_s$ to present the experimental results.

[1] https://www.kaggle.com/ayushkalla1/rotten-tomatoes-movie-database.
[2] https://www.kaggle.com/jyoti1706/imdbmoviesdataset.

Table 1. Statistics of 4 real-world networks.

Network	Nodes	Edge	Attribute
Flickr	6714	7333	3
Myspace	10733	11081	3
Allmovie	6011	124709	14
Imdb	5731	119073	14

Comparison Methods. Our proposed model IDLFA with its variants and the state-of-the-art baseline methods for comparison are listed as following:

PALE [15]: is a network representation technique that learns node embedding by maximizing the co-occurrence likelihood of edge nodes, and then applies a linear or multi-layer perceptron as a mapping function.

REGAL: is a spectral method that models the alignment matrix by topology and feature similarity of nodes, and then accelerates with a low-rank matrix [8].

IsoRank: is a spectral approach and a global alignment method initially with application to protein interaction networks [18].

FINAL: is a spectral method designed for attributed networks, which considers graph structure, node feature, and edge feature [28].

GAlign: is the state-of-the-art alignment mode and proposes a completely unsupervised network alignment framework based on a multi-order GCN embedding model [22].

Hyperparameter Tuning. The margin hyper-parameters γ, γ_1, γ_2 and γ_3 in the relevant loss function are set to 1. And the value of λ is validated in set $\{0.1, 0.03, 0.01, 0.003, 0.001\}$. The embedding dimension is set to 100 and will be further evaluated in the later. We optimize the model with Stochastic Gradient Descent algorithm.

Machines and Repeatability. The results are averaged over 10 runs to mitigate randomness. All experiments are conducted on 8 3.6 GHz Intel Cores with 64 GB RAM and 1 GeForce RTX2080Ti graphic cards. Our proposed algorithm is programmed in Python.

5.2 Experiment Result

To verify the effectiveness of our proposed model, we compares the models with several state-of-the-art models on two real-world datasets, and the experimental results are shown in Table 2. Bold numbers indicate optimal results, and underlined numbers indicate sub-optimal results. The results are obtained with 80% of the anchor nodes as training set and the rest for testing.

Table 2. The performance of network alignment on real-world datasets.

Dataset	Metrics	Ours	GAlign	PALE	REGAL	IsoRank	FINAL
Allmovie-Imdb	MAP	**0.9320**	0.8496	0.7601	0.1888	0.5271	0.8459
	$Success$@1	**0.9068**	0.8214	0.6947	0.0953	0.4653	0.7647
	$Success$@10	**0.9710**	0.9003	0.7159	0.3869	0.6427	0.9609
Flickr-Myspace	MAP	0.1245	**0.1608**	0.0059	0.0090	0.0085	0.0429
	$Success$@1	0.0556	**0.0774**	0.0000	0.0464	0.0000	0.0206
	$Success$@10	0.2615	**0.3127**	0.0206	0.1950	0.0275	0.0722

In general, our proposed IDLFA model outperforms all the baselines on datasets Allmovie-Imdb in terms of MAP, $Success$@1 and $Success$@10. In terms of $Success$@1, IDLFA achieves more than 90%, exceeds GAlign by 8% and exceeds FINAL by nearly 15%. In terms of MAP, IDLFA outperforms the second over 8%. In addition, the $Success$@10 of IDLFA is about 97%.

In Flickr-Myspace, our proposed model achieves the sub-optimal alignment accuracy. Compared with FINAL, MAP of IDLFA increases by more than 0.07, and $Success$@10 is about 15% higher. Compared with the SOTA model GAlign, our proposed method is nearly 2% lower in $Success$@1 and about 0.04 lower in terms of MAP. Although our model IDLFA does not exceed GAlign, we only use structural information, while GAlign uses additional attribute information.

Weakly Supervised Condition. Table 3 further gives the detailed model comparison when the ratio of training set to test set is 0.2:0.8. Our proposed model is compared with the previous SOTA GAlign in Allmovie-Imdb and Flickr-Myspace datasets. Our method IDLFA still has $Success$@1 $\approx$ 78% better than GAlign in Allmovie-Imdb dataset. In Flickr-Myspace dataset, IDLFA outperforms GAlign by a large margin. The results show that the model is still robust and well-performed in a weakly supervised manner. And our proposed model performs better in Allmovie-Imdb than Flickr-Myspace, which indicates that the model has more prominent performance on datasets with abundant structure information in a weakly supervised manner.

Table 3. The performance of our proposed model IDLFA on 20% anchor links.

Dataset	Metrics	Ours	GAlign
Allmovie-Imdb	MAP	**0.8185**	0.7925
	$Success$@1	**0.7785**	0.7399
	$Success$@10	**0.9023**	0.8667
Flickr-Myspace	MAP	**0.0395**	0.0177
	$Success$@1	**0.0140**	0.0044
	$Success$@10	**0.0514**	0.0327

Ablation Study. The local feature argument mechanism is based on knowledge representation model, which is designed for sparse datasets. Although the IDGL model can learn better structure representation, the premise is that the nodes have rich topology information. It can be seen from Table 2 that the alignment accuracy of the same model on different datasets varies greatly. What is the reason for this phenomenon? It can be seen from Table 1 that the network Flickr and Myspace are relative sparse, whose average edges of nodes is about 2. That's to say, these networks have a large number of long-tail nodes. For long-tail nodes, GNN-based models are limited to learn their structure features [26]. So, to verify the effectiveness of our proposed method on sparse datasets, we conduct ablation study on Flickr-Myspace.

IDNA represents our proposed model without local feature augmentation module. IDNA+TransE, IDNA+TransH, IDNA+DistMult, IDNA+ComplEx and IDNA+RotatE denote fusing IDNA with various knowledge representation method to learn better local feature. Table 4 presents the result with 4 metrics. From the table, we can clearly see that local feature augmentation module obtains better performance in terms of MAP, $Success@1$, $Success@3$ and $Success@10$, which indicates the local feature augmentation module works well and learns better structure representation. When the percentage of anchor links is 0.8, IDNA+RotatE outperforms other method and $Success@1$ improves by 2% compared to IDNA. When the percentage of anchor links is 0.2, IDNA+ComplEx performs better and has 1.2% improvement in terms of MAP.

Table 4. The result of ablation study on Flickr-Myspace.

	Model	$Success@1$	$Success@3$	$Success@10$	MAP
Anchor links = 0.8	IDNA	0.0371	0.1111	0.2408	0.1165
	IDNA+TransE	0.0185	0.0556	0.1760	0.0800
	IDNA+TransH	0.0370	0.0370	0.1760	0.0875
	IDNA+DistMult	0.0370	0.0926	0.1945	0.1050
	IDNA+ComplEx	0.0185	0.0278	0.1667	0.0735
	IDNA+RotatE	**0.0556**	**0.1311**	**0.2615**	**0.1245**
Anchor links = 0.2	IDNA	0.0047	0.0140	0.0421	0.0270
	IDNA+TransE	0.0047	0.0094	0.0538	0.0270
	IDNA+TransH	0.0070	0.0280	**0.0631**	0.0355
	IDNA+DistMult	0.0070	0.0210	0.0561	0.0340
	IDNA+ComplEx	**0.0140**	**0.0304**	0.0514	**0.0395**
	IDNA+RotatE	0.0047	0.0140	0.0584	0.0285

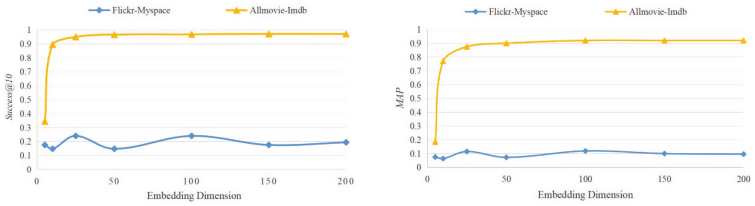

Fig. 4. The relationship between alignment results and embedding dimension.

5.3 Hyperparameter Sensitivity

Figure 4 studies the sensitivity of the embedding dimension. In general, users should not choose a high number of dimensions as it does not increase the performance (*Success*@1) significantly while the time and space complexity definitely become larger. In Flickr-Myspace dataset, as the embedding dimension increases, the model performance will fluctuate to a certain extent, but in general, the effect is better when the embedding dimension is 100. In Allmovie-Imdb, the initial performance increases rapidly over dimension and remains stable when dimension reaches about 100.

6 Conclusion

In this paper, we propose a novel network alignment framework IDLFA, which learns better network structure representation and further solves the networks data sparsity. Comprehensive empirical studies on two pairs of popular real-world datasets show that IDLFA can significantly improve the performance for social network alignment tasks in comparison with existing solutions. On part datasets, such as Allmovie-Imdb, our model shows the superiority, whose *Success*@1 comes to 90%, and can be adopt to practical applications. Our model doesn't take attribute information into consideration. In the future works, we will study relative framework for attributed networks and the proposed framework can be applied to other tasks, e.g., cross-lingual knowledge graph task.

References

1. Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. Rev. Mod. Phys. **74**(1), 47 (2002)
2. Bayati, M., Gerritsen, M., Gleich, D.F., Saberi, A., Wang, Y.: Algorithms for large, sparse network alignment problems. In: 2009 Ninth IEEE International Conference on Data Mining, pp. 705–710. IEEE (2009)
3. Bordes, A., Weston, J., Collobert, R., Bengio, Y.: Learning structured embeddings of knowledge bases. In: Twenty-Fifth AAAI Conference on Artificial Intelligence (2011)

4. Chen, Y., Wu, L., Zaki, M.: Iterative deep graph learning for graph neural networks: better and robust node embeddings. Adv. Neural. Inf. Process. Syst. **33**, 19314–19326 (2020)
5. Gao, H., Zhang, Y., Li, B.: Improving the link prediction by exploiting the collaborative and context-aware social influence. In: Li, J., Wang, S., Qin, S., Li, X., Wang, S. (eds.) ADMA 2019. LNCS (LNAI), vol. 11888, pp. 302–315. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-35231-8_22
6. Goyal, P., Ferrara, E.: Graph embedding techniques, applications, and performance: a survey. Knowl.-Based Syst. **151**, 78–94 (2018)
7. Guzzi, P.H., Milenković, T.: Survey of local and global biological network alignment: the need to reconcile the two sides of the same coin. Brief. Bioinform. **19**(3), 472–481 (2018)
8. Heimann, M., Shen, H., Safavi, T., Koutra, D.: Regal: representation learning-based graph alignment. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 117–126 (2018)
9. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
10. Koutra, D., Tong, H., Lubensky, D.: Big-align: fast bipartite graph alignment. In: 2013 IEEE 13th International Conference on Data Mining, pp. 389–398. IEEE (2013)
11. Levy, O., Goldberg, Y.: Neural word embedding as implicit matrix factorization. In: Advances in Neural Information Processing Systems, vol. 27 (2014)
12. Liu, J., Shao, Y., Su, S.: Multiple local community detection via high-quality seed identification over both static and dynamic networks. Data Sci. Eng. **6**(3), 249–264 (2021)
13. Liu, L., Cheung, W.K., Li, X., Liao, L.: Aligning users across social networks using network embedding. In: IJCAI, pp. 1774–1780 (2016)
14. Man, T., Shen, H., Jin, X., Cheng, X.: Cross-domain recommendation: an embedding and mapping approach. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence, pp. 2464–2470 (2017)
15. Man, T., Shen, H., Liu, S., Jin, X., Cheng, X.: Predict anchor links across social networks via an embedding approach. In: IJCAI, vol. 16, pp. 1823–1829 (2016)
16. Ou, M., Cui, P., Pei, J., Zhang, Z., Zhu, W.: Asymmetric transitivity preserving graph embedding. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1105–1114 (2016)
17. Qiu, J., Dong, Y., Ma, H., Li, J., Wang, K., Tang, J.: Network embedding as matrix factorization: unifying Deepwalk, Line, PTE, and Node2Vec. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, pp. 459–467 (2018)
18. Singh, R., Xu, J., Berger, B.: Global alignment of multiple protein interaction networks with application to functional orthology detection. Proc. Natl. Acad. Sci. **105**(35), 12763–12768 (2008)
19. Sun, Z., Hu, W., Zhang, Q., Qu, Y.: Bootstrapping entity alignment with knowledge graph embedding. In: IJCAI, vol. 18, pp. 4396–4402 (2018)
20. Sun, Z., Deng, Z.H., Nie, J.Y., Tang, J.: Rotate: knowledge graph embedding by relational rotation in complex space. In: International Conference on Learning Representations (2018)
21. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: International Conference on Machine Learning, pp. 2071–2080. PMLR (2016)

22. Trung, H.T., Van Vinh, T., Tam, N.T., Yin, H., Weidlich, M., Hung, N.Q.V.: Adaptive network alignment with unsupervised and multi-order convolutional networks. In: 2020 IEEE 36th International Conference on Data Engineering (ICDE), pp. 85–96. IEEE (2020)
23. Viswanath, B., Mislove, A., Cha, M., Gummadi, K.P.: On the evolution of user interaction in Facebook. In: Proceedings of the 2nd ACM Workshop on Online Social Networks, pp. 37–42 (2009)
24. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 28 (2014)
25. Yang, B., Yih, W.T., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575 (2014)
26. Zeng, W., Zhao, X., Wang, W., Tang, J., Tan, Z.: Degree-aware alignment for entities in tail. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 811–820 (2020)
27. Zhang, J., Philip, S.Y.: Multiple anonymized social networks alignment. In: 2015 IEEE International Conference on Data Mining, pp. 599–608. IEEE (2015)
28. Zhang, S., Tong, H.: Final: fast attributed network alignment. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1345–1354 (2016)
29. Zhou, C., Liu, Y., Liu, X., Liu, Z., Gao, J.: Scalable graph embedding for asymmetric proximity. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31 (2017)
30. Zhou, F., Liu, L., Zhang, K., Trajcevski, G., Wu, J., Zhong, T.: Deeplink: a deep learning approach for user identity linkage. In: IEEE INFOCOM 2018-IEEE Conference on Computer Communications. pp. 1313–1321. IEEE (2018)