IJCNN 2019. International Joint Conference on Neural Networks. Budapest, Hungary. 14-19 July 2019

Noise-Aware Network Embedding for Multiplex Network

Xiaokai Chu[1,2], Xinxin Fan[1], Di Yao[1,2], Chen-Lin Zhang[3], Jianhui Huang[1], Jingping Bi[1]
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China[1]
University of Chinese Academy of Sciences, China[2]
National Key Laboratory for Novel Software Technology, Nanjing University, China[3]
{chuxiaokai,fanxinxin,yaodi,huangjianhui,bjp}@ict.ac.cn, zhangcl@lamda.nju.edu.cn

***Abstract*—Network embedding aims at learning the latent representations of nodes while preserving the complex structure of the underlying graph. Real-world networks are usually related with each other via common nodes, the so-called multiplex network. To make the data mining work on the multiplex network more actionable, it become urgent and essential to transform it into low-dimension vector space. Recently, several works have been proposed to leverage the complementary information for embedding. However, they suffer from sacrificing distinct properties of the counterparts in different layers, as they preserve much noise information into embedding vectors. In this paper, we propose a Noise-Aware Network Embedding approach for Multiplex Network, namely NANE. Unlike previous works, NANE considers the roles of an identical node in different layers, and adopts a more robust and flexible strategy to rationally integrate the cross-layer information while keeping the unique characteristic of each layer. We perform extensive evaluations on several real-world datasets. The experimental results demonstrate that our NANE can achieve better performance on link prediction task and significantly outperform previous methods especially in noisy multiplex network scenarios.**

I. Introduction

In recent years, the rise of big data has generated a large volume of network data, such as social networks, citation networks, biological networks, etc. Accordingly, there has been a wealth of research on mining the rich information in networks, including link prediction [1] and vertex classification [2]. As a fundamental component in these downstream network analysis, the representation of the node plays a critical role in network mining. Currently, there has been a tremendous surge of research interest in learning the node representations in a network, i.e., network embedding [3]–[6].

Despite the gratifying results achieved, previous methods are mainly designed for the single-layer network. However, real-world networks are usually associated by sharing the same set of nodes [7], i.e., multiplex network. The multiplex network is referred to contain multiple types of relationships among nodes, and each layer depicts one topological network structure corresponding to a particular relationship. Multiplex networks are common to be observed in real-life. For example, people are usually involved in several online social platforms, such as WeChat, Facebook, Twitter, and the same set of the people can be expressed as a multiplex network, where each layer illustrates one type of the social relationship corresponding to the platform. Another similar example can be observed in bioinformatics, such as the multiple genetic and protein interactions network, where each layer represents one type of genetic interaction. In recent years, many works have been proposed to mine the rich information in multiplex networks, including network fusion [8], [9], cross-network recommendation [10], etc. As the structures of the layers are related, it is necessary to incorporate the information across multiple layers to improve the node embedding vectors. However, previous works only concern one type of interaction between nodes [4]–[6], so they fail to preserve the correlations among the layers.

To tackle this problem, several literatures have been proposed recently [11]–[15]. However, there still exist some important issues which need further concern. First, most of them [11]–[13] hold a complementary assumption, which indicates that the structures of the layers are similar, and the edges are complementary among the layers. Nevertheless, the related layers can show quite diverse structures due to the differences in their semantic meanings, and this assumption can be easily violated in many scenarios [16], [17]. Second, previous works consider all the cross-layer information is useful while ignoring the potential noise in this information. For example, the surroundings of a person in a professional network may become noise information for the same person in a friend network. Therefore, preserving this structural noise can inevitably sacrifice the distinctive properties of the identical node in different layers.

For solving these problems, we, in this paper, rethink how to utilize the cross-layer information more rationally in a fine-grained perspective. We thus propose a Noise-Aware Network Embedding approach for Multiplex Network, namely NANE. The core idea of NANE is to measure the confidence level of an extra edge from the other layers by comparing the two nodes' roles between the layers. NANE has two attractive characters: (i) NANE considers more general scenario, where the layers in the multiplex network can be (un)directed, (un)weighted or even uncompleted, which means some nodes in a layer may not have corresponding nodes in other layers. (ii) NANE is more flexible and has stronger noise immunity compared to existing works, because our NANE can

Jingping Bi (bjp@ict.ac.cn) is the corresponding author

978-1-7281-2009-6/$31.00 ©2019 IEEE paper N-19593.pdf

reasonably integrate the complementary information among the layers. We summarize our contributions as follows:

- We propose a noise-aware network embedding method for the multiplex network (NANE), which rationally integrates the cross-layer information of a given multiplex network while preserving the distinctive properties of each layer to boost the quality of node embedding.
- NANE can filter out the potential noise as much as possible by considering the roles of the identical node in different layers. Compared to existing methods, our NANE is more flexible as well as anti-noise, also appropriate for the multiple layers with diverse structures.
- Extensive experiments show that our NANE significantly outperforms other baselines, i.e., achieve 5% improvement with PierreAuger dataset on link prediction task. Furthermore, NANE can achieve 5% to 15% improvement with all noisy multiplex networks compared to existing multiplex network embedding methods.

We organize the remains of this paper as follows. Section 2 states the problems of multiplex network embedding, and investigate the inherent vulnerabilities of previous works. We formally describe our NANE in Section 3 and then validate our approach by analyzing extensive experiments in Section 4. We present related work in Section 5 and finally conclude our work in Section 6.

II. Problem Statement

Multiplex network embedding aims at mapping multiple types of related layers into continuous latent space, which benefits a wide range of multiplex network mining tasks. Previous single-layer network embedding methods cannot work for the multiplex network, as they only consider the correlations among nodes in one perspective, ignoring the relationships among different layers. Therefore, recent works [11]–[15] have been proposed to handle the representation learning problem on the multiplex network. However, there still exist two facets of vulnerabilities:

(i) Most of the previous works [11]–[14] make a complementary assumption, which indicates that the structures of the layers are similar and the edges are complementary across the layers. However, this assumption exists vulnerability and can be easily offended in practice. For example, in bioinformatics, different types of genetic interactions can construct quite diverse network structures [17]. Another similar example can also be found in online social network sites [16], where users' behavior may be divergent and platform dependent, making different social platforms show various connection relationships.

(ii) Previous works use all of the cross-layer information but ignore the potential noise in the information, which sacrifices the quality of the learned embedding vectors. Fig. 1 shows a toy example with three layers. According to previous works, layer L_1 can provide an extra information for the other two layers that there can be an edge e_{v_1,v_2} in layers L_2 and L_3 because the two nodes have been connected in L_1. However, it is unreasonable in this example. Considering the two nodes v_1 and v_2 in layer L_2, each of them has a similar local structure with its counterpart in layer L_1, so it is natural to infer that there can be a potential edge e_{v_1,v_2} in L_2. But we may not intuitively get the same conclusion for layer L_3, as the two nodes are both in separate communities with new neighbors. Therefore, in this case, even if they have been connected in layer L_1, we still cannot infer that they will have interaction in L_3. So the cross-layer information of edge e_{v_1,v_2} from layer L_1 is useful for L_2, but can be the noise if we incorporate it into L_3. Previous works ignore this critical issue, so they can bring excessive noise into node embedding vectors, sacrificing the distinctive properties of the identical nodes in different layers.

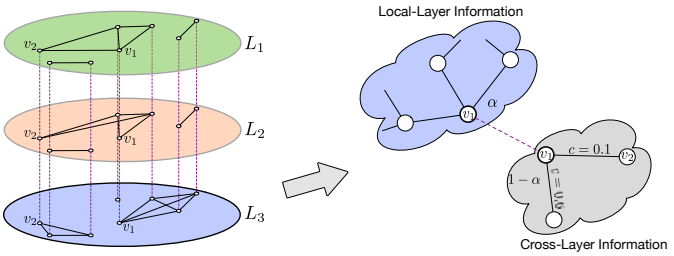

Fig. 1. Left: A toy example of a multiplex network with three layers. Right: The biased learning of NANE on layer L_3. The parameter α controls the balance of learning from local-layer information and cross-layer information. The value c in cross-layer information denotes the confident level of each edge.

There are two challenges to tackle the issues above: First, how to determine if the information from other layers is the complementary information or a noise; Second, how to balance the preference of preserving information from local layer and cross-layer. Thus, we propose a Noise-Aware Network Embedding method NANE. Unlike previous works, NANE holds such an assumption: an edge in a layer can be the complementary information for another layer if the two nodes take similar roles between the layers, where the role of a node is determined by its local structure. This assumption is very intuitive and interpretable, i.e., if a node plays similar roles in two different layers, its structural information in one layer will be useful to the other. Holding this assumption, NANE can make rational use of the cross-layer information in a more flexible and robust way, by comparing the roles of the counterparts to filter out the noise information and controlling the tendency of learning from local-layer or cross-layer.

III. Multiplex Network Embedding

In this section, we first define some concepts and propose the notations used in this paper. After that, we formally illustrate our method.

A. Problem Formulation

In this paper, we only consider the network structure of each layer. We define a multiplex network as follow:

Definition 1. *Multiplex Network is a multi-layer network therein all layers share the same set of nodes. To be more general, we assume that each layer can be (un)directed, (un)weighted or even uncompleted, which means nodes in a given layer can have no counterpart in other layers. We define a multiplex network as* $\mathcal{G} = \{L^1, ..., L^i, ..., L^n\}$*, where* $L^i = (V, E^i, W^i)$ *denotes the* i*-th layer in* $\mathcal{G}$ *and* n *is the number of layers.* V *is the set of nodes of whole network* $\mathcal{G}$*. As each layer* L^i *may not contain all the nodes in* V*, we assume the nodes that do not exist in layer* L^i *as the isolated nodes.* E^i *denotes the set of edges in layer* L^i *and* W^i *denotes the weight of each edge. We use* G^i *to represent the adjacency matrix of layer* L^i*, where each entry in* G^i *denotes the weight of an edge. For concise, we also use* u*,*v*,*z *to denote nodes.*

Definition 2. *Cross-Layer Information. For each given layer* L^i *in a multiplex network* $\mathcal{G}$*, the structural information from this layer is called the local-layer information, while the information from other layers is the cross-layer information for* L^i*, which we also call the extra information for* L^i*.*

Definition 3. *Noise Information. The noise information for each layer* L^i *is referred as the redundant or unreasonable information from cross-layer information, e.g. the edge* e_{v_1,v_2} *from layer* L_1 *for layer* L_3 *in Fig. 1.*

Definition 4. *Local Structure. The local structure of a node* v *contains its surrounding nodes, e.g. nodes reachable in* K*-hop, and the connections among them.*

Multiplex network embedding aims at learning the latent representation for each node in each layer, preserving both local-layer information and the cross-layer information of counterparts.

B. NANE: Noise-Aware Network Embedding for Multiplex Network

The core of NANE is to integrate the cross-layer information while filtering out the noise edges. Depend on the assumption mentioned above, NANE judges whether the extra information of an edge $e_{u,v}$ from other layers is a noise by considering the local structures of the two nodes u, v in different layers. So the first step in NANE is to calculate the similarities of the counterparts and give the confidence level of each extra information of edge from cross-layer information.

1) Measuring the Similarity of Counterpart: NANE measures the local structural similarity between a node and its counterparts by comparing a K-step context vector, which reflects the proximity between a node and its surroundings reachable within K steps in a layer. Considering the adjacency matrix G^i of i-th layer L^i, if there is an edge between node u and v in L^i then $G^i_{u,v} > 0$, otherwise $G^i_{u,v} = 0$. We first calculate the following diagonal matrix D^i for each layer, which is known as the degree matrix of the adjacency matrix G^i,

$$D^i = diag(d^i), \tag{1}$$

where d^i is the sum of each row in G^i and each entry d^i_u is:

$$d^i_u = \sum_z G^i_{uz}. \tag{2}$$

Then we get the row-normalized adjacency matrix of i-th layer as follow:

$$A^i = (D^i)^{-1} G^i, \tag{3}$$

where each entry $A^i_{u,v}$ can be regarded as the probability of a transition from u to v in one step, called the 1-step transition matrix of layer L^i.

We can then define the k-step transition matrix as follow:

$$(A^i)^k = \underbrace{A^i ... A^i}_{k}. \tag{4}$$

For each node u in layer L^i, the closer a node v is to u, the more important this node v is to u. Therefore, we can calculate the K-step context matrix through a weighted average as follows:

$$C^i = \frac{1}{K} \sum_{k=1}^{K} \frac{(A^i)^k}{k}. \tag{5}$$

The u-th row in C^i is called the K-step context vector of node u, which denotes the proximity between u and each other node in L^i, and can reflect the role of node u in this layer. To remove the influence of the center node itself, we set the diagonal entries in C^i to 0. Therefore, we can use this to measure the similarities of the counterparts in different layers. In this work, we intuitively take cosine similarity to calculate the structural similarity of each common node u between layers L^i and L^j as :

$$s^{(i,j)}_u = cos(C^i_{u,:}, C^j_{u,:}), \tag{6}$$

It is obvious that if node u takes similar role in L^i and L^j, the $s^{(i,j)}_u$ will tend to be 1. Note that, the K-step context vector of each isolated node is a zero vector, so the problem of uncompleted layer do not affect the calculation of Equation (6). We thus can give the confidence level of each extra edge $e_{u,v}$ from layer L^j to layer L^i as:

$$c^{(i,j)}_{u,v} = s^{(i,j)}_u \cdot s^{(i,j)}_v, \tag{7}$$

and for each extra edge $e_{u,v}$ from other layers, NANE records the biggest confidence level for layer L_i:

$$c^i_{u,v} = max(\{c^{(i,j)}_{u,v}, j \in \{1, 2, \cdots, n\}, j \neq i\}). \tag{8}$$

2) Noise-Aware Network Embedding: As illustrated in Figure 1, for each layer L^i, NANE takes a biased learning between the local-layer information of layer L^i and the cross-layer information from other layers. In NANE, a hyper-parameter α is used to balance the information from the two parts, and a biased random walk is performed on each layer L^i: considering a random walk now resides at node u, the transition probability for it to travel to the next node v can be defined as:

$$\pi(u,v) = \begin{cases} \alpha \cdot w^i_{u,v} & \text{if } e_{u,v} \in E^i \\ (1-\alpha) \cdot c^i_{u,v} & \text{if } e_{u,v} \in E \bigcap e_{u,v} \notin E^i, \end{cases} \tag{9}$$

TABLE I
STATISTICS OF DATASETS

Dataset	AM-LK	SacchCere	ArXiv	PierreAuger
# layers	2	7	13	16
# nodes	45,732	6,571	14,489	514
# edges	218,594	282,755	59,026	7,153

where the weight $w^i_{u,v}$ of each edge is normalized. Parameter α controls the likelihood of traveling through an edge in the local layer or other layers. A larger α make NANE tend to explore in the local layer, while a smaller one encourages the cross-layer exploration. For the local-layer exploration, at each step, the random walk is simply biased towards the weight of edges. For the cross-layer exploration, the random walk tends to travel through the edge with a high confidence level. Therefore, this flexible sampling strategy benefits the model to preserve useful cross-layer information and avoid potential noise edges as much as possible. To speed up the bias random walk across the layers, we adopt Alias sampling strategy in [6].

3) Optimization: We have obtained sets of node sequences on each layer and then we can perform the Skip-Gram algorithm over the sequences to learn embeddings. For a random walk on i-th layer L^i, we take w as half of the window size. Thus, in each window $n^i_{j-w}, n^i_{j-w+1}, \cdots, n^i_{j+w-1}, n^i_{j+w}$, where n^i_j denotes the j-th node in this sequence, our objective is to minimize the following negative log-likelihood:

$$-\prod_{k=j-c}^{k=j+c} logP(n^i_k|n^i_j), \tag{10}$$

where $P(n^i_k|n^i_j)$ is defined as the conditional probability of node n^i_k generated by node n^i_j as:

$$P(n^i_k|n^i_j) = \frac{exp(\mathbf{v}^i_{n_k} \cdot \mathbf{v}^i_{n_j})}{\sum_{n_z} exp(\mathbf{v}^i_{n_z} \cdot \mathbf{v}^i_{n_j})}, \tag{11}$$

where $\mathbf{v}^i_{n_j}$ represents the node embedding vector of node n^i_j. To speed up the training process, following [18], we use negative sampling to approximate the objective function as:

$$E = -log\sigma(\mathbf{v}^i_{n_k} \cdot \mathbf{v}^i_{n_j}) - \sum_{n^i_z \in \mathcal{N}_{n^i_j}} log\sigma(-\mathbf{v}^i_{n_z} \cdot \mathbf{v}^i_{n_j}), \tag{12}$$

where $\sigma(x) = 1/(1 + exp(x))$ is the sigmoid function and $\mathcal{N}_{n^i_j}$ is set of randomly negative nodes for node n^i_j in layer L^i.

IV. EXPERIMENT EVALUATION

In this section, we conduct extensive experiments to evaluate our method. Previous works on multiplex network embedding verify the effectiveness of embedding vectors mainly using link prediction task while the work [13] also test the effectiveness via multi-label classification. The noise in cross-layer information has a great influence on link prediction while has a relatively small impact on classification task, so in this work we verify the effectiveness of our method on link prediction task. Specifically, we compare with several state-of-art methods on four real-world networks in both normal environment and noisy scenarios. We first explain the details of datasets and baseline methods. Then we show the experimental results and conduct a parameter study of our NANE.

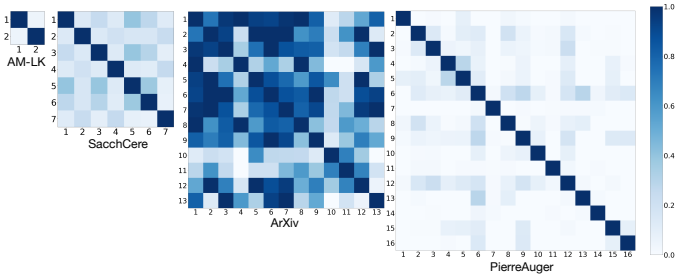

Fig. 2. An illustration of the similarity between layers. Each grid denotes the global structural similarity between each two layers in the dataset. The darker the color, the more similar the two layers are in structure.

A. Experiment Configuration

1) Data Set: We perform experiments on four real-world networks from academic, bioinformatics, and online social platform and summarize the statistics of the datasets in Table I. We also present the structural similarities between each pair of layers in Fig. 2 by comparing the common neighbors reachable in three steps of the counterparts. We detail the datasets as follows:

ArnetMiner-Linkedin. ArnetMiner[1] [19] is an expertise search and mining service for the academic community. Linkedin[2] is a world's professional network where users can maintain their profile page and connections. We extract a subgraph from [20] where the links in ArnetMiner represent the co-author relationships and the links in Linkedin denote that if two profiles of users are co-viewed by others. This network contains 45,732 nodes, 218,594 edges.

SacchCere[3] [21], [22] is a subset of BioGRID, which is a public database that archives and disseminates genetic and protein interaction data from humans and model organisms. It contains 6,571 nodes, 282,755 connections, and 7 layers, where each layer represents one type of genetic interaction.

ArXiv[3] [23] consists of the layers corresponding to different ArXiv categories. There are 14,489 nodes, 59,026 coauthorship connections, and 13 layers. The weight of each connection is the times of cooperation between the two authors.

PierreAuger[3] [24] contains 16 layers corresponding to different working tasks within the Pierre Auger Collaboration. There are 514 nodes and 7,153 coauthorship connections.

2) Baseline: We compare NANE with five competitive baselines, which can be categorized into two groups: single-layer network based methods and multiplex network based methods.

[1]https://www.aminer.cn/
[2]https://www.linkedin.com
[3]https://comunelab.fbk.eu/data.php

TABLE II
RESULT OF LINK PRECTION.

Method	DeepWalk	LINE	PMNE	MTNE	MELL	NANE
ArXiv	96.12	94.75	98.24	97.93	98.84	**99.18**
SacchCere	78.29	68.87	81.35	81.19	80.21	**83.10**
PierreAuger	81.01	74.25	78.81	79.87	81.83	**87.11**
AM-LK	89.56	73.97	88.67	87.27	87.54	**93.25**

Single-layer network based methods contain two recent works for embedding nodes in each layer separately:

DeepWalk [4] is a typical network embedding method that learns vertex representations based on the network structure. It performs a random walk on the network to obtain vertex sequences and conducts Skip-Gram model to train the sequences.

LINE [6] is another model which aims at learning node embeddings in large-scale networks. It minimizes a loss function to preserve both first-order and second-order proximity between nodes.

Multiplex network based methods contain three baselines which not only consider the local layer information but the cross-layer information:

PMNE [11] proposes three methods to obtain one overall embedding for each node. In this experiment, we compare with its final Co-Analysis method, which performs a biased random walk across the layers and conducts skip-gram model to train the node sequences.

MTNE [13] proposes a family of algorithms to learn multi-task network embedding. We compare with MTNE-C, which builds a bridge among different layers by sharing a common embedding among the counterparts.

MELL [14] is a multi-layer embedding model which simultaneously learns the node embedding vectors and a layer embedding for each layer.

We perform a five-fold cross-validation on each dataset. We randomly sample an unconnected node pair as a negative edge for each positive edge in each test set and use both of them for testing. Following previous works, we here adopt a standard evaluation metric ROC-AUC [25] for each layer and present the average score. Furtherly, to verify the robustness of models in the face of noisy networks, we randomly add noise edges which do not exist in any layer into training set and also calculate the average AUC score. For fair comparison, we set the dimension d=100 for all methods.

B. Parameter Settings

The parameters of all the baselines are set to the best settings as they reported. For Deepwalk, we set walks per vertex to 20, window size to 10 and walk length to 80. For LINE, we employ both first-order and second-order proximity and obtain representations via concatenation, and we set the number of negative samples to 5. For PMNE, we follow their default settings $\{\alpha, p, q\}$=$\{0.5, 0.5, 0.5\}$. For MTNE, we follow the default settings where the regularization coefficients are set as 1.0. For MELL, the objective function of it is applicable to the network where each layer has the same number of nodes, which is violated in our datasets. This makes it show poor performance in our experiments. Therefore, we modify its regularization to make it suitable for our datasets. We set $\{k, \lambda, \beta, \gamma\}$=$\{4, 1, 1, 1\}$

For our NANE, we set $\alpha = 0.8$ for datasets SacchCere and PierreAuger, $\alpha = 0.4$ for ArXiv and 0.6 for ArnetMiner-Linkedin. To accelerate the training process, we set the number of negative sample to 1. For the other parameters in our method, we set the same as in DeepWalk.

C. Performance Evaluation

We first compare our NANE with all the baselines on link prediction task. After that, we perform another set of experiments to analyze the influence of noise on each multiplex network embedding model.

1) Link Predition: We present the results of link prediction in Table II. Our NANE continually outperforms all baselines methods significantly. Taking PierreAuger as an example, we can even gain 5% improvement compared to all the baselines. This observation illustrates the effectiveness of our NANE on preserving cross-layer information. Moreover, we observe that the model designed for the single-layer network can even achieve better results than the multiplex network embedding methods, e.g., DeepWalk can achieve better performance than the other three multi-layer embedding methods on dataset ArnetMiner-Linkedin. This is mainly because the layers of this dataset tend to show diverse global structures, which violates the complementary assumption, making these multi-layer embedding methods preserve much noise information into node embedding vectors. Therefore, this phenomenon also indicates that not all the cross-layer information is essential to be preserved. Unlike previous works, our NANE can measure the confidence level of each extra information, learning from the local layer information while preserving useful cross-layer information. Thus, NANE can achieve much better results than all the baselines and be more flexible and practical than previous works on these complex datasets.

2) Influence of Noise: We perform a set of experiments to explore the influence of structural noise on each multiplex network embedding model and report the results in Fig. 3. We first add different level of noise edges into training sets and calculate the average AUC score of link prediction. For each positive edge in test set, we then replace its corresponding negative edge in AUC calculation with an edge of other layers from the set of the added noise edges, i.e. -ne in Fig. 3. For fair comparison, we use the same hyper-parameter in this set of experiments. Upon the experimental results, we have the following observations:

(i) The performance of all methods decreases with the increase of noise level. This demonstrates that the structural noise has an influence on the quality of the learned embeddings of each model designed for multiplex networks. However, our NANE can produce the stable performance with the increase of the noise level and achieves 5% to 15% improvement than other methods on all datasets.

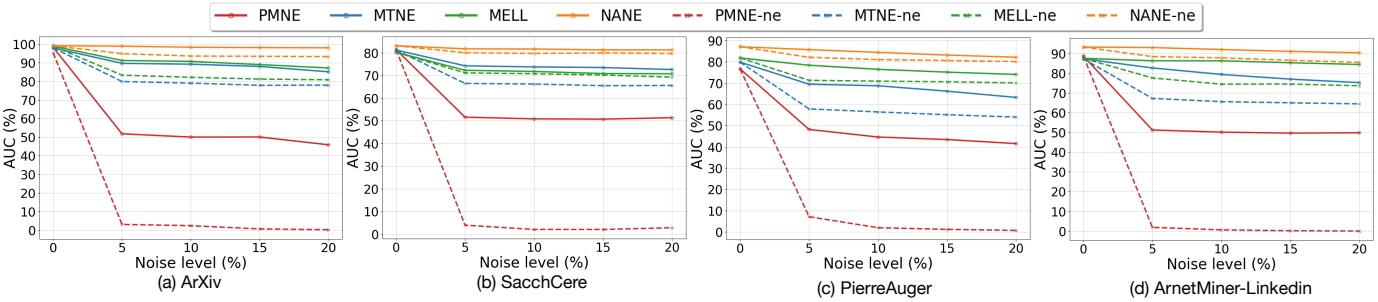

Fig. 3. The results of link prediction with different levels of noise. The lines of -ne denote the results of replacing the randomly sampled edges in AUC calculation with the added noise edges. (This set of figures is best viewed in color.)

TABLE III
PARAMETER STUDY OF α.

Dataset	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
ArXiv	98.27	98.37	98.40	**99.18**	98.91	98.65	98.54	98.44	97.22
SacchCere	75.28	77.99	80.04	80.89	81.48	82.55	82.63	**83.10**	80.65
PierreAuger	72.39	76.87	77.44	79.04	80.14	82.93	84.72	**87.11**	84.14
AM-LK	91.79	92.13	92.33	92.54	92.62	**93.25**	92.69	91.14	90.84

(ii) The performance of the other three methods drops sharply when we take the noise edges as negative edges for AUC calculation. This indicates that the learned vectors preserve much noise from the cross-layer information, losing the specific properties of a node in its selected layer. For PMNE, as it merges multiple layers into one overall embeddings vectors, it inevitably keeps much noise and reduces the quality of learned embedding. Thus it shows poor performance in noisy scenarios. For MTNE and MELL, although they can preserve much distinguish information for each layer, they integrate all the cross-layer structural information while failing to judge which is essential. This makes them also preserve many noisy edges from other layers and cannot achieve better performance. Unlike these methods, the performance of our NANE declines slightly even using the noise edges for AUC calculation. This outperformance demonstrates that our NANE can learn the complementary information while filtering out the unnecessary or noisy cross-layer information, which is practical and robust in real-life multi-layer network applications.

To summarize, on all datasets, our NANE can always achieve better performance and outperform all the baselines in noisy scenarios. These experimental results demonstrate our NANE is flexible and robust with noise in various multi-layer network scenarios.

D. Parameter Sensitivity

In this section, we study the parameter α in our NANE. We vary the value of α from 0.1 to 0.9 with an equal interval of 0.1 and check its impact on link prediction task with each dataset. We report the results in Table III.

We can observe that the best setting of α is different in each dataset. Combined with Fig. 2, we find that the value of the best setting is related to the global similarities among the layers. For the layers with similar global structures, e.g., ArXiv, α tends to be set as a small value, which means NANE can accept more of the cross-layer information, while for the networks with diverse layers, a larger α can make NANE focus on preserving distinct structural information of each layer and less cross-layer information is integrated. When the value of α is close to 1, our method will degenerate into network embedding on each layer seperately and the results wil be close to DeepWalk in Table II.

V. RELATED WORK

The rapid growth of network-based analysis tasks has brought significant concerns on the representations of the vertices. Network embedding has been emerged as an effective and efficient approach for learning low-dimension distributed representations for the nodes in the network and is studied intensively in recent years.

A. Single-Layer Network Embedding

Inspired by Skip-Gram [18], a widely adopted word representation learning model in natural language processing, DeepWalk [4] performs a random walk over networks to generate vertex sequences and conducts Skip-Gram to obtain node embeddings. On top of DeepWalk, Node2vec [5] modifies the random walk strategy into a biased random walk to explore network structure more efficiently. In addition to the Skip-Gram based methods, some other works design particular objective function to refine node embeddings. LINE [6] optimizes two objective functions to separately approximate first-order and second-order proximity in the large-scale networks. Tri-Party [26] proposes three parts of objectives to incorporate

text contents and label information into node embedding vectors. Another general approach for obtaining node embeddings is matrix factorization. GreRap [27] proposed a matrix factorization based methods to encode k-step representations, where each step reflects different local information. TADW [28] incorporates text contents into network embedding under the framework of matrix factorization. Some other works propose to take advantage of the Deep Neural Network (DNN) to preserve network structure in high-order. SDAE [29] and SDNE [30] both propose an AutoEncoder based network embedding method to learn representations for vertices.

B. Multiplex Network Embedding

Although previous methods have achieved satisfactory performance on many downstream network analysis tasks, they only consider the single-layer network, ignoring the multiplex network. Multiplex network is a multi-layer network, where the layers share the same set of nodes. Each layer in the multiplex network characterizes one facet of the node while the structures of the layers are usually associated. Nowadays, multiplex networks have been extensively studied as a special type of the networks, and many works have been proposed to focus on graph mining on it [10], [31]–[34]. Because of the complex relationship among the layers, existing single-layer network embedding methods cannot work for the multiplex networks.

To make the data mining tasks on the multiplex network more actionable, several works have been proposed to transform the multiplex network into a low-dimension vector space. Most of the previous works assume that the layers are similar and complementary. PMNE [11] proposes two simple merge-based methods which only consider inter-layer edges or intra-layer edges, and one cross-layer method which performs a biased random-walk across each layer, to obtain one overall embedding for each node. MTNE [13] builds a bridge among different layers by sharing a common embedding vector for the counterparts in different layers. SMNE [12] also shares a common embedding across layers, but propose one high-dimensional common embedding and one low-dimensional additional vector for each node to save memory occupation. MELL [14] proposes the method of simultaneously learning node embeddings and layer embeddings using all of the layer structures. However, real-life networks do not always satisfy this complementary assumption. For example, the professional relationship may not align well with friendship [15]. Until very recently, mvn2vec [15] introduces the concepts of preservation (exclusiveness) and collaboration (complementarity) between different types of layers and argues that preservation and collaboration can co-exist in a multiplex network. Mvn2vec manipulates the extent of model's preference for preserving the complementary information via controlling a hyperparameter.

However, all existing methods fail to consider the noise information among different layers, especially when the counterparts take entirely different roles. Preserving this noisy cross-layer information can sacrifice the distinct properties of each layer. To handle these problems, we propose NANE to jointly learn from multiple layers while filtering out the noisy information.

In addition to the works mentioned above, DMNE [35] considers the scenario where each layer may have different sets of nodes and the cross-network relationships can be many-to-many which are associated with weights. MVE [36] proposes an attention-based method to learn the representation of multi-view networks in supervised schemas. However, the scenarios of these works are different from this work.

VI. Conclusion

We have introduced a flexible embedding model for multiplex networks, namely NANE. It considers the local structure of each node in the layers as its roles to reasonably incorporate cross-network relationships and boost the quality of learned node representations. NANE is generic for multiple networks and robust with noise. A set of experimental results on real-world datasets show analytically that NANE can achieves better performance on all of the datasets and significantly outperforms existing methods on noisy network scenarios.

VII. ACKNOWLEDGMENTS

This work has been supported by the National Natural Science Foundation of China under Grant No.: 61702470.

References

[1] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 6, pp. 1150–1170, 2011.

[2] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective classification in network data," *AI Magazine*, vol. 29, no. 3, pp. 93–106, 2008.

[3] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[4] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, 2014, pp. 701–710.

[5] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, San Francisco, CA, USA, August 13-17, 2016*, 2016, pp. 855–864.

[6] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: large-scale information network embedding," in *Proceedings of the 24th International Conference on World Wide Web, WWW '15, Florence, Italy, May 18-22, 2015*, 2015, pp. 1067–1077.

[7] L. Solá, M. Romance, R. Criado, J. Flores, A. García del Amo, and S. Boccaletti, "Eigenvector centrality of nodes in multiplex networks," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 23, no. 3, pp. 175–1250, 2013.

[8] J. Zhang and P. S. Yu, "Multiple anonymized social networks alignment," in *Proceedings of IEEE International Conference on Data Mining, ICDM '15, Atlantic City, NJ, USA, November 14-17, 2015*, 2015, pp. 599–608.

[9] X. Kong, J. Zhang, and P. S. Yu, "Inferring anchor links across multiple heterogeneous social networks," in *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, CIKM '13, San Francisco, CA, USA, October 27 - November 1, 2013*, 2013, pp. 179–188.

[10] J. Zhang, P. S. Yu, and Z. Zhou, "Meta-path based multi-network collective link prediction," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, 2014, pp. 1286–1295.

[11] W. Liu, P. Chen, S. Yeung, T. Suzumura, and L. Chen, "Principled multilayer network embedding," in *Proceedings of the 17th IEEE International Conference on Data Mining Workshops, ICDM Workshops '17, New Orleans, LA, USA, November 18-21, 2017*, 2017, pp. 134–141.

[12] H. Zhang, L. Qiu, L. Yi, and Y. Song, "Scalable multiplex network embedding," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI '18, July 13-19, 2018, Stockholm, Sweden.*, 2018, pp. 3082–3088.

[13] L. Xu, X. Wei, J. Cao, and P. S. Yu, "Multi-task network embedding," in *Proceedings of the 4th IEEE International Conference on Data Science and Advanced Analytics, DSAA '17, Tokyo, Japan, October 19-21, 2017*, 2017, pp. 571–580.

[14] R. Matsuno and T. Murata, "MELL: effective embedding method for multiplex networks," in *Proceedings of the 27th World Wide Web Conference on World Wide Web, WWW '18, Lyon, France, April 23-27, 2018*, 2018, pp. 1261–1268.

[15] Y. Shi, F. Han, X. He, C. Yang, J. Luo, and J. Han, "Mvn2vec: Preservation and collaboration in multi-view network embedding," *CoRR*, vol. abs/1801.06597, 2018.

[16] S. Liu, S. Wang, F. Zhu, J. Zhang, and R. Krishnan, "HYDRA: large-scale social identity linkage via heterogeneous behavior modeling," in *Proceedings of the 41th International Conference on Management of Data, SIGMOD '14, Snowbird, UT, USA, June 22-27, 2014*, 2014, pp. 51–62.

[17] J. Ni, H. Tong, W. Fan, and X. Zhang, "Inside the atoms: Ranking on a network of networks," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA, August 24 - 27, 2014*, 2014, pp. 1356–1365.

[18] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013.

[19] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: Extraction and mining of academic social networks," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '08, Las Vegas, Nevada, USA, August 24-27, 2008*, 2008, pp. 990–998.

[20] Y. Zhang, J. Tang, Z. Yang, J. Pei, and P. S. Yu, "COSNET: connecting heterogeneous social networks with local and global consistency," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15 Sydney, NSW, Australia, August 10-13, 2015*, 2015, pp. 1485–1494.

[21] M. De Domenico, V. Nicosia, A. Arenas, and V. Latora, "Structural reducibility of multilayer networks," *Nature Communications*, vol. 6, p. 6864, 2015.

[22] C. Stark, B. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers, "Biogrid: a general repository for interaction datasets," *Nucleic Acids Research*, vol. 34, no. Database issue, pp. 535–539, 2006.

[23] M. De Domenico, A. Lancichinetti, A. Arenas, and M. Rosvall, "Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems," *Physical Review X*, vol. 5, no. 1, p. 011027, 2015.

[24] ——, "Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems," *Physical Review X*, vol. 5, no. 1, p. 011027, 2015.

[25] J. A. Hanley and B. J. Mcneil, "The meaning and use of the area under a receiver operating characteristic ROC curve," *Radiology*, vol. 143, no. 1, p. 29, 1982.

[26] S. Pan, J. Wu, X. Zhu, C. Zhang, and Y. Wang, "Tri-party deep network representation," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence, IJCAI '16, New York, NY, USA, 9-15 July 2016*, 2016, pp. 1895–1901.

[27] S. Cao, W. Lu, and Q. Xu, "Grarep: Learning graph representations with global structural information," in *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM '15, Melbourne, VIC, Australia, October 19 - 23, 2015*, 2015, pp. 891–900.

[28] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Y. Chang, "Network representation learning with rich text information," in *Proceedings of the 24th International Joint Conference on Artificial Intelligence, IJCAI '15, Buenos Aires, Argentina, July 25-31, 2015*, 2015, pp. 2111–2117.

[29] S. Cao, W. Lu, and Q. Xu, "Deep neural networks for learning graph representations," in *Proceedings of the 13th Conference on Artificial Intelligence, AAAI '16, February 12-17, 2016, Phoenix, Arizona, USA.*, 2016, pp. 1145–1152.

[30] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proceedings of the 22nd ACM International Conference on Knowledge Discovery and Data Mining, SIGKDD '16, San Francisco, CA, USA, August 13-17, 2016*, 2016, pp. 1225–1234.

[31] D. Cai, Z. Shao, X. He, X. Yan, and J. Han, "Community mining from multi-relational networks," in *Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD '05, Portugal, October 3-7, 2005*, 2005, pp. 445–452.

[32] C. Chen, H. Tong, L. Xie, L. Ying, and Q. He, "FASCINATE: fast cross-layer dependency inference on multi-layered networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, San Francisco, CA, USA, August 13-17, 2016*, 2016, pp. 765–774.

[33] H. Cho, B. Berger, and J. Peng, "Compact integration of multi-network topology for functional analysis of genes," *Cell systems*, vol. 3, no. 6, pp. 540–548, 2016.

[34] P. Bródka, K. Skibicki, P. Kazienko, and K. Musial, "A degree centrality in multi-layered social network," in *Proceedings of the 3th International Conference on Computational Aspects of Social Networks, CASoN '11, Salamanca, Spain, October 19-21, 2011*, 2011, pp. 237–242.

[35] J. Ni, S. Chang, X. Liu, W. Cheng, H. Chen, D. Xu, and X. Zhang, "Co-regularized deep multi-network embedding," in *Proceedings of the 27th World Wide Web Conference on World Wide Web, WWW '18, Lyon, France, April 23-27, 2018*, 2018, pp. 469–478.

[36] M. Qu, J. Tang, J. Shang, X. Ren, M. Zhang, and J. Han, "An attention-based collaboration framework for multi-view network representation learning," in *Proceedings of the 25th ACM on Conference on Information and Knowledge Management, CIKM '17, Singapore, November 06 - 10, 2017*, 2017, pp. 1767–1776.