Representation learning on multi-layered heterogeneous network

Delvin Ce ZHANG
Singapore Management University, cezhang.2018@smu.edu.sg

Hady W. LAUW
Singapore Management University, hadywlauw@smu.edu.sg

Representation Learning on Multi-Layered Heterogeneous Network

Delvin Ce Zhang[0000−0001−5571−9766] (✉) and Hady W. Lauw[0000−0002−8245−8677]

School of Computing and Information Systems, Singapore Management University, Singapore
{cezhang.2018,hadywlauw}@smu.edu.sg

Abstract. Network data can often be represented in a multi-layered structure with rich semantics. One example is e-commerce data, containing user-user social network layer and item-item context layer, with cross-layer user-item interactions. Given the dual characters of homogeneity within each layer and heterogeneity across layers, we seek to learn node representations from such a multi-layered heterogeneous network while jointly preserving structural information and network semantics. In contrast, previous works on network embedding mainly focus on single-layered or homogeneous networks with one type of nodes and links. In this paper we propose intra- and cross-layer proximity concepts. Intra-layer proximity simulates propagation along homogeneous nodes to explore latent structural similarities. Cross-layer proximity captures network semantics by extending heterogeneous neighborhood across layers. Through extensive experiments on four datasets, we demonstrate that our model achieves substantial gains in different real-world domains over state-of-the-art baselines.

Keywords: Representation Learning · Heterogeneous Network · Dimensionality Reduction.

1 Introduction

Much of the data on the Web can be represented in a network structure, ranging from social and biological to academic networks, etc. Network analysis recently attracts escalating research attention due to its importance and wide applicability. Diverse problems could be formulated as network tasks, e.g., recommending items to users on e-commerce [12]. As the primary information is the inherent structure of the network itself, one promising direction known as the *network embedding* problem is to learn the representation of each node, which could in turn fuel tasks such as node classification, node clustering, and link prediction.

Figure 1 illustrates an example network with various object types (users, movies, movie actors). These objects are connected via various links, e.g., a user may friend other users, favor some movies, and follow some actors, while a movie may share similar contexts as another (being added to the same preference folder, or recommended in the same movie list) or feature some actors. Network embedding learns a low-dimensional representation for each node (user, movie, or actor), which preserves the network information. In turn, the node representations may be used in applications such

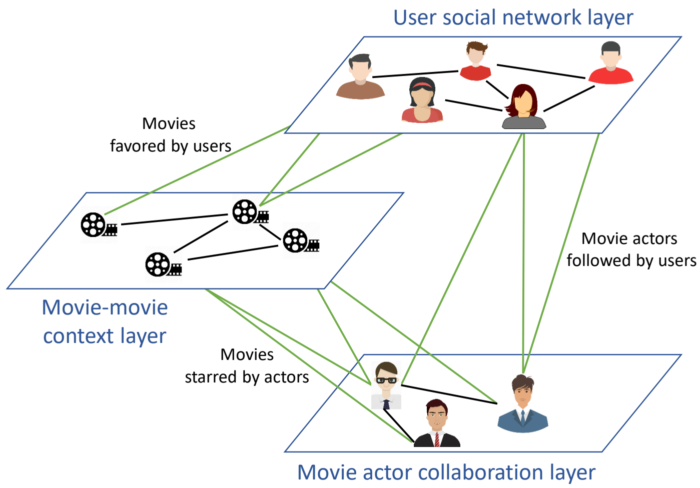


Fig. 1. Illustration of multi-layered heterogeneous network. Three homogeneous layers (user social network layer, *movie-movie* context layer, movie actor collaboration layer) are connected by heterogeneous interactions.

as predicting whether a user is likely to favor a movie, or whether a user is likely to friend another user.

Present work and challenges. Previous works on network embedding focus on *homogeneous* networks [19, 23]. They treat all nodes and edges as the same type, regardless of varying relations. Such homogeneous treatment may miss out on the nuances that arise from the diversity of associations (e.g., user favoring a movie has different semantics from movie featuring an actor).

More recent works recognize the value of absorbing the varying semantics into the node representation, modeling a *heterogeneous* network. However, to encode semantics, models such as Metapath2vec [6] rely on the notion of meta-path scheme (sequence of node types that make up a path). These are to be prespecified in advance, requiring domain-specific knowledge (incurring manual costs) or exhaustive enumeration of schemes (incurring computational costs). Other models [11] only consider each edge relation as one type of connections, but ignore the two end-point nodes are sometimes mutually homogeneous, thereby losing structural information in node embeddings.

Proposed approach and contributions. We observe that complex networks simultaneously exhibit homogeneous and heterogeneous tendencies. The interplay between the two gives rise to a *multi-layered* structure, whereby each layer encodes the structural connectivity of objects of the same type, and connections across layers bear rich semantics between objects of different types. Figure 1 can be seen as a *multi-layered heterogeneous network* with three layers. We offer a formal definition in Section 3.

Given the dual characters of multi-layered network, we seek to learn node embeddings that preserve both structure and semantics to facilitate downstream tasks, e.g., item recommendation in e-commerce, user alignment across multiple social networks [15]. In contrast to heterogeneous models that rely on prespecified schemes [6], the cross-layer proximity of our model naturally 'induces' various schemes by how it models layers, and its maximum order controls the semantics learning. In contrast to heterogeneous models [11] that do not consider that the two end-point nodes are sometimes mutually homogeneous, we use nodes of the same type to jointly preserve semantics, so as to embody structural proximity.

In this paper, we propose **Multi-Layered H**eterogeneous **N**etwork **E**mbedding, or (MULTILAYEREDHNE), describing how it models both intra-layer proximities to explore structural similarities in a breadth-wise propagation manner and cross-layer proximities for depth-wise semantics capture in Section 4. In a nutshell, like ripples expanding across the water, intra-layer proximities broadcast one node's homogeneous neighborhood hop by hop to investigate latent structural relationships. Cross-layer proximities iteratively extend heterogeneous relations layer by layer and leverage intra-layer proximities to *jointly* preserve network semantics.

Our contributions in this paper are as follows:

- Though "multi-layered" notion may have appeared in prior, here we articulate a concrete definition in the context of heterogeneous network. Importantly, we define the novel notions of intra- and cross-layer proximities underlining our approach.
- To capture network homogeneity and heterogeneity jointly, we propose a novel framework that encodes both structural proximity and network semantics into unified node embeddings by higher-order intra- and cross-layer proximities.
- We conduct extensive experiments on four real datasets, and the results validate the effectiveness of our model over state-of-the-art baselines.

2 Related Work

Here we review related research works for homogeneous, heterogeneous, and multi-layered network embedding.

Homogeneous Network Embedding. Homogeneous networks are those with one single type of nodes and links. DeepWalk [19] generates random walk on the network as corpus and applies skip-gram model to train the nodes. Node2vec [9] extends DeepWalk by simulating biased random walk to explore diverse neighborhoods. LINE [23] learns node representations by preserving first- and second-order proximities. GraRep [2] generalizes LINE to incorporate higher-order proximities, but may not scale efficiently to very large networks. These methods mainly focus on embedding network topology to preserve structural information.

Meanwhile, there are also models dealing with attributed homogeneous networks with task-specific supervision (e.g., GCN [26], GAT [24]). They are different from our model that embeds network in an unsupervised manner to support arbitrary downstream tasks. Others that operate on attributed graph for multi-modal learning (EP [8]) and are designed specifically for document network (Adjacent-Encoder [27]) are also not directly comparable.

Heterogeneous Network Embedding. Some heterogeneous network models leverage meta-path-based random walks to capture network semantics, such as Metapath2vec [6] and HIN2vec [7]. The applications of meta-path-based models (e.g. recommender systems) are also widely studied [20]. Some of them simulate meta-paths of specified

Table 1. Summary of main notations.

Notation	Explanation
$\mathcal{G}$	the input network
$\mathcal{V}, \mathcal{E}$	the node set and edge set, resp.
$\mathcal{O}, \mathcal{R}$	the node type set and edge type set, resp.
$\mathcal{L}$	the layer set
$\mathcal{I}_v^m$	the m^{th}-order intra-layer proximity of node v
$\mathcal{C}_v^n$	the n^{th}-order cross-layer proximity of node v
M, N	maximum order of intra- and cross-layer proximity, resp.
$\mathcal{K}$	number of negative samples

schemes on each network to preserve complex semantics. To this end, the cross-layer proximity of our model does not restrict to specific schemes, and its maximum order controls the semantics learning. There also exist some methods that do not require specific meta-paths, such as HeGAN [11], which utilizes GAN to generate fake nodes to train discriminator. More recently, Graph Neural Networks have been successfully applied to attributed heterogeneous networks with satisfactory results [25].

Multi-Layered Network Embedding. Multi-layered networks, as a set of interdependent network layers, appear in real-world scenarios including recommender and academic systems, cross-platform social networks, etc. Previous works focus on cross-layer links inference [5, 4] and network ranking [18]. MANE [14] studies representation learning on multi-layered networks by seeking low-dimensional node embeddings by modeling each intra-layer and cross-layer links. Our model has a couple of distinctions. For one, we incorporate higher-order proximities. For another, the manner in which we model proximities integrates nodes with similar structures, instead of predicting links individually. These differences do make a difference to the effectiveness of our node embeddings (see Section 5).

There exists another definition of "multi-layered" network [16], which really is *multiplex* or *multi-view* network [3], where there are multiple relationships between the same set of nodes. In contrast, multi-layered network in this paper refers to a set of interdependent network layers, each with a different set of nodes.

3 Definitions and Problem Formulation

We introduce intra- and cross-layer proximities, and formalize our problem. Table 1 lists the notations.

Definition 1 *A **Heterogeneous Information Network** (HIN) $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{O}, \mathcal{R}\}$ consists of a node set $\mathcal{V}$ and an edge set $\mathcal{E}$. This network is associated with a node type mapping function ϕ: $\mathcal{V} \rightarrow \mathcal{O}$ and an edge type mapping function φ: $\mathcal{E} \rightarrow \mathcal{R}$. $\mathcal{O}$ and $\mathcal{R}$ represent the sets of predefined node types and edge types respectively, where $|\mathcal{O}| + |\mathcal{R}| > 2$.*

We use the terms *edge* and *link* interchangeably, ditto for *network* and *graph*. Multi-layered network is defined over HIN, with an additional requirement of layers.

Definition 2 *A* ***Multi-Layered Heterogeneous Network*** $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{O}, \mathcal{R}, \mathcal{L}\}$ *is a connected HIN that contains a layer set* $\mathcal{L}$ *of* $|\mathcal{L}| > 1$ *homogeneous network layers. In addition to* ϕ *and* φ*, we have two more mapping functions. The node mapping function* θ*:* $\mathcal{V} \rightarrow \mathcal{L}$ *projects each node* $v \in \mathcal{V}$ *to a certain layer* $l_v \in \mathcal{L}$*. The edge mapping function* ϑ*:* $\mathcal{E} \rightarrow \mathcal{L} \times \mathcal{L}$ *places each edge* $e \in \mathcal{E}$ *between two layers* $(l_{e,1}, l_{e,2}) \in \mathcal{L} \times \mathcal{L}$*.*

$\mathcal{L} \times \mathcal{L} = \{(l_{e,1}, l_{e,2}) | l_{e,1}, l_{e,2} \in \mathcal{L}\}$ *represents the Cartesian product of two sets. Thus* $l_{e,1}$ *and* $l_{e,2}$ *could be the same, and edge* e *is intra-layer, otherwise cross-layer.*

Figure 1 illustrates a multi-layered network with three homogeneous layers (*user–user*, *movie–movie*, *actor–actor*) and three heterogeneous interactions (*user–movie*, *movie–actor*, *user–actor*). Intra-layer edges (black) connect nodes of the same type. Cross-layer edges (green) of different relations connect arbitrary type of nodes. Multi-layered networks are a subset of HIN, as each layer contains intra-layer edges.

Problem 1. Given a multi-layered heterogeneous network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{O}, \mathcal{R}, \mathcal{L}\}$, the goal of **Representation Learning on Multi-Layered Heterogeneous Network** is to learn a mapping function to project each node $v \in \mathcal{V}$ to a low-dimensional space $\mathbb{R}^d$ where $d \ll |\mathcal{V}|$. The node representation in the new space should preserve both structural proximities and network semantics within $\mathcal{G}$.

To harness contributions from intra-layer edges containing structural information within layers and from cross-layer edges capturing semantics, we propose the MULTI-LAYEREDHNE framework built on *intra-layer proximity* and *cross-layer proximity*.

Definition 3 *The* m^{th}*-order* ***Intra-Layer Proximity*** *of node* v *is defined as the set of nodes that can be reached by* m *intra-layer edges from* v*:*

$$\mathcal{I}_v^m = \{v^m | v^{m-1} \in \mathcal{I}_v^{m-1}, (v^{m-1}, v^m) \in \mathcal{E}, l_{v^m} = l_{v^{m-1}}\}, \tag{1}$$

where $m = 1, 2, ..., M$*, and* $\mathcal{I}_v^0 = \{v\}$*.*

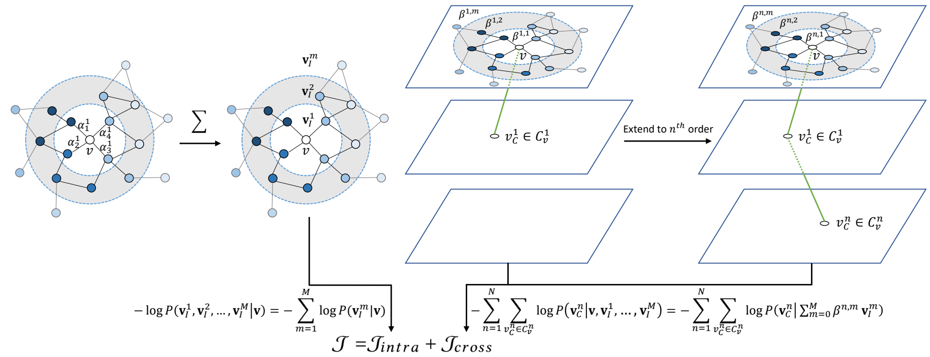
This concept is illustrated by Figure 2(a) (best seen in color). Here we suppose this network is homogeneous within a network layer. Node v's first-order intra-layer proximity consists of four nodes inside the inner white circle with black links connecting them. Similarly nodes lying in the gray annulus represent v's second-order intra-layer proximity. We can extend this concept up to M^{th} order and obtain $\mathcal{I}_v^M$.

Definition 4 *The* n^{th}*-order* ***Cross-Layer Proximity*** *of node* v *is defined as the set of nodes that can be reached by* n *cross-layer edges from* v*:*

$$\mathcal{C}_v^n = \{v^n | v^{n-1} \in \mathcal{C}_v^{n-1}, (v^{n-1}, v^n) \in \mathcal{E}, l_{v^n} \neq l_{v^{n-1}}\}, \tag{2}$$

where $n = 1, 2, ..., N$*, and* $\mathcal{C}_v^0 = \{v\}$*.*

To illustrate this concept, we use Figure 2(d) (best seen in color). v_C^1 represents one node in v's first-order cross-layer proximity with a cross-layer green link connecting them. Extending this example up to N^{th} order, we obtain $\mathcal{C}_v^N$.



(a) Evaluate intra-layer attention α_i^m between $\mathbf{v}$ and nodes in each intra-layer proximity. (b) Weighted aggregation and generate $\mathbf{v}_I^1, \mathbf{v}_I^2, ..., \mathbf{v}_I^M$ for each proximity. (c) Evaluate cross-layer attention $\beta^{1,m}$ between $\mathbf{v}_C^1$ and $\mathbf{v}, \mathbf{v}_I^1, ..., \mathbf{v}_I^M$. (d) Extend to n^{th}-order cross-layer proximity and use $\mathbf{v}, \mathbf{v}_I^1, ..., \mathbf{v}_I^M$ to *jointly* evaluate loss.

Fig. 2. Illustration of intra- and cross-layer proximity modeling.

4 Model Architecture

We now describe our proposed model MULTILAYEREDHNE. It consists of two modeling components. First, intra-layer proximity modeling (Figures 2(a) and (b)) simulates the breadth-wise propagation across homogeneous neighbors within layers to explore structural similarities. Second, cross-layer proximity modeling (Figure 2(c) and (d)) captures semantics by extending heterogeneous neighborhood across layers.

4.1 Intra-Layer Proximity Modeling

Suppose that $\mathbf{v} \in \mathbb{R}^d$ is the embedding of a node v. This is the quantity that we seek to derive. Intra-layer proximity concerns the relationships between v and its homogeneous neighbors from the same layer. We first consider the first-order proximity ($m = 1$). This effectively concerns the direct neighbors of v, collectively denoted $\mathcal{I}_v^1$. The embedding of each first-order neighbor $v_i^1 \in \mathcal{I}_v^1$ is denoted $\mathbf{v}_i^1 \in \mathbb{R}^d$. Since intra-layer proximities contain nodes of the same type, it is reasonable to treat these nodes homogeneously. Thus we derive a representation of v's first-order intra-layer proximity $\mathbf{v}_I^1$ as a weighted aggregation of its neighbors' embeddings.

$$\mathbf{v}_I^1 = \sum_{i=1}^{|\mathcal{I}_v^1|} \alpha_i^1 \mathbf{v}_i^1. \tag{3}$$

Not all neighbors are equally important to v. Some may be of greater importance. Therefore, the aggregation in Eq. 3 factors in an attention coefficient w.r.t. v.

$$a_i^1 = \sigma(\mathbf{v}^T \mathbf{v}_i^1), \qquad \alpha_i^1 = \frac{\exp(a_i^1)}{\sum_{i=1}^{|\mathcal{I}_v^1|} \exp(a_i^1)}, \tag{4}$$

where $i = 1, 2, ..., |\mathcal{I}_v^1|$, and σ is sigmoid function. The attention values α_i^1 can be regarded as the similarity between v and v_i^1, as illustrated by Figure 2(a). This aggregation is illustrated in Figure 2(b). Here $\mathbf{v}_I^1$ can be seen as the first-order propagation of v, or the first "ripple" of intra-layer proximity.

Extending beyond the first-order proximity, we repeat the process above (Eqs. 3–4) for each m^{th}-order intra-layer proximity up to the a specified maximum order M, propagating to the subsequent ripples. This generates a set of representations $\{\mathbf{v}_I^m\}_{m=1}^M = \{\mathbf{v}_I^1, \mathbf{v}_I^2, ..., \mathbf{v}_I^M\}$. For simplicity, we let $\mathbf{v}_I^0 = \mathbf{v}$.

Proximities indicate a shared relationship. Nodes within maximum proximity from v would likely have similar representation with v. Thus, given node v and $\{\mathcal{I}_v^m\}_{m=1}^M$, our objective is to minimize the following negative log-likelihood.

$$-\log P(\mathbf{v}_I^1, \mathbf{v}_I^2, ..., \mathbf{v}_I^M|\mathbf{v}) = -\sum_{m=1}^{M} \log P(\mathbf{v}_I^m|\mathbf{v}). \tag{5}$$

4.2 Cross-Layer Proximity Modeling

Beyond a single layer, connections across layers encode network semantics. Node v and its intra-layer proximities $\{\mathcal{I}_v^m\}_{m=1}^M$ are mutually homogeneous, they are expected to reflect the same identity. They would carry information to *jointly* preserve semantics w.r.t. v's cross-layer proximities. Formally, for each node $v_C^n \in \mathcal{C}_v^n$ in n^{th}-order cross-layer proximity, we have the following negative log-likelihood.

$$-\log P(\mathbf{v}_C^n|\mathbf{v}_I^0, ..., \mathbf{v}_I^M) = -\log P(\mathbf{v}_C^n|\sum_{m=0}^{M} \beta^{n,m}\mathbf{v}_I^m). \tag{6}$$

Given $\mathbf{v}$ and its intra-layer proximities $\{\mathbf{v}_I^m\}_{m=1}^M$, we force them to predict the observation of cross-layer node $\mathbf{v}_C^n$ together. As we increase the order of intra-layer proximity M and reach more nodes, more noisy nodes may inadvertently be included. Thus we expect that different $\mathbf{v}_I^m$ may affect this prediction to different degrees. This is the intuition behind $\beta^{n,m}$, which measures the relative importance of $\mathbf{v}_I^m$.

$$b^{n,m} = \sigma(\mathbf{v}_C^{nT}\mathbf{v}_I^m), \qquad \beta^{n,m} = \frac{\exp(b^{n,m})}{\sum_{m=0}^{M}\exp(b^{n,m})}. \tag{7}$$

This is illustrated by Figure 2(c), where we evaluate attention between $\mathbf{v}_C^1$ and $\{\mathbf{v}_I^m\}_{m=0}^M$. $\beta^{n,m}$ is specific to each cross-layer node, since different nodes capture semantics from different aspects. For example, a user and his friends may like superhero movies, but suppose he is the only one who likes it because of the actors. In this case, $\beta^{n,m}$ should be assigned equally between the user and his friends in terms of superhero genre, but biased to only the user in terms of actors.

Similarly as for intra-layer, we extend cross-layer proximity to specified maximum N^{th} order, and obtain the following objective, which is also illustrated by Figure 2(d).

$$-\sum_{n=1}^{N}\sum_{v_C^n \in \mathcal{C}_v^n} \log P(\mathbf{v}_C^n|\sum_{m=0}^{M}\beta^{n,m}\mathbf{v}_I^m). \tag{8}$$

Table 2. Dataset statistics.

Dataset	#nodes	#intra-layer links	#cross-layer links	#layers	#labels
ACM	30,126	77,484	38,772	3	7
Aminer	17,504	72,237	35,229	3	8
TF	3,218	25,811	1,609	2	N.A.
LastFM	19,524	301,015	92,834	2	N.A.

4.3 Learning Strategy

As in [6], the conditional probabilities in Eq. 5 and 8 are defined as the heterogeneous softmax function.

$$P(\mathbf{v}_j|\mathbf{v}_i) = \frac{\exp(\mathbf{v}_j^T\mathbf{v}_i)}{\sum_{l_{v_k}=l_{v_j}} \exp(\mathbf{v}_k^T\mathbf{v}_i)}, \tag{9}$$

where $\mathbf{v}_k$ comes from the same network layer as $\mathbf{v}_j$. Here we use $P(\mathbf{v}_j|\mathbf{v}_i)$ to denote both conditional probabilities for simplicity. Finally, we leverage heterogeneous negative sampling to approximate both objective functions, and obtain Eq. 10, where $\mathcal{K}$ is the number of negative samples. v_k is a negative sample, randomly drawn from a noise distribution $P_l(v_k)$ defined on the node set of each proximity's corresponding layer. v_C^n is one node from v's n^{th}-order cross-layer proximity, sampled at each iteration.

$$\begin{aligned}\mathcal{J} =&\mathcal{J}_{intra} + \mathcal{J}_{cross}\\ =&-\sum_{m=1}^{M}\left(\log\sigma(\mathbf{v}_I^{mT}\mathbf{v}) + \sum_{k=1}^{\mathcal{K}}\mathbb{E}_{v_k\sim P_l(v_k)}\log\sigma(-\mathbf{v}_k^{mT}\mathbf{v})\right)\\ &-\sum_{n=1}^{N}\mathbb{E}_{v_C^n\sim\mathcal{C}_v^n}\left(\log\sigma(\mathbf{v}_C^{nT}\sum_{m=0}^{M}\beta^{n,m}\mathbf{v}_I^m) + \sum_{k=1}^{\mathcal{K}}\mathbb{E}_{v_k\sim P_l(v_k)}\log\sigma(-\mathbf{v}_k^T\sum_{m=0}^{M}\beta^{n,m}\mathbf{v}_I^m)\right).\end{aligned} \tag{10}$$

Complexity. We use $\mathcal{I}_{\max}$ to denote the maximum size of intra-layer proximity, and $|\mathcal{E}_{cross}|$ to denote the number of cross-layer links in the network, thus we have $O(|\mathcal{E}_{cross}|Md(\mathcal{I}_{\max}+\mathcal{K}))$ per iteration for intra-layer proximity modeling, where d represents the dimensionality of node embeddings. The complexity of cross-layer proximity modeling is $O(|\mathcal{E}_{cross}|Nd(M+\mathcal{K}))$ on a training iteration. Putting two components together, we have $O(|\mathcal{E}_{cross}|d(M\mathcal{I}_{\max}+MN+\mathcal{K}M+\mathcal{K}N))$ per iteration.

5 Experiments

Our experimental objective is to validate the node embeddings learned by MULTILAYEREDHNE as compared to baselines.

5.1 Setup

We conduct experiments on four publicly available datasets from different domains. Table 2 summarizes their statistics. ACM [21] and Aminer [14] are two academic datasets

Table 3. Micro-F1 and Macro-F1 scores of node classification on ACM.

Model	Micro-F1				Macro-F1			
	20%	40%	60%	80%	20%	40%	60%	80%
DeepWalk	0.916	0.920	0.919	0.919	0.872	0.877	0.876	0.878
LINE (1st+2nd)	0.924	0.926	0.927	0.927	0.875	0.879	0.880	0.879
Metapath2vec	0.921	0.921	0.922	0.926	0.887	0.887	0.888	0.887
HIN2vec	0.936	0.938	0.938	0.937	0.902	0.908	0.907	0.906
HeGAN	0.938	0.940	0.941	0.941	0.903	0.908	0.910	0.918
MANE	0.842	0.850	0.854	0.859	0.711	0.742	0.756	0.759
GAT	0.867	0.908	0.927	0.921	0.786	0.853	0.883	0.875
HAN	0.828	0.869	0.900	0.905	0.728	0.773	0.824	0.834
MULTILAYEREDHNE	**0.951***	**0.954***	**0.956***	**0.953***	**0.919***	**0.928***	**0.933***	**0.929***

Table 4. Micro-F1 and Macro-F1 scores of node classification on Aminer.

Model	Micro-F1				Macro-F1			
	20%	40%	60%	80%	20%	40%	60%	80%
DeepWalk	0.959	0.962	0.963	0.964	0.922	0.930	0.934	0.931
LINE (1st+2nd)	0.964	0.967	0.968	0.969	0.925	0.930	0.933	0.935
Metapath2vec	0.962	0.963	0.964	0.964	0.870	0.876	0.886	0.893
HIN2vec	0.960	0.962	0.963	0.963	0.922	0.925	0.926	0.927
HeGAN	0.955	0.960	0.963	0.966	0.875	0.892	0.895	0.905
MANE	0.949	0.953	0.956	0.955	0.876	0.893	0.900	0.903
GAT	0.946	0.958	0.965	0.969	0.867	0.919	0.927	0.925
HAN	0.908	0.942	0.956	0.959	0.888	0.911	0.918	0.931
MULTILAYEREDHNE	**0.972***	**0.974***	**0.975***	**0.975***	**0.926***	**0.935***	**0.943***	**0.944***

with three network layers: co-authorship, paper citation, and venue citation layer. Two types of cross-layer links are *author-paper* and *paper-venue* links. Twitter-Foursquare (TF) [29] is a cross-platform social network dataset, containing two social networks: Twitter and Foursquare. Each node only has one cross-layer link, representing his identity across two platforms. LastFM [12] is a recommendation dataset with two layers: *user-user* social network and *artist-artist* context network. TF and LastFM are reserved for link prediction task only, since they do not have labels for nodes.

Baselines. To investigate the efficacy of modeling heterogeneity, we compare to two *homogeneous baselines* that treat all nodes and links as the same type: **DeepWalk** [19] and **LINE** [23]. For LINE, we consider the advanced version with first- and second-order proximities with $d/2$ dimensions each. To study the effects of homogeneity in addition to heterogeneity, we compare to three *heterogeneous baselines*: **Metapath2vec** [6], **HIN2vec** [7], and **HeGAN** [11]. To see if higher-order proximities are useful, we compare to a *multi-layer baseline*: **MANE** [14]. Although GCN-based models are designed with task-specific supervision, and different from our unsupervised model, for completeness, we still compare to **GAT** [24] and **HAN** [25].

Implementation details. Hyperparameters are chosen based on validation set. For MULTILAYEREDHNE, intra-layer proximity order M is 1 on all datasets. The cross-

Table 5. NMI on node clustering.

Model	ACM	Aminer
DeepWalk	0.519	0.787
LINE (1st+2nd)	0.458	0.800
Metapath2vec	0.358	0.570
HIN2vec	0.201	0.589
HeGAN	0.322	0.586
MANE	0.473	0.789
GAT	0.497	0.832
HAN	0.499	0.781
MULTILAYEREDHNE	**0.534***	**0.862***

layer proximity order N is 4 for ACM and Aminer, 1 for TF and LastFM. The number of negative samples $\mathcal{K}$ is 16. For random walk models, as in [25], the number of walks per node is 40, the walk length is 100, the window size is 5. For Metapath2vec, the combination of meta-path schemes APVPA and APPVPPA has the best performance on ACM and Aminer. TTTF and TFFF produce the best results on TF, while for LastFM we combine UAUA, UUUA, and UAAA. For other baselines, we follow the hyperparameter settings in the original paper. For fair comparison, as in [11], the embedding dimension is set to 64 for all methods.

5.2 Node Classification

We expect a good model to embed nodes from the same category closely, while separating different categories. We train a logistic regression based on the embeddings, varying the ratio of the training set from 20% to 80% (of these, 10% is further reserved for validation). We report *Micro-F1* and *Macro-F1* scores on the testing sets in Table 3 and 4 for the two respective datasets that are applicable. In this paper we use "*" to denote that the performance of our model is significantly different from the best baseline model's based on the paired t-test at the significance level of 0.01.

MULTILAYEREDHNE consistently outperforms the baselines across all training splits. As the training ratio increases, all models tend to perform better, as expected. It is worth noting that HIN-based models, including MULTILAYEREDHNE, generally classify nodes more accurately than those models working solely on homogeneous networks, highlighting the effectiveness of modeling network semantics. Among HIN embedding models, Metapath2vec based on only specific cross-layer links performs the worst, emphasizing the necessity of modeling both intra-layer and cross-layer links.

5.3 Node Clustering

Intuitively, good node embeddings would put "similar" nodes together. We apply K-means algorithm [1] to perform clustering on the node embeddings. Since for ACM and Aminer, nodes are labeled, we can assess whether nodes in a cluster tend to share the same labels. We evaluate the clustering quality using *Normalized Mutual Information (NMI)* w.r.t. the true labels (not used in training).

Table 6. Intra-layer link prediction results.

Model	ACM			Aminer			TF			LastFM		
	AUC	AP	F1	AUC	AP	F1	AUC	AP	F1	AUC	AP	F1
DeepWalk	0.900	0.908	0.688	0.880	0.860	0.685	0.678	0.651	0.683	0.587	0.600	0.693
LINE (1st+2nd)	0.962	0.972	0.668	0.811	0.731	0.739	0.701	0.725	0.661	0.665	0.729	0.666
Metapath2vec	0.786	0.830	0.672	0.851	0.840	0.667	0.620	0.621	0.666	0.789	0.778	0.502
HIN2vec	0.871	0.872	0.671	0.579	0.544	0.667	0.770	0.749	0.667	**0.884**	0.876	0.682
HeGAN	0.509	0.517	0.667	0.641	0.626	0.667	0.512	0.510	0.667	0.507	0.504	0.665
MANE	0.973	0.978	0.675	0.871	0.858	0.688	0.750	0.693	0.679	0.864	0.873	0.667
GAT	0.674	0.675	0.589	0.854	0.812	0.583	-	-	-	-	-	-
HAN	0.592	0.607	0.585	0.647	0.638	0.608	-	-	-	-	-	-
MULTILAYEREDHNE	**0.979***	**0.983***	**0.799***	**0.897***	**0.890***	**0.795***	**0.798***	**0.823***	**0.722***	0.880	**0.892***	**0.768***

Table 5 presents the results. Overall, MULTILAYEREDHNE outperforms baseline models significantly. In comparison to DeepWalk and LINE that model all nodes and links homogeneously, we observe that our distinctive treatment of intra-layer and cross-layer proximities is helpful. MULTILAYEREDHNE also clusters nodes more effectively than MANE, demonstrating that higher-order proximities could help better explore network structure. Overall, MULTILAYEREDHNE achieves performance gains over the closest baseline by 2.8% and 7.7%, respectively.

5.4 Link Prediction

Here we predict intra- and cross-layer links, respectively. For *intra-layer link prediction*, we predict the *author-author* link on ACM and Aminer [22], *user-user* link on Twitter of TF [28], and *artist-artist* link on LastFM. As in leave-one-out evaluation [10], for nodes with more than one intra-layer links, we hide one as the ground truth positives, and randomly sample the same number of disconnected node pairs as negative instances. The remaining network is our training set. Since this is a binary classification for the held-out links, we adopt inner product [13] to make predictions, and report *AUC*, *Average Precision (AP)*, and *F1 score* in Table 6. For *cross-layer link prediction*, we predict *author-paper* links on ACM and Aminer [11], *user-user* links on TF, and *user-artist* links on LastFM. We hide cross-layer links similarly with intra-layer. Table 7 presents the results. Since GAT and HAN are designed with label supervision to learn embeddings, they do not have link prediction results on TF and LastFM.

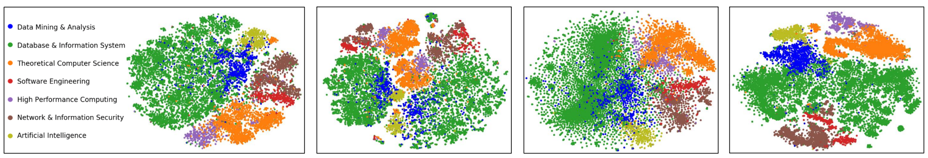
MULTILAYEREDHNE generally outperforms baselines significantly on all evaluation metrics, except for the sole case of the LastFM dataset. For intra-layer link prediction, compared with DeepWalk and LINE, this task verifies the effectiveness of MULTILAYEREDHNE on predicting links between homogeneous nodes. We attribute this to the network heterogeneity captured by our model. For cross-layer link prediction, MULTILAYEREDHNE benefits from the structure-preserving embeddings learned via intra-layer proximity as compared with heterogeneous baselines.

5.5 Network Visualization

Visualization provides an intuitive sense of how nodes are embedded. We visualize node embeddings using t-SNE [17], and color nodes using their corresponding labels. Figure

Table 7. Cross-layer link prediction results.

Model	ACM			Aminer			TF			LastFM		
	AUC	AP	F1	AUC	AP	F1	AUC	AP	F1	AUC	AP	F1
DeepWalk	0.891	0.887	0.690	0.923	0.919	0.687	0.707	0.720	0.669	0.606	0.617	0.647
LINE (1st+2nd)	0.912	0.920	0.685	0.894	0.842	0.781	0.725	0.750	0.663	0.714	0.746	0.646
Metapath2vec	0.780	0.798	0.704	0.819	0.773	0.691	0.916	0.911	0.752	**0.923**	0.906	0.476
HIN2vec	0.929	0.939	0.223	0.921	0.924	0.173	0.524	0.561	0.173	0.265	0.397	0.276
HeGAN	0.530	0.534	0.268	0.683	0.683	0.545	0.705	0.685	0.168	0.535	0.527	0.027
MANE	0.923	0.913	0.670	0.906	0.857	0.673	0.724	0.727	0.646	0.736	0.765	0.645
GAT	0.653	0.629	0.549	0.880	0.832	0.574	-	-	-	-	-	-
HAN	0.606	0.590	0.554	0.690	0.669	0.667	-	-	-	-	-	-
MULTILAYEREDHNE	**0.950***	**0.948***	**0.812***	**0.936***	**0.925***	**0.827***	**0.954***	**0.952***	**0.799***	0.919	**0.911***	**0.837***



(a) LINE (b) HIN2vec (c) MANE (d) Multi-LayeredHNE

Fig. 3. t-SNE visualization on ACM dataset.

3 presents four models on ACM dataset. By encoding network structural proximity and semantics, MULTILAYEREDHNE provides denser clusters with clearer category boundaries than others.

5.6 Model Analysis

Here we conduct several analysis on MULTILAYEREDHNE to better understand the underlying mechanism of it.

Homogeneity and heterogeneity. To investigate if MULTILAYEREDHNE effectively leverages network homogeneity and heterogeneity, we conduct ablation analysis here. MULTILAYEREDHNE-homo removes the intra-layer proximity modeling, and only maintains cross-layer proximity. Conversely, MULTILAYEREDHNE-hetero assumes all nodes and links are of the same type, and discards network layer concept to investigate network semantics.

Results in Figure 4 reveal three insights. First, MULTILAYEREDHNE-homo performs worse than MULTILAYEREDHNE, showcasing the advantage of modeling structural information. Second, MULTILAYEREDHNE can indeed encode semantics, since MULTILAYEREDHNE-hetero, which ignores heterogeneity, leads to worse performance compared to MULTILAYEREDHNE. Third, by comparing MULTILAYEREDHNE-homo and MULTILAYEREDHNE-hetero, we conclude that network structural proximity is more informative than semantics, as MULTILAYEREDHNE-homo drops more than MULTILAYEREDHNE-hetero from MULTILAYEREDHNE.

Parameter sensitivity. We vary maximum order of intra- and cross-layer proximity to investigate performance sensitivity. We report the results of clustering (NMI) on ACM dataset in Figure 5. We first test intra-layer proximity order M. Compared with

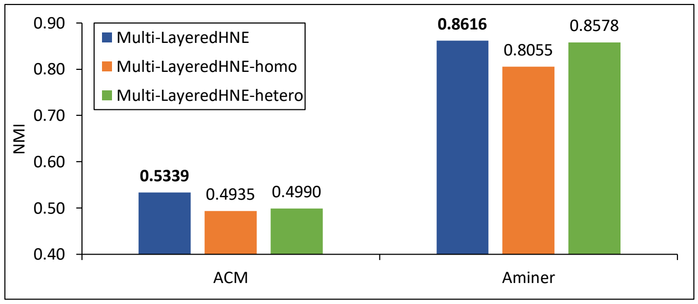
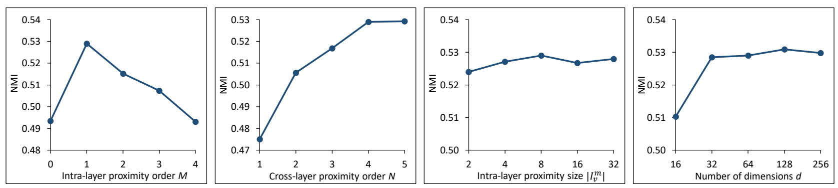


Fig. 4. Impact of homogeneity and heterogeneity



(a) Intra-layer proximity order M (b) Cross-layer proximity order N (c) Intra-layer proximity size $|\mathcal{I}_v^m|$ (d) Number of dimensions d

Fig. 5. MULTILAYEREDHNE parameter sensitivity

$M = 0$ where no intra-layer proximity is modeled, MULTILAYEREDHNE achieves notable performance gain at $M = 1$. However, our model deteriorates its clustering when M is greater than 1, since more noisy neighbors are involved.

We then vary the order of cross-layer proximity N. Too small N apparently could not effectively explore network semantics. The clustering quality is boosted as the order increases, emphasizing the efficacy of modeling cross-layer proximity to capture network heterogeneity and semantics.

Intra-layer proximity size $|\mathcal{I}_v^m|$. We limit the size of each intra-layer proximity to further investigate the robustness of MULTILAYEREDHNE on sparse scenarios. Figure 5(c) shows the results. With the increase of the size of intra-layer proximity, the performance of MULTILAYEREDHNE is improved at first, because a larger set of neighbors can encode more structural information on the network. But the clustering results decrease slightly and then stay flat when the size is too large. Overall, the performance is stable w.r.t. different sizes.

Number of dimensions d. To check the impact of different embedding dimensions d on model performance, we vary the value of d and report the results (Figure 5 (d)). With the growth of d from 16 to 32, NMI rises at first, and fluctuates slightly when $d > 32$. Since small dimensions cannot fully encode the rich information embodied by the networks, increasing d could potentially capture more features, thereby boosting experiment results. When d is overly large, e.g., $d = 256$, over-fitting problem may happen, and the performance decreases. Overall, our model still performs relatively stable with different dimensions.

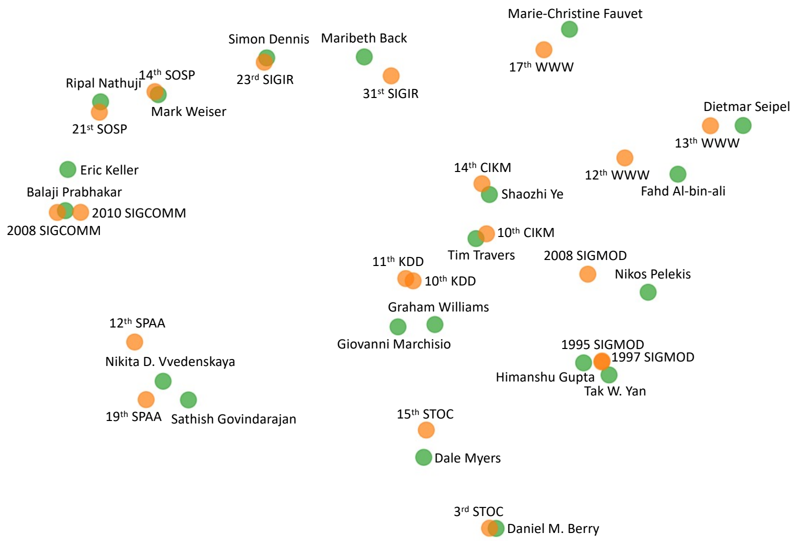


Fig. 6. Case study on ACM dataset (best seen in color). t-SNE visualization of various conferences (orange) from different years and most active authors (green) in those years.

5.7 Case Study

As an illustration of how MULTILAYEREDHNE encodes homogeneity and heterogeneity, we conduct a case study on ACM dataset. We randomly select two or three years of each conference, and draw the most active authors in those years. Figure 6 shows the t-SNE visualization. Interestingly, the distance between 12^{th} and 13^{th} WWW conferences (top right corner) is shorter than their distances to 17^{th} WWW conference. SIGMOD (bottom right corner) also has similar observations, where 1995 and 1997 are almost overlapping, but far from 2008. That closer years are more related is quite intuitive. Researchers tend to cite more recent papers, authors also collaborate with recently active researchers. Due to intra-layer modeling, our model is able to capture these homogeneous connections.

Figure 6 also depicts close relationships between conferences and their highly-profiled authors. Moreover, different areas tend to display some separation. Data points from Data Mining, Databases, and Artificial Intelligence dominate the right-hand side, while left-hand side has more from Information Security, Operating Systems, and Computer Architecture. This layout among conferences from diverse domains, and among authors actively involved in conferences of different years, demonstrates the embedding ability of MULTILAYEREDHNE to preserve network heterogeneity.

6 Conclusion

We formalize the multi-layered heterogeneous network embedding problem, and propose a novel framework MULTILAYEREDHNE to model intra- and cross-layer proximity. Due to the dual characters of multi-layered networks on homogeneity and heterogeneity, our model learns node embeddings that preserve network topology and semantics jointly. Extensive experiments verify the effectiveness of our model on four public datasets. With ablation analysis, we show that our model could effectively benefit from both modeling components.

Acknowledgments

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its NRF Fellowship Programme (Award No. NRF-NRFF2016-07).

References

1. Bishop, C. M.: Pattern recognition and Machine learning, Springer (2006).
2. Cao, S., Lu, W., Xu, Q.: Grarep: Learning graph representations with global structural information. In: Proceedings of the 24th ACM international on conference on information and knowledge management. pp. 891-900 (2015).
3. Cen, Y., Zou, X., Zhang, J., Yang, H., Zhou, J., Tang, J.: Representation learning for attributed multiplex heterogeneous network. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1358-1368 (2019).
4. Chen, C., Tong, H., Xie, L., Ying, L., He, Q.: 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. pp. 765-774 (2016).
5. Chen, C., Tong, H., Xie, L., Ying, L., He, Q.: Cross-dependency inference in multi-layered networks: A collaborative filtering perspective. ACM Transactions on Knowledge Discovery from Data (TKDD), 11(4), 1-26 (2017).
6. Dong, Y., Chawla, N. V., Swami, A.: metapath2vec: Scalable representation learning for heterogeneous networks. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 135-144 (2017).
7. Fu, T. Y., Lee, W. C., Lei, Z.: Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. pp. 1797-1806 (2017).
8. Garcia Duran, A., Niepert, M.: Learning graph representations with embedding propagation. In: Advances in neural information processing systems. pp. 5119-5130 (2017).
9. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 855-864 (2016).
10. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T. S.: Neural collaborative filtering. In: Proceedings of the 26th international conference on world wide web. pp. 173-182 (2017).
11. Hu, B., Fang, Y., Shi, C.: Adversarial learning on heterogeneous information networks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 120-129 (2019).
12. Hu, B., Shi, C., Zhao, W. X., Yu, P. S.: Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1531-1540 (2018).
13. Kipf, T. N., Welling, M.: Variational graph auto-encoders. In: NIPS workshop on Bayesian Deep Learning (2016).

14. Li, J., Chen, C., Tong, H., Liu, H.: Multi-layered network embedding. In Proceedings of the 2018 SIAM International Conference on Data Mining. pp. 684-692 (2018).
15. Liu, L., Cheung, W. K., Li, X., Liao, L.: Aligning Users across Social Networks Using Network Embedding. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence. pp. 1774-1780 (2016).
16. Liu, W., Chen, P. Y., Yeung, S., Suzumura, T., Chen, L.: Principled multilayer network embedding. In: 2017 IEEE International Conference on Data Mining Workshops (ICDMW). pp. 134-141 (2017).
17. Van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. In: Journal of machine learning research, 9, 2579-2605 (2008).
18. Ni, J., Tong, H., Fan, W., Zhang, X.: Inside the atoms: ranking on a network of networks. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 1356-1365 (2014).
19. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 701-710 (2014).
20. Shi, C., Hu, B., Zhao, W. X., Philip, S. Y.: Heterogeneous information network embedding for recommendation. IEEE Transactions on Knowledge and Data Engineering, 31(2), 357-370 (2018).
21. Shi, C., Kong, X., Huang, Y., Philip, S. Y., Wu, B.: Hetesim: A general framework for relevance measure in heterogeneous networks. IEEE Transactions on Knowledge and Data Engineering, 26(10), 2479-2492 (2014).
22. Sun, Y., Barber, R., Gupta, M., Aggarwal, C. C., Han, J.: Co-author relationship prediction in heterogeneous bibliographic networks. In: 2011 International Conference on Advances in Social Networks Analysis and Mining. pp. 121-128 (2011).
23. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: Large-scale information network embedding. In: Proceedings of the 24th international conference on world wide web. pp. 1067-1077 (2016).
24. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. In: Proceedings of International Conference on Learning Representations. (2018).
25. Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., Yu, P. S.: Heterogeneous graph attention network. In: The World Wide Web Conference. pp. 2022-2032 (2019).
26. Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., Leskovec, J.: Graph convolutional neural networks for web-scale recommender systems. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 974-983 (2018).
27. Zhang, C., Lauw, H. W.: Topic modeling on document networks with adjacent-encoder. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 34, No. 04, pp. 6737-6745 (2020).
28. Zhang, J., Kong, X., Philip, S. Y.: Predicting social links for new users across aligned heterogeneous social networks. In: 2013 IEEE 13th International Conference on Data Mining. pp. 1289-1294 (2013).
29. Zhang, J., Philip, S. Y.: Integrated anchor and social link predictions across social networks. In: Twenty-fourth international joint conference on artificial intelligence. pp. 2125-2131 (2015).