Meta-Graph Based HIN Spectral Embedding: Methods, Analyses, and Insights

Carl Yang, Yichen Feng, Pan Li, Yu Shi, Jiawei Han

University of Illinois, Urbana Champaign, 201 N Goodwin Ave, Urbana, Illinois 61801, USA

{jiyang3, feng36, panli2, yushi2, hanj}@illinois.edu

***Abstract*—Heterogeneous information network (HIN) has drawn significant research attention recently, due to its power of modeling multi-typed multi-relational data and facilitating various downstream applications. In this decade, many algorithms have been developed for HIN modeling, including traditional similarity measures and recent embedding techniques. Most algorithms on HIN leverage *meta-graphs* or *meta-paths* (special cases of meta-graphs) to capture various semantics. Given any arbitrary set of meta-graphs, existing algorithms either consider them as equally important or study their different importance through supervised learning. Their performance largely relies on prior knowledge and labeled data. While unsupervised embedding has shown to be a fundamental solution for various homogeneous network mining tasks, for HIN, it is a much harder problem due to such a presence of various meta-graphs.**

In this work, we propose to study the utility of different meta-graphs, as well as how to simultaneously leverage multiple meta-graphs for HIN embedding in an unsupervised manner. Motivated by prolific research on homogeneous networks, especially spectral graph theory, we firstly conduct a systematic empirical study on the spectrum and embedding quality of different meta-graphs on multiple HINs, which leads to an efficient method of meta-graph assessment. It also helps us to gain valuable insight into the higher-order organization of HINs and indicates a practical way of selecting useful embedding dimensions. Further, we explore the challenges of combining multiple meta-graphs to capture the multi-dimensional semantics in HIN through reasoning from mathematical geometry and arrive at an embedding compression method of autoencoder with $\ell_{2,1}$-loss, which finds the most informative meta-graphs and embeddings in an end-to-end unsupervised manner. Finally, empirical analysis suggests a unified workflow to close the gap between our meta-graph assessment and combination methods. To the best of our knowledge, this is the first research effort to provide rich theoretical and empirical analyses on the utility of meta-graphs and their combinations, especially regarding HIN embedding. Extensive experimental comparisons with various state-of-the-art neural network based embedding methods on multiple real-world HINs demonstrate the effectiveness and efficiency of our framework in finding useful meta-graphs and generating high-quality HIN embeddings.

I. Introduction

Networks are widely used to model relational data such as web pages with hyperlinks and people with social connections. Recently, increasing research attention has been paid to the heterogeneous information network (HIN), due to its power of accommodating rich semantics in terms of multi-typed nodes (vertices) and links (edges), which enables the integration of real-world data from various sources and facilitates wide downstream applications [1], [2], [3], [4], [5], [6].

For capturing the complex semantics in HIN, the concepts of meta-paths and meta-graphs have been developed, which are subsets of HIN schemas [7]. Since each particular meta-graph indicates an essential semantic unit that can be potentially useful for various tasks, they have become the de facto tool of HIN modeling, leveraged by various existing works [1], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18]. In this work, to be general, we refer meta-paths to special cases of meta-graphs, and study them under the same framework.

Since there can be various meta-graphs on a given HIN, the key problems for leveraging them are: (1) what meta-graphs are useful (*assessment*), and (2) how to jointly leverage multiple meta-graphs (*combination*). To our surprise, however, no existing work explicitly studies the first problem, while no satisfactory solution exists to the second, especially regarding general-purpose unsupervised network embedding.

To get around the first problem, existing HIN models mostly assume that useful meta-graphs can be manually composed based on domain knowledge [1], [8], [11], [12], [13], [14], [15], [16], [18], [3], [4], while such knowledge can be expensive and not always available for arbitrary unfamiliar HINs. To break this limitation, a few algorithms attempt to generate all legitimate meta-graphs up to a certain size through heuristic mechanisms [9], [10], [17], [19], but they again fail to further select the more useful ones before sending all of them into a subsequent combination model.

As for the second problem, most algorithms rely on supervised learning towards specific tasks to tune the weights on different meta-graphs [8], [3], [20], [21], [5], [10], [11]. Their performance heavily relies on labeled data. On the other hand, while general-purpose unsupervised network embedding has received tremendous attention recently due to the huge success of neural network based models like [22], [23], [24], [25], there exists no method to properly combine multiple meta-graphs for unsupervised HIN embedding, except for simply adding up their instance counts [13], [15] or looking for the proper weights through exhaustive grid search [14].

In this work, we extend the rich theoretical and empirical studies on homogeneous networks to the HIN setting. Specifically, we provide a series of *methods*, *analyses* and *insights* towards *meta-graph based HIN spectral embedding*, which serves as solutions to both of the aforementioned assessment and combination problems. Our main contributions are summarized in the following.

Contribution 1: Meta-Graph based HIN Spectral Embedding. Motivated by prolific studies on homogeneous networks, we review and introduce several key conclusions from spectral graph theory, and propose to leverage meta-graphs to compute the *projected networks* of HIN. It facilitates HIN spectral embedding, which serves as a great tool for various subsequent theoretical and empirical analyses (Section II and III).

Contribution 2: Meta-Graph Assessment. Based on well-established *spectral graph theory*, we compute the graph spectra of projected networks, which in principle capture the key network properties. Through a systematic empirical study on three real-world HINs, we discover two essential properties that have significant impacts on the general quality of HIN embedding. Theoretical interpretations of these properties provide valuable insights into the high-order organizations of HINs and their implications towards embedding quality, which further allows efficient assessment of meta-graph utility (Section IV).

Contribution 3: Meta-Graph Combination. Since different meta-graphs essentially capture different semantic information of a HIN, it is necessary to properly combine multiple useful meta-graphs. To simultaneously solve the intrinsic *dimension reduction* and *meta-graph selection* problems in an *unsupervised* manner, we devise an autoencoder with $\ell_{2,1}$-loss. It is able to end-to-end select the important meta-graphs from a set of candidates by capturing the embedding dimensions with large variance grouped by the corresponding meta-graphs. We also provide rich theoretical and empirical analyses towards its effectiveness (Section V).

Contribution 4: Comprehensive Evaluations. Through extensive experiments in comparison with various state-of-the-art HIN embedding methods on three large real-world datasets towards two traditional downstream tasks, we demonstrate the supreme performance of our proposed method for general-purpose unsupervised HIN embedding (Section VI).

II. Related Work and Preliminaries

A. Heterogeneous Information Network (HIN)

Networks provide a natural and generic way of modeling data with interactions. Among them, HIN has drawn increasing research attention in the recent decade, due to its capability of retaining rich type information [7]. A HIN can be defined as $\mathcal{N}_H = (\mathcal{V}_H, \mathcal{E}_H, \phi, \psi)$, where $\mathcal{V}_H$ is the vertex set and $\mathcal{E}_H$ is the edge set. In addition to traditional homogeneous networks, $\phi : \mathcal{V}_H \rightarrow \mathcal{T} = (t_1, t_2, ...)$ and $\psi : \mathcal{E}_H \rightarrow \mathcal{R} = (r_1, r_2, ...)$ are two mapping functions that assign vertices and edges with the type information. Such extensions, while making the networks much more complicated, have shown to be very powerful in modeling real-world multi-modal multi-aspect data [1], [8], [10], [19], [12] and beneficial to various downstream applications [2], [3], [4], [5], [6]. To model HIN with typed vertices and edges, [1] proposes to leverage the tool of meta-paths, which is later on generalized to meta-graphs [9]. They are adopted by almost all HIN models due to the capture of fine-grained type-and-structure-aware semantics.

Recently, network embedding algorithms based on the advances in neural networks (NN) have been extremely popular [22], [23], [24], [25]. They aim to compute distributed representations of vertices that capture both neighborhood and structural similarity [26], [27]. Following this trend, many HIN embedding methods have also been developed [13], [14], [15], [16], [17], [18], [11]. Most of them, while guided by meta-graphs, mainly leverage well-developed NN models (*e.g.*, Skip-gram [28]). While they are shown to work well in certain cases, their performances are not stable and hard to track.

In this work, we get inspired by prolific studies on homogeneous networks. For the first time, we provide a series of theoretically sound and empirically effective methods towards HIN embedding, together with extensive analyses and valuable insights, based on the well-established spectral graph theory.

B. Spectral Graph Theory

Spectral embedding, also termed as the Laplacian eigenmap, has been widely used for homogeneous network embedding [29], [30]. Mathematically, it can be computed as follows: Given a weighted homogeneous network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the vertex set and $\mathcal{E}$ is the edge set. We also define an adjacency matrix $\mathcal{A}$. For any $e_{ij} \in \mathcal{E}$, $a_{ij} > 0$ denotes its edge weight, and for any $e_{ij} \notin \mathcal{E}$, $a_{ij} = 0$. Let $\mathcal{D}$ be the diagonal degree matrix where $d_{ii} = \sum_{e_{ij} \in \mathcal{E}} a_{ij}$. Then, the *normalized Laplacian matrix* follows

$$\mathcal{L} = I - \mathcal{D}^{-\frac{1}{2}} \mathcal{A} \mathcal{D}^{-\frac{1}{2}}, \tag{1}$$

where I is the identity matrix. Suppose $\mathcal{L}$ has eigenvalues ordered as $\lambda_1 = 0 \leq \lambda_2 \leq \cdots \leq \lambda_{|\mathcal{V}|}$, which is also termed as the spectrum of graph $\mathcal{G}$. For each eigenvalue λ_i, we denote the corresponding eigenvector as $u_i = (u_{ij})_{v_j \in \mathcal{V}}$. Then, the k-dimensional embedding of vertice v_j can be expressed as $\mathbf{h}_j = [u_{1j}, u_{2j}, ..., u_{kj}]^T$. Spectral graph theory connects the spectrum of $\mathcal{L}$ to the properties of $\mathcal{G}$ and further gives plentiful results that are useful in both theory and practice.

For later usage, in the following, we review some key theories and definitions related to our work while refer interested readers to [31], [32], [33] for more results.

Theorem 2.1 ([31]): The number of zero eigenvalues of $\mathcal{L}$ is equal to the number of connected components of $\mathcal{G}$.

Suppose the number of zero eigenvalues is p. One step further, the first p dimensions of h_i are orthogonal to those of h_j if v_i and v_j lie in different connected components. So spectral embedding naturally encodes the connectivity between vertices in the embedding space.

We next introduce some results on the concept of *nodal domain* [34], [35] that may be used to understand the embedding space. We start with the definition of some key concepts.

Definition 2.2: For a subset of vertices $\mathcal{S} \subseteq \mathcal{V}$, we denote the *induced subgraph* of $\mathcal{G}$ by $\mathcal{S}$ as $\mathcal{G}(\mathcal{S}) = (\mathcal{S}, \mathcal{E}(\mathcal{S}))$, such that for any pair of vertices $v_i, v_j \in \mathcal{S}$, the edge $e_{ij} \in \mathcal{E}(\mathcal{S})$ if and only if $e_{ij} \in \mathcal{E}$.

Definition 2.3: Given a function $f(\cdot) : \mathcal{V} \rightarrow \mathbb{R}^{|\mathcal{V}|}$. A subset $\mathcal{S} \subset \mathcal{V}$ is a *strong nodal domain* of $\mathcal{G}$ induced by f if the induced subgraph $\mathcal{G}(\mathcal{S})$ is a maximal connected component of either $\{v_i \in \mathcal{V} : f(v_i) > 0\}$ or $\{v_i \in \mathcal{V} : f(v_i) < 0\}$.

Next we introduce two powerful results from spectral graph theory that characterize the nodal domains of an eigenvector.

The first one gives the bound on the number of nodal domains of an eigenvector.

Theorem 2.4 ([34]): If the network is connected, the number of strong nodal domains of v_i is not greater than the number of eigenvalues that are not greater than λ_i.

It implies that for small λ_i, the number of nodal domains induced by v_i is also small.

The next one is the high-order Cheeger inequality.

Theorem 2.5 ([35]): Suppose $\{\mathcal{S}_l\}_{1\leq l\leq m}$ are strong nodal domains induced by v_i. Then

$$\max_{1\leq l\leq m} \frac{\sum_{j\in\mathcal{S}_l, j'\notin\mathcal{S}_l} a_{jj'}}{\sum_{j\in\mathcal{S}_l, j'\in V} a_{jj'}} \leq \sqrt{2\lambda_i}. \tag{2}$$

It implies that each nodal domain associated with small eigenvalues corresponds to the community structures of $\mathcal{G}$, whose inside is densely connected with few out-edges.

These two theorems indicate how spectral embedding represents the topology of $\mathcal{G}$ in the Euclidean space. As we will see in the remainder of this work, these theories lay the solid foundation of our methods, guide our fruitful data analyses and lead to quite a few valuable insights.

III. Meta-Graph Based HIN Spectral Embedding

In order to generalize the key theoretical studies and empirical analyses on homogeneous networks to HIN, we introduce our basic HIN spectral embedding method.

Traditional graph theory studies the adjacency matrices of homogeneous networks. As we discussed in Section II.A, the additional type information endows HIN with advantageous modeling capability but also makes it much more complicated and inappropriate to be represented by a single adjacency matrix. To this end, we leverage the powerful tool of meta-graphs that encode various fine-grained HIN semantics by designing a HIN projection process. While spectral embedding has been widely studied [36], [37], [38], [39], no previous work has connected it with the utility of meta-graphs on HIN.

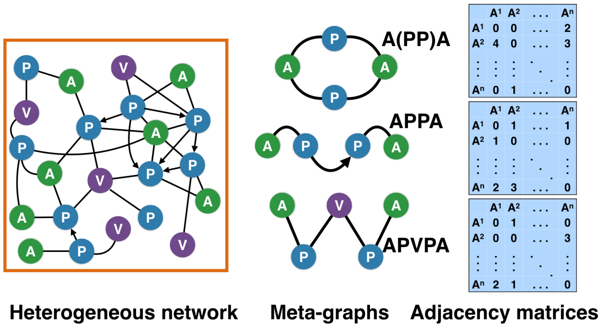
Figure 1 shows an example of the academic publication network, where we use three different meta-graphs to project the HIN and get three different adjacency matrices for the corresponding homogeneous networks of authors. The edge weights are generated by the number of matched meta-graph instances between each pair of vertices. We call the homogeneous networks obtained in this way the *projected networks*. Note that, during this procedure, the type information is captured by meta-graphs which may further be encoded into the edge-weights of the projected networks. Therefore, one may expect to obtain a good vertex embedding as long as the meta-graphs are chosen properly.

In Figure 1, we also give a few examples of meta-graphs and their notations. For simplicity, we only consider edges connecting to the pairs of vertices (*e.g.*, authors on the two sides here), and do not differentiate directed and undirected edges, while our methods trivially generalize to those cases.

Based on the homogeneous projected networks, we can compute the standard spectral embedding as described in Section II.B. Note that spectral embedding is mathematically equivalent to the PCA of a degree normalized adjacency matrix $\mathcal{A} = I + D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ [40], so approximating the original graph in the optimal sense such that $H = (h_1^T, h_2^T, ..., h_{|\mathcal{V}|}^T)^T$ gives the solution to the approximation problem

$$\min_{H\in\mathcal{R}^{|\mathcal{V}|\times k}, W\in\mathbb{R}^{k\times k}} \|\mathcal{A} - HWH^T\|_F^2, \tag{3}$$

where W is a diagonal matrix and $\|\cdot\|_F$ is Frobenius norm.



	A¹	A²	...	Aⁿ
A¹	0	0	...	2
A²	4	0	...	3
⋮	⋮	⋮	⋱	⋮
Aⁿ	0	1	...	0

	A¹	A²	...	Aⁿ
A¹	0	1	...	1
A²	1	0	...	0
⋮	⋮	⋮	⋱	⋮
Aⁿ	2	3	...	0

	A¹	A²	...	Aⁿ
A¹	0	1	...	0
A²	0	0	...	3
⋮	⋮	⋮	⋱	⋮
Aⁿ	2	1	...	0

Fig. 1. The process of obtaining the projected homogeneous networks and corresponding adjacency matrices from HINs with different meta-graphs.

In contrast to those complex NN-based approaches, spectral embedding holds superiority in several aspects. First, it is computationally cheaper. For a k-dimensional embedding, one may require $O(k|\mathcal{E}|)$ number of scalar sum and product operations on the projected networks based on power iterations [41], while one single epoch of training the NN-based models costs such amount of computation on the original HIN with much more vertices and edges. More importantly, unlike the NN-based approaches that implicitly factorize the adjacency matrices [42], spectral embedding directly provides linear approximation for the PCA problem in Eq. 3. Its performance is more tractable through the well-established spectral graph theory, which makes it a good tool to understand the underlying structures and principal properties of networks, as well as the function of different meta-graphs.

In spectral embedding, an eigenvector u_i can be viewed as a one-dimensional embedding of vertices. Conceptually, based on Theorem 2.4 and 2.5, for a small eigenvalue λ_i, the vertices from one nodal domain of v_i typically lie within a densely connected community of $\mathcal{G}$. Correspondingly, due to the definition of nodal domains, all the vertices in this nodal domain will be embedded into the same quadrant in the embedding space. This relation gives a direct mapping from the densely connected communities of $\mathcal{G}$ to a quadrant of the embedding space. As each of the eigenvectors can be viewed as a one-dimensional embedding as described above, the spectral embedding based on the concatenation of eigenvectors with small λ_i actually gives a fine embedding of the whole graph $\mathcal{G}$ in the sense that vertices topologically close on $\mathcal{G}$ are essentially more likely to be embedded into the same quadrant. Moreover, it also makes the change and tuning of embedding sizes extremely efficient. To increase the embedding size by k', only $O(k'|\mathcal{E}|)$ time is required, while decreasing the embedding size takes no time. On the other hand, the NN-based models need to be totally retrained whenever the embedding sizes are changed.

Our method is also closely related to the spectral methods leveraged for the investigation of higher-order organizations of homogeneous networks in [43], [36]. However, in HIN, the high-order connectivity patterns are carried by meta-graphs that encode various semantic information. Moreover, the projection process is quite different, since meta-graphs do not always lead to cliques as the network motifs in [43]. A very recent work on hypergraphs shows that the spectral clustering based on inhomogeneous projections of *hyperedges* keeps good approximation of the cheeger isoperimetric constant of hypergraphs [44]. Since hyperedges can be viewed as a mathematical abstract of our meta-graphs, this implies that our method essentially puts vertices lying on many common meta-graphs close to each other in the embedding space.

IV. Meta-Graph Assessment

While meta-graphs are widely used for HIN modeling, different meta-graphs encode diverse semantics that essentially leads to rather different utilities, which might be understood by looking into the structures of the underlying projected networks [45], [19]. To this end, we present our spectral embedding method, which naturally serves as a great tool to facilitate such assessment in an efficient way.

We notice that in spectral graph theory, eigenvalues are closely related to many essential graph properties [31]. However, it is unknown what properties are indeed impactful, *i.e.*, important for meta-graph utilities, especially regarding HIN embedding. To understand this, we conduct a systematic empirical study on various real-world HINs towards multiple traditional network mining tasks. Specifically, for each projected network, we visualize and study the correlations between its spectrum and embedding quality. As we will soon see, the results are indeed highly interpretable and insightful.

The datasets we use include HINs in different domains, *i.e.*, DBLP from an academic publication collection[1], IMDB from a movie rating platform[2], and Yelp from a business review website[3]. Details of these datasets are as follows.

1) **DBLP**: We use the Arnetminer dataset V8[4] collected by [46]. It contains four types of vertices, *i.e.*, author (A), paper (P), venue (V), and year (Y).
2) **IMDB**: We use the MovieLens-100K dataset[5] made public by [47]. There are four types of vertices, *i.e.*, user (U), movie (M), actor (A), and director (D).
3) **YELP**: We use the public dataset from the Yelp Challenge Round 11[6]. Following an existing work that models the YELP data with heterogeneous networks [5], we extract five types of vertices, *i.e.*, business (B), user (U), location (L), category (C), and star (S).

A. FPP (First-Positive-Point) - Network Connectivity

Empirical Observations. Figure 2 shows the spectrum and embedding quality of different meta-graphs on the three datasets. The spectrum is computed via SVD[7] on the normalized Laplacian defined in Eq. 1 and sorted in ascending orders. The embedding quality is evaluated towards node classification through an off-the-shelf SVM[8] model with standard five-fold cross validation on labeled nodes. We compute the commonly used $F1$ score for evaluating the classification performance. Other tasks like standard link prediction and clustering show similar trends and are omitted due to space limit.

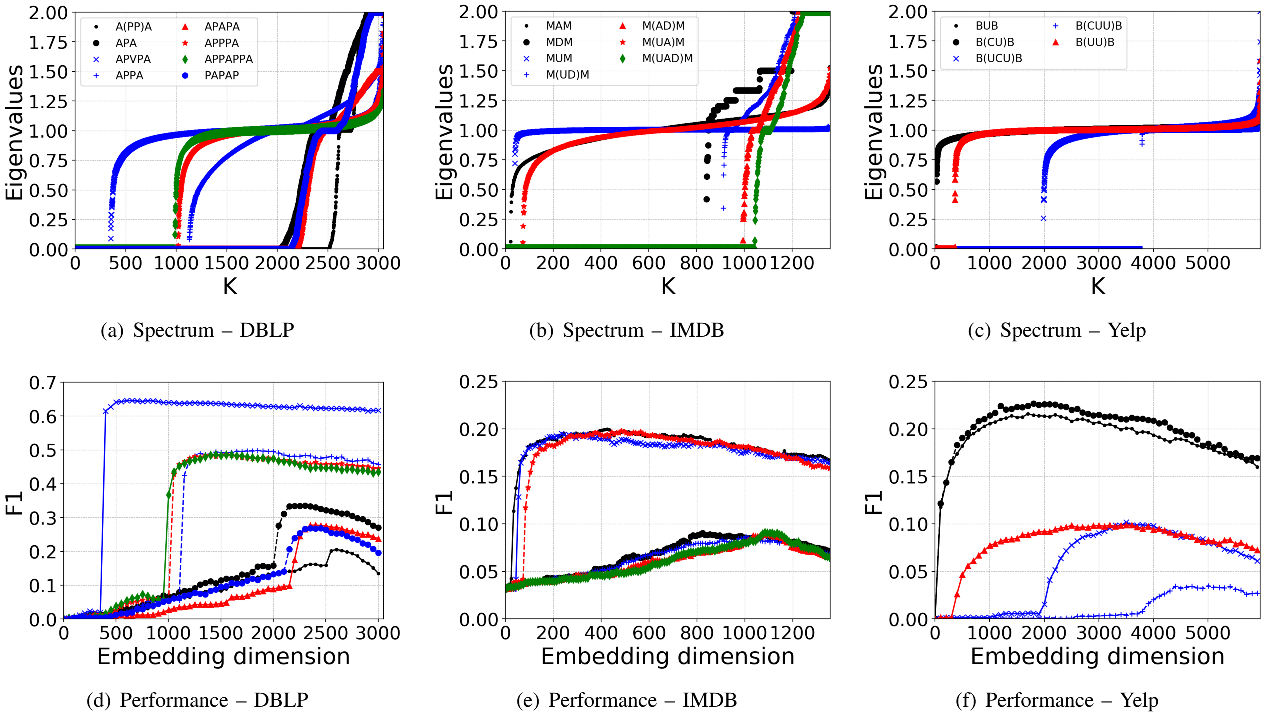
As we can observe, the spectrum curve always starts from zero, and increase to positive values at some point, which we refer to as *FPP (First-Positive-Point)*. Its position has a clear correlation with the embedding quality, *i.e.*, (1) the spectrum curve and performance curve mostly start to grow at the same point, and (2) the earlier the spectrum curve starts to grow, the higher the performance curve can reach.

Theoretical Interpretations. Looking at Figure 2 from a graph theory point of view, we find the strong correlations quite revealing. According to Theorem 2.1, the number of zeros in the spectrum is exactly the number of disconnected components in the corresponding network. Hence, the results in Figure 2 clearly indicate that meta-graphs leading to better connected projected networks usually have better HIN embedding quality. The second observation is more interesting. Again, according to Theorem 2.1, the first several embedding dimensions that correspond to zero eigenvalues actually work as the features that identify whether the corresponding vertex belongs to a single connected component. As the number of embedding dimensions increases from 0 to FPP, the performance hardly improves, which implies that the identity of each connected component might be not useful for the HIN embedding. This observation is quite opposite to the recent significant findings in homogeneous networks [43], [44], where in practice, with a good high-order connectivity pattern, the identity of connected component itself may have already been a strong feature for vertex embedding.

The results further show that a small number of eigenvectors associated with small non-zero eigenvalues may help greatly towards the overall HIN embedding performance. According to Theorem 2.4, within each connected component, these newly added eigenvectors begin to characterize the nodal domains within the connected components. Theorem 2.5 further implies that these nodal domains are essentially good network communities (*i.e.*, densely connected parts) within the connected components. Therefore, the results can be understood as most of the connected components hold good community structures within themselves, and thus these components can be well represented by only a few eigenvectors associated with the small positive eigenvalues right after the FPP.

Efficient Assessments. Based on the systematic empirical study and theoretical interpretations, we are able to efficiently

[1] https://dblp.uni-trier.de/
[2] http://www.imdb.com/
[3] https://www.yelp.com/
[4] https://aminer.org/citation
[5] https://grouplens.org/datasets/movielens/100k/
[6] https://www.yelp.com/dataset
[7] https://docs.scipy.org/doc/numpy.linalg.svd.html
[8] http://scikit-learn.org/stable/modules/svm.html



(a) Spectrum – DBLP

(b) Spectrum – IMDB

(c) Spectrum – Yelp

(d) Performance – DBLP

(e) Performance – IMDB

(f) Performance – Yelp

Fig. 2. FPP of the spectra clearly correlates with the embedding performance.

assess the meta-graph utility regarding HIN embedding by simply looking at the leading eigenvalues of the corresponding projected network. Particularly, meta-graphs corresponding to early FPP of the spectrum curves are generally more useful. Moreover, spectral embeddings corresponding to the positive eigenvalues are more important than those of the zero ones.

B. Curvature - Network Low-Rank Property

Empirical Observations. Besides FPP, is any other spectrum property indicative to the embedding quality? To rule out the influence of FPP, we focus on each pair of meta-graphs and pick out their *LC3 (Largest Common Connected Component)* as illustrated in Figure 3. Suppose subnetworks $\mathcal{S}_1$ and $\mathcal{S}_2$ are the connected components on the projected networks of meta-graph $\mathcal{M}_1$ and $\mathcal{M}_2$, respectively. Then subnetwork $\mathcal{S}_3$ is a common connected component of $\mathcal{M}_1$ and $\mathcal{M}_2$. LC3 is the one with the largest number of vertices. On LC3, the spectra of two meta-graphs are aligned, in a way that they both only have a single zero value.

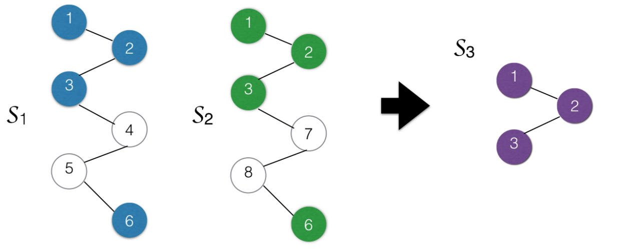


Fig. 3. The spectra aligning process of finding LC3 of two projected networks.

Throughout our systematic empirical study, we find the spectrum curvature highly correlated with the embedding quality. Particularly, as shown in Figure 4, (1) the faster the eigenvalues grow in the beginning (*i.e.*, larger curvature), the better the embedding quality is, and (2) the embedding quality degenerates with larger embedding sizes. Although we only present the performance towards author classification on DBLP due to space limit, we find exactly similar phenomena for other network mining tasks on all three datasets we use.

Theoretical Interpretations. The observations again can be interpreted through references to spectral graph theory. First, better embedding based on the faster growth of eigenvalues can be explained from the perspective of PCA. With simple linear algebra, we know the optimal loss of PCA (Eq. 3) equals to $\sum_{i=k+1}^{|\mathcal{V}|}(2-\lambda_i)^2$. The fast growth of eigenvalues means for some small i, λ_i can be already large and hence $2-\lambda_i$ is small. One step further, it implies that the projected network has preferable low-rank properties, *i.e.*, a steep curvature indicates the energy mostly concentrates on a few eigenvalues of the normalized adjacency matrix $\mathcal{A}$. Therefore, the loss of PCA can be small for some small embedding dimension k, and $\mathcal{A}$ can be well approximated by the inner product of low-dimensional embedding vectors of different vertices, *i.e.*, HWH^T. When the eigenvalues achieve the medium value (almost 1), the nodal domains of the corresponding eigenvectors can hardly express the community structures of the network according to the inequality in Eq. 2, as the RHS is greater than 1 and becomes trivial. Therefore, the eigenvectors *w.r.t.* large eigenvalues (>1) may not be a good representation that encodes the topology of the network. As a consequence, eigenvectors of large λ_i are not informative for the HIN embedding. In fact, adding them as the embedding features may cause significant overfitting and hence the degenerated

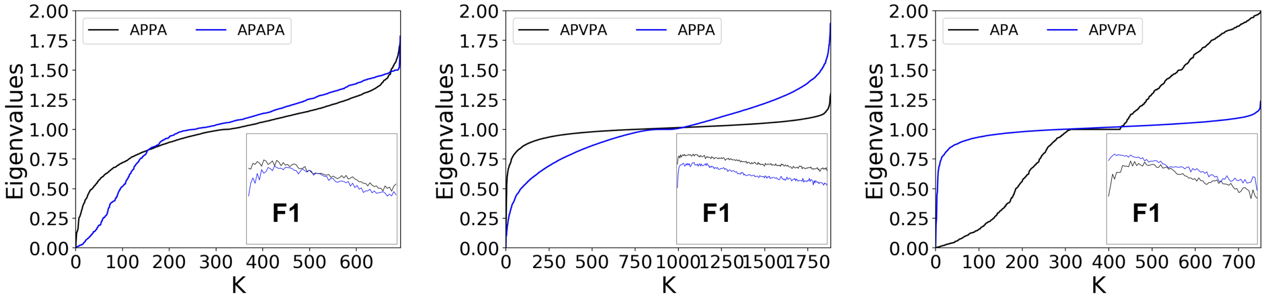


Fig. 4. The curvature of the spectrum clearly correlates with the embedding performance.

learning performance.

Efficient Assessments. Besides FPP, the curvature of spectrum allows additional efficient assessment of the meta-graph utility, *i.e.*, steeper eigenvalue growth indicates better embedding performance. Moreover, spectral embeddings corresponding to the first several non-zero eigenvalues carry the most useful structure information, while the subsequent ones are less useful and may easily lead to model overfitting.

C. Implied Assessment Method

Given a meta-graph $\mathcal{M}$ on an arbitrary HIN $\mathcal{N}$, without knowing the downstream task, we simply need to compute the projected network and its leading eigenvalues, based on which we can then quickly assess the utility of $\mathcal{M}$ and select the most informative embedding dimensions. Note that, this method is efficient due to several facts: (1) Given $\mathcal{N}$ and $\mathcal{M}$, it is not always necessary to compute the projected networks from scratch. In fact, many real-world network companies nowadays maintain the graph databases to constantly track and store the instances of certain high-order structures for analytical usage [48]. (2) Finding instances of $\mathcal{M}$ on $\mathcal{N}$ is a well-studied problem, which can be efficiently solved by algorithms like [9]. (3) Our method only works with the projected networks, which are much smaller than the original HINs. (4) We only need to check the leading K eigenvalues and do not require the networks to be fully decomposed [41].

V. Meta-Graph Combination

A. Motivations and Challenges

In HIN, each meta-graph captures particular semantics. Take Figure 5 (a) as an example. In a movie-review HIN, suppose *Alice* is connected with *Bob* by meta-graph $\mathcal{M}_1$ (*UDU*), and with *Carl* by $\mathcal{M}_2$ (*UGU*). Thus, the underlying semantics are, *Alice* and *Bob* like movies directed by the same *director*, while she and *Carl* like movies of the same *genre*. For the general purpose of HIN embedding, it is natural that we want the embeddings to capture all "useful" semantics by simultaneously considering multiple meta-graphs.

However, simply concatenating the individual embeddings of multiple meta-graphs may actually lead to poor results. To illustrate this, we continue with the example in Figure 5. In the concatenated embedding space of $\mathcal{M}_1$ and $\mathcal{M}_2$, *Bob* and *Carl* might be far away, since they do not like the same movies. As a consequence, *Alice* can only lie between the two of them while being close to neither of them, due to the triangle inequality property of metric spaces as shown in (b). It implies that, in order to capture the essentially useful semantics, we need to wisely distort the embedding space, by throwing away redundant, noisy and non-discriminative information. Eventually, we want a model that is able to automatically trade-off different meta-graphs and their embedding dimensions, and arrive at an embedding space like one of those in (c), where *Alice* can be close to either *Bob* or *Carl*, depending on which meta-graph is found to be more important.

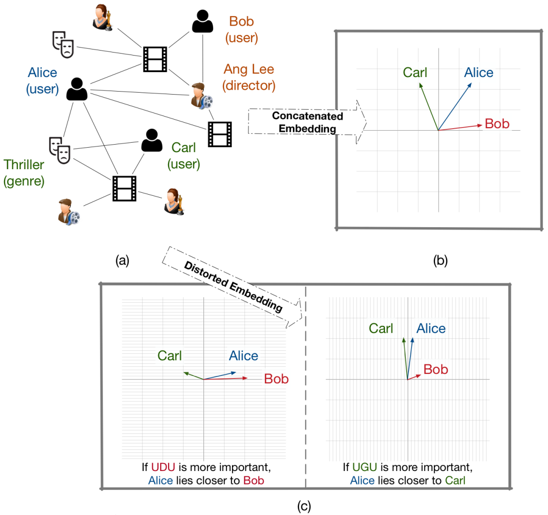


Fig. 5. A toy example of a movie-review HIN.

We find that the problem of unsupervised meta-graph combination essentially boils down to two challenging subproblems as follows:

1) Dimension Reduction: As we have just explained, simply concatenating the individual embeddings ignores the interactions and correlations among meta-graphs, and results in high dimensionality and data redundancy. Moreover, as we can observe from our analyses in Section IV, the individual embeddings can be quite noisy, which together with the high dimensionality can easily lead to model overfitting.
2) Meta-graph Selection: As we also observe in Section IV, the utilities of meta-graphs towards HIN embedding can be rather different. While they can be efficiently assessed individually, there is no end-to-end systematic method for the selection of important meta-graphs by considering them together in an unsupervised way, so

as to capture all essentially useful semantics in a HIN.

B. Autoencoder with $\ell_{2,1}$-Loss

To simultaneously solve the above two problems, we propose the method of autoencoder with $\ell_{2,1}$-loss. The overall framework is shown in Figure 6.

For unsupervised dimension reduction, we take the spirit of [49], [40] in preserving the most representative features by variance maximization. Further, we get motivated by recent advances in neural networks and deep learning, particularly, the unsupervised deep denoise autoencoders [50], [51]. They have been shown effective in feature composition due to the proven advantages in capturing the intrinsic features within high-dimensional noisy redundant inputs in a non-linear way.

One step further, we design a specific $\ell_{2,1}$-loss to further require grouped sparsity on the embedding dimensions *w.r.t.* each meta-graph, so as to effectively select the more useful meta-graphs in an end-to-end fashion. It helps us to put more stress on the important meta-graphs to improve the final embedding quality. Moreover, it also enables better understanding of the meta-graph utilities, and allows further validation of our meta-graph assessment methods.

In what follows, we go through our model design in details.

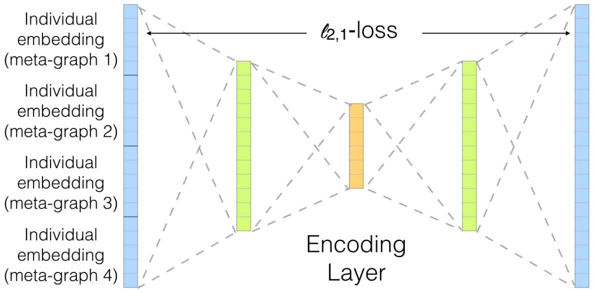

Individual embedding (meta-graph 1)
Individual embedding (meta-graph 2)
Individual embedding (meta-graph 3)
Individual embedding (meta-graph 4)
$\ell_{2,1}$-loss
Encoding Layer

Fig. 6. Our joint embedding framework for meta-graph combination.

For each vertex $v_i \in \mathcal{V}$, given its spectral embedding of the k-th projected network $\mathbf{h}_i^k$, the input of our meta-graph combination framework is thus $\mathbf{x}_i = \Phi(\mathbf{h}_i^1, \ldots, \mathbf{h}_i^K)$, which is a vector concatenation of K spectral embeddings.

To leverage the power of autoencoders, given $\mathbf{x}_i$, we first apply an encoder, which consists of multiple laters of fully connected feedforward neural networks with *LeakyReLU* activations. The neural networks are in decreasing sizes and after them we get a Q-dim compressed embedding $\mathbf{q}_i$ as

$$\mathbf{q}_i = \mathbf{f}_e^P(\ldots \mathbf{f}_e^2(\mathbf{f}_e^1(\mathbf{x}_i))\ldots), \tag{4}$$

where P is the number of hidden layers in the encoder, and

$$\mathbf{f}_e^p(\mathbf{x}) = \text{LeakyReLU}(\mathbf{W}_e^p \text{Dropout}(\mathbf{x}) + \mathbf{b}_e^p). \tag{5}$$

To ensure that $\mathbf{q}_i$ captures the important information in $\mathbf{x}_i$, we compute the reconstruction $\tilde{\mathbf{x}}_i$ of $\mathbf{x}_i$ through stacking a decoder, which also consists of multiple layers of fully connected feedforward neural networks. The sizes of neural networks are in an increasing order, exactly the opposite as in the encoder. So we have

$$\tilde{\mathbf{x}}_i = \mathbf{f}_d^P(\ldots \mathbf{f}_d^2(\mathbf{f}_d^1(\mathbf{q}_i))\ldots), \tag{6}$$

$$\mathbf{f}_d^p(\mathbf{x}) = \text{LeakyReLU}(\mathbf{W}_d^p \text{Dropout}(\mathbf{x}) + \mathbf{b}_d^p). \tag{7}$$

The number of hidden layers in the decoder is also H, the same as in the encoder.

After the decoder, a reconstruction loss for embedding is computed as

$$\mathcal{J} = \sum_{i=1}^{n} l(\mathbf{x}_i, \tilde{\mathbf{x}}_i), \tag{8}$$

which is a summation over all vertices in $\mathcal{V}$.

For regular autoencoders, l is implemented either as a cross entropy for binary features, or a mean squared error for continuous features. However, per reasons we have just discussed in Section V.A, we apply a specific $\ell_{2,1}$-loss [52] to l as

$$l(\mathbf{x}_i, \tilde{\mathbf{x}}_i) = ||\mathbf{x}_i - \tilde{\mathbf{x}}_i||_{2,1} \tag{9}$$

$$= \sum_{k=1}^{K} ||\mathbf{h}_i^k - \tilde{\mathbf{h}}_i^k||_2, \tag{10}$$

where $\tilde{\mathbf{h}}_i^k$ is the reconstructed embedding of $\mathbf{h}_i^k$.

C. Theoretical Justification

Autoencoder is a non-linear generalization of PCA. Particularly, consider an ℓ_2-loss in Eq. 8. It is exactly the same as the PCA loss in Eq. 3, if we remove the non-linear activation layers. From the mathematical geometry point of view, consider the original embedding space as a ball, PCA distorts this ball into an ellipsoid by picking out the directions of the greatest variance in the dataset. This process necessarily incurs an information loss, but the variance maximization process ensures the lost information to be more of the redundant part.

One step further, our leverage of autoencoder further enables the utilization of the expressiveness of non-linear feedforward neural networks. It allows us to efficiently explore more complex interactions of different embedding dimensions and distort the embedding space with more flexibility [53].

Beyond the standard autoencoder, our $\ell_{2,1}$-loss is built on group-wise feature selection via group lasso [54]. The setting of the ℓ_1-loss only imposes sparsity in the group level while the ℓ_2-loss within each group expresses that all features of one group should be selected or rejected simultaneously. The mathematical property of $\ell_{2,1}$-loss coincides with our target of combining the individual embeddings of different meta-graphs: When compressing the embeddings, we want the model to ensure that only some grouped dimensions are exactly reconstructed, which allows it to ignore certain useless meta-graphs and instead focus on the more important ones. In this way, our model is able to select important meta-graphs in an end-to-end fashion.

D. Empirical Analysis

We conduct a series of empirical analyses to specifically study our meta-graph combination model. The autoencoder we use in this subsection has only one encoding layer with no additional hidden layer. For input, we take an 80-dimensional individual spectral embedding for each meta-graph. Due to space limit, we focus the analyses on the DBLP dataset with four meta-graphs: *APVPA, APPA, APPPA* and *APPAPPA*.

$\lvert Q \rvert$	10	20	40	70	100	200
Linear	0.582	0.611	0.628	0.631	0.637	0.642
Non-linear	0.654	0.668	0.673	0.669	0.668	0.676

TABLE I
COMPARING THE F1 SCORES OF LINEAR AND NON-LINEAR MODELS.

Meta-graphs	*APVPA*	*APPA*	*APPPA*	*APPAPPA*	F1
$\ell_{2,1}$	0.214	0.256	5.222	5.243	0.695
ℓ_2	2.420	3.133	3.349	3.340	0.668

TABLE II
COMPARING AUTOENCODERS WITH $\ell_{2,1}$-LOSS AND ℓ_2-LOSS.

The first analysis we did is on the comparison between linear and non-linear autoencoders. In Table I, the *non-linear* results are constantly better than the *linear* ones, which is generated by the exact same architectures with the non-linear activation functions removed. The differences are more significant for smaller encoding dimensions. It clearly indicates the power of non-linear embedding and supports our selection of autoencoder as the basic model.

Subsequently, we study the efficacy of our $\ell_{2,1}$-loss. In Table II, we compare two models with the $\ell_{2,1}$-loss and standard ℓ_2-loss. The encoding dimension is fixed to 200, and the losses are all group-wise computed after vector mean-shifting and normalization. As we can see, the $\ell_{2,1}$-losses can effectively differentiate the utilities of *APVPA* and *APPA* from *APPPA* and *APPAPPA*, while the ℓ_2-losses are more uniform over all meta-graphs. The final embedding quality regarding the classification F1 score with the $\ell_{2,1}$-loss is also significantly better. It confirms our intuition of leveraging the group lasso for end-to-end meta-graph selection.

Moreover, such results from our combination method clearly deem the meta-graphs *APVPA* and *APPA* to be more important than *APPPA* and *APPAPPA*, which aligns with our assessment method in Section IV. Such observation allows us to close the gap between these two methods, and further propose a unified framework for meta-graph based HIN embedding. To be specific, given a large number of candidate meta-graphs (due to the lack of precise domain knowledge), our assessment method can be firstly applied for an efficient but coarse selection of individual candidate meta-graphs as well as promising embedding dimensions. Then our combination method can be applied to fine-tune the combined embedding, which results in low-dimensional high-quality representations capturing the most important information across multiple meta-graphs.

VI. EXPERIMENTAL EVALUATION

We comprehensively evaluate the performance of our proposed method in comparison with various state-of-the-art NN-based HIN embedding algorithms on the same three large real-world datasets as we described in Section IV. Extensive experimental results show that our method can effectively select and combine useful meta-graphs for general-purpose unsupervised HIN embedding, which leads to supreme performance on multiple traditional network mining tasks.

A. Experimental Settings

Datasets. The datasets we use are DBLP, IMDB and Yelp, as described in Section IV, with statistics shown in Table III.

Dataset	Size	#Types	#Nodes	#Links	#Classes
DBLP	4.33GB	4	335,185	2,704,655	4
IMDB	16.1MB	4	45,913	153,645	23
Yelp	6.52GB	5	1,123,649	8,912,736	6

TABLE III
STATISTICS OF THE FOUR PUBLIC DATASETS WE USE.

Baselines. We compare with various unsupervised HIN embedding algorithms to comprehensively evaluate the performance of our proposed method.

- **PTE** [25]: It decomposes the heterogeneous network into a set of bipartite networks and then captures first and second order proximities for HIN embedding.
- **Meta2vec** [13]: It leverages heterogeneous random walks and negative sampling for HIN embedding.
- **ESim** [14]: It leverages meta-path guided path sampling and noise-contrastive estimation for HIN embedding.
- **HINE** [15]: It captures w-hop neighborhoods under meta-path constrained path counts fo HIN embedding.
- **Hin2vec** [16]: It jointly learns the node embeddings and meta-path embeddings through relation triple prediction.
- **AspEm** [17]: It selects meta-graphs based on a heuristic incompatibility score and combine the embedding of multiple induced graphs through vector concatenation.

Evaluation protocols. We study the embedding quality of all algorithms on two traditional network mining tasks, *i.e.*, node classification and link prediction. The class labels and evaluation links are generated as follows. For DBLP, we use the manual class labels of authors from four research areas, *i.e.*, database, data mining, machine learning and information retrieval provided by [1]. For IMDB, we follow [17] to use all 23 available genres such as drama, comedy, romance, thriller, crime and action as class labels. For Yelp, we extract six sets of businesses based on some available attributes, *i.e.*, good for kids, take out, outdoor seating, good for groups, delivery and reservation. Following the common practice in [9], [10], for each dataset, we assume that authors (movies, businesses) within each semantic class are similar in certain ways, and generate pairwise links among them for the evaluation of link prediction.

All algorithms learn the embeddings on the whole network. For node classification, we split the class labels in half for training and testing. We train a standard SVM[9] on the training data and compute the *F1* and *Jaccard* scores towards all class labels on the testing data. For link prediction, we compute the cosine similarity of each node pair, and rank all nodes for each node to compute the *precision at* K and *recall at* K. All algorithms are run on a server with one GeForce GTX TITAN X GPU and two Intel Xeon E5-2650V3 10-core 2.3GHz CPUs. While scalability is not our focus in this work, we also measure the training time of all algorithms. Our audoencoder-based embedding model can be efficiently trained on GPU and consumes no significantly more time than most baselines.

[9] http://scikit-learn.org/stable/modules/svm.html

Parameter settings. Our method only has a few parameters. The sizes of spectral embeddings are set *w.r.t.* our assessment method (*i.e.*, 80 for DBLP, 150 for IMDB and 800 for Yelp). For the autoencoder, we empirically set the number of both encoder and decoder layers to 2, each halving (doubling) the size of the previous layer. The drop out ratio is 0.2.

For each HIN, we firstly enumerate all meta-graphs up to size 5 and visualize their spectra to select a few most promising meta-graphs by our assessment method[10]. These meta-graphs are then given as input to our combination method. Since most promising meta-graphs are actually meta-paths, they are also given as input to all compared baselines. All other parameters of the baselines are either set as given in the original work on the same datasets, or tuned to the best through standard five-fold cross validation on each dataset.

B. Performance Comparison with Baselines

As we can see from Table IV and V, the HIN embeddings produced by our method constantly lead to better performance on both node classification and link prediction tasks. The results on node classification are all averaged over 10 random training-testing splits, and the improvements of our method over the compared algorithms all passed the paired t-tests with significance value $p < 0.01$. The link prediction results are averaged across all nodes in the networks.

Firstly, by comparing the results in this section with those in Section IV, we can clearly see that properly combining multiple meta-graphs leads to better overall performances, especially on more complicated HINs like IMDB and Yelp. Secondly, the relative performance of baselines varies across different datasets and tasks, while our method is able to constantly yield more than 10% relative improvements compared with the strongest baselines on all datasets and both tasks, which clearly demonstrates its advantage.

C. Embedding Efficiency

Now we conduct an in-depth study on the effects of different embedding sizes and training data on the performance of our method, in order to further demonstrate our embedding efficiency. As we can see in Figure 7, for all algorithms, when the embedding size is large, the task performance relies much on the amount of training data, due to the effect of overfitting. This justifies our intuition of efficient feature selection to reduce the embedding size. On the other hand, for all baselines, small-size embeddings can hardly capture all useful information and always perform much worse than the large-size ones. Our method is the only one that efficiently captures the most important information with small embedding sizes, which is especially useful when training data are limited.

VII. Conclusions

In this work, we systematically study the assessment and combination of meta-graphs for unsupervised HIN embedding.

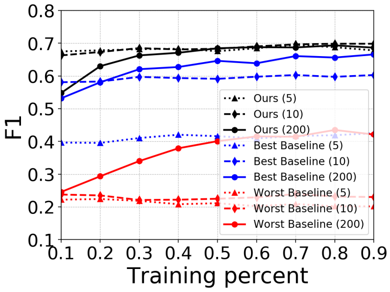


Fig. 7. Node classification performance with varying embedding sizes and training data. The highlighted trends in our texts are even more clearly observed on IMDB and Yelp, which have more complicated network structures and larger individual embedding sizes. Due to space limit, we only present the results on DBLP.

For future work, we would like to see how our methods can generally benefit various HIN models through better meta-graph selection. Moreover, our methods, while producing high-quality HIN embedding for various downstream tasks, also indicate the importance of each meta-graph in the spectral embedding process and is of great interest to in-depth studies of HIN high-order organizations in particular domains.

Acknowledgement

Research was sponsored in part by U.S. Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), DARPA under Agreement No. W911NF-17-C-0099, National Science Foundation IIS 16-18481, IIS 17-04532, and IIS-17-41317, DTRA HDTRA11810026, and grant 1U54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative (www.bd2k.nih.gov). The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

[1] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, “Pathsim: Meta path-based top-k similarity search in heterogeneous information networks,” *VLDB*, vol. 4, no. 11, pp. 992–1003, 2011.

[2] Z. Liu, V. W. Zheng, Z. Zhao, F. Zhu, K. Chang, M. Wu, and J. Ying, “Semantic proximity search on heterogeneous graph by proximity embedding.” in *AAAI*, 2017, pp. 154–160.

[3] S. Hou, Y. Ye, Y. Song, and M. Abdulhayoglu, “Hindroid: An intelligent android malware detection system based on structured heterogeneous information network,” in *KDD*, 2017, pp. 1507–1515.

[4] Y. Sun, B. Norick, J. Han, X. Yan, P. S. Yu, and X. Yu, “Integrating meta-path selection with user-guided object clustering in heterogeneous information networks,” *TKDD*, vol. 7, no. 3, p. 11, 2013.

[5] H. Zhao, Q. Yao, J. Li, Y. Song, and D. L. Lee, “Meta-graph based recommendation fusion over heterogeneous information networks,” in *KDD*, 2017, pp. 635–644.

[6] C. Yang, C. Zhang, J. Han, X. Chen, and J. Ye, “Did you enjoy the ride: Understanding passenger experience via heterogeneous network embedding,” in *ICDE*.

[7] Y. Sun and J. Han, “Mining heterogeneous information networks: principles and methodologies,” *Synthesis Lectures on Data Mining and Knowledge Discovery*, vol. 3, no. 2, pp. 1–159, 2012.

[8] C. Wang, Y. Sun, Y. Song, J. Han, Y. Song, L. Wang, and M. Zhang, “Relsim: relation similarity search in schema-rich heterogeneous information networks,” in *SDM*, 2016, pp. 621–629.

[9] Y. Fang, W. Lin, V. W. Zheng, M. Wu, K. Chang, and X.-L. Li, “Semantic proximity search on graphs with metagraph-based learning,” in *ICDE*, 2016, pp. 277–288.

[10] C. Meng, R. Cheng, S. Maniu, P. Senellart, and W. Zhang, “Discovering meta-paths in large heterogeneous information networks,” in *WWW*, 2015, pp. 754–764.

[11] Y. Shi, P.-W. Chan, H. Zhuang, H. Gui, and J. Han, “Prep: Path-based relevance from a probabilistic perspective in heterogeneous information networks,” in *KDD*, 2017, pp. 425–434.

[10] IMDB: *MDM*, *MAM*, *MUM*, *M(UD)M*, *M(AD)M*, *M(UA)M*, *M(UAD)M*; DBLP: *A(PP)A*, *APA*, *APPA*, *APVPA*, *APAPA*, *APPPA*, *PAPAP*, *APPAPPA*; Yelp: *BUB*, *B(UC)B*, *B(UCU)B*, *B(CUU)B*, *B(UU)B*

Algorithm	F1			Jaccard		
	DBLP	IMDB	Yelp	DBLP	IMDB	Yelp
PTE	0.326 ± 0.011	0.299 ± 0.024	0.031 ± 0.023	0.363 ± 0.013	0.401 ± 0.039	0.035 ± 0.031
Meta2vec	0.235 ± 0.012	0.045 ± 0.008	0.018 ± 0.018	0.262 ± 0.014	0.256 ± 0.022	0.022 ± 0.031
ESim	0.596 ± 0.022	0.290 ± 0.011	0.296 ± 0.012	0.612 ± 0.022	0.405 ± 0.014	0.244 ± 0.011
HINE	0.645 ± 0.013	0.233 ± 0.007	0.245 ± 0.021	0.648 ± 0.013	0.368 ± 0.018	0.212 ± 0.021
Hin2vec	0.595 ± 0.013	0.123 ± 0.007	0.166 ± 0.015	0.606 ± 0.013	0.287 ± 0.017	0.149 ± 0.017
AspEm	0.536 ± 0.014	0.181 ± 0.028	0.064 ± 0.007	0.591 ± 0.012	0.290 ± 0.013	0.049 ± 0.011
Ours	**0.689 ± 0.011**	**0.321 ± 0.021**	**0.339 ± 0.015**	**0.685 ± 0.019**	**0.445 ± 0.013**	**0.268 ± 0.012**

TABLE IV

NODE CLASSIFICATION PERFORMANCE IN COMPARISON WITH ALL BASELINES.

Algorithm	Precision@10			Recall@10		
	DBLP	IMDB	Yelp	DBLP	IMDB	Yelp
PTE	0.337	0.701	0.212	0.0044	0.0079	0.0023
Meta2vec	0.334	0.623	0.203	0.0043	0.0070	0.0022
ESim	0.538	0.710	0.278	0.0070	0.0081	0.0033
HINE	0.566	0.727	0.255	0.0073	0.0085	0.0030
Hin2vec	0.548	0.674	0.236	0.0070	0.0075	0.0026
AspEm	0.577	0.663	0.149	0.0074	0.0075	0.0016
Ours	**0.612**	**0.735**	**0.296**	**0.0079**	**0.0085**	**0.0036**

TABLE V

LINK PREDICTION PERFORMANCE IN COMPARISON WITH ALL BASELINES.

[12] C. Wang, Y. Song, H. Li, M. Zhang, and J. Han, "Knowsim: A document similarity measure on structured heterogeneous information networks," in *ICDM*, 2015, pp. 1015–1020.

[13] Y. Dong, N. V. Chawla, and A. Swami, "metapath2vec: Scalable representation learning for heterogeneous networks," in *KDD*, 2017, pp. 135–144.

[14] J. Shang, M. Qu, J. Liu, L. M. Kaplan, J. Han, and J. Peng, "Meta-path guided embedding for similarity search in large-scale heterogeneous information networks," *arXiv preprint arXiv:1610.09769*, 2016.

[15] Z. Huang and N. Mamoulis, "Heterogeneous information network embedding for meta path based proximity," *arXiv preprint arXiv:1701.05291*, 2017.

[16] T.-y. Fu, W.-C. Lee, and Z. Lei, "Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning," in *CIKM*, 2017, pp. 1797–1806.

[17] Y. Shi, H. Gui, Q. Zhu, L. Kaplan, and J. Han, "Aspem: Embedding learning by aspects in heterogeneous information networks," in *SDM*, 2018.

[18] M. Wan, Y. Ouyang, L. Kaplan, and J. Han, "Graph regularized meta-path based transductive regression in heterogeneous information network," in *SDM*, 2015, pp. 918–926.

[19] H. Jiang, Y. Song, C. Wang, M. Zhang, and Y. Sun, "Semi-supervised learning over heterogeneous information networks by ensemble of meta-graph guided random walks," in *IJCAI*, 2017, pp. 1944–1950.

[20] T. Chen and Y. Sun, "Task-guided and path-augmented heterogeneous network embedding for author identification," in *WSDM*.

[21] X. Yu, Y. Sun, B. Norick, T. Mao, and J. Han, "User guided entity similarity search using meta-path selection in heterogeneous information networks," in *CIKM*, 2012, pp. 2025–2029.

[22] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *KDD*, 2014, pp. 701–710.

[23] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *WWW*, 2015, pp. 1067–1077.

[24] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *KDD*, 2016, pp. 855–864.

[25] J. Tang, M. Qu, and Q. Mei, "Pte: Predictive text embedding through large-scale heterogeneous text networks," in *KDD*, 2015, pp. 1165–1174.

[26] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *KDD*, 2016, pp. 1225–1234.

[27] T. Lyu, Y. Zhang, and Y. Zhang, "Enhancing the network embedding quality with structural similarity," in *CIKM*, 2017.

[28] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013, pp. 3111–3119.

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

[30] S. White and P. Smyth, "A spectral clustering approach to finding communities in graphs," in *SDM*, 2005, pp. 274–285.

[31] F. R. Chung, *Spectral graph theory*. American Mathematical Soc., 1997, no. 92.

[32] J. R. Lee, S. O. Gharan, and L. Trevisan, "Multiway spectral partitioning and higher-order cheeger inequalities," *JACM*, vol. 61, no. 6, p. 37, 2014.

[33] P. Li and O. Milenkovic, "Submodular hypergraphs: p-laplacians, cheeger inequalities and spectral clustering," in *ICML*, 2018.

[34] E. B. Davies, G. M. Gladwell, J. Leydold, and P. F. Stadler, "Discrete nodal domain theorems," *Linear Algebra Appl*, vol. 336, no. 1-3, pp. 51–60, 2001.

[35] F. Tudisco and M. Hein, "A nodal domain theorem and a higher-order cheeger inequality for the graph p-laplacian," *Journal of Spectral Theory*, 2017.

[36] H. Yin, A. R. Benson, J. Leskovec, and D. F. Gleich, "Local higher-order graph clustering," in *KDD*, 2017, pp. 555–564.

[37] D. Zhou, S. Zhang, M. Y. Yildirim, S. Alcorn, H. Tong, H. Davulcu, and J. He, "A local algorithm for structure-preserving graph cut," in *KDD*, 2017, pp. 655–664.

[38] B. Long, Z. M. Zhang, X. Wu, and P. S. Yu, "Spectral clustering for multi-type relational data," in *ICML*, 2006, pp. 585–592.

[39] S. Sengupta and Y. Chen, "Spectral clustering in heterogeneous networks," *Statistica Sinica*, pp. 1081–1106, 2015.

[40] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[41] E. R. Davidson, "The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices," *Journal of Computational Physics*, vol. 17, no. 1, pp. 87–94, 1975.

[42] J. Qiu, Y. Dong, H. Ma, J. Li, K. Wang, and J. Tang, "Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec," in *WSDM*, 2018, pp. 459–467.

[43] A. R. Benson, D. F. Gleich, and J. Leskovec, "Higher-order organization of complex networks," *Science*, vol. 353, no. 6295, pp. 163–166, 2016.

[44] P. Li and O. Milenkovic, "Inhomogeneous hypergraph clustering with applications," in *NIPS*, 2017, pp. 2305–2315.

[45] X. Li, B. Kao, Y. Zheng, and Z. Huang, "On transductive classification in heterogeneous information networks," in *CIKM*, 2016, pp. 811–820.

[46] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: extraction and mining of academic social networks," in *KDD*, 2008, pp. 990–998.

[47] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *TIIS*, vol. 5, no. 4, p. 19, 2016.

[48] R. Angles and C. Gutierrez, "Survey of graph database models," *ACM Computing Surveys (CSUR)*, vol. 40, no. 1, p. 1, 2008.

[49] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Advances in neural information processing systems*, 2006, pp. 507–514.

[50] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *ICML*, 2008, pp. 1096–1103.

[51] Q. V. Le, "Building high-level features using large scale unsupervised learning," in *ICASSP*, 2013, pp. 8595–8598.

[52] G. Ma, C.-T. Lu, L. He, P. S. Yu, and A. B. Ragin, "Multi-view graph embedding with hub detection for brain network analysis," in *AAAI*, 2018.

[53] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[54] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.