AspEm: Embedding Learning by Aspects in Heterogeneous Information Networks

Yu Shi[†] Huan Gui[‡*] Qi Zhu[†] Lance Kaplan[§] Jiawei Han[†]

[†]Dept. of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL USA

[‡]Facebook Inc., Menlo Park, CA USA [§]U.S. Army Research Laboratory, Adelphi, MD USA

[†]{yushi2, qiz3, hanj}@illinois.edu [‡]huangui@fb.com [§]lance.m.kaplan.civ@mail.mil

Abstract

Heterogeneous information networks (HINs) are ubiquitous in real-world applications. Due to the heterogeneity in HINs, the typed edges may not fully align with each other. In order to capture the semantic subtlety, we propose the concept of aspects with each aspect being a unit representing one underlying semantic facet. Meanwhile, network embedding has emerged as a powerful method for learning network representation, where the learned embedding can be used as features in various downstream applications. Therefore, we are motivated to propose a novel embedding learning framework—AspEm—to preserve the semantic information in HINs based on multiple aspects. Instead of preserving information of the network in one semantic space, AspEm encapsulates information regarding each aspect individually. In order to select aspects for embedding purpose, we further devise a solution for AspEm based on dataset-wide statistics. To corroborate the efficacy of AspEm, we conducted experiments on two real-words datasets with two types of applications—classification and link prediction. Experiment results demonstrate that AspEm can outperform baseline network embedding learning methods by considering multiple aspects, where the aspects can be selected from the given HIN in an unsupervised manner.

Keywords: Heterogeneous information networks, network embedding, graph mining, representation learning.

1 Introduction

In real-world applications, objects of different types interact with each other, forming heterogeneous relations. Such objects and relations, acting as strongly-typed nodes and edges, constitute numerous *heterogeneous information networks (HINs)* [16, 19]. HINs have received increasing interests in the past decade due to its capability of retaining the rich type information, as well as the accompanying wide applications such as recommender system [25], clustering [20], and outlier detection [28]. As an example, the IMDb network is an HIN containing information about users' preferences over movies and have five different node types: user, movie, actor, director, and genre.

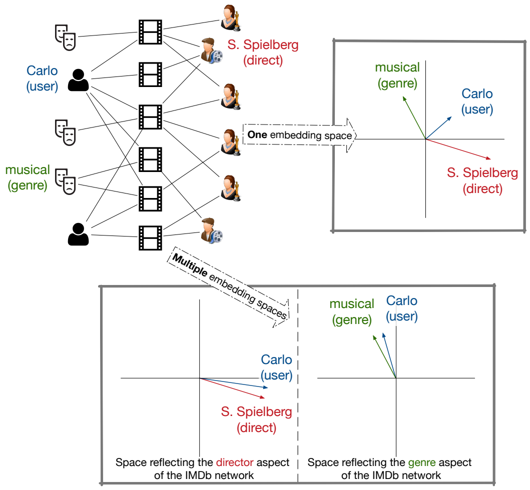


Figure 1: A toy example of node embeddings in an HIN. The upper left of the figure depicts the interactions among nodes, where users review movies and movies have various attributes. Carlo likes both musicals and movies directed by S. Spielberg. If all nodes were embedded to one space, Carlo would be close to neither musical nor S. Spielberg due to the dissimilarity between musical and S. Spielberg. However, by embedding the aspect related to director and that related to genre into separate spaces, Carlo could be close to S. Spielberg in one space, and close to musical in another.

Meanwhile, network embedding has recently emerged as a scalable unsupervised representation learning method [4, 6, 12, 14, 21, 22, 24]. In particular, network embedding learning projects the network into low-dimensional space, where each node is represented using a corresponding embedding vector and the relativity among nodes is preserved. With the semantic information transcribed from the networks, the embedding vectors can be directly used as node features in various downstream applications. We therefore use the two terms—the embedding of a node and

[*]The work was done when Huan Gui was a graduate student at UIUC.

the learned feature of a node—interchangeably in this paper.

The heterogeneity in HINs poses a specific challenge for data mining and applied machine learning. We hence propose to study the problem of learning embedding in HINs with an emphasis on leveraging the rich and intrinsic type information. There are multiple attempts in studying HIN embedding or tackling specific application tasks using HIN embedding [2, 3, 7, 21]. Though these studies formulate the problem differently with respective optimization objectives, they share a similar underlining philosophy: using a unified objective function to embed all the nodes into *one* low-dimensional space.

Embedding all the nodes into *one* low-dimensional space, however, may lead to information loss. Take the IMDb network as example, where users review movies based on their preferences. Since each movie has several facets, users may review movies with emphasis over different facets. For instance, both Alice and Bob may like the movie Star Wars, but Alice likes it because of Carrie Fisher *(actor)*; while Bob likes it because it is a fantasy movie *(genre)*. Furthermore, suppose user Carlo likes both movies directed by Steven Spielberg and musicals. Due to the semantic dissimilarity between Steven Spielberg and musical, if this HIN were projected into one embedding space as visualized in the upper part of Figure 1, musical *(genre)* and Steven Spielberg *(director)* would be distant from each other, while the user Carlo would be in the middle and close to neither of them. Therefore, it is of interest to obtain an embedding that can reflect Carlo's preference for both musicals and Spielberg's movies. To this end, we are motivated to embed the network into two distinct spaces: one for the aspect of genre information whereas the other for that of director information. In this case, Carlo could be close to musical *(genre)* in the first space and close to Steven Spielberg *(director)* in the second space as in the lower part of Figure 1.

In this paper, we propose a flexible embedding learning framework—AspEm—for HINs that mitigates the incompatibility among aspects via considering each aspect separately. The use of aspects is motivated by the intuition that very distinct relationship could exist between components of a typed network, which has been observed in a special type of HIN [18]. Moreover, we demonstrate the feasibility of selecting a set of representative aspects for any HIN using statistics of the network without additional supervision.

It is worth noting that most existing embedding learning methodologies can be extended based on AspEm using the principle that different aspects should reside in different embedding spaces. Therefore, AspEm is a principled and flexible framework that has the potential of inheriting the merits of other embedding learning methods. To the best of our knowledge, this is the first work to study the property of multiple aspects in HIN embedding learning. Lastly, we summarize our contributions as follows:

1. We provide a key insight regarding incompatibility in HINs that each HIN can have multiple aspects that do not align with each other. We thereby identify that embedding algorithms employing only one embedding space may lose subtlety of the given HIN.
2. We propose a flexible HIN embedding framework, named AspEm, that can mitigate the incompatibility among multiple aspects via considering the semantic information regarding each aspect separately.
3. We propose an aspect selection method for AspEm, which demonstrates that a set of representative aspects can be selected from any HIN using statistics of the network without additional supervision.
4. We conduct quantitative experiments on two real-world datasets with various evaluation tasks, which validate the effectiveness of the proposed framework.

2 Related Work

Heterogeneous information networks. Heterogeneous information network (HIN) has been extensively studied as a powerful and effective paradigm to model networked data with rich and informative type information [16, 19]. Following this paradigm, a great many applications such as classification, clustering, recommendation, and outlier detection have been studied [16, 17, 19, 20, 25, 28]. However, many of these existing works rely on feature engineering [20, 25, 28]. Meanwhile, we aim at proposing an unsupervised feature learning method for general HINs that can serve as the basis for different downstream applications.

Network embedding. Network embedding has recently emerged as a representation learning approach for networks [6, 10, 12, 14, 22, 24]. Unlike traditional unsupervised feature learning approaches [1, 15, 23] that typically arise from the spectral properties of networks, recent advances in network embedding are mostly based on local properties of networks and are therefore more scalable. The designs of many homogeneous network embedding algorithms [6, 12, 14, 22] trace to the skip-gram model [9] that aims to learn word representations in natural language processing. Beyond skip-gram, embedding methods for preserving certain other network properties have also been studied [10, 24].

Heterogeneous information network embedding. There is a line of research on embedding learning for HINs, while the necessity of modeling aspects of an HIN and embedding them into different spaces has been rarely discussed. On top of the LINE algorithm [22], Tang et al. propose to learn embedding by traversing all edge types and sampling one edge at a time for each edge type [21], where the use of type information is shown to be instrumental. Chang et al. propose to embed HIN with additional node features via deep architectures [2], which does not suit for typical HINs consisting of only typed nodes and edges. Gui et al. devise an HIN embedding algorithm to model a special type of HINs with hyperedges, which does not apply to general HINs [7]. More recently, an HIN embedding algorithm is proposed, which transcribes semantics in HINs by meta-paths [4]. However, this work does not employ multiple embedding spaces for different aspects. Moreover, it requires the involved meta-paths to be specified as input, while our method is completely unsupervised and can automatically select aspect using statistics of the given HIN. Embedding in the context of HIN has also been studied to address various application tasks with additional supervision [3, 8, 11, 26, 27]. These methods either yield features specific to given tasks or do not generate node features, and therefore fall outside of the scope of unsupervised HIN embedding that we study.

Additionally, we review the related work on multi-sense embedding in the supplementary file for this paper, which is related but cannot be directly applied to the task of HIN embedding learning with aspects.

3 Problem Definition

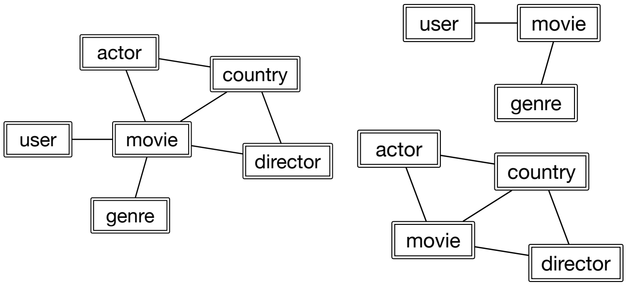
In this section, we formally define the problem of learning embedding from aspects of HINs and related notations.

DEFINITION 3.1. (HIN) *An* ***information network*** *is a directed graph $G = (\mathcal{V}, \mathcal{E})$ with a node type mapping $\phi : \mathcal{V} \to \mathcal{T}$ and an edge type mapping $\psi : \mathcal{E} \to \mathcal{R}$. Particularly, when the number of node types $|\mathcal{T}| > 1$ or the number of edge types $|\mathcal{R}| > 1$, the network is called a* ***heterogeneous information network*** *(HIN) [19].*

In addition, when the network is weighted and directed, we use $W_{uv}^{(r)}$ to denote the weight of an edge $e \in \mathcal{E}$ with type $r \in \mathcal{R}$ that goes out from node u and into node v. $D_u^{O(r)}$ and $D_u^{I(r)}$ represent the outward degree of node u (*i.e.*, the sum of weights associated with all edges in type r going outward from u) and the inward degree of node u (*i.e.*, the sum of weights associated with all edges in type r going inward to u), respectively. For unweighted networks, the degrees can be similarly defined. In case a network is undirected, it can be converted to the directed case by simply decomposing every edge to two directed edges with equal weights and opposite directions.

Given the typed essence, an HIN can be abstracted using a network schema $\tilde{G} = (\mathcal{T}, \mathcal{R})$ [19], which provides meta-information regarding the node types and edge types in the HINs. Figure 2a gives an example of the schema of a movie reviewing network as an HIN.

DEFINITION 3.2. (ASPECT OF HIN) *For a given HIN G with network schema $\tilde{G} = (\mathcal{T}, \mathcal{R})$, an* ***aspect*** *of G is defined as a subgraph of the network schema $\tilde{G}$. For an aspect a, we use $\mathcal{T}^a \subseteq \mathcal{T}$ to denote the node types involved in this aspect, and $\mathcal{R}^a \subseteq \mathcal{R}$ as the edge types involved in this aspect .*



(a) Schema (b) Two aspects

Figure 2: The schema and two aspects of an toy HIN with six node types: movie, director, actor, genre, country, and user.

As an example, we illustrate two aspects from the schema in Figure 2a: one on users' preferences for movies based on genre information (upper part in Figure 2b); and the other on the semantics of movies based on the composite information of directors, actors and their countries (lower part in Figure 2b). If we denote $\mathcal{A}$ a set of representative aspects generated by a certain method, where information is compatible within each aspect and is not redundant across different aspects, then an HIN with only one aspect will have $|\mathcal{A}| = 1$, $\mathcal{T}^a = \mathcal{T}$, and $\mathcal{R}^a = \mathcal{R}$.

DEFINITION 3.3. (HIN EMBEDDING FROM ASPECTS) *Suppose that an HIN $G = (\mathcal{V}, \mathcal{E})$ and a set of representative aspects $\mathcal{A}$ are given. For one aspect $a \in \mathcal{A}$, embedding learning in HIN from one aspect a is to learn a node embedding mapping $f^a : \{u \in \mathcal{V} : \phi(u) \in \mathcal{T}^a\} \to \mathbb{R}^{d(a)}$, where $d(a)$ is the embedding dimension for a and $d(a) \ll |\mathcal{V}|$. For all aspects in $\mathcal{A}$ and all nodes $u \in \mathcal{V}$, the problem of* ***embedding learning from aspects in HIN*** *is to learn corresponding feature vector $\mathbf{f}_u$, such that $\mathbf{f}_u = \bigoplus_{a \in \mathcal{A} \colon \phi(u) \in \mathcal{T}^a} \mathbf{f}_u^a$, where $\mathbf{f}_u^a$ is the embedding of node u in aspect a.*

We remark that, for nodes of different types, the corresponding $\mathbf{f}_u$ might be of different dimensions by definition.

4 The AspEm Framework

To address the problem of embedding learning from aspects in HIN, we propose a flexible framework to distinguish the semantic information regarding each aspect. Specifically, for a node u, the corresponding embedding vectors $\mathbf{f}_u^a$ are inferred independently for different aspects in $\{a \in \mathcal{A} : \phi(u) \in \mathcal{T}^a\}$. We name the new framework as AspEm, which is short for Aspect Embedding. AspEm includes three components: (i) selecting a set of representative aspects for the HIN of interest, (ii) learning embedding vectors for each aspect, and (iii) integrating embeddings from multiple aspects. We introduce these components as follows.

4.1 Aspect Selection in HINs

Since different aspects are expected to reflect distinct semantic facets of an HIN, an aspect of representative capability should consist of compatible edge types in terms of the information carried by the edges. Therefore, even without supervision from downstream applications, the incompatibility within each aspect can be leveraged to determine the quality of the aspect, and such incompatibility can be inferred from dataset-wide statistics.

Before introducing the proposed incompatibility measure, $\mathrm{Inc}(\cdot)$, we first describe the properties that we posit a proper measure should have as follows.

PROPERTY 4.1. (NON-NEGATIVITY) *For any aspect* a, $\mathrm{Inc}(a) \geq 0$.

PROPERTY 4.2. (MONOTONICITY) *For two aspects* a_1 *and* a_2, *if* $a_1 \subseteq a_2$, *then* $\mathrm{Inc}(a_1) \leq \mathrm{Inc}(a_2)$.

PROPERTY 4.3. (CONVEXITY) *For two aspects* a_1 *and* a_2, *if their graph intersection has empty edge set, i.e.,* $\mathcal{E}(a_1 \cap a_2) = \varnothing$, *then* $\mathrm{Inc}(a_1) + \mathrm{Inc}(a_2) \leq \mathrm{Inc}(a_1 \cup a_2)$.

We note that the intuition of Property 4.3 is that the incompatibility arises from the co-existence of multiple types of edges. As a result, generating an aspect by the union of a_1 and a_2 could only introduce more incompatibility.

To propose our incompatibility measure, we start from the simplest incompatibility-prone scenario: since the incompatibility arises from the co-existence of edge types, the simplest incompatible-prone aspects are those with two edge types joined by a common node type. In particular, an aspect in this form can be uniquely determined by a schema-level representation $\phi_l \xrightarrow{\psi_l} \phi_c \xrightarrow{\psi_r} \phi_r$, where $\phi_l, \phi_c, \phi_r \in \mathcal{T}$ are (not necessarily distinct) node types and $\psi_l, \psi_r \in \mathcal{R}$ are edge types. Once the incompatibility measure $\mathrm{Inc}(\cdot)$ is defined for this scenario, it can then be generalized to any aspect a by

$$\mathrm{Inc}(a) := \sum_{\langle \phi_l, \psi_l, \phi_c, \psi_r, \phi_r \rangle \subseteq a} \mathrm{Inc}(\phi_l \xrightarrow{\psi_l} \phi_c \xrightarrow{\psi_r} \phi_r), \tag{4.1}$$

where $\langle \phi_l, \psi_l, \phi_c, \psi_r, \phi_r \rangle \subseteq a$ represents enumerating all such sub-aspects in aspect a. For undirected networks, we do not distinguish $\langle \phi_l, \psi_l, \phi_c, \psi_r, \phi_r \rangle$ and $\langle \phi_r, \psi_r, \phi_c, \psi_l, \phi_l \rangle$ in this enumeration process. Note that such generalization meets the criteria in Property 4.2 and 4.3.

Incompatible edge types result in inconsistent information. To reflect such intuition, we define the incompatibility measure on aspects of the form $\phi_l \xrightarrow{\psi_l} \phi_c \xrightarrow{\psi_r} \phi_r$ with a Jaccard coefficient–based formulation over each node of type ϕ_c—the node type that joins two edge types. Specifically, for node u of type ϕ_c, we calculate the inconsistency in information observed from ψ_l and ψ_r by

$$\gamma(u) := \frac{\sum_{\phi(\tilde{u})=\phi_c} \max\left\{\mathbf{P}^{\psi_r}_{u,:}(\mathbf{P}^{\psi_r}_{\tilde{u},:})^\top, \mathbf{P}^{\psi_l^{-1}}_{u,:}(\mathbf{P}^{\psi_l^{-1}}_{\tilde{u},:})^\top\right\}}{\sum_{\phi(\tilde{u})=\phi_c} \min\left\{\mathbf{P}^{\psi_r}_{u,:}(\mathbf{P}^{\psi_r}_{\tilde{u},:})^\top, \mathbf{P}^{\psi_l^{-1}}_{u,:}(\mathbf{P}^{\psi_l^{-1}}_{\tilde{u},:})^\top\right\}} - 1, \tag{4.2}$$

where $\mathbf{M}^{\psi_i}$ is the adjacency matrix of edge type ψ_i and $\mathbf{P}^{\psi_i}$ is $\mathbf{M}^{\psi_i}$ after row-wise normalization. We remark that this formulation, with a difference of minus 1, is essentially the inverse of Jaccard coefficient over the one-hop neighbors that u can reach via edge type ψ_l and edge type ψ_r. The inverse is taken since greater Jaccard coefficient implies more similarity while we expect more inconsistency, and the minus 1 is appended so that $\gamma(u) = 0$ when $\mathbf{P}^{\psi_r} = \mathbf{P}^{\psi_l^{-1}}$, *i.e.*, no inconsistency if two edge types are identical. Lastly, we average over all such nodes to find incompatibility score of a simplest incompatible-prone aspect

$$\mathrm{Inc}(\phi_l \xrightarrow{\psi_l} \phi_c \xrightarrow{\psi_r} \phi_r) := \frac{1}{|\phi_c^*|} \sum_{u \in \phi_c^*} \gamma(u),$$

where ϕ_c^* is the set of all u in ϕ_c such that the denominator in Eq. (4.2) is nonzero and $\gamma(u)$ is thereby well-defined. Note that this definition satisfies Property 4.1.

To select a set $\mathcal{A}$ of representative aspects for given HIN under any threshold $\theta \in \mathbb{R}_{\geq 0}$, (i) an aspect with incompatible score greater than θ is not eligible to be selected into $\mathcal{A}$, because it is not semantically consistent enough; (ii) in case both aspects a_1 and a_2 have incompatible score below θ and $a_1 \subset a_2$, we do not select a_1 into $\mathcal{A}$. We note that the second requirement is intended to keep $\mathcal{A}$ concise, so that the information across different aspects is not redundant. Note that when both computation resource and overfitting in downstream application are not of concern, one may explore the potential of trading in model size for gaining additional performance boost by including both a_1 and a_2 to $\mathcal{A}$.

We will demonstrate by experiments in Section 5 that this proposed aspect selection method is effective in the sense that (i) ASPEM built atop this method can outperform baselines that do not model aspects; and (ii) the set of aspects selected using this statistics-based unsupervised method can outperform other comparable sets of aspects.

4.2 Embedding Learning from One Aspect

To design the embedding algorithm for one aspect, we extend the skip-gram model [9] in an approach inspired by existing network embedding studies [7, 21, 22]. We note that ASPEM is a flexible framework that can be directly integrated with other homogeneous network embedding methods [6, 12, 14, 24], other than the adopted skip-gram–based approach, while still enjoying the benefits of modeling aspects in HINs.

For an aspect $a \in \mathcal{A}$, the associated node embeddings can be denoted as $\{\mathbf{f}^a_u\}_{\phi(u) \in \mathcal{T}^a}$. Recall that $\mathcal{T}^a$ corresponds to the set of node types included in the aspect a. We model the probability of observing edge e with edge type $r \in \mathcal{R}^a$ from node u to node v as

$$p^a(v|u,r) = \frac{\exp\left(\mathbf{f}^a_u \cdot \mathbf{f}^a_v\right)}{\sum_{v' \in \mathcal{V}:\, \phi(v')=\phi(v)} \exp\left(\mathbf{f}^a_u \cdot \mathbf{f}^a_{v'}\right)}. \tag{4.3}$$

This equation can be interpreted as the probability of observing v given u and the edge type r. On the other hand, the empirical conditional probability observed from aspect a is

$$\hat{p}^a(v|u,r) = W_{uv}^{(r)} / D_u^{O(r)}. \tag{4.4}$$

To obtain embeddings that reflect the network topology, we seek to minimize the difference between the probability distribution derived from the learned embedding Eq. (4.3) and the empirical probability distribution observed in data Eq. (4.4). Therefore, the embedding learning is reduced to minimizing the following objective function

$$\mathcal{O}^a = \sum_{r\in\mathcal{R}^a} \sum_{u\in\mathcal{V}_{O(r)}} \lambda_u^{(r)} d\big(\hat{p}^a(\cdot|u,r), p^a(\cdot|u,r)\big), \tag{4.5}$$

where $\mathcal{V}_{O(r)} \subseteq \mathcal{V}$ is the set of all nodes with outgoing type-r edges, $\lambda_u^{(r)}$ is the relative importance of node u in the context of edges with type r, and $d(\cdot,\cdot)$ is the KL-divergence. Furthermore, we set $\lambda_u^{(r)} \propto D_u^{O(r)}$ with $\lambda_u^{(r)}$ sum up to 1 for a given edge type r. Putting pieces together, Eq. (4.5) can be rewritten as

$$\mathcal{O}^a = -\sum_{r\in\mathcal{R}^a} \frac{1}{\Omega^{(r)}} \sum_{u\in\mathcal{V}_{O(r)}} W_{uv}^{(r)} \log p^a(v|u,r), \tag{4.6}$$

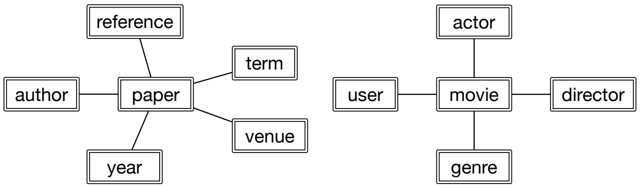
where $\Omega^{(r)} = \sum_{u,v} W_{uv}^{(r)}$. Consequently, the problem of learning embedding from an aspect $a \in \mathcal{A}$ is equivalent to solving the following optimization problem

$$\min_{\{\mathbf{f}_u^a\}_{u:\phi(u)\in\mathcal{T}^a}} \mathcal{O}^a. \tag{4.7}$$

With this formulation, information from each aspect of an HIN is transcribed into a different embedding space.

4.3 Compositing Node Embedding and Edge Embedding By solving the optimization problem Eq. (4.7), we are able to obtain a feature vector $\mathbf{f}_u^a$ for each node $u \in \mathcal{V}^a$ from the aspect $a \in \mathcal{A}$, and the final embedding for node u is thereby given by the concatenation of the learned embedding vectors from all aspects involving u, *i.e.*, $\mathbf{f}_u := \bigoplus_{a\in\mathcal{A}:\, \phi(u)\in\mathcal{T}^a} \mathbf{f}_u^a$. To characterize edges for applications such as link prediction, we follow the method in existing work [6] and define the edge embedding mapping g with domain in $\mathcal{V}\times\mathcal{V}$ as $g(u,v) = \mathbf{g}_{uv} := \bigoplus_{a\in\mathcal{A}:\, \phi(u),\phi(v)\in\mathcal{T}^a} \mathbf{f}_u^a \circ \mathbf{f}_v^a$, where $\circ$ is Hadamard product between two vectors of commensurate dimensions. We discuss this choice of edge embedding definition in the supplementary file, since it is not the main focus or contribution of our paper.

4.4 Model Inference It is computationally expensive to directly optimize the objective function Eq. (4.6) since the partition function in Eq. (4.3) sums over all the nodes in $\mathcal{V}$. Therefore, we approximate it with negative sampling [9] and resort to asynchronous stochastic gradient descent (ASGD) [13] for optimization as with the common practice in skip-gram–based embedding methods [6, 12, 21, 22]. For each iteration in ASGD, we first sample an edge type r from $\mathcal{R}^a$; then sample an edge $e = (u,v)$ of type r with the sampling probability proportional to $W_{uv}^{(r)}$; and finally obtain negative samples from the noise distribution $P_n^{(r)}(v) \propto \left[D_v^{I(r)}\right]^{3/4}$ [9]. The optimization objective for each iteration is therefore $\log\sigma(\mathbf{f}_u^a \cdot \mathbf{f}_v^a) + \sum_{i=1}^{K} \mathbb{E}_{v_i'\sim P_n^{(r)}} \log\sigma(-\mathbf{f}_u^a \cdot \mathbf{f}_{v_i'}^a)$, where $\sigma(\cdot)$ is the sigmoid function $\sigma(x) = \exp(x)/\big(1+\exp(x)\big)$. This optimization procedure shares the same spirit with some existing network embedding algorithms, and one may refer to the network embedding paper by Tang *et al.* [22] for further details.



(a) DBLP (b) IMDb

Figure 3: The network schemas of DBLP and IMDb.

5 Experiments

In order to provide evidence for the efficacy of AspEm, we experiment with two real-world HINs in this section. Specifically, the learned embeddings are fed into two types of downstream applications—multi-class classification and link prediction—to answer the following two questions:

Q1 Does exploiting aspects in HIN embedding learning help better capture the semantics of typed networks in both link prediction and classification tasks?

Q2 Without supervision, is it feasible to select a set of representative aspects just using dataset-wide statistics.

5.1 Data Description We use two publicly available real-world HIN datasets: DBLP and IMDb. **DBLP** is a bibliographical information network in the computer science domain[1]. There are six types of nodes in the network: author (A), paper (P), reference (R), term (T), venue (V), and year (Y), where reference corresponds to papers being referred by other papers. The terms are extracted and released by Chen et al. [3]. The edge types include: author writing paper, paper citing reference, paper containing term, paper publishing in venue, and paper publishing in year. The corresponding network schema is depicted in Figure 3a. Note that we distinguish the node type of reference, so that a paper have a different embedding when acting as a reference. **IMDb** is an HIN built by linking the movie-attribute information from IMDb

[1] https://aminer.org/citation

Table 1: Basic statistics for the DBLP and IMDb networks.

DBLP	Author	Paper	Reference	Term	Venue	Year
	1,003,836	1,756,680	693,406	402,687	7,528	62
IMDB	User	Movie	Actor	Director	Genre	
	943	1,360	42,275	918	23	

and the user-reviewing-movie relationship from MovieLens-100K.[2] There are five types of nodes in the network: user (U), movie (M), actor (A), director (D), and genre (G). The edge types include: user reviewing movie, actor featuring in movie, director directing movie, and movie being of genre. The network schema can be found in Figure 3b. We summarize the statistics of the datasets in Table 1.

We use the node types to represent an aspect in these two HINs. For example, APY in the DBLP network refers to the aspect involving author, paper, and year, and UMA in IMDb represents the aspect involving user, movie, and actor. The schema of each aspect can be easily inferred based on the holistic network schema, as shown in Figure 3.

5.2 Baseline Methods and Experiment Setting To answer Q1 at the beginning of the section, we compare AspEm against several unsupervised embedding methods. **SVD** [5]: a matrix factorization based method, where singular value decomposition is performed on the adjacent matrix of the homogeneous network and the first d singular vectors are taken as the node embeddings of the network, where d is the dimension of the embedding. **DeepWalk** [12]: a homogeneous network embedding method, which samples multiple walks starting from each node, and then applies the skip-gram model to learn embedding. **LINE** [22]: a homogeneous network embedding method, which treats the neighbors of a node as its context, and then applies the skip-gram model to learn embedding. **OneSpace**: as a heterogeneous network embedding method, OneSpace serves as a direct comparison against the proposed AspEm algorithm to validate the utility of embedding different aspects into multiple spaces. This method is given by the proposed AspEm framework with the full HIN schema as the only selected aspect. We note that the OneSpace method embeds all nodes into only one low-dimensional space. In the special case of HINs with star-schema, OneSpace is identical to PTE proposed in [21]. We remark that DeepWalk is identical to node2vec [6] under default hyperparameters.

For the baselines developed for homogeneous networks, we treat the HIN as a homogeneous network by neglecting the node types. Additionally, we apply the same downstream learners onto the embeddings yielded by different embedding methods for fair comparison.

Parameters. While AspEm is capable of using different dimensions for different aspects, we employ the same dimension for all aspects out of simplicity. In other words, we set $d(a_1) = \ldots = d(a_{|\mathcal{A}|}) = d, a_1, \ldots, a_{|\mathcal{A}|} \in \mathcal{A}$. In particular, we set $d = 100$ for DBLP and $d = 10$ for IMDb. For fair comparison, we experiment with two dimensions for every baseline method: (i) the dimension of one aspect used by AspEm (*i.e.*, d) and (ii) the total dimension of all aspects employed by AspEm (*i.e.*, $|\mathcal{A}| \cdot d$). We report the better result between the two choices of dimension for every baseline method. $1,000$ million edges are sampled to learn the embedding on DBLP, and 100 million edges are sampled on IMDb. The number of negative samples is set to 5 following the common practice in network embedding [22].

Table 2: Classification accuracy in two DBLP tasks.

Dataset/task	DBLP-group		DBLP-area	
Classifier	LR	SVM	LR	SVM
SVD	0.7566	0.7550	0.8158	0.8008
DeepWalk	0.6629	0.7077	0.8308	0.8390
LINE	0.7037	0.7314	0.8526	0.8540
OneSpace	0.7685	0.8333	0.8758	0.8731
AspEm	**0.8425**	**0.8889**	**0.8786**	**0.8813**

Selected aspects. Since all our experiments on DBLP involve the node type author (A), we set the threshold for incompatibility measure θ to be the *smallest possible value* such that all node types co-exist with the node type author (A) in at least one aspect eligible to be selected to $\mathcal{A}$ as per the two requirements discussed in Section 4.1. As a result, θ is set to be 221267 on DBLP, and the set of selected representative aspects, $\mathcal{A}$, is {APRTV, APT}. Similarly for IMDb, considering that all its experiments involve the node type user (U), θ is set to be 1927.68, and the set of selected representative aspects, $\mathcal{A}$, is {UMA, UMD, UMG}.

The detailed presentation on the calculations and figures involving threshold and aspect selection for both HINs can be found in the supplementary file for this paper.

5.3 Classification For classification tasks, we use the learned embeddings as node features and then classify the nodes into different categories using off-the-shelf classifiers. The classification performance is evaluated using accuracy. For a set of concerned nodes $\mathcal{X}$ and node $x \in \mathcal{X}$, denote $l(x)$ the predicted label of x and denote $l^*(x)$ the ground truth label. Then accuracy is defined as $\mathrm{Acc.} = \frac{1}{|\mathcal{X}|}\sum_{x \in cX} \delta\big(l(x) = l^*(x)\big)$, where $|\mathcal{X}|$ is the cardinality of $\mathcal{X}$ and $\delta(\cdot)$ is the indicator function.

Due to the availability of trustworthy class labels, we perform two classification tasks on DBLP. The first one (**DBLP-group**) is on the research group affiliation of each author. We consider four research groups led by Christos Faloutsos, Dan Roth, Jiawei Han, and Michael I. Jordan. 116 authors in the dataset are labeled with such group affiliation. The second label set (**DBLP-area**) is on the primary research area of authors. 4,040 authors are manually labeled in four research areas: data mining, database, machine learning, and artificial intelligence [20].

[2] https://grouplens.org/datasets/movielens/100k/

Table 3: Link prediction results on DBLP and IMDb.

Dataset	DBLP						IMDb					
Metrics	P@1	P@3	P@10	R@1	R@3	R@10	P@1	P@3	P@10	R@1	R@3	R@10
SVD	0.6648	0.5164	0.2274	0.2939	0.6178	0.8512	0.2470	0.2474	0.2249	0.0152	0.0445	0.1343
DeepWalk	0.7395	0.5297	0.2303	0.3268	0.6329	0.8622	0.3499	0.3605	0.3416	0.0253	0.0774	0.2236
LINE	0.7404	0.5367	0.2299	0.3267	0.6375	0.8596	0.4782	0.4701	0.4130	0.0379	0.1133	0.3137
OneSpace	0.7440	0.5381	0.2279	0.3301	0.6401	0.8519	0.4665	0.4386	0.3852	0.0435	0.1146	0.3038
AspEm	**0.7724**	**0.5645**	**0.2356**	**0.3479**	**0.6749**	**0.8810**	**0.5090**	**0.4853**	**0.4219**	**0.0464**	**0.1296**	**0.3420**

We experiment with two widely used classifiers. One is logistic regression (LR) and the other is support vector machine (SVM). Both classifiers are based on the liblinear implementation.[3] The classification accuracy for different methods are reported in Table 2.

The proposed AspEm method outperformed all four baselines in both tasks with either of the two downstream learners applied. In particular, AspEm yielded better results than OneSpace, which confirms our intuition that there exists incompatibility among aspects, and learning node embeddings independently from different aspects can better preserves the semantics of an HIN. In addition, we observed that the classification results of AspEm were significant better than OneSpace in research group classification; while the improvement of AspEm over OneSpace was less significant in research area classification. This can be partially explained by that the label of research groups is more relevant to temporal information compared with that of research area, and the presence of the aspect APY in AspEm may therefore be more informative for the research group classification task.

Based on the results in Table 2, another observation is that the embedding methods distinguishing node types (OneSpace and AspEm) performed better than those not considering node types. This observation is in line with previous studies [7], and can be explained by the heterogeneity of node types in HINs. The nodes of different types in HINs have different properties, such as degrees distribution. Simply ignoring such information can lead to information loss.

5.4 Link Prediction On experiments with link prediction essence, we perform author identification on the DBLP dataset, and user review prediction on the IMDb dataset. Precision and recall are used for evaluating these tasks. Precision at k ($P@k$) is defined as $P@k = \frac{\text{\# of true instances at top } k}{k}$, and recall at k ($R@k$) is defined as $R@k = \frac{\text{\# of true instances at top } k}{\text{\# of total true instances}}$.

We describe the key facts on deriving features for link prediction, and provide further details in the supplementary file. **DBLP**—The author identification task on DBLP aims at re-identifying the authors of an anonymized paper, where the reference, term, venue, and year information is still available. Since papers in the test set do not appear in the training set, their embeddings are hence not available. Therefore, we use the edge embedding of an author and each attribute of a paper (reference, term, venue, or year) to infer whether this author writes this paper. Specifically, for both train and test sets, we derive the feature of an author–paper pair by (i) first computing the edge embedding of the concerned author and each attribute of the concerned paper; (ii) then averaging all edge embedding vectors with the same edge type (author–reference, author–term, author–venue, or author–year) to find four edge-type-specific vectors; (iii) finally deriving the feature vector for an author–paper pair by concatenating of the previous four averaged edge embedding vectors. **IMDb**—The user review prediction task on IMDb aims at predicting which user reviews a movie. Features for user–movie pairs are likewise derived as with author–paper pairs in DBLP.

Table 4: Link prediction results (P@1) using only one edge.

Edge embedding used	AR	AT	AV	AY
Aspect APRTVY (OneSpace)	0.6933	0.6723	0.6501	0.3166
Aspect APRTV	**0.7566**	**0.6977**	**0.6878**	——
Aspect APR	0.6071	——	——	——
Aspect APT	——	0.6802	——	——
Aspect APV	——	——	0.5836	——
Aspect APY	——	——	——	0.3187

On top of the derived node pair features as well as labels in the training set, logistic regression is trained for inferring the existence of edges in the test set. We choose the scikit-learn[4] implementation with the SAG solver for logistic regression—different from that used for classification—because the SAG solver converges faster than liblinear, and the author identification task on DBLP has a huge number of author–paper pairs as training instances.

From the main results on link prediction presented in Table 3, we have observation consistent with the classification tasks that OneSpace and AspEm had better performance than the methods without considering type information. Also, AspEm outperformed OneSpace.

Predictive power of single edge embedding. In order to better understand the mechanism of AspEm in the link prediction tasks, we dissect each aspect and study the predictive power of a single edge embedding from one aspect. Specifically, we use each edge embedding over an author-attribute pair from one aspect for link prediction. Due to space limita-

[3] https://www.csie.ntu.edu.tw/ cjlin/liblinear/

[4] http://scikit-learn.org/stable/

Table 5: Link prediction results using different 2-combinations aspects on DBLP.

Metrics	$P@1$	$P@3$	$P@10$	$R@1$	$R@3$	$R@10$
{APTV, APRY}	0.7522	0.5476	0.2303	0.3362	0.6524	0.8611
{APRV, APTY}	0.7347	0.5327	0.2257	0.3271	0.6327	0.8425
{APRT, APVY}	0.7579	0.5556	0.2332	0.3385	0.6614	0.8708
{APTVY, APR}	0.7384	0.5360	0.2277	0.3280	0.6372	0.8499
{APRVY, APT}	0.7353	0.5356	0.2271	0.3263	0.6355	0.8474
{APRTY, APV}	0.7366	0.5362	0.2277	0.3274	0.6364	0.8492
{APRTV, APY}	**0.7724**	**0.5645**	**0.2356**	**0.3479**	**0.6749**	**0.8810**

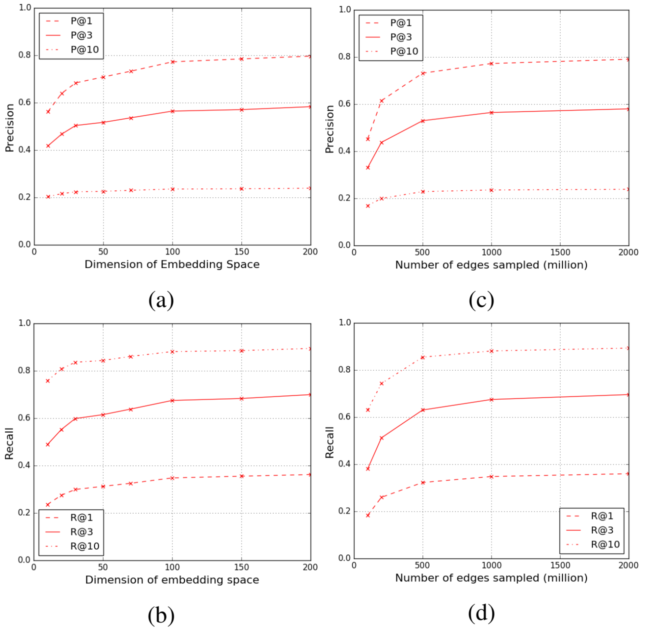
tion, we focus on the link prediction task on DBLP, because it has the largest number of available labels and can thereby yield most reliable conclusions. The experimental results are presented in Table 4, where the rows correspond to the aspect being used for embedding learning and the columns correspond to the edge embedding being used for link prediction.

It can be seen that using the aspect APRTV was better than using the bigger aspect APRTVY for all edge embeddings, where APRTVY was identical to the whole network schema. Such result provides evidence that for certain HIN datasets, using all the information in the network may be less effective than using partial information (i.e., one aspect). We interpret this result as: on the one hand, an author may focus on certain research field that cites certain classic references (R), uses certain terminologies (T), and publishes papers in certain venues (V), *i.e.*, R, T, and V correlate to some extent; on the other hand, an author may be actively publishing papers in a certain range of years (Y). However, the information regarding R, T, and V do not align well with Y. As a result, embedding R, T, V, and Y together into the same space (as in the OneSpace model) led to worse embedding quality even though more types of data were used. This result further consolidated our insight that HIN can have multiple aspects, and one should embed aspects with different semantics into distinct spaces.

To conclude, the results for classification and link prediction give an affirmative answer to Q1—Distinguishing the information from semantically different aspects can benefit HIN embedding learning.

5.5 The Impact of Aspect Selection In the previous section, we have shown that the aspect selection method proposed in Section 4.1 can effectively support the AspEm framework to outperform embedding methods that do not model aspects in HINs. In this section, we further address Q2 and demonstrate the set of representative aspects AspEm selected using the proposed method is of good quality compared with other selections of aspects.

To this end, we again use the link prediction on DBLP as the downstream evaluation task, and experiment with all sets of aspect that are comparable to {APRTV, APY}. Specifically, each of these comparable sets of aspects (i) has two aspects, and (ii) author and paper appear in both aspects, and other node types exist in exactly one of the two aspects.



(a) (c)
(b) (d)

Figure 4: (a) and (b) depict the precision and recall against various dimensions employed for the embedding space. (c) and (d) give the precision and recall against various choices of sampled edge numbers.

From the results presented in Table 5, it can be seen that the set of representative aspects selected by our proposed method, {APRTV, APY}, achieved the best performance among all comparable aspect selections. Note that all the 6 inferior sets of aspects have inconsistency score, $\text{Inc}(\cdot)$, greater than the threshold we set, which can be verified from the numbers provided in the supplementary file. This result further consolidates the feasibility of selecting representative aspects for any HIN solely by dataset-wide statistics without the need of additional task-specific supervision.

5.6 Hyperparameter Study We vary two hyperparameters, one at each time, that play important roles in embedding learning: dimension of embedding spaces and the number of edges sampled in the training phase. All other parameters are set following Section 5.2.

The performance in the link prediction task on DBLP is presented in Figure 4. It can be seen that model performance tended to be better as either the dimension of embedding spaces or the number of edges sampled grew, and the growth became less drastic after dimension reached 100 and number of edges sampled reached 1000 million. Such a pattern agrees with the results in other similar studies [6, 7, 22].

6 Conclusions and Future Work

In this paper, we study the problem of embedding learning in HINs. Particularly, we make the key observation that there are multiple aspects in heterogeneous information networks and there might be incompatibility among different aspects. Therefore, we take advantage of the information encapsulated in each aspect and propose AspEm—a new embedding learning framework from aspects, which comes with an unsupervised method to select a set of representative aspects from an HIN. We conducted experiments to corroborate the efficacy of AspEm in better representing the semantic information in HINs.

To focus on the utility of aspects in HIN embedding, AspEm is designed to be simple and flexible with each aspect embedded independently. For future work, one may explore optimizing the embeddings for all the aspects jointly, in hope of preserving more intrinsic information among nodes and further boost performance in downstream applications. Additionally, it is of interest to investigate into aspect selection methods when supervision is further provided.

Acknowledgments. This work 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, NSF IIS 16-18481, IIS 17-04532, and IIS-17-41317, 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. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] M. Belkin and P. Niyogi, *Laplacian eigenmaps and spectral techniques for embedding and clustering*, in NIPS, 2001.

[2] S. Chang, W. Han, J. Tang, G.-J. Qi, C. C. Aggarwal, and T. S. Huang, *Heterogeneous network embedding via deep architectures*, in KDD, ACM, 2015.

[3] T. Chen and Y. Sun, *Task-guided and path-augmented heterogeneous network embedding for author identification*, in WSDM, ACM, 2017.

[4] Y. Dong, N. V. Chawla, and A. Swami, *metapath2vec: Scalable representation learning for heterogeneous networks*, in KDD, ACM, 2017.

[5] G. H. Golub and C. Reinsch, *Singular value decomposition and least squares solutions*, Numerische Mathematik, (1970).

[6] A. Grover and J. Leskovec, *node2vec: Scalable feature learning for networks*, in KDD, ACM, 2016.

[7] H. Gui, J. Liu, F. Tao, M. Jiang, B. Norick, and J. Han, *Large-scale embedding learning in heterogeneous event data*, in ICDM, IEEE, 2016.

[8] Z. Liu, V. W. Zheng, Z. Zhao, F. Zhu, K. C.-C. Chang, M. Wu, and J. Ying, *Semantic proximity search on heterogeneous graph by proximity embedding*, in AAAI, 2017.

[9] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, *Distributed representations of words and phrases and their compositionality*, in NIPS, 2013.

[10] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu, *Asymmetric transitivity preserving graph embedding*, in KDD, ACM, 2016.

[11] S. Pan, J. Wu, X. Zhu, C. Zhang, and Y. Wang, *Tri-party deep network representation*, in IJCAI, 2016.

[12] B. Perozzi, R. Al-Rfou, and S. Skiena, *Deepwalk: Online learning of social representations*, in KDD, ACM, 2014.

[13] B. Recht, C. Re, S. Wright, and F. Niu, *Hogwild: A lock-free approach to parallelizing stochastic gradient descent*, in NIPS, 2011.

[14] L. F. Ribeiro, P. H. Saverese, and D. R. Figueiredo, *struc2vec: Learning node representations from structural identity*, in KDD, ACM, 2017.

[15] S. T. Roweis and L. K. Saul, *Nonlinear dimensionality reduction by locally linear embedding*, Science, (2000).

[16] C. Shi, Y. Li, J. Zhang, Y. Sun, and S. Y. Philip, *A survey of heterogeneous information network analysis*, TKDE, (2017).

[17] 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, ACM, 2017.

[18] Y. Shi, M. Kim, S. Chatterjee, M. Tiwari, S. Ghosh, and R. Rosales, *Dynamics of large multi-view social networks: Synergy, cannibalization and cross-view interplay*, in KDD, ACM, 2016.

[19] Y. Sun and J. Han, *Mining heterogeneous information networks: a structural analysis approach*, SIGKDD Explorations, (2013).

[20] Y. Sun, Y. Yu, and J. Han, *Ranking-based clustering of heterogeneous information networks with star network schema*, in KDD, ACM, 2009.

[21] J. Tang, M. Qu, and Q. Mei, *Pte: Predictive text embedding through large-scale heterogeneous text networks*, in KDD, ACM, 2015.

[22] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, *Line: Large-scale information network embedding*, in WWW, IW3C2, 2015.

[23] J. B. Tenenbaum, V. De Silva, and J. C. Langford, *A global geometric framework for nonlinear dimensionality reduction*, Science, (2000).

[24] D. Wang, P. Cui, and W. Zhu, *Structural deep network embedding*, in KDD, ACM, 2016.

[25] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han, *Personalized entity recommendation: A heterogeneous information network approach*, in WSDM, ACM, 2014.

[26] C. Zhang, L. Liu, D. Lei, Q. Yuan, H. Zhuang, T. Hanratty, and J. Han, *Triovecevent: Embedding-based online local event detection in geo-tagged tweet streams*, in KDD, ACM, 2017.

[27] C. Zhang, K. Zhang, Q. Yuan, H. Peng, Y. Zheng, T. Hanratty, S. Wang, and J. Han, *Regions, periods, activities: Uncovering urban dynamics via cross-modal representation learning*, in WWW, IW3C2, 2017.

[28] H. Zhuang, J. Zhang, G. Brova, J. Tang, H. Cam, X. Yan, and J. Han, *Mining query-based subnetwork outliers in heterogeneous information networks*, in ICDM, IEEE, 2014.

Supplementary File for “AspEm: Embedding Learning by Aspects in Heterogeneous Information Networks”

Yu Shi[†] Huan Gui[‡*] Qi Zhu[†] Lance Kaplan[§] Jiawei Han[†]
[†]Dept. of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL USA
[‡]Facebook Inc., Menlo Park, CA USA [§]U.S. Army Research Laboratory, Adelphi, MD USA
[†]{yushi2, qiz3, hanj}@illinois.edu [‡]huangui@fb.com [§]lance.m.kaplan.civ@mail.mil

Related Work on Multi-Sense Embedding

The idea of multiple aspects is in a way related to the polysemy of words. There have been some studies on inferring multi-sense embeddings of words [2, 6, 7, 8], which aims at inferring multiple embedding vectors for each word. However, the two tasks differ significantly in the following perspectives. Firstly, each node may have multiple embeddings due to the semantic subtlety associated with each aspect; while in multi-sense word embedding learning, the number of senses for each word varies. Secondly, we aim at studying the embedding in HINs; while multi-sense embeddings word embedding learning is for textual data. Therefore, the methods developed for multi-sense embedding learning cannot be directly applied to the task of HIN embedding learning with aspects.

Discussion on Compositing Edge Embedding

Instead of simply focusing on the node embeddings, another important component of networks is the interactions among nodes, *i.e.*, edges. Characterizing edges is important for downstream applications such as link prediction, which aims to predict whether there is an edge between a pair of nodes for a certain edge type. Therefore, it is of interest to define the embedding for edges. In this paper, we simply refer to a function of embeddings of a node pair as edge embedding, even if there might be no edge between the given node pair. The function of the edge embeddings is a hyperparameter and can be chosen by various designs.

Multiple possible ways exist in building edge embedding from the embedding vectors of the two involved nodes. In the AspEm framework, we bridge node embedding and edge embedding by Hadamard product [5]. We adopt Hadamard product in this design for two reasons: (i) For a pair of nodes, the inner product of the node embeddings is equivalent to the sum of Hadmard product of the two embeddings. As formulated in Eq. (4.3), the inner product of the node embeddings plays a vital role in modeling the proximity of edges between nodes. (ii) Empirical experiments on three datasets from a previous study [4] show that Hadamard product is a choice superior to other options in constructing edge embeddings from node embeddings. Specifically, we define the edge embedding mapping g with domain in $\mathcal{V} \times \mathcal{V}$ as $g(u, v) = \mathbf{g}_{uv} := \bigoplus_{a \in \mathcal{A}\colon \phi(u), \phi(v) \in \mathcal{T}^a} \mathbf{f}_u^a \circ \mathbf{f}_v^a$, where $\circ$ is Hadamard product between two vectors of commensurate dimensions.

We additionally remark that a recent paper [1] specifically addresses the problem of learning edge representation, and defines edge embedding as a parametric function over node embeddings, which is learned from the dataset. Since the focus of our paper is not to tackle the problem of edge embedding, we simply adopt the aforementioned straightforward Hadamard approach.

Additional Details on Link Prediction Feature Derivation

We provide further details on deriving features for link prediction tasks on both DBLP and IMDb in addition to the information available in Section 5.1 from the main content of the paper. **DBLP**—We randomly selected 32,488 papers into the test set, and take the rest as training data. Following the procedure proposed by Chen et al. [3], for each paper in test, we randomly sample a set of negative authors, which together with all the true authors constitute the candidate author set of size 100. **IMDb**—As with the DBLP author identification task, we sampled a candidate set of 100 movies for each user for testing on DBLP.

Incompatibility Score of Each Aspect in DBLP and IMDb

In this section of the supplementary material, we provide the sufficient statistics for calculating incompatibility of each aspect as defined in Section 4.1 from the main content of the paper. That is, we provide the incompatibility of aspects of the form $\phi_l \xrightarrow{\psi_l} \phi_c \xrightarrow{\psi_r} \phi_r$ as in Table 1 for DBLP and Table 2 for IMDb. Note that the proposed AspEm framework selects a set of representative aspects $\mathcal{A}$ for embedding purpose based on their incompatibility, which will be illustrated in the next section.

[*] The work was done when Huan Gui was a graduate student at UIUC.

Table 1: Sufficient statistics for incompability on DBLP.

Aspect	Incompatibility score
$R-P-Y$	52753.6
$A-P-Y$	221267.
$T-P-Y$	10254.4
$V-P-Y$	1830.08
$A-P-R$	307.988
$T-P-R$	6060.62
$V-P-R$	948.654
$T-P-A$	11518.2
$V-P-A$	5724.80
$V-P-T$	3579.59

Table 2: Sufficient statistics for incompability on IMDb.

Aspect	Incompatibility score
$U-M-A$	171.607
$D-M-A$	1689.76
$G-M-A$	12956.6
$D-M-U$	1927.68
$G-M-U$	636.442
$G-M-D$	531.266

Aspect Selection in DBLP and IMDb

Using Eq. (4.1) and the sufficient statistics provide in Table 1 and 2, one can calculate the incompatibility score of any aspect in DBLP and IMDb. We proceed to illustrate the aspect selection results using DBLP as example.

Given any threshold $\theta \in \mathbb{R}_{\geq 0}$, (i) any aspect with incompatible score greater than θ is not eligible to be selected into $\mathcal{A}$, because it is not meaningful and semantically consistent enough to be one representative aspect of the involved HIN; (ii) in case both aspects a_1 and a_2 have incompatible score below θ and $a_1 \subset a_2$, we do not select a_1 into $\mathcal{A}$. We note that the second requirement is intended to keep $\mathcal{A}$ concise and representative in the aspect selection process. However, when both computation resource and overfitting in downstream application are not of concern, one may explore the possibility of gaining additional performance boost by adding both a_1 and a_2 to $\mathcal{A}$.

Aspects in DBLP satisfying the aforementioned two requirements at various threshold θ are presented in Figure 1. Since all our experiments on DBLP involve the node type author (A), we set θ to be the *smallest possible value* such that all node types co-exist with the node type author (A) in at least one aspect eligible to be selected to $\mathcal{A}$ as per the aforementioned two requirements. Therefore, θ is set to be 221267 on DBLP. One can verify this by calculating $\text{Inc}(APRTV) = \text{Inc}(APR) + \text{Inc}(TPA) + \text{Inc}(VPA) + \text{Inc}(TPR) + \text{Inc}(VPR) + \text{Inc}(VPT) = 28139.9$, $\text{Inc}(APY) = 221267$, and $\text{Inc}(YPRTV) = \text{Inc}(RPY) + \text{Inc}(TPY) + \text{Inc}(VPY) + \text{Inc}(TPR) + \text{Inc}(VPR) + \text{Inc}(VPT) = 75426.9$.

Furthermore, aspects not involving author (A) are additionally excluded from $\mathcal{A}$ (those outside of the dotted boxes in Figure 1), because whether or not adding them to $\mathcal{A}$ does not affect the downstream evaluations. As a result, the set of selected representative aspects, $\mathcal{A}$, for DBLP is $\{$APRTV, APT$\}$.

Similarly for IMDb, following the same requirements and the consideration that all its experiments involve the node type user (U), θ is set to be 1927.68, and the set of selected representative aspects, $\mathcal{A}$ is $\{$UMA, UMD, UMG$\}$.

References

[1] S. Abu-El-Haija, B. Perozzi, and R. Al-Rfou, *Learning edge representations via low-rank asymmetric projections*, in CIKM, 2017.

[2] S. Arora, Y. Li, Y. Liang, T. Ma, and A. Risteski, *Linear algebraic structure of word senses, with applications to polysemy*, arXiv:1601.03764, (2016).

[3] T. Chen and Y. Sun, *Task-guided and path-augmented heterogeneous network embedding for author identification*, in WSDM, ACM, 2017.

[4] A. Grover and J. Leskovec, *node2vec: Scalable feature learning for networks*, in KDD, ACM, 2016.

[5] R. A. Horn, *The hadamard product*, in Proc. Symp. Appl. Math, 1990.

[6] S. K. Jauhar, C. Dyer, and E. H. Hovy, *Ontologically grounded multi-sense representation learning for semantic vector space models.*, in HLT-NAACL, 2015.

[7] A. Neelakantan, J. Shankar, A. Passos, and A. McCallum, *Efficient non-parametric estimation of multiple embeddings per word in vector space*, arXiv:1504.06654, (2015).

[8] S. Šuster, I. Titov, and G. van Noord, *Bilingual learning of multi-sense embeddings with discrete autoencoders*, arXiv:1603.09128, (2016).

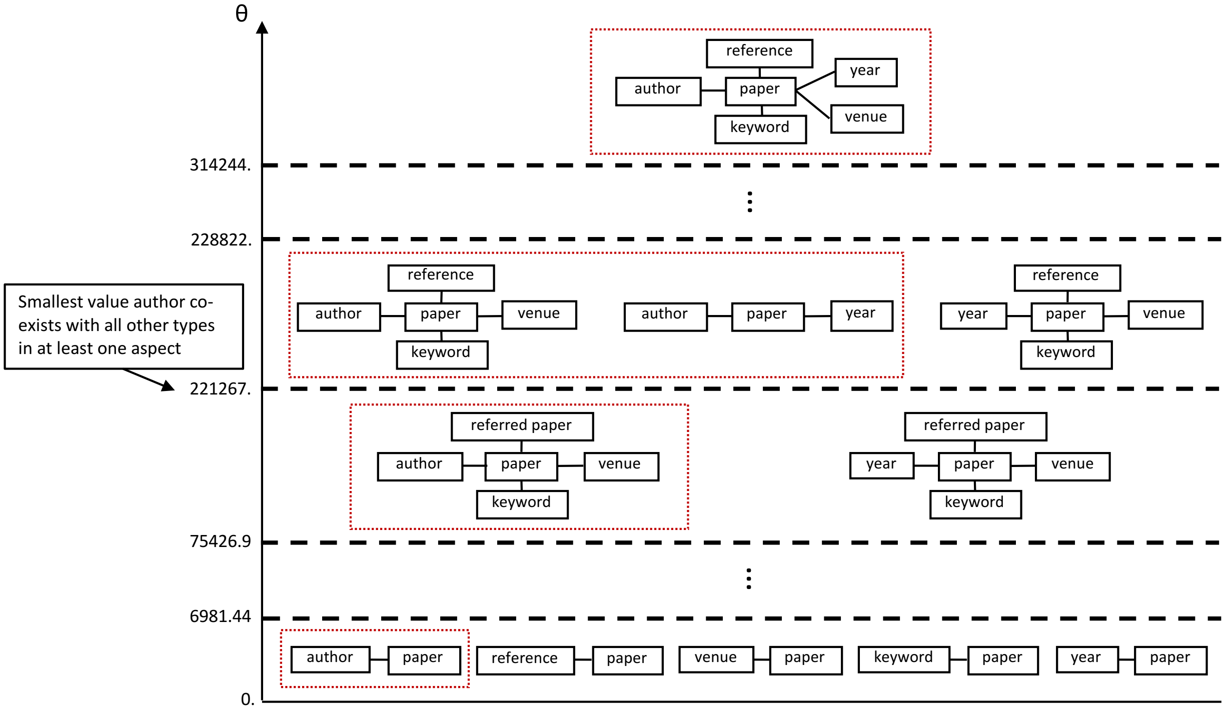


Figure 1: Aspects in DBLP satisfying the two requirements at various threshold θ.