Embedding Learning with Events in Heterogeneous Information Networks

Huan Gui, Jialu Liu, Fangbo Tao, Meng Jiang, Brandon Norick, Lance Kaplan, *Fellow, IEEE*, and Jiawei Han, *Fellow, IEEE*

Abstract—In real-world applications, objects of multiple types are interconnected, forming *Heterogeneous Information Networks*. In such heterogeneous information networks, we make the key observation that many interactions happen due to some *event* and the objects in each event form a complete semantic unit. By taking advantage of such a property, we propose a generic framework called ***H**yper**E**dge-**B**ased **E**mbedding* (HEBE) to learn object embeddings with events in heterogeneous information networks, where a *hyperedge* encompasses the objects participating in one event. The HEBE framework models the proximity among objects in each event with two methods: (1) predicting a target object given other participating objects in the event, and (2) predicting if the event can be observed given all the participating objects. Since each hyperedge encapsulates more information of a given event, HEBE is robust to data sparseness and noise. In addition, HEBE is scalable when the data size spirals. Extensive experiments on large-scale real-world datasets show the efficacy and robustness of the proposed framework.

Index Terms—Heterogeneous information networks, event, object embedding, large scale, noise pairwise ranking

1 Introduction

Learning objects embeddings is to represent each object using a low-dimensional vector. It is an important task in unsupervised learning and in data preprocessing of supervised learning. The low-dimensional vectors, as distributed representations of the relationships among objects, are beneficial for various downstream applications, such as exploratory data analysis, link prediction [1], visualization [2], object clustering [3], classification [4], and recommendation [5]. The objective of embedding techniques is mainly to preserve certain relationships among objects [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22].

Interactions among individual components or agents, such as friendships in social sites, hyperlinks on webpages, word co-occurrences in corpora and citations in bibliographical data, are ubiquitous in real-world applications. Embedding on single-typed interactions (e.g., word co-occurrences, friendships) has been studied extensively. Taking word co-occurrences as an example, given a corpus, there is an interaction between two words if one word (as target) appears near the other word (as context) within a snippet, such as a sentence. The proximity of the interaction between the two words can be modeled as the conditional probability of predicting the observed target word given the context word [7], where the conditional probability is estimated via softmax function, with low-dimensional vectors of words as parameters. This model has also been generalized to network data, such as [8], [11], [23].

On the other hand, recent years have witnessed an increasing interest on studying interactions among *strongly-typed objects*, which form *Heterogeneous Information Networks* (HINs) [24]. Bibliographical data is one such example, the interactions among objects include *authors* writing *paper*, *paper* being published in *venue*, and *paper* containing *terms* as content. Among these strongly-typed interactions, a key observation is that many interactions happen due to some event. For instance, when a paper is published, the interactions among corresponding paper, author(s), venue and terms *forms a complete semantic unit* in the bibliographical HIN. Consequently, we define the publication of a paper as an *event*. Moreover, we represent each event using a *hyperedge*, which encapsulates all the participating objects in the event. Since all the interactions in one event share semantic implications, we take advantage of such a property and attempt the problem of object embedding learning with events in HINs.

Embedding learning with strongly-typed interactions has broad real-world applications [9], [12], [50]. There are different approaches to computing embeddings. However, existing approaches do not take advantage of such event simultaneity property. We use DBLP as an illustration example.

Example 1.1. DBLP (http://dblp.uni-trier.de) is a CS bibliographical information network, where each publication record corresponds to an event. There are three types of participating objects: authors (A), terms (T), and venue (V), with their interactions represented at the schema level as shown in Fig. 1(left). To learn object embeddings, we need to preserve the proximity among all the participating objects (Fig. 1, top right). Previous studies (e.g., [9], [12])

- *H. Gui, F. Tao, M. Jiang, B. Norick, and J. Han are with the University of Illinois at Urbana-Champaign, Urbana, IL 61801. E-mail: {huangui2, ftao2, mjiang89, bnorick, hanj}@illinois.edu.*
- *J. Liu is with Google Research, New York, NY 10011. E-mail: jialu@google.com.*
- *L. Kaplan is with the U.S. Army Research Laboratory, Adelphi, MD 20783. E-mail: lance.m.kaplan.civ@mail.mil.*

Manuscript received 2 Feb. 2017; revised 6 July 2017; accepted 13 July 2017. Date of publication 31 July 2017; date of current version 4 Oct. 2017. (Corresponding author: Huan Gui.) Recommended for acceptance by P. Cui. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TKDE.2017.2733530

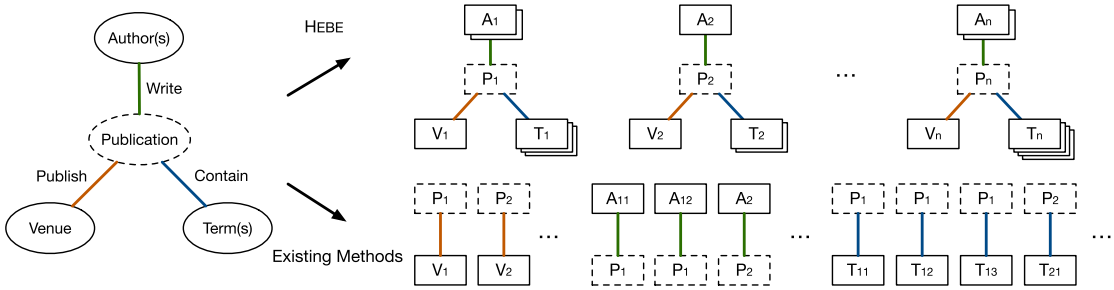


Fig. 1. The interaction schema of DBLP is in the left. In a publication event, the interactions of authors-publication, venue-publication, and terms-publication form a complete semantic unit. Existing methods (in the bottom right) consider each interaction type independently. Our method (in the top right) defines the set of interactions resulted from the same event as a hyperedge, and model each hyperedge as a whole.

decompose the interaction among all participating objects in each event into several scattered pairwise interactions (e.g., Author-Paper, Venue-Paper), as shown in Fig. 1 (bottom right). Object embeddings are then learned by combing embedding learning procedures upon each set of pairwise interactions, using existing embedding learning methodologies developed for single-typed network data. However, such pairwise interactions may miss some important information. Consider Einstein and Hawking may publish in the same venue, using similar terms in astrophysics, but they did not coauthor a paper. Pairwise modeling cannot capture such subtle differences.

In this paper, we propose a generic framework called **H**yper**E**dge **B**ased **E**mbedding (HEBE) that captures multiple interactions as a whole, which is illustrated in the top right of Fig. 1. Inspired from classical hypergraph theory [25] on hyperedges, we define a *hyperedge* as a set of objects forming a consistent and complete semantic unit. It is worth noting that the hyperedge defined here is more general than in classical hypergraph theory since the participating objects might be of different types. For each event, we model the proximity of the interaction among all participating objects in the hyperedge as a whole. Hyperedges provide us with a more complete description of events, therefore our methods preserves more contextual information for embedding learning in HINs.

The hyperedge model encapsulates more information, but also imposes challenges on modeling and optimization. Since interactions with multiple participating objects are modeled as a whole, existing methods cannot be straightforwardly applied. Instead, we propose two methods based on different prediction semantics to model the proximity of each event. The first method HEBE-**P**redict **O**bject (HEBE-PO) is to predict *if a participating object (as target) would be observed in an event given all the other participating objects*. This method is based on the observation that all the participating objects in an event share semantic similarity. Our second HEBE-**P**redict Hyper**E**dge (HEBE-PE) is to predict *if the hyperedge can be observed given all the participating objects*. Similar to HEBE-PO, HEBE-PE is based on the observation of semantic similarity among participating objects. Moreover, HEBE-PE additionally learn the embedding for the event, which provides further guidance on the semantic meaning of the whole event. The embeddings for the events serve as summarization of the event and can filter out noise information in the event observations.

In addition, it is essential for HEBE to be scalable on big data. We leverage recent advancement of asynchronous stochastic optimization [26] to take advantage of the parameter sparsity in embedding learning. Furthermore, we devise a new optimization technique, called *Noise Pairwise Ranking*, on the conditional probability of prediction. In sharp comparison with the existing methods, our method is free of negative sampling hyperparameter [7], [11], [27].

In HEBE, each hyperedge encapsulates more contextual information, leading to more informative and efficient updates. Consequently, HEBE is more robust to data sparseness and noise. We apply HEBE to large-scale HINs to learn object embeddings and measure the quality of the learned embeddings by various classification tasks. We observe that HEBE produces embeddings with better classification accuracy results while being robust to data sparseness and noise.

In sum, the study makes the following contributions:

1) It proposes the problem of learning embeddings for HINs using hyperedges, especially interactions with consistent and complete semantics are modeled as a whole, i.e., an event.
2) A new embedding framework HEBE is established, with two methods (HEBE-PO and HEBE-PE) further proposed to model the proximity among participating objects in each event.
3) A new method *Noise Pairwise Ranking* is developed to optimize the conditional probability based on ranking.
4) Extensive numerical experiments are conducted to demonstrate the effectiveness and robustness of HEBE.

2 PRELIMINARIES

In this section, we define the problem of embedding learning with events in heterogeneous information networks and introduce several related concepts and necessary notations.

2.1 Heterogeneous Information Networks and Events

We first define HIN and events in HINs.

Definition 2.1 (Information Networks). *Given a set of objects belonging to T types $\mathcal{X} = \{X_t\}_{t=1}^{T}$, where X_t represents the set of distinct objects with tth type, a network $G = \langle \mathcal{X}, \mathcal{E} \rangle$ is called* an information network on objects $\mathcal{X}$, *where $\mathcal{E}$ is a set of binary interactions of objects in $\mathcal{X}$. Specifically, such an information network is called a* heterogeneous information network (HIN) *if $T \geq 2$; and* homogeneous information network *if $T = 1$.*

Definition 2.2 (Events). *For a set of objects $V_i \subset \mathcal{X}$, if interactions among objects in V_i form a consistent and complete semantic unit, we define it as an event Q_i and represent the*

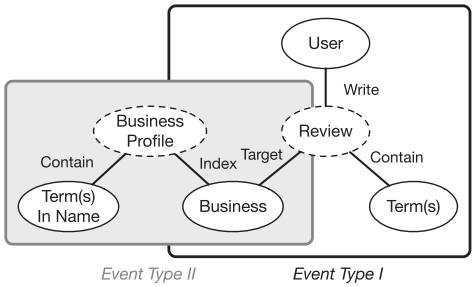


Fig. 2. Event schema of Yelp with two event types, business profile (left) and review (right).

event by a triplet $Q_i = \langle q_i, V_i, \omega_i \rangle$, where q_i is the event identifier serving as the index of the event, $V_i \subseteq \mathcal{X}$ is the set of participating objects, and ω_i is the weight of event Q_i (e.g., *the number of occurrences of this event).*

For each dataset, the semantic unit can be specified by the users. For example, in the DBLP dataset, the semantic unit is defined as the publication of each paper, with the additional information of the authors, terms, and venues.

In later sections we will slightly abuse notation and use X_t to indicate both the set of objects belonging to tth type and the name of the type as well. Besides multiple object types, we also study the general case with multiple event types where each event type is associated with an *event schema*, which is a tool to visualize relationships among the event and participating objects. The network schema of DBLP of Example 1.1 is shown on the left of Fig. 1 with one event type. Yelp data described in the following example contain two event types. Event identifiers are marked in dashed circles.

Example 2.3. Yelp (http://www.yelp.com/) is an online website for users to review various businesses, which can be naturally represented as a HIN. As shown in Fig. 2, there are two types of events. The first event type (left) is business profile, the participating object types of which include Terms in Name and Business; The second (right) is the review event, including User, Business, and Term types. The business objects type participates in both event types.

2.2 Learning Object Embedding

Given a HIN and the event schema, embedding algorithms learn to represent each object of different types using a low-dimensional vector in the same space. The embedding algorithms are to preserve the semantic similarity among objects as well as event topological structures such that objects that are semantically similar and co-occur in the same event will be close in the space, with the distance measured by cosine similarity, for instance.

Object embedding learning for HINs has broad applications [9], [12]. A straightforward method would ignore the object types and learn the object embeddings using network models, such as LINE [11]. In existing studies, heterogeneous event data are modeled as a HIN, in which the objects are of multiple types and the relations between objects are also of multiple types [24]. Chang et al. [9], Tang et al. [12] decompose each event into multiple *binary* relations between objects. Object embeddings are thus learned based on all sets of binary relations independently in a joint optimization framework. However, as we discussed above, such a decomposition method may lose some subtle information within the HINs.

Instead of simply considering each event as a set of independent binary relations between individual participating objects, we define a new structure to encapsulate all the information based on the events. Inspired from classical analysis on hypergraphs and hyperedges [25], [28], for each event Q_i, we use a corresponding *hyperedge* H_i to model the event by viewing all the participating objects as a whole in the hyperedge, i.e., H_i with q_i as its identifier connects the set of objects V_i with edge (event) weight ω_i. When the number of participating objects in each event is two, our model reduces to classical network model with binary interactions. Therefore, each event corresponds to one binary interaction.

In order to model the semantic similarity among participating objects in each event, which consequently preserves the topological structures of events, we propose two methods based on different semantics of prediction. The first insight is that semantically related objects are more likely to participate in the same event. For instance, in the DBLP data, it is more frequently to observe publications with author "Christos Faloutsos" and term "Network" in the venue ICDM. Therefore, we define the object-driven proximity based on the prediction of participating object observation given the other participating objects as context.

Definition 2.4. *The* Object-driven proximity *of an event is defined as the likelihood of observing a target object given all other participating objects on the same hyperedge corresponding to an event.*

Based on Object-Driven Proximity, we aim at predicting a target object. Therefore, the corresponding embedding algorithm is called HEBE **P**redict **O**bject (HEBE-PO) where HEBE stands for hyperedge-based embedding.

Since the HEBE-PO approach considers only the semantic proximity among participating objects on the same event (i.e., hyperedge), we further take the event itself into consideration. The second approach therefore is to predict the event given the set of participating objects. In other words, we additionally assign embeddings to each hyperedge (through event identifier) so that the proximity is well-defined, as follows.

Definition 2.5. *The* hyperedge-driven proximity *is defined as the likelihood of observing a hyperedge given all the participating objects.*

For example, given the set of author (Christos Faloutsos), term (Network) and venue (ICDM), we additionally learn the embedding of the publication event record, and connect the set of participating objects with the right publication record as an observed hyperedge. The corresponding embedding algorithm is called HEBE **P**redict Hyper**E**dge (HEBE-PE). The underlying intuition of HEBE-PO is that the semantic meanings of all participating objects are close to the semantic meaning of the event. Particularly, the event embedding serves as summarization of all the participating objects, which can effectively filter out noise information within each event if exists.

Based on the two kinds of proximity that preserves the event structures as defined above, we formally define the task of object embedding as follows.

Definition 2.6 (Object Embedding for HIN). *Given a hin, represented as a collection of events $\mathcal{D} = \{Q_i\}$, and the event schema,* object embedding *is to learn a function $\mathcal{M}$ that projects each object to a vector in a d-dimension space $\mathbb{R}^d$ that keeps*

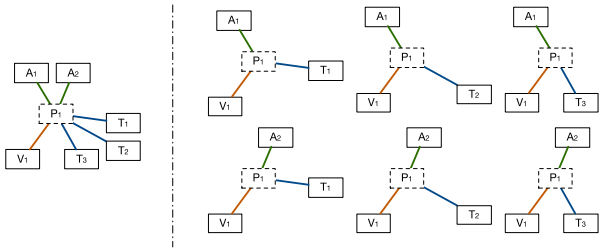


Fig. 3. Illustration of SubEvent sampling. The event (on the left) has six subevents (on the right). The weight of each subevent is the same regarding the event.

either object-driven or hyperedge-driven proximity, where $d \ll |\mathcal{X}|$*, i.e.,* $\mathcal{M} : \mathcal{X} \to \mathbb{R}^d$*, where* $\mathcal{X}$ *is the set of all objects.*

3 HEBE FRAMEWORK

In this section, we introduce the HEBE framework to learn the object embeddings. The major difficulty that lies in embedding learning in heterogeneous event data is how to model and optimize the proximity among participating objects in each event. We will provide the details of estimating the proximity as introduced in Section 2, and the optimization of which will be discussed in Section 4.

3.1 SubEvent Sampling

Before diving into the object learning, we first introduce a concept of *SubEvent Sampling* to simplify the representation of heterogeneous events. Recall that for an event Q_i, we represent it as $\langle q_i, V_i, \omega_i \rangle$, where $V_i \subset \mathcal{X}$. Particularly, we denote $V_i^t \subset V_i$ as the set of objects in type X_t.

In real world scenario, we have that $|V_i^t| \geq 1$. For the case when $|V_i^t| > 1$, we need to additionally include the multiple objects in each type. For instance, in DBLP (as discussed in Example 1.1), for each publication event, we have only one venue but multiple authors and terms. Also, in Yelp (Example 2.3), each review event is about one business but the review text may contain more than one hundred terms. Therefore, there is a count imbalance problem for $|V_i^t|$ with $t = 1, \ldots, T$.

To address the imbalance problem, we propose to sample subevent from each event by uniformly sampling one object from each object type. For instance, given an event of $Q_i = \langle q_i, V_i, \omega_i \rangle$, we have $V_i = \{a_1, a_2\} \cup \{t_1, t_2, t_3\} \cup \{v_1\}$ (where $a.$, $t.$ and $v.$ stand for author, term, and venue objects, respective, as shown in Fig. 3), we can sample a subevent $Q_{i,s} = \{a_2, t_2, v_1\}$ with probability of $1/(2 \times 3)$, consequently, we assign the weight for $Q_{i,s}$ as $\omega_i/(2 \times 3)$.

For a more general case of Q_i, we can sample $S_i = \prod_{t=1}^{T} |V_i^t|$ subevents, with the weight of each subevent as ω_i/S_i. Notably, we can see for each object $v_{i,t} \in V_i^t$, the aggregated weight for the object is $\omega_i/|V_i^t|$, which naturally balances the number of objects in each type. In other words, the more objects are in one type, the less important each object of that type is, vice versa.

We denote $\widetilde{Q}_{i'} = Q_{i,s}$ for $i = 1, \ldots, n$, $s = 1, \ldots, S_i$ with i' obtained sequentially, as one subevent w.r.t. the event Q_i. We flatten the subevent structures within events by defining $\widetilde{\mathcal{Q}} = \{\widetilde{Q}_{i'} = \langle \widetilde{q}_{i'}, \widetilde{V}_{i'}, \widetilde{\omega}_{i'} \rangle\}_{i'=1}^{N}$ with $N = \sum_{i=1}^{n} \prod_{t=1}^{T} |V_i^t|$, where the weight of $\widetilde{Q}_{i'}$ is $\widetilde{\omega}_{i'} = \omega_i/S_i$, since $\widetilde{Q}_{i'}$ is a subevent sampled from Q_i. It is worth noting that different events could generate the same subevent. For instance, both $Q_1 = \{a_1, a_2\} \cup \{t_1, t_2, t_3\} \cup \{v_1\}$ and $Q_2 = \{a_1, a_3\} \cup \{t_1, t_4, t_5\} \cup \{v_1\}$ have $\widetilde{Q}_1 = \{a_1, t_1, v_1\}$ as a subevent.

For implementation, we do subevent sampling on the fly. Suppose event Q_i is sampled with probability proportional to ω_i, we randomly sample one object from each object type and obtain a subevent $\widetilde{Q}_{i'}$. Specifically, the probability of a subevent $\widetilde{Q}_{i'}$ being sampled is propotional to ω_i/S_i as desired. The probability of subevents from more than one event can be aggregated accordingly.

Thereafter, for the following analysis we use event and subevent interchangably and we drop $\widetilde{\ }$ for $\widetilde{Q}, \widetilde{q}, \widetilde{V}, \widetilde{\omega}$ whenever the context is clear. We can also use hyperedges to encapsulate subevent topological structures. Without loss of generality, we assume there are no duplicated subevents $\widetilde{Q}_{i'}$'s and the weight $\widetilde{\omega}_{i'}$ for each subevent $\widetilde{Q}_{i'}$ has been aggregated appropriately.

3.2 HEBE Object Prediction

Our first method is HEBE-PO, which is to predict a target object out of all alternative objects given the other participating objects on the same hyperedge as context. Due to the heterogeneity of the objects, we constrain that the alternative objects are of the same type as the target object. When the target object type X_t is given, the corresponding target object in each subevent is accordingly $u_t \in V_i^t$ where $|V_i^t| = 1$ for each subevent. Without loss of generality, we further assume the target object is of type X_1. We denote the target object as u, context object set as C. Obviously, $|C| = T - 1$ for subevents and $u \notin C$. The conditional probability of predicting the target object u is defined as

$$\mathbb{P}_o(u|C) = \frac{\exp\big(S(u, C)\big)}{\sum_{v \in X_1} \exp\big(S(v, C)\big)}, \tag{3.1}$$

where $S(\cdot)$ is a scoring function reflecting the similarity between target object u and context objects C. Intuitively, (3.1) can be understood as given C selecting u from the pool of candidates X_1. Suppose $C = \{c_2, c_3, \ldots, c_T\}$, we have the scoring function as

$$S(u, C) = \left\langle \mathbf{w}_u, (T-1)^{-1} \sum_{t=2}^{T} \mathbf{w}_{c_t} \right\rangle, \tag{3.2}$$

where $\mathbf{w}_u \in \mathbb{R}^d$ is the embeddings of u.

We remark that the choice of scoring function is a free parameter and can be altered based on specific applications. The HEBE framework does not depend on the choice of the scoring function.

Objective. Suppose the target object type t. To preserve the object-driven proximity, we can naturally minimize Kullback-Leibler (KL) divergence between model distribution $\mathbb{P}_o(\cdot|C_t)$ and empirical distribution $\widehat{\mathbb{P}}_o(\cdot|C_t)$ where C_t is an arbitrary choice of context [11]. We additionally define $\mathcal{P}_t$ as the sample space of $C_{i,t}$ for $i = 1, \ldots, N$, such that $\mathcal{P}_t = \{C_{i,t}\}_{i=1}^{N}$ is the collection of possible values of context C in the empirical observations of $\mathcal{Q}$. Therefore, the objective function is

$$\mathcal{L}_o = -\sum_{t=1}^{T} \sum_{C_t \in \mathcal{P}_t} \lambda_{C_t} \mathrm{KL}\Big(\widehat{\mathbb{P}}_o(\cdot|C_t), \mathbb{P}_o(\cdot|C_t)\Big),$$

where we use λ_{C_t} is the importance of the context C_t, $\lambda_{C_t} = \sum_{i=1}^{N} \omega_i \mathbf{I}_{\{C_t = V_i \setminus u_{i,t}\}}$, where $u_{i,t} \in V_i$ is the target object in type X_t and $\mathbf{I}_{\{\cdot\}}$ is a binary indicator function.

Note that we assign the same weight to each object type. The model can be further extended to distinguish the relative importance for different types, as follows:

$$\mathcal{L}'_o = -\sum_{t=1}^{T} \gamma_t \cdot \sum_{C_t \in \mathcal{P}_t} \lambda_{C_t} \mathrm{KL}\Big(\widehat{\mathbb{P}}_o(\cdot|C_t), \mathbb{P}_o(\cdot|C_t)\Big),$$

with γ_t as importance parameter of object type X_t. For simplicity, we set $\gamma_1 = \cdots = \gamma_T = 1$. We leave the more general cases with different γ_t's for future work.

Lemma 3.1. *Maximizing $\mathcal{L}_o$ is equivalent to maximizing*

$$L_o = \sum_{t=1}^{T} \sum_{i=1}^{N} \omega_i \log \mathbb{P}_o(u_{i,t}|V_i \setminus u_{i,t}). \tag{3.3}$$

Proof.

$$\begin{aligned}
\mathcal{L}_o &= -\sum_{t=1}^{T} \sum_{C_t \in \mathcal{P}_t} \lambda_{C_t} \mathrm{KL}\Big(\widehat{\mathbb{P}}_o(\cdot|C_t), \mathbb{P}_o(\cdot|C_t)\Big) \\
&= -\sum_{t=1}^{T} \sum_{C_t \in \mathcal{P}_t} \sum_{i=1}^{N} \omega_i \mathbf{I}_{\{C_t = V_i \setminus u_{i,t}\}} \\
&\qquad \cdot \sum_{i=1}^{N} \widehat{\mathbb{P}}_o(u_{i,t}|C_t) \log \frac{\widehat{\mathbb{P}}_o(u_{i,t}|C_t)}{\mathbb{P}_o(u_{i,t}|C_t)} \\
&= -\widehat{C}_o + \sum_{t=1}^{T} \sum_{i=1}^{N} \omega_i \log \mathbb{P}_o(u_{i,t}|V_i \setminus u_{i,t}),
\end{aligned}$$

where

$$\widehat{C}_o = \sum_{t=1}^{T} \sum_{C_t \in \mathcal{P}_t} \sum_{i=1}^{N} \omega_i \mathbf{I}_{\{C_t = V_i \setminus u_{i,t}\}} \cdot \widehat{\mathbb{P}}_o(u_{i,t}|C_t) \log \widehat{\mathbb{P}}_o(u_{i,t}|C_t),$$

is a constant and the last equation follows from the fact that

$$\widehat{\mathbb{P}}_o(u_{i,t}|C_t) = \frac{w_i}{\sum_{i'=1}^{N} \omega_{i'} \mathbf{I}_{\{C_t = V_{i'} \setminus u_{i',t}\}}}.$$

The proof completes. □

Since $\mathbb{P}_o(u_{i,t}|C_{i,t})$ is the probability of observing a subevent of with participating objects V_i with weight ω_i, by Lemma 3.1, we have that minimizing the KL divergence is equivalent to maximium likelihood estimation.

3.3 HEBE Hyperedge Prediction

Recall that the hyperedge-driven proximity is to predict the hyperedge (through event identifier) given the set of participating objects. Therefore, we need to estimate the corresponding scoring function between the hyperedge (event identifier) and the set of objects. Desirably, the new scoring function $S(q_i, V_i)$ for event Q_i measures the similarity between the participating object set and the event identifier. The scoring function is defined as follows:

$$S(q_i, V_i) = \left\langle \mathbf{h}_i, T^{-1} \sum_{t=1}^{T} \mathbf{w}_{v_{i,t}} \right\rangle, \tag{3.4}$$

where the embedding of the event identifier is $\mathbf{h}_i$ and $\mathbf{w}_{v_{i,t}}$ is the embedding for object $v_{i,t} \in V_i^t$. Note that in HEBE-PE the context is the set of participating objects. Therefore, the condition probability of predicting the subevent identifier in the hyperedge is as follows:

$$\mathbb{P}_e(q_i|V_i) = \frac{\exp(S(q_i, V_i))}{\sum_{q_j \in \mathcal{Q}} \exp(S(q_j, V_i))}, \tag{3.5}$$

where $\mathcal{Q}$ is the set of all event identifiers. Similarly, $\mathbb{P}_e(q_i|V_i)$ corresponds to given V_i selecting q_i from the pool of candidates $\mathcal{Q}$. To preserve the topological structure of events, we minimize the KL divergence between the empirical distribution of $\widehat{\mathbb{P}}_e(\cdot|V)$ and $\mathbb{P}_e(\cdot|V)$, i.e.,

$$\mathcal{L}_e = -\sum_{V \in \mathcal{V}} \lambda'_V \mathrm{KL}\Big(\widehat{\mathbb{P}}_e(\cdot|V), \mathbb{P}_e(\cdot|V)\Big),$$

where V is a set of participating objects, $\mathcal{V}$ is the sample space of V, and λ'_V is the importance of V, $\lambda'_V = \sum_{i=1}^{N} \omega_i \mathbf{I}_{\{V = V_i\}}$. λ'_V is weighted number of hyperedges connecting with participating objects of V. Since we assume there is no duplicated V_i, for subevent q_i, $\lambda'_{V_i} = \omega_i$.

Lemma 3.2. *Maximizing $\mathcal{L}_e$ is equivalent to maximizing*

$$L_e = \textstyle\sum_{i=1}^{N} \omega_i \mathbb{P}_e(q_i|V_i). \tag{3.6}$$

Proof.

$$\begin{aligned}
\mathcal{L}_e &= -\sum_{V \in \mathcal{V}} \lambda'_V \mathrm{KL}\Big(\widehat{\mathbb{P}}_e(\cdot|V), \mathbb{P}_e(\cdot|V)\Big) \\
&= -\sum_{V \in \mathcal{V}} \sum_{i=1}^{n} \omega_i \mathbf{I}_{\{V = V_i\}} \sum_{i=1}^{n} \widehat{\mathbb{P}}_e(q_i|V) \log \frac{\widehat{\mathbb{P}}_e(q_i|V)}{\mathbb{P}_e(q_i|V)} \\
&= -\widehat{C}_e + \sum_{V \in \mathcal{V}} \sum_{i=1}^{n} \omega_i \widehat{\mathbb{P}}_e(q_i|V_i) \log \mathbb{P}_e(q_i|V_i),
\end{aligned} \tag{3.7}$$

where

$$\widehat{C}_e = \sum_{i=1}^{n} \omega_i \log \widehat{\mathbb{P}}_e(q_i|V_i) \widehat{\mathbb{P}}_e(q_i|V_i),$$

is a constant and the last equation follows from the fact that

$$\widehat{\mathbb{P}}_e(q_i|V) = \frac{\omega_i}{\sum_{i'=1}^{N} \omega_{i'} \mathbf{I}_{\{V = V_{i'}\}}}.$$

The proof completes. □

Since $\mathbb{P}_e(q_i|V_i)$ is the probability of observing a subevent of with participating objects V_i in subevent q_i with weight ω_i, by Lemma 3.2, we have the minimizing the KL divergence is equivalent to maximium likelihood estimation.

3.4 Multiple Event Types

In this section, we consider a more general case, when there are multiple event types, such as Example 2.3. As depicted in Fig. 2, there are two event types in the event schema, i.e., business profile event type and review event type. Assume there are K event types, considering HEBE-PO, for each event type, we have the objective function as $\mathcal{L}_o^k$ for $k = 1, \ldots, K$.

We treat each event type as equally important. Therefore, the overall objective function to be minimized is

$$\mathcal{L}_o^* = \sum_{k=1}^{K} \mathcal{L}_o^k.$$

Similar analysis can be directly applied to HEBE-PE.

4 OPTIMIZATION

We introduce a novel general optimization procedure, called *Noise Pairwise Ranking (NPR)*, for the HEBE framework.

4.1 Noise Pairwise Ranking

Considering the objective function of HEBE-PO in (3.3), direct optimizing $\mathcal{L}_o$ is intractable since the conditional probability (3.1) requires the summation over the entire set of objects of type X_1. The similar challenge exists for HEBE-PE in (3.6), which sums over the entire set of event identifiers. In the real world applications, the size of objects and event identifiers can be tremendous.

To address this challenge, noise contrastive estimation (NCE) [27] and negative sampling (NEG) [7] are proposed. NCE reduces the problem of estimating the conditional probability into a probabilistic classification problem to distinguish samples from the empirical distribution and a noise distribution, where the empirical distribution corresponds to positive samples and the noise distribution corresponds to negative samples. Moreover, based on NCE, [7] introduces negative sampling. Negative sampling also learns the parameters as a binary classification problem, it particularly formulates the objective as logistic regression, which is shown to be effective in embedding learning [7], [8], [11].

Here, we develop a new optimization framework, called *noise pairwise ranking* (NPR), from a pairwise ranking perspective. To illustrate the underlying idea, the conditional probability to be maximized can be abstracted as follows:

$$\mathbb{P}(u|C) = \frac{\exp\big(S(u,C)\big)}{\sum_{u' \in \mathcal{U}} \exp\big(S(u',C)\big)}, \tag{4.1}$$

where $\mathcal{U}$ is the set of targets (which can be instantiated to objects or event identifiers). For HEBE-PO, $\mathcal{U}$ corresponds to X_t where t is the type of the target object; while for HEBE-PE, $\mathcal{U}$ corresponds to $\{q_i\}_{i=1}^N$. Therefore, we have

$$\mathbb{P}(u|C) = \Big[1 + \sum\nolimits_{u' \in \mathcal{U} \setminus u} \exp\big(S(u',C) - S(u,C)\big)\Big]^{-1}, \tag{4.2}$$

which follows from (4.1) via dividing the denominator and numerator by $\exp\big(S(u,C)\big)$. Instead of directly optimizing (4.2) over all $u' \in \mathcal{U} \setminus u$, we sample an example u_n from $\mathcal{U} \setminus u$ as a negative sample; then we update (4.2) using u_n as proxy of $\mathcal{U} \setminus u$. W.r.t. sampling u_n from $\mathcal{U} \setminus u$, similar to NCE and NEG [7], [27], NPR also has a noise distribution P_n as a free parameter.

We denote the sigmoid function as $\sigma(x) = 1/(1+\exp(-x))$. In order to maximize the conditional probability defined in (4.1), we maximize the following noise pairwise ranking function [7] instead,

$$\mathbb{P}(u > u_n|C) = \sigma\big(-S(u_n,C) + S(u,C)\big), \tag{4.3}$$

which can be interpreted as maximizing the probability of observing the target u over the noise u_n, given the context C. Particularly, it can be verified as follows that

$$\mathbb{P}(u|C) > \prod_{u_n \neq u} \mathbb{P}(u > u_n|C),$$

which implies that optimizing $\mathbb{P}(u > u_n|C)$ can be explained as optimizing the lower bound of $\mathbb{P}(u|C)$. It is worth noting that the lower bound is not tight.

Remark 4.1. The derived noise pairwise ranking results in (4.3) is similar to the Bayesian Pairwise Ranking (BPR) proposed in [29]. However, BPR is designed for the personalized ranking in a specific recommender system with the negative samples coming from missing implicit feedback; while our NPR is derived based on approximation from the softmax definition of the conditional probability. Besides, the negative samples are sampled from a noise distribution.

Thus, for all $u_n \in \mathcal{U} \setminus u$, (4.1) can be approximated by

$$\log \mathbb{P}(u|C) \propto \mathbb{E}_{u_n \sim P_n} \log \mathbb{P}(u > u_n|C),$$

where P_n is the noise distribution. We set $P_n \propto d_u^{3/4}$, as proposed in [7], where d_u is the degree of u. For HEBE-PO, the degree d_u of each object is the number of hyperedges that object u involves in; while for HEBE-PE, the degree d_u is set to be the weight of the event (i.e., ω_u).

4.2 Optimization for HEBE-PO

Based on the NPR optimization framework proposed in Section 4.1, we apply it to the method of HEBE-PO in the HEBE framework. Recall that the optimization objective of HEBE-PO is defined in (3.3) with the conditional probability defined in (3.1). By applying the NPR optimization method, we have the following new objective function

$$\widetilde{\mathcal{L}}_o = -\sum_{i=1}^{N} \omega_i \sum_{t=1}^{T} \mathbb{E}_{u_{n,t} \sim P_n(X_t)} \mathbb{P}_o(u_{i,t} > u_{n,t}|C_{i,t}).$$

where $u_{n,t}$ is the sampled noise from $P_n(X_t)$ and the latter is the noise distribution of objects of type X_t. According to (4.3), we have

$$\mathbb{P}_o(u_{i,t} > u_{n,t}|C_{i,t}) = \sigma\big(-S(u_{n,t},C_{i,t}) + S(u_{i,t},C_{i,t})\big).$$

To optimize $\widetilde{\mathcal{L}}_o$, we use the asynchronous stochastic gradient algorithm (ASGD) [26]. ASGD takes advantage of the sparsity of the optimization problem, which means that most gradient updates only modify a small portion of the variables. Define $\mathbf{\Theta} = \{\mathbf{w}_v\}_{v \in \mathcal{X}}$ as the parameters, where $\mathbf{w}_v$ is the embedding for object v, we have the gradient

$$\frac{\partial \widetilde{\mathcal{L}}_o}{\partial \mathbf{\Theta}} = -\sum_{i=1}^{N} \omega_i \sum_{t=1}^{T} \mathbb{E}_{u_{n,t} \sim P_n(\mathcal{X}_t)} \frac{\partial \mathbb{P}_o(u_{i,t} > u_{n,t}|C_{i,t})}{\partial \mathbf{\Theta}}.$$

We define the shorthand notation $\mathbb{P}_o(>_{t,i,n}) = \mathbb{P}_o(u_{i,t} > u_{n,t}|C_{i,t})$. The gradients can be written as follows:

$$\frac{\partial \ln \mathbb{P}_o(>_{t,i,n})}{\partial \mathbf{w}_{u_{i,t}}} = \frac{\sigma(-S_\Delta)\sum_{t=2}^{T}\mathbf{w}_{c_t}}{T-1};$$
$$\frac{\partial \ln \mathbb{P}_o(>_{t,i,n})}{\partial \mathbf{w}_{u_{n,t}}} = -\frac{\sigma(-S_\Delta)\sum_{t=2}^{T}\mathbf{w}_{c_t}}{T-1}; \quad (4.4)$$
$$\frac{\partial \ln \mathbb{P}_o(>_{t,i,n})}{\partial \mathbf{w}_{c_t}} = \frac{\sigma(-S_\Delta)(\mathbf{w}_{u_{i,t}} - \mathbf{w}_{u_{n,t}})}{T-1}.$$

where $S_\Delta = -S(u_{n,t}, C_{i,t}) + S(u_{i,t}, C_{i,t})$.

Gradient Coefficient. Objects in types of smaller sizes are more likely to be sampled when we sample the events. For instance in the example of DBLP, the expected probability of a random venue being sampled is propotional to $1/|X_V|$; while for objects of types author and term, the expected probabilities of being sampled are $1/|X_A|$ and $1/|X_T|$, respectively. Since $|X_A| \gg |X_V|$ and $|X_T| \gg |X_V|$, the expected probability of each venue being sampled is higher than authors and terms. Similar observations can also be made in the Yelp network. This inevitably makes some object types better trained than others as optimization proceeds, resulting in the learned $\mathbf{\Theta}$ being trapped at poor local optima during the optimization procedure.

In order to balance the average step size among different object types, when applying ASGD to learn the object embeddings (and event identifier embeddings for HEBE-PE), we propose to adjust the global step size using a type-wise gradient coefficient. Suppose the global step size is η, given an object type t, the step size for each object in X_t is defined as $\beta_t = \alpha_t \eta$, where α_t is the gradient coefficient,

$$\alpha_t = \frac{|X_t|}{\max_{t'=1}^{T}\{|X_{t'}|\}}. \quad (4.5)$$

We define $\boldsymbol{\beta} = [\beta_t]_{t=1}^{T}$ as the vector of step size for each object type. By (4.5), we have that for object type t, the smaller $|X_t|$, the smaller α_t, corresponding to smaller step-size. Therefore, we slow down the training process for the objects of type t where $|X_t|$ is relatively smaller.

The updating process for a single iteration of HEBE-PO is summarized in Algorithm 1.

Algorithm 1. HEBE-PO($Q_i, \boldsymbol{\beta}$)

1: Sample a subevent $\widetilde{Q}$ from Q_i;
2: Uniformly sample an object type X_t from $\widetilde{Q}$;
3: Draw a random object from $P_n(X_t)$ as noise object;
4: Update Object Embeddings $\mathbf{\Theta}$ of $\widetilde{Q}$ by Gradient Descent (4.4) with type-wise step size $\boldsymbol{\beta}$;
5: **Return:** $\mathbf{\Theta}$

4.3 Optimization for HEBE-PE

Similarly, we apply the NPR optimization framework to HEBE-PE, which yields the new optimization objective of

$$\widetilde{\mathcal{L}}_e = -\sum_{i=1}^{N} \omega_i \mathbb{E}_{q_n \sim P_n(\mathcal{Q})} \mathbb{P}_e(q_i >_{} q_n | V_i),$$

where q_n is the sampled noise event from $P_n(\mathcal{Q})$. In addition, we have

$$\mathbb{P}_e(q_i >_{} q_n | V_i) = \sigma\big(-S(q_n, V_i) + S(q_i, V_i)\big).$$

It is worth noting that in HEBE-PE, we have event identifier embeddings ($\boldsymbol{H} = \{\mathbf{h}_i\}_{i=1}^{N}$) as parameters, in addition to object embeddings $\mathbf{\Theta}$. The gradient $\partial \mathbb{P}_e(>_{i,n})/\partial \mathbf{\Theta}$ can be obtained as follows, with $\mathbb{P}_e(>_{i,n}) = \mathbb{P}_e(q_i >_{} q_n | V_i)$,

$$\frac{\partial \ln \mathbb{P}_e(>_{i,n})}{\partial \mathbf{w}_{v_{i,t}}} = \frac{\sigma(-S_\Delta)(\mathbf{h}_i - \mathbf{h}_n)}{T}. \quad (4.6)$$

where $S_\Delta = -S(q_n, V_i) + S(q_i, V_i)$.

Additionally, we have $\partial \mathbb{P}_e(>_{i,n})/\partial \boldsymbol{H}$ as follows:

$$\frac{\partial \ln \mathbb{P}_e(>_{i,n})}{\partial \mathbf{h}_i} = \frac{\sigma(-S_\Delta)\sum_{t=1}^{T}\mathbf{w}_{v_{i,t}}}{T};$$
$$\frac{\partial \ln \mathbb{P}_e(>_{i,n})}{\partial \mathbf{h}_n} = -\frac{\sigma(-S_\Delta)\sum_{t=1}^{T}\mathbf{w}_{v_{i,t}}}{T}. \quad (4.7)$$

The corresponding updating process for a single iteration of HEBE-PE is presented in Algorithm 2.

Algorithm 2. HEBE-PE($Q_i, \boldsymbol{\beta}$)

1: Sample a subevent $\widetilde{Q}$ from Q_i;
2: Draw an event identifier from $P_n(\mathcal{Q})$ as negative;
3: Update Object Embeddings $\mathbf{\Theta}$ of $\widetilde{Q}$ by Gradient Descent (4.6) with type-wise step size $\boldsymbol{\beta}$;
4: Update Event Identifier Embeddings $\boldsymbol{H}$ of q_i by Gradient Descent (4.7) with type-wise step size $\boldsymbol{\beta}$.
5: **Return:** $\mathbf{\Theta}$, $\boldsymbol{H}$

4.4 Unified Algorithm

The optimization procedures for HEBE-PO and HEBE-PE introduced in the previous sections are applicable when there is only one event type. Here, we consider the more general scenario where there are multiple event types (i.e., $K > 1$). The unified algorithm is described in Algorithm 3, with η_0 and R as the initial step size and the iteration number. When learning embeddings for the objects (and the event identifiers) in the heterogeneous information networks, we opt to use a similar procedure to that used in [12], which is to use all event types jointly and weigh each event type equally. Accordingly, we adopt the strategy that first uniformly samples a event type and then sample a event instance of that type, as shown in Line 8.

5 EXPERIMENTAL STUDY

In this section, we report experimental results of the proposed two HEBE methods, including HEBE-PO and HEBE-PE. To evaluate how well the learned embeddings preserve the proximity between objects in HINs with events, we evaluate the embeddings using both classification and ranking measures. Particularly, via a series of quantitative studies, we aim at answering the following questions:

- Q1: W.r.t. classification tasks, do HEBE methods, including both HEBE-PO and HEBE-PE, learn better object embeddings compared with existing methods?
- Q2: Are HEBE methods robust to random noisy objects included in the event schemata?
- Q3: Are HEBE methods robust to data sparseness?
- Q4: W.r.t. classification tasks, when does HEBE-PO learn better embeddings than HEBE-PE, and vice versa?

TABLE 1
Number of Objects for DBLP and Yelp

DBLP	Author 209,679	Term 165,657	Venue 7,953	Paper 1,938,912
Yelp	Business 12,241	Term (review) 130,259	Term (name) 6,709	Review 905,658

Algorithm 3. HEBE

Initialize: randomly initialize $\boldsymbol{\Theta}$, $\boldsymbol{H}$
for $t = 1, \ldots, T$ **do**
  Calculate α_t via (4.5)
end for
for $i = 0$ to $I_N - 1$ **do**
  $\eta \leftarrow \eta_0 \cdot (I_N - i)/I_N$
  $\boldsymbol{\beta} \leftarrow \eta \cdot [\alpha_o]_{o \in \mathcal{O}}$
  for $k = 1, \ldots, K$ **do**
    Sample a event Q_i of event type k
    if method is HEBE-PO **then**
      $\boldsymbol{\Theta} \leftarrow$ HEBE-PO$(Q_i, \boldsymbol{\beta})$
    else if method is HEBE-PE **then**
      $\boldsymbol{\Theta}, \mathbf{H} \leftarrow$ HEBE-PE$(Q_i, \boldsymbol{\beta})$
    end if
  end for
end for
Return: $\boldsymbol{\Theta}, \mathbf{H}$

5.1 Datasets and Compared Methods

We introduce two datasets on which we conduct experiments: DBLP and Yelp, as of Examples 1.1 and 2.3. The basic statistics of both datasets are summarized in Table 1. The event schemas of DBLP and Yelp can be found in Figs. 1 and 2. In Yelp dataset, user type is removed in event type I due to data sparsity that the number of reviews written by each user is typically small. It is worth noting that we distinguish the terms in the review and terms in the business profile in event type II.

In order to demonstrate the efficacy of the two proposed methods, we use an extensive set of existing methods as baselines. For the sake of convenience, we define some notations before detailing the baselines. Recall that $\mathcal{X}$ is the set of objects and $\mathcal{D}$ is the set of events. We define the coocurrence matrix $\mathbf{M} \in \mathbb{R}^{|\mathcal{X}| \times |\mathcal{X}|}$ such that $\mathbf{M}_{i,j}$ denotes the number of events that two objects are both involved in. It is worth noting that by constructing the coocurrence matrix, we ignore the type information associated with each object. Due to the fact that some methods decompose the data into pairwise interactions, total degrees among different interactions may vary significantly and compromise the embeddings. For fair comparison, we therefore can first apply degree normalizions to these pairwise interaction sets and then merge them to get normalized coocurrence matrix $\widetilde{\mathbf{M}}$ as presented in [30]. The dimensionality of object embeddings is set to be 300 for all methods. In particular, we consider the following methods:

- Singular Value Decomposition (SVD) on $\mathbf{M}$, and singular vectors are used as object representations.
- Normalized SVD (NSVD) on $\widetilde{\mathbf{M}}$.
- Positive shifted PMI (PPMI). As shown in [31], the word embedding with negative sampling is equivalent to approximate the PPMI. Hence, we perform SVD on the PPMI matrix of $\mathbf{M}$. We have $k = 5$ as the negative sampling parameter.
- Non-negative Matrix Factorization (NMF) on $\mathbf{M}$, and matrix factor is used as object representation.
- Normalized NMF (NNMF) on $\widetilde{\mathbf{M}}$.
- LINE [11]: a second-order object embedding approach originally proposed for networked data. We apply LINE to the decomposed pairwise interactions directly, ignoring the object type information.
- PTE [12]: an object embedding approach that applies pairwise modeling in a round-robin fashion within each event, considering the type information.[1]

The implementation of HEBE can be found here (bitbucket.org/hgui/hebe).

5.2 Evaluation Metric

The goal of our experiments is to quantitatively evaluate how well our methods perform in generating proximity-preserved embeddings.

One way to evaluate the quality of the embeddings is through proximity-related object classification tasks. After obtaining the embeddings of the objects, we feed these embeddings into classifiers including linear SVM and logistic regression to perform classification with five-fold cross validation. Supposing $x \in \mathcal{X}$, we define l_x^* as the true label of x while $\widehat{l_x}$ as the predicted label of x. We report the classification metric accuracy (Acc.)

$$\text{Acc.} = |X_l|^{-1} \sum_{x \in X_l} \delta\big(\widehat{l_x} = l_x^*\big),$$

where X_l is the set of objects that have labels and $\delta(\cdot)$ is the indicator function. Due to the space limit, the higher accuracy of linear svm and logistic regression for each method gets reported.

Classification relies on ground truth labels to learn mapping function between embeddings and classes. It may not be able to exploit information underlying all dimensions. For instance, some dimensions may be independent of the class labels. Therefore we further use a ranking metric called area under the curve (AUC) [32] to evaluate the quality of embeddings over all dimensions

$$\text{AUC} = |X_l|^{-1} \mathbb{P}\big(\text{sim}(u, v) > \text{sim}(u', v) | l_v^* = l_u^*, l_v^* \neq l_{u'}^*\big),$$

where $v, u, u' \in X_l$ and $\text{sim}(u, v)$ is the similarity measure between the embeddings of objects u, v. Specifically, we use cosine similarity as the similarity measure [7]. The AUC measure becomes high if embeddings are close for objects sharing the same label, and distant for objects having different labels.

Regarding the DBLP dataset, we have two types of labels of authors. The first is on the *research groups*, with 116 members from four research group manually labelled. These groups are lead by Christos Faloutsos, Dan Roth, Jiawei Han, and Michael I. Jordan, respectively. The other type of labels is on the *research area*, including 4,040 researchers from four research areas including data mining, database, machine learning, and artificial intelligence.

1. In the original presentation of [12], labels are provided. For fair comparison, we do not provide labels for PTE during training.

TABLE 2
Classification Accuracy (%) and AUC on Two Datasets, Respecting Tasks of Research Group (DBLP), Research Area (DBLP), and Restaurant Categories (Yelp)

	Research Group		Research Area		Restaurant Type	
Method	Acc.	AUC	Acc.	AUC	Acc.	AUC
SVD	81.03	0.7137	83.27	0.5720	74.09	0.7147
NSVD	72.41	0.6958	89.75	0.6271	66.45	0.6244
PPMI	70.69	0.7513	90.22	0.7450	82.82	0.6504
NMF	73.28	0.6210	75.69	0.5798	79.64	0.7955
NNMF	72.41	0.7223	88.31	0.7665	72.00	0.7328
LINE	78.45	0.5607	79.48	0.5565	79.82	0.6378
PTE	**87.93**	0.7235	90.27	0.6646	81.91	0.7195
HEBE-PO	84.48	0.7957	**92.18**	0.7905	**88.00**	**0.8961**
HEBE-PE	87.07	**0.8207**	91.66	**0.8417**	87.27	0.8826

TABLE 3
Classification Accuracy (%) and AUC on Two Datasets with Extra *Noisy Object* Types ("Year" for DBLP and "Zipcode" for Yelp)

	Research Group		Research Area		Restaurant Type	
Method	Acc.	AUC	Acc.	AUC	Acc.	AUC
SVD	78.03	0.6846	80.10	0.5374	67.73	0.6902
NSVD	70.69	0.6668	87.48	0.6112	48.81	0.6138
PPMI	68.09	0.7175	88.99	0.7162	81.09	0.6892
NMF	72.73	0.6121	71.96	0.5635	67.00	0.7469
NNMF	71.38	0.6823	86.12	0.7411	43.45	0.6142
LINE	80.17	0.5465	78.94	0.5425	76.09	0.6035
PTE	85.34	0.6297	89.83	0.5873	75.18	0.6702
HEBE-PO	76.72	0.7582	89.11	0.7614	85.91	0.8296
HEBE-PE	**85.34**	**0.8214**	**91.26**	**0.8425**	**86.73**	**0.8834**

As for the Yelp dataset, we select eleven *restaurant categories* including Mexican, Chinese, Italian, American (traditional), American (new), Mediterranean, Thai, French, Japanese, Vietnamese and Indian as labels.[2] For each category, we randomly select 100 restaurants that have at least 50 reviews. Restaurants with multiple labels are excluded.

5.3 Experimental Results

Now we are ready to present the experimental results for the aforementioned tasks and try to answer the three questions raised at the beginning of this section.

5.3.1 Classification Results

Table 2 summarizes the experimental results on classification (Acc.) and ranking (AUC) in DBLP and Yelp.

Considering the results for research group classification in DBLP, we note that PTE and HEBE-PE achieve the best performance. PTE is slightly better than HEBE-PE on accuracy but the latter outperforms the former on AUC by a large margin. HEBE-PO narrowly loses to HEBE-PE on both measures. It is interesting to see that the normalization strategy on $\mathbf{M}$ has a big effect on the performance, but the trend is oppsite between SVD and NMF respecting AUC.

For the task of research area classification in DBLP, HEBE-PO attains the best performance on classification accuracy and HEBE-PE has the highest AUC score. The results on research area are better than the ones on research group for all methods, which means that the research area task is easier than the former task. It's worth noting that two HEBE methods are better than baselines on both measures, confirming the their effectiveness of capturing the proximity. We also observe that both NSVD and NNMF beat their unnormalized versions, implying that the normalization trick works at least for some tasks.

With respect to the Yelp dataset, on classifying the restaurant type, both HEBE methods are significantly better than the baselines, for both measures. A tentative explanation is that HEBE framework models both event types explicitly, the review event and the business profile event, which better captures the proximity among objects. For PTE and the rest methods, this intricate structure will be dropped due to the representation limits of the models.

2. The labels are obtained from www.yelp.com.

To summarize, we positively answer Q1 on the effectiveness of HEBE methods in learning the object embeddings. Among all the competitors, PTE works relatively well for all three tasks, showing its idea of modeling pairwise interactions better than the rest. Meanwhile, by modeling each event as a whole, HEBE achieves even better performance.

5.3.2 Robustness to Noisy Objects

One challenge of modelling events in HINs is to develop a method with anti-interference ability. Hence, we test the robustness of the HEBE framework against artificially inserted object noises. The added noisy objects are designed to convey little knowledge regarding the tasks on both datasets. Consequently, for DBLP data, we include the year of the publication as an additional object type. For Yelp data, the zip code of each restaurant is considered. The results are summarized in Table 3.

For all three tasks, HEBE-PE achieves the best performance and is better than the baselines by a large margin. In addition, HEBE-PO is bested by HEBE-PE for all three tasks, but attain results better than PTE and the rest methods in most cases. These observations verify our expectation that HEBE-PE is more robust to noise than all the rest methods including HEBE-PO. A possible explanation is that HEBE-PO explicitly models the proximity between the noisy object and the context objects, leading to deviation of the object embeddings from the optimal ones. In contrast, HEBE-PE additionally learns the event identifier embeddings. With respect to each event, the event identifier serves as a filter and summarizing the sematic information of all participating objects. Since the noise objects have low semantic similarity with the other participating objects, information related to the noise objects will be droped by the event identifier embedding. While learning embeddings for other objects, information is directly propagated from the event identifier to objects. Consequently, the object embedding learning will not be influenced by the noise objects. The experimental results across all three tasks, we have HEBE-PO achieves the best performance. Moreover, we can observe that by adding noise, HEBE-PO achieves good classification accuracy as when there are no noise objects. On the other hand, we still recognize that HEBE-PO is the second best method in terms of absolute performance.

TABLE 4
The Classification Accuracy and AUC Results on *Sampled* DBLP Data for Research Group and Research Area Classification

Sampling %.	1%		5%		10%		20%		30%		50%	
Density Measure	1.264		2.028		2.882		4.595		6.400		10.315	
Method	Acc.	AUC	Acc.	AUC	Acc.	AUC	Acc.	AUC	Acc.	AUC	Acc.	AUC
Research Group												
SVD	38.46	0.5602	66.67	0.6169	65.59	0.6481	72.55	0.6494	72.86	0.6720	77.28	0.6924
NSVD	43.59	0.5504	58.73	0.5919	68.82	0.6330	70.59	0.6345	72.64	0.6517	74.55	0.6790
PPMI	46.15	0.5502	60.32	0.5993	76.34	0.6557	71.57	0.6703	72.97	0.6792	74.55	0.7192
NMF	41.03	0.5583	57.14	0.5989	56.99	0.5874	54.90	0.6009	66.96	0.5950	70.91	0.6120
NNMF	46.15	0.5462	55.56	0.6601	60.22	0.6806	75.49	0.7167	70.55	0.7197	71.82	0.7294
LINE	**56.41**	0.6004	66.67	0.6254	72.04	0.5877	77.45	0.5619	77.86	0.5669	85.45	0.5871
PTE	**56.41**	0.6190	69.84	0.6727	76.34	0.6434	84.31	0.6778	**85.94**	0.7034	**88.18**	0.6783
HEBE-PO	53.85	0.6034	66.67	0.7082	72.04	0.7151	74.51	0.7515	75.55	0.7640	82.73	0.7841
HEBE-PE	**56.41**	**0.6547**	**73.02**	**0.7434**	**83.87**	**0.7749**	**85.29**	**0.8221**	84.13	**0.8220**	**88.18**	**0.8316**
Research Area												
SVD	47.88	0.5162	62.47	0.5337	66.27	0.5411	71.66	0.5516	75.47	0.5551	79.15	0.5644
NSVD	52.39	0.5076	66.21	0.5004	72.15	0.5021	77.91	0.5157	80.13	0.5299	85.23	0.5600
PPMI	51.67	0.5063	68.00	0.5092	72.66	0.5180	78.15	0.5395	80.59	0.5669	85.91	0.6203
NMF	43.37	0.5143	53.54	0.5329	59.30	0.5391	63.63	0.5493	68.01	0.5560	70.72	0.5637
NNMF	50.50	0.5303	62.50	0.5773	67.73	0.6206	72.37	0.6486	76.51	0.6807	82.91	0.7594
LINE	57.17	0.5552	69.83	0.5764	72.15	0.5716	74.89	0.5501	74.53	0.5339	80.82	0.5822
PTE	53.29	0.5291	**71.54**	0.5858	73.95	0.5782	79.03	0.6015	82.68	0.6356	86.80	0.6340
HEBE-PO	**57.53**	**0.5635**	69.71	0.6108	74.91	0.6798	80.26	0.7199	81.66	0.7293	86.17	0.7817
HEBE-PE	54.64	0.5500	71.09	**0.6282**	**75.90**	**0.6834**	**81.64**	**0.7405**	**83.94**	**0.7645**	**87.84**	**0.8075**

The sparsity is measured by the average number of publication each author is involved in.

5.3.3 Robustness to Sparsity

In general, the sparsity of event data is defined as the average number of events each object is involved in. Thus, if we assume the set of objects to be relatively stable, the sparsity of the heterogenous event data can be altered by sampling a subset of all events. In this section, we randomly sample different percentages (1, 5, 10, 20, 30, 50 percent) of the two datasets and repeat the three tasks mentioned aforehand. Experimental results are reported in Table 4 for the DBLP dataset and Table 5 for the Yelp dataset. The density measures are reported in the first two rows. For DBLP, since the classification is performed on authors, we define *density measure* as the number of publications each author is associated with. For Yelp, because the businesses are of interest, we define *density measure* as the number of reviews each restaurant receives. The density measure increases as the sampling percentage increases, and its incremental rate is slower than the latter due to the long-tail behavior in the event data. In other words, when more events are sampled, the size of the object set will also increase, but having a slower rate of increment.

Across the three tasks in the two datasets, vertically we observe the two HEBE methods achieve the best performance in general among all cases. In DBLP dataset, for both tasks, HEBE-PE is better than HEBE-PO for both measures in most cases. In Yelp dataset, when less than 20 percent of events are sampled, HEBE-PE attains better results than HEBE-PO; when more than 20 percent of events are sampled, HEBE-PO outperforms similar to HEBE-PE; across the different sampling percentages the margin between HEBE-PO and HEBE-PE is relatively small. For different percentages, we observe that PTE is still the most stable method among all baselines while the performances of the rest fluctuate wildly for different tasks. When the density measure is close to 1 such as 1 percent of events being sampled in the DBLP dataset, the

TABLE 5
The Classification Accuracy and AUC Results on *Sampled* Yelp Data

Sampling %.	1%		5%		10%		20%		30%		50%	
Density Measure	1.963		4.791		8.155		15.09		22.32		37.01	
Method	Acc.	AUC	Acc.	AUC	Acc.	AUC	Acc.	AUC	Acc.	AUC	Acc.	AUC
SVD	64.12	0.6133	70.85	0.6786	73.44	0.7001	73.98	0.7100	73.82	0.7121	74.82	0.7134
NSVD	62.07	0.6081	63.36	0.6236	65.17	0.6308	66.97	0.6275	67.00	0.6280	67.36	0.6259
PPMI	59.35	0.5561	65.01	0.5484	69.94	0.5626	75.43	0.5824	78.55	0.6089	80.55	0.6253
NMF	63.61	0.6790	71.23	0.7381	75.09	0.7594	76.34	0.7877	78.09	0.7907	78.18	0.7991
NNMF	60.71	0.6710	66.76	0.7022	68.47	0.7082	70.79	0.7213	70.73	0.7297	70.73	0.7312
LINE	60.88	0.5337	71.72	0.5367	77.32	0.5689	80.71	0.6665	80.91	0.6789	81.27	0.6833
PTE	64.29	0.6315	72.89	0.6758	76.28	0.6993	79.25	0.7163	81.00	0.7043	80.91	0.7266
HEBE-PO	71.09	0.7576	79.01	0.8316	82.63	0.8621	85.08	**0.8825**	**86.36**	**0.8845**	**86.82**	**0.8938**
HEBE-PE	**73.30**	**0.7747**	**79.69**	**0.8434**	**83.06**	**0.8746**	**85.44**	0.8779	85.82	0.8765	86.36	0.8862

TABLE 6
Considering the Optimization Algorithms, NPR and NEG, Classification Accuracy (%) and AUC

Method	Research Group		Research Area		Restaurant Type	
	Acc.	AUC	Acc.	AUC	Acc.	AUC
HEBE-PO	84.48	0.7957	92.18	0.7905	**88.00**	**0.8961**
HEBE-PE	87.07	**0.8207**	91.66	**0.8417**	87.27	0.8826
NEG-PO-1	86.05	0.7960	84.31	0.7587	87.82	0.8579
NEG-PO-5	87.46	0.7908	85.76	0.7477	87.55	0.8102
NEG-PO-10	88.24	0.7645	86.62	0.7444	86.91	0.7932
NEG-PO-20	87.92	0.7318	86.93	0.7357	86.73	0.7769
NEG-PO-40	87.79	0.7136	86.74	0.7337	86.27	0.7670
NEG-PE-1	87.92	0.7324	90.52	0.8406	86.00	0.8878
NEG-PE-5	88.24	0.7265	**93.47**	0.8388	87.00	0.8721
NEG-PE-10	**88.59**	0.7133	93.38	0.8346	86.73	0.8587
NEG-PE-20	87.82	0.6911	93.10	0.8282	86.55	0.8453
NEG-PE-40	87.25	0.6812	93.10	0.8339	86.48	0.8380

AUC scores are close to random (0.5). This is because with a density measure of 1.29, the average number of events an object is involved in is only slightly higher than 1 and the co-occurrence observations are not sufficient to capture proximity among objects.

Based on the vertical comparison from Table 3, with regard to Q2, the HEBE framework is relatively robust to noise and data sparsity. For the scenarios (i) when there is noise objects and (ii) when the observation data is sparse, HEBE consistently outperforms the baselines.

Horizontally, we observe that when more events are observed, the accuracies of the classification tasks increases as well. The increment rate is the largest when sampling percentage changes from 1 to 5 percent. Similarly, the performance improvements from 10 to 20 percent are more significant than from 20 to 30 percent. Particularly, we are interested in the case, when the sampled percentage of events exceeds 20 percent, the performance of HEBE-PO becomes comparable with HEBE-PE, and is even slightly better. When the density measure increases, HEBE-PO becomes more effective in modeling the semantic relatedness. In other words, HEBE-PO is more effective when there are enough observations. It is worth noting that even though HEBE-PO better performs than HEBE-PE when the sampling percentage is larger than 20 percent, HEBE-PE is only surpassed by a small margin.

Hence, we answer Q3 based under two scenarios. If the data is noisy, HEBE-PE is more robust than HEBE-PO. If the data is relatively sparse, HEBE-PE is more effective than HEBE-PO; otherwise, if the data is relatively dense, both methods are robust in preserving the proximity among objects.

5.3.4 Noise Pairwise Ranking

In this section, we study the proposed NPR, and compare NPR with NEG. We consider different choices of the negative sampling parameter. The experimental results are reported in Table 6, where NEG-PO-k and NEG-PE-k correspond to HEBE-PO and HEBE-PE with NEG of parameter k, respectively.

As shown in Table 6, we have the following observations. (i) Regarding accuracy, the performance of HEBE with NPR is comparable with HEBE with NEG; regarding AUC, NPR significantly outperforms NEG. This is due to the fact that NPR is ranking based while AUC is also a ranking based measure. (ii) When increasing the negative sampling parameter k, the classification accuracy of HEBE with NEG increases in general. The perform is good when k is large enough. Such an observation aligns with previous studies, such as [33], which also uses accuracy as evaluation metric. (iii) When increasing the negative sampling parameter k, the AUC result of HEBE with NEG decreases. This can be partially explained by the fact that the optimization objective of negative sampling does not align with the AUC measure. Additionally, we remark that the larger k is, the longer time the optimization takes.

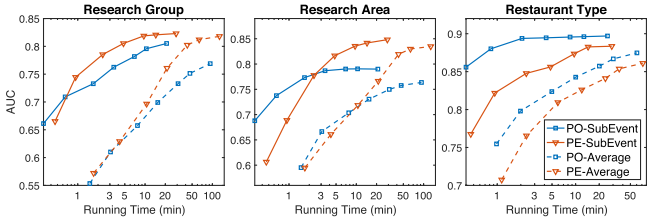


Fig. 4. Regarding subevent sampling, performance comparison in terms of AUC verse the number of running time.

5.3.5 SubEvent Sampling

In Section 4.1, we introduced SubEvent sampling to address the count imbalance problem. In this section, we empirically validate the efficacy of SubEvent Sampling. We compare HEBE with SubEvent sampling with the baseline method, which is to do pooling of all the object embeddings with regard to each event, i.e., averaging. For each iteration, the average baseline method needs longer optimization time. For fair comparison, we vary the iteration number and compare the performance of the two approaching based on running time. The results are plotted in Fig. 4. First, we have the observation that increasing the running time (which is equivalent to increase the iteration number), the performance of all methods increases and converges. Second, for the evaluation metric of AUC, with the same running time, HEBE methods with subevent sampling achieves better performance than the averaging methods without subevent event sampling. Therefore, SubEvent sampling helps with faster convergence and addresses the count imbalance problem.

5.3.6 Hyperparameters

In this section, we study the hyperparameter in the HEBE model. Particularly we study three aspects: the dimensionality of the embedding, the type-wise gradient coefficient, and the number of threads.

We plot the AUC results against dimensionality of the learned embeddings in Fig. 6. An increasing and converging performance pattern is observed for both methods, which is a common pattern that has been observed in previous work [11], [12].

In Section 4, we proposed a type-wise gradient coefficient for ASGD. We verify the effectiveness of the proposed gradient coefficient, compared with a global gradient for all types, the results are reported in Fig. 5, which clearly shows the superiority of the proposed gradient coefficient for step size adjustment, especially on the Yelp dataset.

Regarding the efficiency, we have tested the number of events processed per second against the number of threads, which is shown in Fig. 7. The experiments were conducted on a machine with 2 Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80 GHz (20 Cores). One can observe that the more threads we have, the larger the number of events processed per

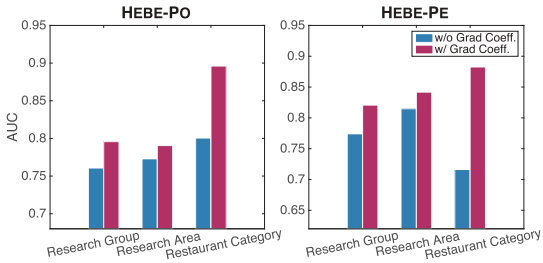


Fig. 5. Performance variations in terms of AUC verse the choice of the gradient for updates.

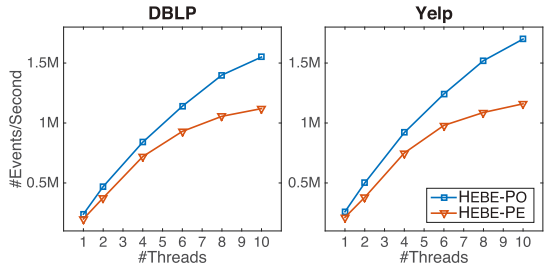


Fig. 7. Number of events processed per second verse the number of threads.

second. Therefore, our method can be easily scaled to extremely large information networks. However, it is worth mentioning that the incremental speed-up of HEBE-PE is smaller than HEBE-PO. Our explanation is that HEBE-PE has many more parameters than HEBE-PO due to the embeddings of hyperedge, resulting in slower performance due to the caching mechanism among different threads when they are accessing random objects and hypeedges.

6 Related Work

Heterogeneous information networks ubiquitously exist in real word and have been investigated in previous studies [24], [30]. Quite many methods were developed towards various applications including classification [30], clustering [24], and similarity search [34]. Recall that when the number of object types in each event is one, the heterogeneous information networks reduce to homogeneous. However, in previous studies of heterogeneous information networks, only binary interactions are studied. In this paper, we model the semantic relatedness among objects based on the concept of events, which correspond to a complete semantic unit.

In particular for the embedding task, both [12] and [9] study the problem of object embedding in heterogeneous information networks. But instead of modeling proximity among objects in each event as a whole, [9], [12] decompose each event into several binary interactions and then do the pairwise modeling separately. Li et al. [21] studies the problem of a new angle by considering the multi-faceted representation of objects in information networks. Zhang et al. [20] learns object embedding by considering the hetergenity of both nodes and relations, based on knowledge base, which is specifically developed for the task of recommender system. Our model is substantially different since we directly model each hyperedge as a whole so that the proximity among objects can be better preserved. Chen et al. [35] studies the problem of embedding in heterogeneous information network, specifically for the task of anomaly detection, with each event defined as a collection of categorical values. Our framework is more general and consider two different methods of modeling the proximity among objects. Zhang et al. [36] shares similar flavor as our modeling for task of modeling people's activities in urban space. These two papers are tailed for particular tasks; while our framework is general, which is can be adopted for various tasks via feeding corresponding task labels.

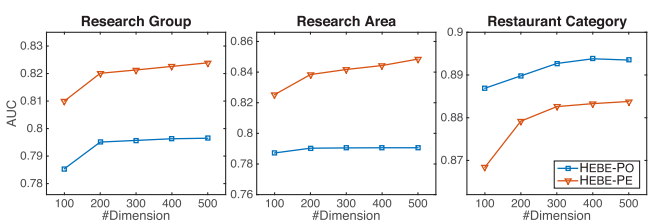


Fig. 6. Performance variations in terms of AUC verse the dimension of the embeddings.

The hyperedge-based framework is also related to tensor analysis, with each event corresponding to an element int the tensor. Such studies in higher-order data [37], [38] have recently emerged for some tasks, such as recommender system [29], multi-relational learning [39], and clustering [40]. In [29], a tensor factorization model is designed specifically for tag recommendation; while we explore a more general framework for embedding from which two methods are designed to model the object-driven and hyperedge-driven proximity respectively. Benson et al. [40] defines higher-order network structures, such as cycles and feed-forward loops, and uses tensor to model the heterogenous event data. In sharp contrast, most of these methods cannot scale to the datasets used in this paper and meanwhile our framework is more general in the sense that it allows multiple event types. In addition, [40] only models the events with one type of object; while HEBE supports multiple object types in multiple event types. However, in order to perform tensor decomposition, the tensor needs to be materalized. Due to curse of dimension, such a method is not computationally feasible. In our case, each time we sample an event, which is independent of the size of dimensions of the corresponding tensor. Moreover, we adopted ASGD, which is designed for distributed computation, leading to better scability.

Another line of research is embedding learning in Knowledge Bases (KBs) [41], [42], [43], [44], [45], which is also relevant to embedding learning in HINs. One pioneer work in knowledge base completion (KBC) is [41]. Not only the entities are embedded but also the relations. In the proposed TransE model, the relations serve as translations between entities. Wang et al. [44] extends the TransE model and proposes a TransH model, which enjoys more mapping properties of relations. In addition, [45] proposes to jointly embed entities and words into the same continuous vector space to improve accuracy in predicting facts. To improve the results for link prediction, TransR is proposed which projects the entities and relations into two different vector spaces [43]. Furthermore, instead of direct modeling of relations among entities, [42] considers to take multiple-step relation paths as translations between entities in KBs while learning embeddings. To summarize, embedding learning in KBs is to model the pairwise relations among entities and is aimed at adding new facts by making link predictions. Meanwhile, our model models interactions among objects in the network as a whole and uses embeddings to represent the semantic information of objects.

In addition, some dimension reduction methods can be adapted for object embedding learning in heterogeneous

information networks, such as principal component analysis [46], singular value decomposition [46], and non-negative matrix factorization [47]. In [48], a new algorithm is proposed to mine both context and content links in social media networks for semantics understanding, which address the problem of sparse context links. Meanwhile, [49] learn object embeddings based on a joint matrix factorization framework. However, these methods ignores the intrinsic event types and fails to model the participating objects collectively, and thus cannot capture the intricate proximity in heterogeneous information networks.

7 Conclusion

In this paper, we proposed to learn object embedding in heterogeneous information networks with event. In detail, we proposed a generic framework called Hebe, which models participant objects in each event as a whole, resulting in more efficient information propagation. Two methods were presented based on the concept of hyperedge: Hebe-Po, modeling the proximity among the participating objects themselves on the same hyperedge, and Hebe-Pe modeling proximity between the hyperedge and the participating objects. Within the Hebe framework, we presented a parameter-free ranking-based method to efficiently optimize the conditional probabilities via noise sampling. Extensive quantitative experiments have been conducted to corroborate the efficacy of the proposed model in learning the object embeddings, particularly robustness towards noisy observations and data sparseness.

We identify some future work for the Hebe framework. First, it is general and could be adapted to many downstream applications, including recommender system and link prediction. Second, Hebe prefers term entities from short text due to the operations of subevent sampling. Some additional work needs to be done in order to adapt it to those data having longer text. Third, it could be of interest to learn the relative importance of different event types, based on specific applications. Finally, this work focuses on learning embeddings in an unsupervised manner. Exploring how to incorporate labels and generate predictive embeddings is a another promising direction.

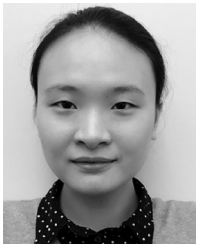
Acknowledgments

Huan Gui and Jialu Liu made equal contributions. The work was done when Jialu Liu was a graduate student at the University of Illinois at Urbana Champaign.

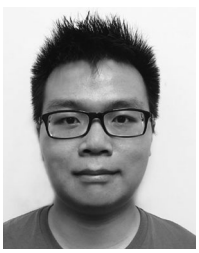
References

[1] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 58, no. 7, pp. 1019–1031, 2007.
[2] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.
[3] T. Opsahl and P. Panzarasa, "Clustering in weighted networks," *Social Netw.*, vol. 31, no. 2, pp. 155–163, 2009.
[4] S. Bhagat, G. Cormode, and S. Muthukrishnan, "Node classification in social networks," in *Social Network Data Analytics*. Berlin, Germany: Springer, 2011, pp. 115–148.
[5] Y. Koren, "The bellkor solution to the netflix grand prize," 2009.
[6] A. Globerson, G. Chechik, F. Pereira, and N. Tishby, "Euclidean embedding of co-occurrence data," in *Proc. Advances Neural Inf. Process. Syst.*, 2004, pp. 497–504.
[7] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Advances Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.
[8] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 701–710.
[9] S. Chang, W. Han, J. Tang, G.-J. Qi, C. C. Aggarwal, and T. S. Huang, "Heterogeneous network embedding via deep architectures," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 119–128.
[10] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 855–864.
[11] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information network embedding," in *Proc. Int. Conf. World Wide Web*, 2015, pp. 1067–1077.
[12] J. Tang, M. Qu, and Q. Mei, "PTE: Predictive text embedding through large-scale heterogeneous text networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 1165–1174.
[13] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, 1964.
[14] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. Advances Neural Inf. Process. Syst.*, 2001, pp. 585–591.
[15] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Sci.*, vol. 290, no. 5500, pp. 2319–2323, 2000.
[16] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1225–1234.
[17] C. Li, X. Guo, and Q. Mei, "DeepGraph: Graph structure predicts network growth," arXiv:1610.06251, 2016.
[18] S. Cao, W. Lu, and Q. Xu, "GraRep: Learning graph representations with global structural information," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, 2015, pp. 891–900.
[19] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu, "Asymmetric transitivity preserving graph embedding," in *Proc. 20th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, 2016, pp. 1105–1114.
[20] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proc. 20th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, 2016, pp. 353–362.
[21] J. Li, A. Ritter, and D. Jurafsky, "Learning multi-faceted representations of individuals from heterogeneous evidence using neural networks," arXiv:1510.05198, 2015.
[22] C. Huang, C. C. Loy, and X. Tang, "Local similarity-aware deep feature embedding," in *Proc. Advances Neural Inf. Process. Syst.*, 2016, pp. 1262–1270.
[23] H. Gui, J. Liu, F. Tao, M. Jiang, B. Norick, and J. Han, "Large-scale embedding learning in heterogeneous event data," in *Proc. IEEE Int. Conf. Data Mining*, 2016, pp. 429–438.
[24] Y. Sun and J. Han, "Mining heterogeneous information networks: Principles and methodologies," *Synthesis Lectures Data Mining Knowl. Discovery*, vol. 3, no. 2, pp. 1–159, 2012.
[25] C. Berge, *Hypergraphs: Combinatorics of Finite Sets*. Amsterdam, The Netherlands: Elsevier, 1984.
[26] B. Recht, C. Re, S. Wright, and F. Niu, "Hogwild: A lock-free approach to parallelizing stochastic gradient descent," in *Proc. Advances Neural Inf. Process. Syst.*, 2011, pp. 693–701.
[27] A. Mnih and Y. W. Teh, "A fast and simple algorithm for training neural probabilistic language models," in *Proc. Int. Conf. Mach. Learn.*, 2012, pp. 1751–1758.
[28] J. Silva and R. Willett, "Hypergraph-based anomaly detection of high-dimensional co-occurrences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 3, pp. 563–569, Mar. 2009.
[29] S. Rendle and L. Schmidt-Thieme, "Pairwise interaction tensor factorization for personalized tag recommendation," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2010, pp. 81–90.
[30] M. Ji, J. Han, and M. Danilevsky, "Ranking-based classification of heterogeneous information networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2011, pp. 1298–1306.
[31] O. Levy and Y. Goldberg, "Neural word embedding as implicit matrix factorization," in *Proc. Advances Neural Inf. Process. Syst.*, 2014, pp. 2177–2185.
[32] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *Proc. Int. Conf. Mach. Learn.*, 2006, pp. 233–240.
[33] Y. Goldberg and O. Levy, "word2vec explained: Deriving Mikolov et al.'s negative-sampling word-embedding method," arXiv:1402.3722, 2014.

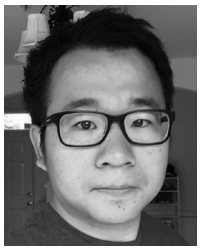
[34] G. Wang, Q. Hu, and P. S. Yu, "Influence and similarity on heterogeneous networks," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage.*, 2012, pp. 1462–1466.
[35] T. Chen, L.-A. Tang, Y. Sun, Z. Chen, and K. Zhang, "Entity embedding-based anomaly detection for heterogeneous categorical events," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 1396–1403.
[36] C. Zhang, et al., "Regions, periods, activities: Uncovering Urban dynamics via cross-modal representation learning," in *Proc. 26th World Wide Web Conf.*, 2017, pp. 361–370.
[37] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.
[38] M. Jiang, P. Cui, F. Wang, X. Xu, W. Zhu, and S. Yang, "FEMA: Flexible evolutionary multi-faceted analysis for dynamic behavioral pattern discovery," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 1186–1195.
[39] R. Jenatton, N. L. Roux, A. Bordes, and G. R. Obozinski, "A latent factor model for highly multi-relational data," in *Proc. Advances Neural Inf. Process. Syst.*, 2012, pp. 3167–3175.
[40] A. R. Benson, D. F. Gleich, and J. Leskovec, "Tensor spectral clustering for partitioning higher-order network structures," in *Proc. IEEE Int. Conf. Data Mining*, 2015, pp. 118–126.
[41] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proc. Advances Neural Inf. Process. Syst.*, 2013, pp. 2787–2795.
[42] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao, and S. Liu, "Modeling relation paths for representation learning of knowledge bases," arXiv:1506.00379, 2015.
[43] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 2181–2187.
[44] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proc. AAAI Conf. Artif. Intell.*, 2014, pp. 1112–1119.
[45] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph and text jointly embedding," in *Proc. Conf. Empirical Methods Natural Language Process.*, 2014.
[46] M. E. Wall, A. Rechtsteiner, and L. M. Rocha, "Singular value decomposition and principal component analysis," in *A Practical Approach to Microarray Data Analysis*. Berlin, Germany: Springer, 2003, pp. 91–109.
[47] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Advances Neural Inf. Process. Syst.*, 2001, pp. 556–562.
[48] G.-J. Qi, C. Aggarwal, Q. Tian, H. Ji, and T. Huang, "Exploring context and content links in social media: A latent space method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 850–862, May 2012.
[49] S. Zhu, K. Yu, Y. Chi, and Y. Gong, "Combining content and link for classification using matrix factorization," in *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2007, pp. 487–494.
[50] 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 *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 361–370.



Huan Gui received the BS degrees in computer science and economics from Peking University, China, in 2012. She is working toward the PhD degree at the University of Illinois at Urbana-Champaign, supervised by Prof. Jiawei Han. Her main research interests include information network analysis, distributed representation learing, and matrix estimation.



Jialu Liu received the PhD degree from the University of Illinois at Urbana Champaign, in 2016, supervised by Prof. Jiawei Han. He is working with Google Research New York on structured data and social data management. His primary research interests include scalable information extraction and text mining.



Fangbo Tao received the BS degree from the School of Software, Tsinghua University, China, in 2012. He is working toward the PhD degree at the University of Illinois at Urbana-Champaign, supervised by Prof. Jiawei Han. His main research interests include large-scale data mining, text data cube, and natural language processing.

Meng Jiang received the BE and PhD degrees from the Department of Computer Science and Technology, Tsinghua University, in 2010 and 2015, respectively. He is now a postdoctoral research associate with the University of Illinois at Urbana-Champaign. He visited Carnegie Mellon University from 2012 to 2013. He has more than 15 published papers on data-driven behavioral analytics for recommendation, prediction and suspicious behavior detection in top conferences and journals of the relevant field. He got the best paper finalist in ACM SIGKDD 2014.

Brandon Norick received the BS degrees in computer science and mathematics from Montana State University, in 2011. He is working toward the PhD degree at the University of Illinois at Urbana-Champaign, supervised by Prof. Jiawei Han. His research interests include personalized recommendation, as well as mining heterogeneous information networks.

Lance Kaplan received the BS (with distinction) degree from Duke University, Durham, North Carolina, in 1989 and the MS and PhD degrees from the University of Southern California, Los Angeles, in 1991 and 1994, respectively, all in electrical engineering. Currently, he is a team leader in the Networked Sensing and Fusion branch of the U.S. Army Research Laboratory. He serves as an associate editor-in-chief and EO/IR systems editor for the *IEEE Transactions on Aerospace and Electronic Systems (AES)*. In addition, he is the tutorials editor for the *IEEE AES Magazine*, and he also serves on the Board of Governors of the IEEE AES Society. He is a three time recipient of the Clark Atlanta University Electrical Engineering Instructional Excellence Award from 1999-2001. His current research interests include signal and image processing, automatic target recognition, data fusion, and resource management. He is a fellow of the IEEE.

Jiawei Han is Abel Bliss professor in the Department of Computer Science, University of Illinois. He has been researching into data mining, information network analysis, and database systems, with more than 600 publications. He served as the founding editor-in-chief of the *ACM Transactions on Knowledge Discovery from Data (TKDD)*. He has received ACM SIGKDD Innovation Award (2004), IEEE Computer Society Technical Achievement Award (2005), IEEE Computer Society W. Wallace McDowell Award (2009), and Daniel C. Drucker Eminent Faculty Award at UIUC (2011). He is a fellow of the ACM and the IEEE. He is currently the director of the Information Network Academic Research Center (INARC) supported by the Network Science-Collaborative Technology Alliance (NS-CTA) program of the U.S. Army Research Lab. His co-authored textbook *Data Mining: Concepts and Techniques* (Morgan Kaufmann) has been adopted worldwide.