Large-Scale Embedding Learning in Heterogeneous Event Data

Huan Gui*†, Jialu Liu*‡, Fangbo Tao†, Meng Jiang†, Brandon Norick† and Jiawei Han†
†University of Illinois at Urbana-Champaign ‡Google Research
†{huangui2, ftao2, mjiang89, bnorick, hanj}@illinois.edu ‡ jialu@google.com

***Abstract*—*Heterogeneous events*, which are defined as events connecting *strongly-typed* objects, are ubiquitous in the real world. We propose a *HyperEdge-Based Embedding* (HEBE) framework for *heterogeneous event data*, where a *hyperedge* represents the interaction among a set of involving objects in an event. The HEBE framework models the proximity among objects in an event by predicting a target object given the other participating objects in the event (hyperedge). Since each hyperedge encapsulates more information on a given event, HEBE is robust to data sparseness. In addition, HEBE is scalable when the data size spirals. Extensive experiments on large-scale real-world datasets demonstrate the efficacy and robustness of HEBE.**

I. INTRODUCTION

Learning embeddings of objects is to represent each object as 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 objects, are beneficial for various downstream applications, such as exploratory data analysis, link prediction [1], object clustering [2], classification [3], and recommendation [4]. The objective of embedding techniques is mainly to preserve certain relationships among objects [5]–[12].

Interactions among individual components or agents, such as friendships in social sites, hyperlinks on webpages, word co-occurrences, 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) in a snippet, such as a sentence. The proximity between the two words can be modeled as the conditional probability of predicting the observed target given the context [6], where the conditional probability is estimated using a softmax function. This model has also been generalized to network data, such as [7], [10].

On the other hand, recent years have witnessed an increasing interest on studying interactions among *strongly-typed objects* (i.e., the participating objects in an event belong to a number of types) [13]. Bibliographical data is one such example, where a publication implies a *simultaneous* interaction among paper, author, venue and terms: *Authors* write *paper*, *paper* publishes in *venue*, and *paper* contains *terms* as content. The publication of a paper can be viewed as an *event*, which can be abstracted by a *hyperedge* encapsulating all the participating objects in the event. In this paper, we propose an embedding learning framework based on a collection of *heterogeneous event data*. More generally, we consider that the participating objects of the events are of *different types*.

Embedding learning with strongly-typed interactions has broad real-world applications [8], [11]. There are different approaches to computing embeddings as shown below.

Example I.1. DBLP (http://dblp.uni-trier.de) is a CS bibliographical data set, 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., [8], [11]) decompose the simultaneous interaction among all objects into several scattered pairwise interactions (e.g., Author-Paper, Venue-Paper). Object embeddings are learned by combing embedding learning procedures upon each set of pairwise interactions, using conventional embedding learning in single-typed network data. However, such decomposition 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 new framework called **H**yper**E**dge **B**ased **E**mbedding (HEBE) that captures each strongly-typed object interaction as a whole, as illustrated in the top right of Figure 1. Inspired from classical hypergraph theory [14] on hyperedges, we define the interaction among a set of objects as a *hyperedge*. HEBE models each hyperedge as a whole. Compared with [8], [11], HEBE preserves more contextual information for embedding learning.

As every coin has two sides, the hyperedge model encapsulates all the contextual information with respect to each event, it also imposes challenges on modeling the proximity and optimization. Since interactions with multiple participating objects are modeled as a whole, existing methods cannot be straightforwardly applied. Instead, we propose to model the proximity of each hyperedge based on prediction, i.e., the probability that *a participating object (as target) would be predicted given all the remaining objects (as context) in the event*. In other words, the higher proximity of objects in an event is, the more likely we can recover a specific involving object given the remaining. Vice versa.

Moreover, it is essential for HEBE to be scalable in the big data era. We leverage recent advancement of asynchronous stochastic optimization [15] to take advantage of the parameter sparsity in embedding learning. Furthermore, we devise a new technique to efficiently optimize the conditional probability of prediction. Compared with existing methods, our method alleviates the negative sampling hyperparameter [6], [10], [16].

In HEBE, each hyperedge encapsulates more contextual information, leading to more informative and efficient updates. Therefore, HEBE is more robust to data sparseness. We apply HEBE to large-scale real word datasets to learn object embeddings and measure the quality of the learned embeddings based on various classification tasks. Experimental results verify the efficacy of HEBE.

In sum, the study makes the following contributions:

1) It proposes the problem of learning object embeddings for heterogeneous event data using hyperedges, especially when each strongly-typed event is modeled as a whole.
2) A new embedding framework HEBE is established, with a proposed method to model the proximity among participating objects in each event based on prediction.

[*] Equal Contribution. The work was done while Jialu Liu was a graduate student at UIUC.

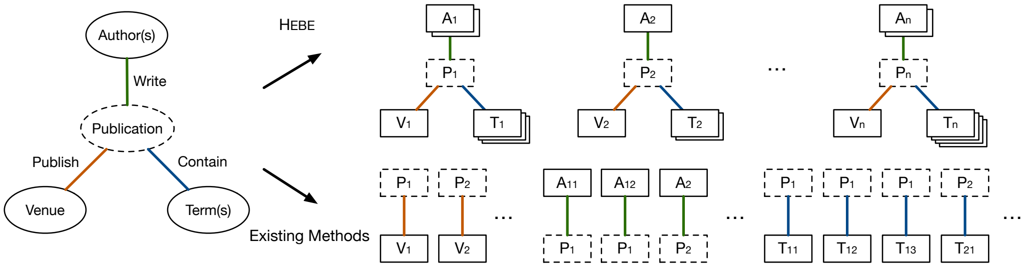

Fig. 1: The interaction schema of DBLP is in the left. A publication event results in the interactions of authors-publication, venue-publication, and terms-publication at the same. Existing methods (in the bottom right) consider each interaction type independently. Our method (in the top right) defines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

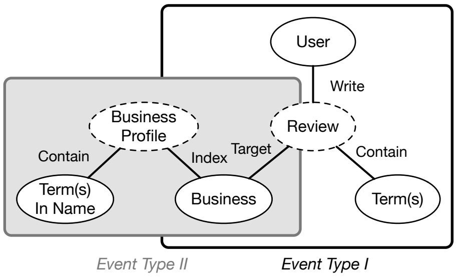

Fig. 2: Event schema of heterogeneous event data, DBLP, with two event types, business profile (left) and review (right).

3) A new method called *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.

II. Preliminaries

In this section, we define the problem of object embedding learning in heterogeneous event data and introduce several related concepts and necessary notations.

A. Heterogeneous Event Data

Definition II.1. 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 of t-th type, we define an event Q_i as a pair represented as $\langle V_i, \omega_i \rangle$, in which $V_i = \cup\{V_i^t\}_{t=1}^{T}$ with $V_i^t \subseteq X_t$ as a set of participating objects in t-th type and ω_i is the weight of event Q_i (*e.g.*, the number of occurrences of this event). Specifically, if $T \geq 2$, such event is defined as a **heterogeneous event**; otherwise ($T = 1$), it is defined as a **homogeneous event**. A collection of heterogeneous events is defined as **heterogeneous event data**.

We slightly abuse the notation and use X_t to represent both the set of objects of the t-th type and the name of the type as well. Besides multiple object types, we further allow multiple event types. Each event type is defined by *event schema* to visualize relationships among objects in the corresponding event type. The event schema of DBLP mentioned in Example I.1 is shown on the left of Figure 1 with one event type. Yelp data described in the following example contain two event types. Event identifiers are marked in dashed circles.

Example II.2. Yelp (http://www.yelp.com/) is an online website for users to review various businesses. Based on schema shown in Figure 2, there are two types of heterogeneous 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 Terms. The business objects type participates in both event types.

B. Learning Object Embeddings

Given heterogeneous event data and the event schemata, 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 such that objects that are semantically similar will be close in the space, with the distance measured by cosine similarity, for instance.

Accordingly, to conduct the object embedding in heterogeneous event data, the event structure must be preserved. Instead of simply considering each event as a set of scattered pairwise interactions between the event identifier and individual participating objects (or between individual participating objects), we define a new structure to encapsulate all the information in the event. We use a corresponding **hyperedge** H_i to model the event Q_i by viewing all the participating objects as a whole, i.e., H_i connecting the set of objects V_i with edge weight ω_i. It is worth noting that the concept of hyperedge come from the classical analysis on hypergraphs and hyperedges [14], [17]. We further generalize the concept of hyperedge by considering the heterogeneous types of the objects.

In order to model the semantic similarity among participating objects in each event, we propose a method based on prediction. The 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 terms of "Network" in the venue ICDM. Therefore, we define proximity based on the prediction of participating object observation.

Definition II.3. The **proximity** of an event is defined as the likelihood of observing a target object given all other participating objects in the same event.

Based on the definition of proximity preserving the event structures, we define the task of object embedding as follows.

Definition II.4 (Object Embedding for Heterogeneous Event Data)**.** Given heterogeneous event data $\mathcal{D} = \{Q_i\}$, 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 proximity of a given event, where $d \ll |\mathcal{X}|$, i.e., $\mathcal{M} : \mathcal{X} \to \mathbb{R}^d$, where $\mathcal{X}$ is the set of all objects.

III. 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 the modeling and optimization of proximity among participating objects in each event. We will provide the details of estimating the proximity as discussed in Section II, the optimization procedure of which is be discussed in Section IV.

A. Optimization Objective

As defined in Definition II.3, HEBE 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 the alternative objects are of the same type as the target object. Assuming the target object is u, we use C to denote the context objects. Without loss of generality, we further assume the target object is of type X_1 and $u \notin C$. The conditional probability of predicting the target object u of type X_1 given C is defined as

$$\mathbb{P}(u|C) = \frac{\exp\left(S(u,C)\right)}{\sum_{v \in X_1} \exp\left(S(v,C)\right)}, \tag{III.1}$$

where $S(\cdot)$ is a scoring function reflecting the similarity between target object u and context objects C such as the aggregation of inner products among every paired elements from the union of u and C. However, in case of an object type having many more objects than the other types in a single event, we take the averaged embeddings of the object set for each type before applying inner product.

Objective. To preserve the proximity among objects, we can naturally minimize Kullback-Leibler (KL) divergence between $\mathbb{P}(\cdot|C)$ and the empirical distribution $\widehat{\mathbb{P}}(\cdot|C)$. Suppose the target object type t, we use C_t to denote the corresponding context, and $\mathcal{P}_t$ as the sample space of C_t. Hence, the objective function can be defined as:

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

where we use λ_{C_t} is the importance of the context C_t

$$\lambda_{C_t} = \sum_{i=1}^{N} \omega_i \mathbf{I}_{\{C_t \in V_i\}} / |\mathcal{P}_{i,t}|, \tag{III.2}$$

where $\mathcal{P}_{i,t}$ is the subset of $\mathcal{P}_t$ constrained on V_i which is defined on event Q_i, and $\mathbf{I}_{\{\cdot\}}$ is a binary indicator function. λ_{C_t} can be intuitively understood as the weighted number of hyperedges that have C_t as an object subset.

Lemma III.1. Based on the definition of λ_{C_t} in (III.2),

$$\mathcal{L} = -\sum_{i=1}^{N} \omega_i \sum_{t=1}^{T} \frac{1}{|\mathcal{P}_{i,t}|} \sum_{C_t \in \mathcal{P}_{i,t}} \mathbb{P}(u|C_t). \tag{III.3}$$

where $u = V_i \setminus C_t$ is the target object.

Proof. Proof omitted due to space constraint. □

B. Multiple Event Types

We consider the scenario there are multiple event types in the heterogeneous event data, such as Example II.2. Suppose there are K heterogeneous event types, the overall objective function ($\mathcal{L}^*$) is defined as the (weighted) sum of objective function $\mathcal{L}^k$ corresponding to the k-th event type.

IV. Optimization

In this section, we first introduce the optimization procedure for Hebe with only one event type, followed by the case with multiple event types.

A. Noise Pairwise Ranking

Considering the objective function of Hebe in (III.3), direct optimization of $\mathcal{L}$ is intractable since the conditional probability (III.1) requires the summation over the entire set of objects with type X_1.

To address this challenge, noise contrastive estimation (NCE) [16] and negative sampling (NEG) [6] 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. While 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 [6], [7], [10].

As [18], [19] shows, the hyperparameter of negative sampling value k [6] plays an important role in obtaining the optimal embeddings. To get rid of the hyperparameter, we develop a new optimization framework from a pairwise ranking perspective, *noise pairwise ranking* (NPR). In comparison, NCE and NEG are discriminative models, while our model is a generative model in optimizing the conditional probability. Recall that the conditional probability to be maximized is defined in (III.1). Therefore,

$$\mathbb{P}(u|C) = \Big(1 + \sum_{v \neq u} \exp\left(S(v,C) - S(u,C)\right)\Big)^{-1}, \tag{IV.1}$$

which follows from (III.1) via dividing the denominator and numerator by $\exp\left(S(u,C)\right)$. Instead of directly optimizing (IV.1) over all $v \in X_1 \setminus u$, we update (IV.1) with respect to a small set of noise samples in $X_1 \setminus u$, where an individual sample is denoted as v_n. With $\sigma(\cdot)$ representing the sigmoid function that $\sigma(x) = 1/(1+\exp(-x))$, we maximize the following probability instead,

$$\mathbb{P}(u > v_n|C) = \sigma(-S(v_n, C) + S(u, C)), \tag{IV.2}$$

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

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

which implies that optimizing $\mathbb{P}(u > v_n|C)$ can be explained as optimizing the lower bound of $\mathbb{P}(u|C)$.

Remark IV.1. The derived pairwise ranking results in (IV.2) is similar to the Bayesian Pairwise Ranking (BPR) proposed in [20]. 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 noise distribution.

Thus, for all $v_n \in X_1 \setminus u$, (III.1) can be approximated by

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

where P_n is the noise distribution. Similar to NCE and NEG, NPR also has the noise distribution P_n as a free parameter. We set $P_n \propto d_u^{3/4}$ as proposed in [6], where d_u is the degree of u, i.e., the number of hyperedges involving object u.

B. Single Event Type

Based on the NPR optimization framework proposed in Section IV-A, we apply it to Hebe, considering single event type. Recall that the objective of Hebe is defined in (III.3) with the conditional probability defined in (III.1). By applying the NPR optimization framework to the conditional probability in (III.1), we have the new objective function as

$$\widetilde{\mathcal{L}} = -\sum_{i=1}^{N} \sum_{t=1}^{T} \frac{\omega_i}{|\mathcal{P}_{i,t}|} \sum_{C_t \in \mathcal{P}_{i,t}} \mathbb{E}_{u_n \sim P_n(X_t)} \ell(C_t, u, u_n).$$

where $\ell(C_t, u, u_n) = \log \mathbb{P}(u > u_n|C_t)$, u_n is the sampled noise from $P_n(X_t)$ and the latter is the noise distribution of objects of type X_t. Based on NPR, we have

$$\mathbb{P}(u > u_n|C_t) = \sigma(-S(u_n, C_t) + S(u, C_t)).$$

To optimize $\widetilde{\mathcal{L}}$, we use the asynchronous stochastic gradient algorithm (ASGD) [15] due to the sparsity of the optimization problem, which means that most gradient updates only modify a small portion of the variables. Define $\mathbf{\Theta} = \{\boldsymbol{\theta}_v\}_{v \in \mathcal{X}}$ as the parameters, where $\boldsymbol{\theta}_v$ is the embedding for object v, we have the gradient

$$\frac{\partial \widetilde{\mathcal{L}}}{\partial \mathbf{\Theta}} = -\sum_{i=1}^{N} \sum_{t=1}^{T} \frac{\omega_i}{|\mathcal{P}_{i,t}|} \sum_{C_t \in \mathcal{P}_{i,t}} \mathbb{E}_{u_n \sim P_n(\mathcal{S}_j)} \frac{\partial \ell(C_t, u, u_n)}{\partial \mathbf{\Theta}}.$$

In specific,

$$\frac{\partial \ell(C_t, u, u_n)}{\partial \boldsymbol{\theta}_{u_i}} = \sigma(S_\Delta) \frac{\partial S(u, C_t)}{\partial \boldsymbol{\theta}_{u_i}}; \qquad \frac{\partial \ell(C_t, u, u_n)}{\partial \boldsymbol{\theta}_{u_n}} = -\sigma(S_\Delta) \frac{\partial S(u, C_t)}{\partial \boldsymbol{\theta}_{u_n}};$$

$$\frac{\partial \ell(C_t, u, u_n))}{\partial \bar{\boldsymbol{\theta}}_{C_t}} = \sigma(S_\Delta) \frac{\partial (S(u, C_t) - S(u_n, C_t))}{\partial \boldsymbol{\theta}_{C_t}}.$$

where $S_\Delta = S(u, C_t) - S(u_n, C_t)$.

Gradient coefficient. Objects in types of smaller size have larger coefficient due to the averaging of embeddings for each object type

Algorithm 1 HEBE.

```
1:  Initialize: randomly initialize $\mathbf{\Theta}$, $\mathbf{\Gamma}$
2:  for $t = 1, \ldots, T$ do
3:      $\alpha^t$ is obtained via (IV.3)
4:  end for
5:  for $i = 0$ to $I_N - 1$ do
6:      $\eta \leftarrow \eta_0 \cdot (I_N - i)/I_N$
7:      $\boldsymbol{\beta} \leftarrow \eta \cdot [\alpha_o]_{o \in O}$
8:      for $k \in \mathcal{K}$ do
9:          Sample a event $Q_i$ of event type k
10:         Sample an object type $t$
11:         Draw a random object from $P_n(X_t)$ as negative
12:         Update Object Embeddings $\mathbf{\Theta}$ by Gradient Descent
            (GD) with type-wise step size $\boldsymbol{\beta}$
13:     end for
14: end for
15: Return: $\mathbf{\Theta}$
```

when calculating the scoring function. 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. In order to balance the average step size among different object types, when applying ASGD to learn the embedding, 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,

$$\alpha^t = |X_t| \,/\, \max_{t'=1}^{T} \{|X_{t'}|\}. \tag{IV.3}$$

We define $\boldsymbol{\beta} = [\beta^t]_{t=1}^{T}$ as the vector of step size for each object type. The updating process for a single iteration of HEBE is summarized in Line 10-12 in Algorithm 1.

C. Multiple Event Types

The optimization procedures for HEBE introduced in the previous sections are applicable when there is only one event type, i.e., $|\mathcal{K}| = 1$, where $\mathcal{K}$ is the set of event types. Here, we consider the scenario when $|\mathcal{K}| > 1$. The unified algorithm with multiple event types is shown in Algorithm 1, with η_0 and I_N as the initial step size and the iteration number. When learning embeddings for the objects (and the event identifiers), we opt to use a similar procedure to that used in [11], which is to use all event types jointly. 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.

V. EXPERIMENTAL STUDY

In this section, we report experimental results of the proposed HEBE framework. To evaluate whether the learned embeddings preserve the proximity between objects in heterogeneous event data, we evaluate the embeddings based on various classification tasks. Particularly, via a series of quantitative studies, we aim at answering the following two questions:

Q1: Does HEBE method learn better object embeddings compared with existing methods?

Q2: Is HEBE method robust when data become sparse?

A. Datasets and Compared Methods

We introduce two datasets on which we conduct experiments: DBLP and Yelp. The basic statistics of both datasets are summarized in Table I. **DBLP** is a collection of bibliographic information on major computer science journals and proceedings, from which we extracted three types of objects and one event type, with the event schema presented in Figure 1. Each event corresponds to a publication, and each publication involves authors, venue, and terms used in the paper.

The **Yelp** dataset provides business reviews and we extracted two event types as presented in Figure 2 with review and business profile as their event identifiers. In event type I, there are three object types including user, business and term; while for event type II, we have two object types, business and term used in its name. It is worth noting that we distinguish the terms in the review and terms in the business profile. User is removed from the review event type due to its sparsity that the number of reviews written by each user is typically small.

TABLE I: Number of objects for DBLP and Yelp.

DBLP	Author	Term	Venue	Paper
	209,679	165,657	7953	1,938,912
Yelp	Business	Term (review)	Term (name)	Review
	12,241	130,259	6,709	905,658

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 in different types 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. 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. Thus we can first apply degree normalizions to these interaction and then merge them to get normalized $\widetilde{\mathbf{M}}$ as described in [21]. The dimensionality is set to be 300 for all methods. In particular, the following methods are considered:

1) Singular Value Decomposition (SVD) on $\mathbf{M}$, and singular vectors are used as object representation.
2) Normalized SVD (NSVD) on $\widetilde{\mathbf{M}}$.
3) Positive shifted PMI (PPMI). As shown in [22], the word embedding with negative sampling is equivalent to approximate the PPMI. Hence, we perform SVD on the PPMI matrix of $\mathbf{M}$.
4) Non-negative Matrix Factorization (NMF) on $\mathbf{M}$, and matrix factor is used as object representation.
5) Normalized NMF (NNMF) on $\widetilde{\mathbf{M}}$.
6) LINE [10]: a second-order object embedding approach originally proposed for networked data. We apply LINE to the decomposed pairwise interactions directly.
7) PTE [11]: an object embedding approach that applies pairwise modeling in a round-robin fashion within each event.[1]

B. Evaluation Metric

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

One way to evaluate the quality of the embeddings is through the proximity-related object classification task. 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. Due to the space limit, we only report the higher accuracy between linear svm and logistic regression under different settings. 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. Therefore we further use a ranking metric called area under the curve (AUC) [23] to evaluate the quality of embeddings over all dimensions. Specifically, we use cosine similarity as the similarity measure. The AUC measure becomes high if embeddings are close for objects sharing the same label, while distant for objects having different labels.

Regarding the DBLP dataset, we have two types of labels over 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**,

[1] The labels are not provided during the training.

TABLE II: 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	84.48	**0.7957**	**92.18**	**0.7905**	**88.00**	**0.8961**

including 4,040 researchers from four research areas including data mining, database, machine learning, and artificial intelligence.

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. For each category, we randomly select 100 restaurants that have at least 50 reviews. Restaurants with multiple labels are excluded.

C. 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.

1) Classification Results: Table II summarizes the experimental results on classification (Acc.) and ranking (AUC) in DBLP and Yelp.

Considering the results for research group in DBLP, we note that PTE and Hebe achieve the best performance. PTE is slightly better than Hebe on accuracy but the latter outperforms the former on AUC by a large margin. It is interesting to see that the normalization strategy on **M** has a big effect on the performance, but the trend is oppsite between SVD and NMF.

For the task of research area in DBLP, Hebe attains the best performance on both classification accuracy and AUC score, confirming the their effectiveness of capturing the proximity. 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. We also observe that both NSVD and NNMF beat their unnormalized versions, implying that the normalization trick works for some tasks.

With respect to the Yelp dataset, on classifying the restaurant type, we observe that Hebe is significantly better than the baselines for both measures. A tentative explanation is that Hebe framework models the two 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.

To summarize, we positively answer Q1 on the effectiveness of Hebe 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. But compared to our framework, by modeling the heterogeneous event as a whole, one can achieve even better performance.

2) 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%) of the two datasets and repeat the three tasks mentioned aforehand. Experimental results are reported in Table III for the DBLP dataset and Table IV 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, leading to a slower rate of increment. Considering the fact that classification needs to learn the mapping function between embeddings and classes based on some certain assumptions, which may not agree with the embedding data, we opt to report AUC results, which provides more comprehensive evaluation of the embeddings across all dimensions.

TABLE III: The AUC results on **sampled** DBLP data considering both research group and research area classification. The sparsity is measured by the average number of publication each author is involved in (similar below).

Sampling %.	1%	5%	10%	20%	30%	50%
Density	1.264	2.028	2.882	4.595	6.400	10.315
Research Group						
SVD	0.5602	0.6169	0.6481	0.6494	0.6720	0.6924
NSVD	0.5504	0.5919	0.6330	0.6345	0.6517	0.6790
PPMI	0.5502	0.5993	0.6557	0.6703	0.6792	0.7192
NMF	0.5583	0.5989	0.5874	0.6009	0.5950	0.6120
NNMF	0.5462	0.6601	0.6806	0.7167	0.7197	0.7294
LINE	0.6004	0.6254	0.5877	0.5619	0.5669	0.5871
PTE	0.6190	0.6727	0.6434	0.6778	0.7034	0.6783
Hebe	**0.6034**	**0.7082**	**0.7151**	**0.7515**	**0.7640**	**0.7841**
Research Area						
SVD	0.5162	0.5337	0.5411	0.5516	0.5551	0.5644
NSVD	0.5076	0.5004	0.5021	0.5157	0.5299	0.5600
PPMI	0.5063	0.5092	0.5180	0.5395	0.5669	0.6203
NMF	0.5143	0.5329	0.5391	0.5493	0.5560	0.5637
NNMF	0.5303	0.5773	0.6206	0.6486	0.6807	0.7594
LINE	0.5552	0.5764	0.5716	0.5501	0.5339	0.5822
PTE	0.5291	0.5858	0.5782	0.6015	0.6356	0.6340
Hebe	**0.5635**	**0.6108**	**0.6798**	**0.7199**	**0.7293**	**0.7817**

TABLE IV: AUC results on **sampled** Yelp data.

Sampling %.	1%	5%	10%	20%	30%	50%
Density	1.963	4.791	8.155	15.09	22.32	37.01
SVD	0.6133	0.6786	0.7001	0.7100	0.7121	0.7134
NSVD	0.6081	0.6236	0.6308	0.6275	0.6280	0.6259
PPMI	0.5561	0.5484	0.5626	0.5824	0.6089	0.6253
NMF	0.6790	0.7381	0.7594	0.7877	0.7907	0.7991
NNMF	0.6710	0.7022	0.7082	0.7213	0.7297	0.7312
LINE	0.5337	0.5367	0.5689	0.6665	0.6789	0.6833
PTE	0.6315	0.6758	0.6993	0.7163	0.7043	0.7266
Hebe	**0.7576**	**0.8316**	**0.8621**	**0.8825**	**0.8845**	**0.8938**

Across the three tasks in the two datasets, vertically we observe Hebe achieves the best performance in general among all cases. This is due to the fact that Hebe models each event as a whole, which preserve more information. This property is particularly important when the observed data is sparse. 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% of events being sampled in the DBLP dataset, the 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 III and Table IV, with regard to Q2, we conclude that Hebe framework is relatively more robust to data sparsity.

VI. Related Work

Heterogeneous event data ubiquitously exist in real word and have been investigated in previous studies. Due to the heterogeneity of the objects involved in each event, [13], [21] proposed to abstract such data as heterogeneous information networks. Similar to the viewpoint of this paper, a network becomes heterogeneous if it contains more than two node types. Quite many methods were developed towards various applications including classification [21], clustering [13], and similarity search [13]. Recall that when the number of object types in each event is one, the heterogeneous event data reduce to homogeneous.

In particular for the embedding task, both [11] and [8] utilize the above abstraction to represent the heterogeneous event data in heterogeneous information networks. But instead of modeling proximity among objects in each event as a whole, [8], [11] decompose the multi-way interaction in each event into several pairwise interactions and then do the pairwise modeling separately. The same problem exists with some previously mentioned methods but for different tasks [13], [21]. Our model is substantially different since we directly model each hyperlink as a single component so that the proximity among objects can be better preserved.

In order to model the heterogeneous event data, we developed a hyperedge-based framework. Studies of similar flavor of higher-order data [24] have recently emerged for some tasks, such as recommender system [20], multi-relational learning [25], prediction [26], and clustering [27]. In [20], a tensor factorization model is designed specifically for tag recommendation; while we explore a more general framework for embedding to model the proximity of each event as a whole. [27] 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, [27] only models the events with one type of object; while HEBE supports multiple object types in multiple event types.

On the other hand, some dimension reduction methods can be adapted for event data embedding, such as principal component analysis [28], singular value decomposition [28], and non-negative matrix factorization [29]. 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 event data.

VII. CONCLUSION

In this paper, we proposed to learn object embedding in heterogeneous event data. In detail, we proposed the HEBE framework, which models participant objects in each event as a whole, resulting in more efficient information propagation. Based on the concept of hyperedge: HEBE models the proximity among the participating objects in the same hyperedge. 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 data sparseness.

We identify some future work for the HEBE framework. Firstly, it is general and could be adapted to many downstream applications, including recommender system and link prediction. Secondly, HEBE prefers term entities from short text. Additional work are needed to apply it to data with longer text. Thirdly, HEBE tends to fail as other exiting methods when there are noise object types. How to eliminate noise for meaningful object embedding learning remains an open problem. 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.

Acknowledgements. We wish to thank the anonymous reviewers for their helpful feedback of the manuscript. In addition, we wish to thank Jim Cai and Jingjing Wang for their patient proofreading. Research was sponsored in part by the U.S. Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), National Science Foundation IIS-1320617, IIS-1354329 and IIS 16-18481, HDTRA1-10-1-0120, 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 document are those of the author(s) and should not be interpreted as representing the official policies of the U.S. Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

REFERENCES

[1] D. Liben-Nowell and J. Kleinberg, “The link-prediction problem for social networks,” *JASIST*, vol. 58, no. 7, pp. 1019–1031, 2007.

[2] T. Opsahl and P. Panzarasa, “Clustering in weighted networks,” *Social networks*, vol. 31, no. 2, pp. 155–163, 2009.

[3] S. Bhagat, G. Cormode, and S. Muthukrishnan, “Node classification in social networks,” in *Social network data analytics*, 2011, pp. 115–148.

[4] Y. Koren, “The bellkor solution to the netflix grand prize,” 2009.

[5] A. Globerson, G. Chechik, F. Pereira, and N. Tishby, “Euclidean embedding of co-occurrence data,” in *NIPS*, 2004, pp. 497–504.

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

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

[8] S. Chang, W. Han, J. Tang, G.-J. Qi, C. C. Aggarwal, and T. S. Huang, “Heterogeneous network embedding via deep architectures,” in *KDD*, 2015, pp. 119–128.

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

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

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

[12] J. B. Tenenbaum, V. De Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

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

[14] C. Berge, *Hypergraphs: combinatorics of finite sets*. Elsevier, 1984, vol. 45.

[15] B. Recht, C. Re, S. Wright, and F. Niu, “Hogwild: A lock-free approach to parallelizing stochastic gradient descent,” in *NIPS*, 2011, pp. 693–701.

[16] A. Mnih and Y. W. Teh, “A fast and simple algorithm for training neural probabilistic language models,” in *ICML*, 2012, pp. 1751–1758.

[17] J. Silva and R. Willett, “Hypergraph-based anomaly detection of high-dimensional co-occurrences,” *PAMI, IEEE Transactions on*, vol. 31, no. 3, pp. 563–569, 2009.

[18] O. Levy, Y. Goldberg, and I. Dagan, “Improving distributional similarity with lessons learned from word embeddings,” *TACL*, vol. 3, pp. 211–225, 2015.

[19] Y. Goldberg and O. Levy, “word2vec explained: deriving mikolov et al.’s negative-sampling word-embedding method,” *arXiv preprint arXiv:1402.3722*, 2014.

[20] S. Rendle and L. Schmidt-Thieme, “Pairwise interaction tensor factorization for personalized tag recommendation,” in *WSDM*, 2010, pp. 81–90.

[21] M. Ji, J. Han, and M. Danilevsky, “Ranking-based classification of heterogeneous information networks,” in *KDD*, 2011, pp. 1298–1306.

[22] O. Levy and Y. Goldberg, “Neural word embedding as implicit matrix factorization,” in *NIPS*, 2014, pp. 2177–2185.

[23] J. Davis and M. Goadrich, “The relationship between precision-recall and roc curves,” in *ICML*. ACM, 2006, pp. 233–240.

[24] Q. Gu, H. Gui, and J. Han, “Robust tensor decomposition with gross corruption,” in *NIPS*, 2014, pp. 1422–1430.

[25] R. Jenatton, N. L. Roux, A. Bordes, and G. R. Obozinski, “A latent factor model for highly multi-relational data,” in *NIPS*, 2012, pp. 3167–3175.

[26] 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 *KDD*. ACM, 2014, pp. 1186–1195.

[27] A. R. Benson, D. F. Gleich, and J. Leskovec, “Tensor spectral clustering for partitioning higher-order network structures,” in *ICDM*, 2015, pp. 118–126.

[28] M. E. Wall, A. Rechtsteiner, and L. M. Rocha, “Singular value decomposition and principal component analysis,” in *A practical approach to microarray data analysis*. Springer, 2003, pp. 91–109.

[29] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in *NIPS*, 2001, pp. 556–562.