CARL: Content-Aware Representation Learning for Heterogeneous Networks

Chuxu Zhang
University of Notre Dame
czhang11@nd.edu

Ananthram Swami
Army Research Laboratory
ananthram.swami.civ@mail.mil

Nitesh V. Chawla
University of Notre Dame
nchawla@nd.edu

ABSTRACT

Heterogeneous networks not only present a challenge of heterogeneity in the types of nodes and relations, but also the attributes and content associated with the nodes. While recent works have looked at representation learning on homogeneous and heterogeneous networks, there is no work that has collectively addressed the following challenges: (a) the heterogeneous structural information of the network consisting of multiple types of nodes and relations; (b) the unstructured semantic content (e.g., text) associated with nodes; and (c) online updates due to incoming new nodes in growing network. We address these challenges by developing a Content-Aware Representation Learning model (CARL). CARL performs joint optimization of heterogeneous SkipGram and deep semantic encoding for capturing both heterogeneous structural closeness and unstructured semantic relations among all nodes, as function of node content, that exist in the network. Furthermore, an additional online update module is proposed for efficiently learning representations of incoming nodes. Extensive experiments demonstrate that CARL outperforms state-of-the-art baselines in various heterogeneous network mining tasks, such as link prediction, document retrieval, node recommendation and relevance search. We also demonstrate the effectiveness of the CARL's online update module through a category visualization study.

KEYWORDS

Heterogeneous Information Networks, Representation Learning, Network Embedding

1 INTRODUCTION

Heterogeneous information networks (HetNets) [16, 17], e.g., academic networks, encode rich information through multi-typed nodes, relationships, and attributes or content associated with nodes. For example, the academic networks can represent human-human relationship (authors), human-object relationship (author-paper or author-venue or author-organization), and object-object relationship (paper-paper, paper-venue, paper-organization). The nodes in this case (human and object) can carry attributes or semantic content (such as paper abstract). Given the multi-typed nodes, relationships, and content at the nodes, feature engineering has presented a unique challenge for network mining tasks such as relation mining [15, 23], relevance search [7, 16], personalized recommendation [9, 14, 27]. The typical feature engineering activity responds to the requirements of the network mining task for an application, requiring both a domain understanding and large exploratory search space for possible features. Not only this is expensive, but it also may not result in optimal performance.

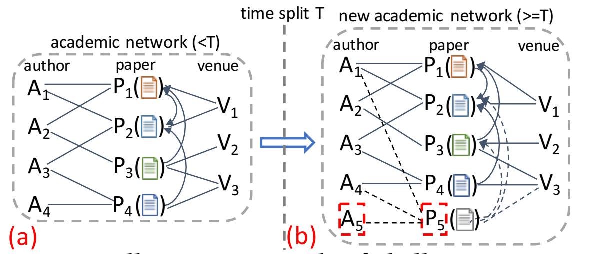


Figure 1: An Illustrative example of challenges in content-aware heterogeneous network representation learning.

To that end, we ask the question: *Can we generalize the feature engineering activity through representation learning on HetNets that addresses the complexity of multi-typed data in HetNets?* With the advent of deep learning, significant effort has been devoted to network representation learning in the last few years, starting with a focus on homogeneous networks [5, 12, 20] and more recently on HetNets [4]. The underlying theme of the models developed in these works is to automate the discovery of useful node latent features that can be further utilized in various network mining problems such as link prediction and node recommendation. However, these methods are limited in truly addressing the challenges of HetNets:

- (**C1**) HetNets include multiple types of nodes and relations. For example, in Figure 1(a), an academic network involves three types of nodes, i.e., *author*, *paper* and *venue*, which are connected by three types of relations, i.e., *author-write-paper*, *paper-cite-paper* and *paper-publish-venue*. Most of the previous models (e.g., Deepwalk and node2vec) employ homogeneous language models which make application to HetNets difficult. Thus challenge 1 is: how to extend homogeneous language model to heterogeneous network representation learning for maintaining structural closeness among multiple types of nodes and relations? We build on our prior work metapath2vec [4] for this.
- (**C2**) HetNets include both structural content (e.g., node type and relation connection) and unstructured semantic content (e.g., text). For example, in Figure 1(a), paper in academic network connects to author & venue and contains semantic text. The current models purely depend on structural content yet can not leverage unstructured content to infer semantic relations that are far away in network. To be more specific (as we will show in Section 4.5), given query author "Jure Leskovec", conventional techniques (e.g., node2vec and metapath2vec) tend to

XXX, XXX, XXX

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

return authors who collaborated with "Jure" due to structural relations bridged by paper or return authors who are different from "Jure" in specific research interests due to structural relations bridged by venue. Thus challenge 2 is: how to effectively incorporate unstructured content of nodes into a representation learning framework for capturing both structural closeness and unstructured semantic relations among all nodes?

- (**C3**) HetNets can grow with time. Current models are not able to handle this due to lack of an update strategy and it is impractical to re-run the model for each new node. For example, in Figure 1(b), new author A_5 co-authors with A_1 and A_4 on new paper P_5 after a given time split T. Thus challenge 3 is: how to efficiently learn representations of new nodes in a growing network?

Our proposed method CARL, a **c**ontent-**a**ware **r**epresentation **l**earning model for HetNets, addresses these challenges. Specifically, first, we develop a heterogeneous SkipGram model to maintain structural closeness among multiple types of nodes and relations. Next, we design two effective ways based on deep semantic encoding to incorporate unstructured content (i.e., text) of some types of nodes into heterogeneous SkipGram for capturing semantic relations. The negative sampling technique and the walk sampling based strategy are utilized to optimize and train the proposed models. Finally, we develop an online update module to efficiently learn representation of each new node by using its relations with existing nodes and the learned node representations.

To summarize, the main contributions of our work are:

- We formalize the problem of content-aware representation learning in HetNets and develop a model, i.e., CARL, to solve the problem. CARL performs joint optimization of heterogeneous SkipGram and deep semantic encoding.
- We design the corresponding optimization strategy and training algorithm to effectively learn node representations. The output representations are further utilized in various HetNet mining tasks, such as link prediction, document retrieval, node recommendation and relevance search, which demonstrate the superior performance of CARL over state-of-the-art baselines.
- We propose an update module in CARL to handle growing networks and conduct the category visualization study to show the effectiveness of this module.

2 PROBLEM DEFINITION

We first introduce the concepts of HetNets and random & meta-path walks, then formally define the problem of content-aware representation learning in HetNets.

Definition 2.1. (**Heterogeneous Networks**) A heterogeneous network [17] is defined as a network $G = (V, E, O_V, R_E)$ with multiple types of nodes V and links E. O_V and R_E represent the sets of object types and relation types. Each node $v \in V$ and each link $e \in E$ is associated with a node type mapping function $\psi_v : V \longrightarrow O_V$ and a link type mapping function $\psi_e : E \longrightarrow R_E$.

For example, in Figure 2(b), the academic network can be seen as a HetNet. The set of node types O_V includes *author* (A), *paper* (P) and *venue* (V). The set of link types R_E includes *author-write-paper*, *paper-cite-paper* and *paper-publish-venue*.

Definition 2.2. (**Random Walk**) A random walk [5] is defined as a node sequence $S_{v_0} = \{v_0, v_1, v_2, ..., v_{L-1}\}$ wherein the i-th node v_{i-1} in the walk is randomly selected from the neighbors of its predecessor v_{i-2}.

Definition 2.3. (**Meta-path Walk**) A meta-path walk [4] in HetNet is defined as a random walk guided by a specific meta-path scheme with the form of $\mathcal{P} \equiv o_1 \xrightarrow{r_1} o_2 \xrightarrow{r_2} \cdots \xrightarrow{r_{m-1}} o_m$, where $o_i \in O_V$, $r_i \in R_E$ and $r = r_1 * r_2 \cdots * r_{m-1}$ represents a compositional relation between relation types r_1 and r_m. Each meta-path walk recursively samples a specific $\mathcal{P}$ until it meets the given length.

Figure 2(b) shows examples of random walk and "APVPA" meta-path walk in the academic network.

Definition 2.4. (**Content-Aware Representation Learning in Heterogeneous Networks**) Given a HetNet with both structural and unstructured content at each node, the task is to design a model to learn a d-dimensional feature representations $\theta \in \mathbb{R}^{|V| \times d} (d \ll |V|)$, which can encode both structural closeness and unstructured semantic relations. Furthermore, the model is able to efficiently infer representation $\theta'_{v'} \in \mathbb{R}^d$ for each new node v' by using the learned representations θ.

For example, in the network of Figure 1, author and venue nodes contain structural content, i.e., node id, node type as well as link relations with others, and paper node contains both structural content and unstructured semantic content, e.g., abstract text. The output θ denotes representations of all existing nodes via the same latent space, which can be further utilized in various HetNet mining tasks. Besides, the learned representation $\theta'_{v'}$ of each new node v' can benefit different tasks for v' such as category assignment.

3 CARL FRAMEWORK

We present the framework of content-aware representation learning which will address the three challenges described in Section 1.

3.1 Heterogeneous Network Embedding (C1)

Inspired by word2vec [10] for learning distributed representation of words in corpus, Deepwalk [12] and node2vec [5] leverage SkipGram and random walks to learn node representations. However, we argued that those techniques focus on homogeneous networks and proposed metapath2vec [4] for HetNets by feeding meta-path walks to SkipGram. Similar to metapath2vec, we formulate the heterogeneous network representation learning as heterogeneous SkipGram (HSG) to address challenge **C1**. Specifically, given a HetNet $G = (V, E, O_V, R_E)$, the objective is to maximize the likelihood of each type of context node given the input node v:

$$o_1 = \arg\max_{\theta} \prod_{v \in V} \prod_{t \in O_V} \prod_{v_c \in N_t(v)} p(v_c | v; \theta) \tag{1}$$

where θ contains the representations of all nodes and $N_t(v)$ is the set of t-type context node of v which can be collected in different ways such as one-hop neighbors or surrounding neighbors in random walks. For example, in Figure 2(b), A_3 is structurally close to other authors (e.g., A_1 & A_4), papers (e.g., P_2 & P_4) and venues (e.g., V_1 & V_3). Thus objective o_1 is able to maintain structural closeness among multiple types of nodes and relations in G.

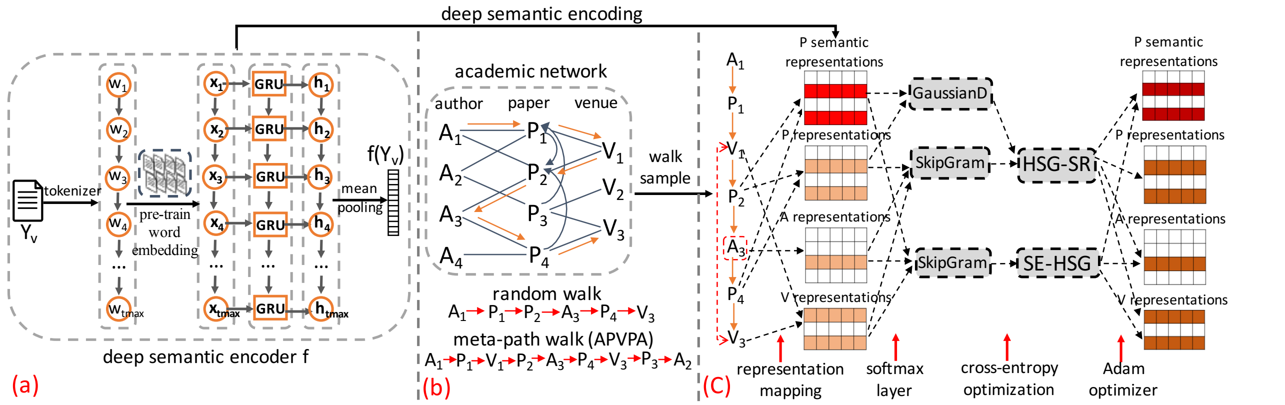


Figure 2: An illustrative example of CARL in the academic network: (a) paper semantic encoder based on gated recurrent neural network; (b) academic network and random & meta-path walks; (c) framework of the proposed models.

3.2 Incorporating Deep Semantic Encoder (C2)

The objective o_1 formulates structural closeness but ignores unstructured semantic relations. To address challenge **C2**, we design two ways to incorporate unstructured content of some types of nodes into heterogeneous network representation learning.

3.2.1 HSG with Unstructured Semantic Regularization (HSG-SR). One way is to tightly join HSG with the conditional probability of semantic constraint, leading to unstructured semantic regularization onto objective o_1. The objective is defined as:

$$o_2 = \arg\max_{\theta,\Phi} \prod_{v \in V} \prod_{t \in O_V} \prod_{v_c \in N_t(v)} p(v_c|v;\theta) \prod_{v \in V_S} p(\theta_v|Y_v;\Phi) \quad (2)$$

where V_S is the set of nodes with unstructured semantic content, Y_v represents unstructured content of node v and Φ are parameters of a deep semantic encoder that will be described later. The conditional probability $p(v_c|v;\theta)$ is defined as the heterogeneous softmax function: $p(v_c|v;\theta) = \frac{e^{\theta_{v_c} \cdot \theta_v}}{\sum_{v_k \in V_t} e^{\theta_{v_k} \cdot \theta_v}}$, where V_t is the set of t-type nodes. Besides, we model the conditional probability $p(\theta_v|Y_v;\Phi)$ as Gaussian prior: $p(\theta_v|Y_v;\Phi) = N(\theta_v|E_v, \sigma^2 I)$, where E_v denotes v's semantic representation encoded by deep learning architecture f: $E_v = f(Y_v)$. For example, in the network of Figure 2, V_S in o_2 is the set of papers and the formulation involves four kinds of representations, i.e., author representations, venue representations, paper representations and paper semantic representations. Notice that, there are two kinds of paper representations and we will use paper semantic representation for evaluation in Section 4.

另一种方法是将深度语义编码器的输出与HSG的输入连接起来，从而导致对o1的非结构语义增强

3.2.2 Unstructured Semantic Enhanced HSG (SE-HSG). Another way is to concatenate the output of deep semantic encoder with the input of HSG, leading to unstructured semantic enhancement onto objective o_1. The objective is defined as:

$$o_3 = \arg\max_{\theta,\Phi} \prod_{v \in V} \prod_{t \in O_V} \prod_{v_c \in N_t(v)} p(v_c|v;\theta;Y;\Phi) \quad (3)$$

where Y is the set of all unstructured semantic content of V_S. The conditional probability $p(v_c|v;\theta;Y;\Phi)$ is defined as the semantic enhanced heterogeneous softmax function: $p(v_c|v;\theta;Y;\Phi) = \frac{e^{\Theta_{v_c} \cdot \Theta_v}}{\sum_{v_k \in V_t} e^{\Theta_{v_k} \cdot \Theta_v}}$, where Θ denotes the enhanced representations. That is, $\Theta_v = E_v = f(Y_v)$ for $v \in V_S$ otherwise $\Theta_v = \theta_v$. For example, in the network of Figure 2, Y in o_3 is text content of all papers and the formulation involves three kinds of representations, i.e., author representations, venue representations and paper semantic representations. Notice that, θ in o_3 only denotes representations of nodes without unstructured content (e.g., author and venue in academic network), which is a bit different from o_2.

3.2.3 Unstructured Semantic Content Encoder. Both objectives o_2 and o_3 involve deep semantic encoding architecture. To encode unstructured content of some types of nodes into fixed length representations $E \in \mathbb{R}^{|V_S| \times d}$, we introduce gated recurrent units (GRU), a specific type of recurrent neural network, which has been widely adopted for many applications such as machine translation [3]. Figure 2(a) gives an illustrative example of this encoder for papers in the academic network. To be more specific, each paper's abstract is represented as a sequence of words: $\{w_1, w_2, \cdots, w_{t_{max}}\}$, followed by the word embeddings sequence: $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{t_{max}}\}$ trained by word2vec [10], where t_{max} is the maximum length of text. For each step t with the input word embedding $\mathbf{x}_t$ and previous hidden state vector $\mathbf{h}_{t-1}$, the current hidden state vector $\mathbf{h}_t$ is updated by $\mathbf{h}_t = \mathbf{GRU}(\mathbf{x}_t, \mathbf{h}_{t-1})$, where the GRU module is defined as:

$$\begin{aligned}
\mathbf{z}_t &= \sigma(\mathbf{A}_z \mathbf{x}_t + \mathbf{B}_z \mathbf{h}_{t-1}) \\
\mathbf{r}_t &= \sigma(\mathbf{A}_r \mathbf{x}_t + \mathbf{B}_r \mathbf{h}_{t-1}) \\
\hat{\mathbf{h}}_t &= \tanh[\mathbf{A}_h \mathbf{x}_t + \mathbf{B}_h(\mathbf{r}_t \circ \mathbf{h}_{t-1})] \\
\mathbf{h}_t &= \mathbf{z}_t \circ \mathbf{h}_{t-1} + (1 - \mathbf{z}_t) \circ \hat{\mathbf{h}}_t
\end{aligned} \quad (4)$$

where σ is the sigmoid function, $\mathbf{A}$ and $\mathbf{B}$ are parameter matrices of GRU network (i.e., Φ in objectives o_2 and o_3 includes $\mathbf{A}$ and $\mathbf{B}$), operator $\circ$ denotes element-wise multiplication, $\mathbf{z}_t$ and $\mathbf{r}_t$ are update gate vector and reset gate vector, respectively. The GRU network encodes word embeddings to deep semantic embeddings $\mathbf{h} \in \mathbb{R}^{t_{max} \times d}$, which is concatenated with a mean pooling layer to obtain the general semantic representation of paper. All of these steps construct the deep semantic encoder f. We have also explored other encoding architectures such as LSTM, bidirectional GRU and attention-based GRU, and obtain similar results. Thus we choose GRU since it has a concise structure and reduce training time.

3.3 Model Optimization and Training

We leverage the negative sampling technique [10] to optimize model and introduce the walk sampling based strategy for model training.

3.3.1 Optimization of HSG-SR. By applying negative sampling to the construction of softmax function, we can approximate the logarithm of $p(v_c|v;\theta)$ in objective o_2 as:

$$\log \sigma(\theta_{v_c} \cdot \theta_v) + \sum_{m=1}^{M} \mathbb{E}_{v_{c'} \sim P_t(v_{c'})} \log \sigma(-\theta_{v_{c'}} \cdot \theta_v) \qquad (5)$$

where M is the negative sample size and $P_t(v_{c'})$ is the pre-defined sampling distribution w.r.t. the t-type node. In our case, M makes little impact on the performance of proposed models. Thus we set $M = 1$ and obtain the cross entropy loss for optimization:

$$\log p(v_c|v;\theta) = \log \sigma(\theta_{v_c} \cdot \theta_v) + \log \sigma(-\theta_{v_{c'}} \cdot \theta_v) \qquad (6)$$

That is, for each context node v_c of v, we sample a negative node $v_{c'}$ according to $P_t(v_{c'})$. Besides, as $p(\theta_v|Y_v;\Phi) = N(\theta_v|E_v, \sigma^2 I)$, the logarithm of $p(\theta_v|Y_v;\Phi)$ in objective o_2 is equivalent to:

$$\log p(\theta_v|Y_v;\Phi) = -\left[\theta_v - E_v\right]^T \left[\theta_v - E_v\right] \qquad (7)$$

where $E_v = f(Y_v)$ and f is the deep semantic encoder. Therefore we rewrite objective o_2 as:

$$\begin{aligned} o_2 = \sum_{\langle v, v_c, v_{c'} \rangle \in T_{walk}} \Big\{ & \log \sigma(\theta_{v_c} \cdot \theta_v) + \log \sigma(-\theta_{v_{c'}} \cdot \theta_v) \\ & - \gamma \sum_{v_* \in T^S_{tri}} \left[\theta_{v_*} - f(Y_{v_*})\right]^T \left[\theta_{v_*} - f(Y_{v_*})\right] \Big\} \end{aligned} \qquad (8)$$

where γ is a trade-off factor and T_{walk} denotes the set of triplets $\langle v, v_c, v_{c'} \rangle$ collected by walk sampling on HetNet, which will be described later. Besides, T^S_{tri} is the set of nodes with unstructured content in each triplet of T_{walk} for semantic regularization. That is, $max\{|T^S_{tri}|\} = 3$.

3.3.2 Optimization of SE-HSG. Similar to HSG-SR, the logarithm of $p(v_c|v;\theta;Y;\Phi)$ in objective o_3 is approximated by:

$$\log\ p(v_c|v;\theta;Y;\Phi) = \log \sigma(\Theta_{v_c} \cdot \Theta_v) + \log \sigma(-\Theta_{v_{c'}} \cdot \Theta_v) \qquad (9)$$

where $\Theta_v = E_v = f(Y_v)$ for nodes with unstructured content otherwise $\Theta_v = \theta_v$. Therefore we rewrite objective o_3 as:

$$o_3 = \sum_{\langle v, v_c, v_{c'} \rangle \in T_{walk}} \log \sigma(\Theta_{v_c} \cdot \Theta_v) + \log \sigma(-\Theta_{v_{c'}} \cdot \Theta_v) \qquad (10)$$

As in HSG-SR, T_{walk} is the set of triplets $\langle v, v_c, v_{c'} \rangle$ collected from walk sequences on HetNet.

3.3.3 Model Training. Both optimized objectives o_2 and o_3 are accumulated on set T_{walk}. Similar to Deepwalk, node2vec and metapath2vec, we design a walk sampling strategy to generate T_{walk}. Specifically, first, we uniformly generate a set of random walks or meta-path walks S in HetNet. Then, for each node v in $S_i \in S$, we collect context node v_c which satisfies: $dist(v, v_c) \leq \tau$. That is, v's neighbors within distance τ in S_i. For example, in Figure 2(c), the context node of A_3 ($\tau = 2$) in the sample walk are V_1, P_2, P_4 and V_3. Finally, for each v_c, we sample a negative node $v_{c'}$ with the same node type of v_c according to $P_t(v_{c'}) \propto dg^{3/4}_{v_{c'}}$, where $dg_{v_{c'}}$ is the frequency of $v_{c'}$ in S. Furthermore, we design a mini-batch based Adam Optimizer [8] to train the model. Specifically, at each iteration, we sample a mini-batch of triplets in T_{walk} and accumulate the objective according to equation (8) or (10), then update the parameters via Adam. We repeat the training iterations until the change between two consecutive iterations is sufficiently small. Figure 2(c) shows an illustration of the framework of HSG-SR and SE-HSG on the academic network. The output representations θ and E can be utilized in various HetNet mining tasks, as we will show in Section 4.

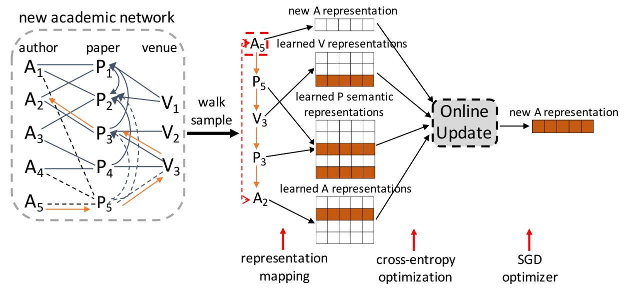


Figure 3: An Illustrative example of online update for new author in academic network.

3.4 Online Update for New Nodes (C3)

The optimization and training strategies in previous section learn representations of all existing nodes but they cannot be employed in an online situation. Considering the growing property of HetNets, we aim to design an online update module to efficiently learn representation of each new node and address challenge **C3**.

3.4.1 Representation of New Node with Unstructured Semantic Content. As we introduce deep architecture f to encode semantic representations of nodes with unstructured content, the optimized parameters Φ^* can be directly applied to infer representation of such new node without an extra training step. That is, $E_{v'} = f^*(Y_{v'})$ for new node v', where $Y_{v'}$ is the text content of v' and f^* is the learned semantic encoder with Φ^*. For example, in the academic network, we can use the learned paper semantic encoder to infer the representation of each new paper.

最优参数可直接应用于新节点

3.4.2 Representation of New Node without Unstructured Semantic Content. Inspired by eALS [6] for online recommendation, we make a reasonable assumption that each new node should not change the learned node representations too much from a global perspective. Accordingly, for each incoming node v' without unstructured content, we leverage objective o_3 to formulate the following objective:

$$o_4 = \sum_{\langle v', v_c, v_{c'} \rangle \in T^{v'}_{walk}} \log \sigma(\Theta^*_{v_c} \cdot \theta'_{v'}) + \log \sigma(-\Theta^*_{v_{c'}} \cdot \theta'_{v'}) \qquad (11)$$

where θ' denotes representation of v', Θ^* is the learned node representations via SE-HSG, i.e., $\Theta^*_v = E^*_v = f^*(Y_v)$ for $v \in V_S$ otherwise $\Theta^*_v = \theta^*_v$. That is, we take Θ^* as constant values in o_4. In addition, we leverage meta-path walks rooted at v' to collect triplet set $T^{v'}_{walk}$ for the training process of update module. Specifically, Figure 3 gives an illustrative example of online update for new author in academic network. First, we randomly sample a number of "APVPA" meta-path walks rooted at new author node A_5. The length of each walk equals the window distance τ ($\tau = 4$) and nodes in the walk are context node of A_5, e.g., P_5, V_3, P_3 and A_2. For each context node v_c, we randomly sample a negative node $v_{c'}$ with the same node type as v_c. Furthermore, the SGD optimizer is utilized to repeatly update θ' for each triplet in $T^{v'}_{walk}$ until the change between two consecutive iterations is sufficiently small. The proposed update module is efficient since only simple meta-path walk sampling and limited update steps on new node representation are performed. In the experimental evaluation, we only consider new author who writes new paper at a venue that exists in network, making the "APVPA" meta-path walk always feasible.

4 EXPERIMENTS

In this section, we conduct extensive experiments with the aim of answering the following research questions:

- (**RQ1**) How does CARL perform vs. state-of-the-art network representation learning models for different HetNet mining tasks, such as link prediction (**RQ1-1**), document retrieval (**RQ1-2**) and node recommendation (**RQ1-3**)? In addition, how do hyper-parameters impact CARL's performance in each task?
- (**RQ2**) What is the performance difference between CARL and baselines in relevance search w.r.t. each task in **RQ1**?
- (**RQ3**) What is the performance of CARL's online update module on the task for new nodes such as category assignment?

Notice that, although our model can be applied to or modified for different HetNets, we focus on experiments on the academic HetNet due to data availability.

4.1 Experimental Setup

4.1.1 Data. We use the AMiner computer science dataset [21], which is publicly available[1]. To avoid noise, we remove the papers published in venues (e.g., workshop) with limited publications and the instances without abstract text. In addition, topic of each area changes over time. For example, according to our analysis of the data, the most popular topics in data mining change from web mining and clustering (1996∼2005) to network mining and learning (2006∼2015). To make a thorough evaluation of CARL and verify its effectiveness for networks in different decades, we independently conduct experiments on two datasets, i.e., AMiner-*I* (1996∼2005) and AMiner-*II* (2006∼2015). As a result, AMiner-*I* contains 160,713 authors, 111,409 papers and 150 venues, AMiner-*II* contains 571,693 authors, 483,449 papers and 492 venues. The structure of the academic network used in this work is shown in Figure 1.

4.1.2 Comparison Baselines. We compare CARL with four state-of-the-art models, i.e., Deepwalk [12], LINE [20], node2vec [5] and metapath2vec [4]. Notice that, we use either random walk (rw) or meta-path walk (mw) to collect context node in HSG-SR and SE-HSG, resulting in four variants of CARL: $\text{CARL}^{rw}_{HSG-SR}$, $\text{CARL}^{mw}_{HSG-SR}$, $\text{CARL}^{rw}_{SE-HSG}$ and $\text{CARL}^{mw}_{SE-HSG}$.

4.1.3 Reproducibility. For fairness comparison, we use the same representation dimension d = 128 for all models. The window size τ = 7, the number of walks per node N = 10 and the walk length L = 30 are used for Deepwalk, node2vec, metapath2vec and CARL. The same set of parameters is used for CARL's online update module. The size of negative samples M is set to 5 for node2vec, LINE and metapath2vec. In addition, $\gamma = 1.0$ for CARL_{HSG-SR} and three meta-path schemes "APA", "APPA" and "APVPA" are jointly used to generate meta-path walks for CARL^{mw}. We employ TensorFlow to implement all variants of CARL and further conduct them on NVIDIA TITAN X GPU. Code will be available upon publication.

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

4.2 Link Prediction (RQ1-1)

Who will be your academic collaborators? As a response to **RQ1-1**, we design an experiment to evaluate CARL's performance on the author collaboration link prediction task.

4.2.1 Experimental Setting. Unlike past work [5] that randomly samples a portion of links for training and uses the remaining for evaluation, we consider a more realistic setting that splits training/test data via a given time stamp T. Specifically, first, the network before T is utilized to learn node representations. Then, the collaboration links before T are used to train a binary logistic classifier. Finally, the collaboration relations after T with equal number of random non-collaboration links are used to evaluate the trained classifier. In addition, only new collaborations among current authors (who appear before T) are considered and duplicated collaborations are removed from evaluation. For example, in Figure 1(b), A_1 and A_4 co-author a new paper P_5 after T. The classifier tends to predict new collaboration between A_1 and A_4 using previous links, e.g., collaboration between A_1 and A_3. The representation of link is formed by element-wise multiplication between representations of two end nodes. We use **Accuracy** and **F1 score** of binary classification as the evaluation metrics. Besides, T is set as 2003/2004 and 2013/2014 for AMiner-*I* and AMiner-*II*, respectively.

4.2.2 Results. The performances of different models are reported in Table 1. According to the table: (a) All variants of CARL perform better than baselines, demonstrating the effectiveness of incorporating unstructured semantic content to learn author representations; (b) $\text{CARL}^{mw}_{SE-HSG}$ achieves the best performances in all cases. The average improvements of $\text{CARL}^{mw}_{SE-HSG}$ over different baselines range from 10.9% to 41.0% and 6.7% to 30.9% on AMiner-*I* and AMiner-*II*, respectively. (c) CARL^{mw} outperforms CARL^{rw}, showing that meta-path walk is better than random walk for collecting context node in CARL. In addition, CARL_{SE-HSG} has better performance than CARL_{HSG-SR}, indicating that concatenating text encoder with heterogeneous SkipGram is more significant than taking text encoding as semantic regularization.

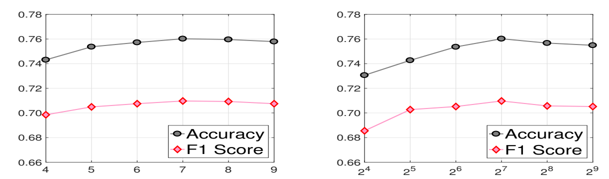
4.2.3 Parameter Sensitivity. We conduct experiment to analyze the impact of two key parameters, i.e., the window size τ of walk sampling and the representation dimension d. We investigate a specific parameter by changing its value and fixing the others. The prediction results of $\text{CARL}^{mw}_{SE-HSG}$ as a function of τ and d on AMiner-*II* (T = 2013) are shown in Figure 4. We see that: (a) With increasing of τ, accuracy and F1 score increase at first since a larger window means more useful context information. But when τ goes beyond a certain value, performances decrease slowly with τ possibly due to uncorrelated noise. The best τ is around 7; (b) Similar to τ, an appropriate value should be set for d such that the best node representations are learned. The optimal d is around 128.

4.3 Document Retrieval (RQ1-2)

Which relevant papers should be retrieved for your query? As a response to **RQ1-2**, we design an experiment to evaluate CARL's performance on the paper retrieval task.

Table 1: Collaboration prediction results comparison.

AMiner-*I*	T = 2004		T = 2003		Gain
	Accuracy	F1	Accuracy	F1	
Deepwalk	0.6341	0.4323	0.6244	0.4058	41.0%
LINE	0.6722	0.5263	0.6714	0.5231	20.8%
node2vec	0.6758	0.5291	0.6821	0.5409	18.9%
metapath2vec	0.7013*	0.5914*	0.7041*	0.5935*	10.9 %
$\text{CARL}^{rw}_{HSG-SR}$	0.7302	0.6561	0.7378	0.6623	–
$\text{CARL}^{mw}_{HSG-SR}$	0.7367	0.6618	0.7401	0.6635	–
$\text{CARL}^{rw}_{SE-HSG}$	0.7388	0.6579	0.7419	0.6648	–
$\text{CARL}^{mw}_{SE-HSG}$	**0.7482**	**0.6753**	**0.7525**	**0.6881**	–
AMiner-*II*	T = 2014		T = 2013		Gain
	Accuracy	F1	Accuracy	F1	
Deepwalk	0.6559	0.5024	0.6487	0.4833	30.9%
LINE	0.7034	0.6048	0.6956	0.5898	14.3%
node2vec	0.7136	0.6122	0.7066	0.5965	12.7%
metapath2vec	0.7299*	0.6628*	0.7254*	0.6512*	6.7%
$\text{CARL}^{rw}_{HSG-SR}$	0.7498	0.7061	0.7495	0.6946	–
$\text{CARL}^{mw}_{HSG-SR}$	0.7511	0.7084	0.7503	0.6962	–
$\text{CARL}^{rw}_{SE-HSG}$	0.7562	0.7125	0.7546	0.6978	–
$\text{CARL}^{mw}_{SE-HSG}$	**0.7627**	**0.7208**	**0.7602**	**0.7097**	–



(a) Window size τ (b) Representation dimension d

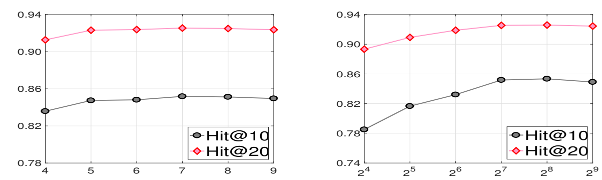
Figure 4: Parameter sensitivity in collaboration prediction.

4.3.1 Experimental Setting. As in the previous task, the network before T is utilized to learn node representations. The ground truth of relevance is assumed as the co-cited relation between two papers after T. For example, in Figure 1(b), new paper P_5 cites both previous papers P_2 and P_3. The model tends to retrieve P_2 when querying P_3 or retrieve P_3 when querying P_2. The relevant score of two papers is defined as the cosine similarity between representations of two papers. We use **HitRatio@k** as the evaluation metric. Due to large number of candidate papers, we follow the sampling strategy in [26] to reduce evaluation time. Specifically, for each evaluated paper, we randomly generate 100 negative samples for comparison with the true relevant paper. The hit ratio equals 1 if the true relevant paper is ranked in the top-k list of relevant score, otherwise 0. The overall result is the average value of HitRatio@k among all evaluated papers. The duplicated co-cited relations are removed from evaluation and k is set to 10 or 20. In addition, T is set as 2003/2004 and 2013/2014 for AMiner-*I* and AMiner-*II*, respectively.

4.3.2 Results. The results are reported in Table 2. From the table: (a) All variants of CARL achieve better performance than baselines, demonstrating the benefit of incorporating unstructured semantic content to learn paper representations; (b) The average improvements of $\text{CARL}^{mw}_{SE-HSG}$ over different baselines range from 2.3% to 16.4% and 6.3% to 29.1% on AMiner-*I* and AMiner-*II*, respectively; (c) CARL_{SE-HSG} outperforms CARL_{HSG-SR}, showing that semantic enhanced SkipGram is better than SkipGram with semantic regularization. However, CARL^{rw} has performance close to that of CARL^{mw}, indicating that meta-path walk has little impact for this task. It is reasonable since the paper representations depend on the deep semantic encoder in our model.

Table 2: Paper retrieval results comparison.

AMiner-*I*	T = 2004		T = 2003		Gain
	Hit@10	Hit@20	Hit@10	Hit@20	
Deepwalk	0.8120	0.8816	0.8217	0.8967	6.7%
LINE	0.7320	0.8130	0.7485	0.8380	16.4%
node2vec	0.8552*	0.9148*	0.8653*	0.9250*	2.3%
metapath2vec	0.8239	0.8910	0.8366	0.9081	5.3%
$\text{CARL}^{rw}_{HSG-SR}$	0.8685	0.9351	0.8696	0.9375	–
$\text{CARL}^{mw}_{HSG-SR}$	0.8673	0.9342	0.8722	0.9389	–
$\text{CARL}^{rw}_{SE-HSG}$	0.8741	0.9412	**0.8788**	0.9455	–
$\text{CARL}^{mw}_{SE-HSG}$	**0.8751**	**0.9425**	0.8783	**0.9470**	–
AMiner-*II*	T = 2014		T = 2013		Gain
	Hit@10	Hit@20	Hit@10	Hit@20	
Deepwalk	0.7460	0.8366	0.7392	0.8316	13.2%
LINE	0.6502	0.7453	0.6353	0.7382	29.1%
node2vec	0.8041*	0.8785*	0.7981*	0.8749*	6.3%
metapath2vec	0.7214	0.8136	0.7257	0.8201	15.9%
$\text{CARL}^{rw}_{HSG-SR}$	0.8263	0.9162	0.8141	0.9101	–
$\text{CARL}^{mw}_{HSG-SR}$	0.8287	0.9184	0.8153	0.9096	–
$\text{CARL}^{rw}_{SE-HSG}$	0.8619	**0.9281**	0.8516	0.9243	–
$\text{CARL}^{mw}_{SE-HSG}$	**0.8625**	0.9278	**0.8518**	**0.9254**	–



(a) Window size τ (b) Representation dimension d

Figure 5: Parameter sensitivity in paper retrieval.

4.3.3 Parameter Sensitivity. Following the same setup in link prediction, we investigate the impact of window size τ and representation dimension d on $\text{CARL}^{mw}_{SE-HSG}$'s performance on AMiner-*II* (T = 2013), as shown by Figure 5. It can be seen that: (a) The results are little sensitive to τ when $\tau \geq 7$. As we noted above, paper representations depend on the deep semantic encoder; (b) The dimension d plays significant role on generating paper representations. The best representations are learned when d is around 128 for the paper retrieval task.

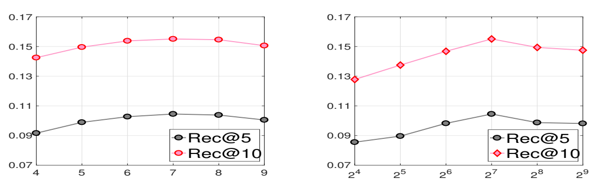
4.4 Node Recommendation (RQ1-3)

Which venues should be recommended to you? As a response to **RQ1-3**, we design an experiment to evaluate CARL's performance on the venue recommendation task.

4.4.1 Experimental Setting. As in the previous two tasks, the network before T is utilized to learn node representations. The ground truth of recommendation is based on author's appearance in venue after T. For example, in Figure 1(b), author A_1 writes

Table 3: Venue recommendation results comparison.

AMiner-*I*	T = 2004		T = 2003		Gain
	Rec@5	Rec@10	Rec@5	Rec@10	
Deepwalk	0.1051*	0.1628*	0.0864*	0.1403*	18.9%
LINE	0.0376	0.0677	0.0388	0.0717	178.7%
node2vec	0.0945	0.1570	0.0774	0.1386	27.2%
metapath2vec	0.0878	0.1527	0.0714	0.1395	33.3%
$\text{CARL}^{rw}_{HSG-SR}$	0.1156	0.1772	0.0988	0.1593	–
$\text{CARL}^{mw}_{HSG-SR}$	0.1208	0.1813	0.1050	0.1631	–
$\text{CARL}^{rw}_{SE-HSG}$	0.1225	0.1831	0.1054	0.1648	–
$\text{CARL}^{mw}_{SE-HSG}$	**0.1250**	**0.1852**	**0.1073**	**0.1667**	–
AMiner-*II*	T = 2014		T = 2013		Gain
	Rec@5	Rec@10	Rec@5	Rec@10	
Deepwalk	0.1039*	0.1549*	0.0925*	0.1384*	11.7%
LINE	0.0297	0.0470	0.0267	0.0433	275.6%
node2vec	0.0766	0.1182	0.0691	0.1074	47.8%
metapath2vec	0.0654	0.1077	0.0608	0.1001	66.1%
$\text{CARL}^{rw}_{HSG-SR}$	0.1092	0.1643	0.0977	0.1466	–
$\text{CARL}^{mw}_{HSG-SR}$	0.1116	0.1713	0.1032	0.1526	–
$\text{CARL}^{rw}_{SE-HSG}$	0.1107	0.1725	0.1023	0.1534	–
$\text{CARL}^{mw}_{SE-HSG}$	**0.1136**	**0.1742**	**0.1045**	**0.1551**	–



(a) Window size τ (b) Representation dimension d

Figure 6: Parameter sensitivity in venue recommendation.

new paper P_5 on V_3 after T, indicating A_1 is likely to accept V_3 recommendation before T. The preference score is defined as the cosine similarity between representations of author and venue. We use **Recall@k** as the evaluation metric and k is set to 5 or 10. In addition, the duplicated author-venue pairs are removed from evaluation. The reported score is the average value over all evaluated authors.

4.4.2 Results. The results are reported in Table 3. From the table: (a) All variants of CARL achieve better performance than baselines, showing the benefit of incorporating unstructured semantic content for learning author and venue representations; (b) The average improvements of $\text{CARL}^{mw}_{SE-HSG}$ over different baselines are significant, and range from 18.9% to 178.7% and 11.7% to 275.6% on AMiner-*I* and AMiner-*II*; (c) The results of different variants of CARL are close due to relative small recall values. However, we find that $\text{CARL}^{rw}_{HSG-SR}$ is the worst among four, indicating both meta-path walk and semantic enhanced SkipGram help improve the performance of CARL in the venue recommendation task.

4.4.3 Parameter Sensitivity. Figure 6 shows the impact of window size τ and feature dimension d on the performance of $\text{CARL}^{mw}_{SE-HSG}$ on AMiner-*II* (T = 2013). Accordingly, $\text{CARL}^{mw}_{SE-HSG}$ achieves the best results when τ is around 7 and d is around 128 for the venue recommendation task.

4.5 Relevance Search: Case Study (RQ2)

To answer **RQ2**, we present three case studies of relevance search on AMiner-*II* (T = 2013) to show the performance differences between $\text{CARL}^{mw}_{SE-HSG}$ and baselines. The ranking of each search result is based on the cosine similarity of representations.

4.5.1 Relevant Author Search. Table 4 lists the top-5 returned authors for query author **"Jure Leskovec"** of $\text{CARL}^{mw}_{SE-HSG}$ and two baselines, i.e., node2vec and metapath2vec, which achieves relatively better performances in the collaboration prediction task. According to this table: (a) most of returned authors of node2vec have collaboration relations with "Jure" before T, indicating that node2vec highly depends on structural closeness and cannot find relevant authors who are far away from "Jure" in the network; (b) metapath2vec returns some authors (e.g., P. Nguyen) who are different from "Jure" in their specific research interests, illustrating that "APVPA" meta-path walks (used by metapath2vec) may collect context node that are different from target node since it is common that authors bridged by the same venue study different research topics; (c) $\text{CARL}^{mw}_{SE-HSG}$ not only returns structurally close authors who have collaboration relations with "Jure" before T but also finds farther authors (e.g., D. Romero) who share similar research interest with "Jure", demonstrating $\text{CARL}^{mw}_{SE-HSG}$ captures both structural closeness and unstructured semantic relations for learning author representations.

4.5.2 Relevant Paper Search. Table 5 lists the top-5 returned papers for query paper **"When will it happen?"** of $\text{CARL}^{mw}_{SE-HSG}$ and the best baseline node2vec in paper retrieval task. From this table: (a) all returned papers of node2vec are written by at least one author in query paper, showing that node2vec only returns structurally close papers but has difficulty finding farther semantically related papers; (b) $\text{CARL}^{mw}_{SE-HSG}$ not only returns structurally close papers which have common authors with query paper but also finds semantically related papers without authorship overlapping, showing that $\text{CARL}^{mw}_{SE-HSG}$ utilizes both structural content and unstructured semantic content for learning paper representations.

4.5.3 Relevant Author-Venue Search. Table 6 lists the top-10 returned venues for query author **"Chi Wang"** of $\text{CARL}^{mw}_{SE-HSG}$ and two baselines, i.e., Deepwalk and node2vec, which have relatively better performances on the venue recommendation task. We find that: (a) Deepwalk and node2vec recommend both data mining (e.g., KDD & ICDM) and database (e.g., SIGMOD & PVLDB) venues to "Chi" since some of his works cite database papers and some of his co-authors focus on database research, which illustrates that both Deepwalk and node2vec return structurally close venues, some of which are not the most suitable ones; (b) most of the venues in $\text{CARL}^{mw}_{SE-HSG}$'s recommendation list belong to data mining related areas, demonstrating that incorporating semantic content helps learn better representations of author and venue.

4.6 Category Visualization of New Nodes (RQ3)

The previous experiments and case studies demonstrate the effectiveness of CARL in learning representations of current nodes, i.e., nodes that exist in HetNet before T. As described in Section 3.4, the learned semantic encoder of CARL can be directly applied to infer representation of each incoming node with semantic content.

Table 4: Case study of relevant author search. "Coauthor-b" denotes whether two authors have a collaboration relation before T and "Similar-I" represents whether two authors have similar research interests.

Query: Jure Leskovec (2009 SIGKDD Dissertation Award, Research Interest: Network Mining & Social Computing)									
Rank	node2vec			metapath2vec			CARL$^{mw}_{SE-HSG}$		
	Author	Coauthor-b ?	Similar-I ?	Author	Coauthor-b ?	Similar-I ?	Author	Coauthor-b ?	Similar-I ?
1	S. Kairam	✓	✓	L. Backstrom	✓	✓	J. Kleinberg	✓	✓
2	M. Rodriguez	✓	✓	P. Nguyen	✗	✗	D. Romero	✗	✓
3	D. Wang	✓	✓	S. HanhijÃďrvi	✗	✗	A. Dasgupta	✓	✓
4	J. Yang	✓	✓	S. Myers	✓	✓	L. Backstrom	✓	✓
5	A. Jaimes	✗	✗	V. Lee	✗	✓	G. Kossinets	✗	✓

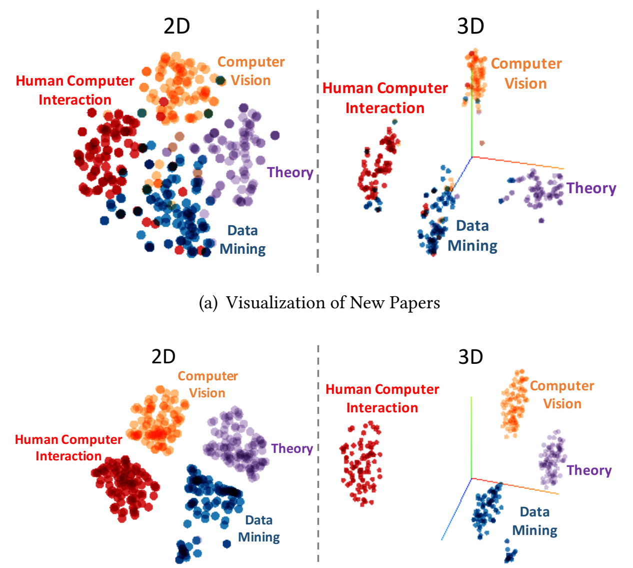
Table 5: Case study of relevant paper search.

Query: When will it happen?: relationship prediction in heterogeneous information networks (WSDM2012, Citation > 180), A: Y. Sun, J. Han, C. Aggarwal, N. Chawla		
Model	Rank	Returned Paper
node2vec	1	Co-author relationship prediction in heterogeneous bibliographic networks (ASONAM2011), A: Y. Sun, R. Barber, M. Gupta, C. Aggarwal, J. Han
	2	A framework for classification and segmentation of massive audio data streams (KDD2007), A: C. Aggarwal
	3	Mining heterogeneous information networks: the next frontier (KDD2012), A: J. Han
	4	Ranking-based classification of heterogeneous information networks (KDD2011), A: M. Ji, J. Han, M. Danilevsky
	5	Evolutionary clustering and analysis of bibliographic networks (ASONAM2011), A: M. Gupta, C. Aggarwal, J. Han, Y. Sun
CARL$^{mw}_{SE-HSG}$	1	Collective prediction of multiple types of links in heterogeneous information networks (ICDM2014), A: B. Cao, X. Kong, P. Yu
	2	Community detection in incomplete information networks (WWW2012), A: W. Lin, X. Kong, P. Yu, Q. Wu, Y. Jia, C. Li
	3	Meta path-based collective classification in heterogeneous information networks (CIKM2012), A: X. Kong, P. Yu, Y. Ding, D. Wild
	4	Ranking-based classification of heterogeneous information networks (KDD2011), A: M. Ji, J. Han, M. Danilevsky
	5	Fast computation of SimRank for static and dynamic information networks (EDBT2010), A: C. Li, J. Han, G. He, X. Jin, Y. Sun, Y. Yu, T. Wu

Table 6: Case study of relevant author-venue search.

Query: Chi Wang (2015 SIGKDD Dissertation Award) Research Interest: Unstructured Data/Text Mining			
Rank	Deepwalk	node2vec	CARL$^{mw}_{SE-HSG}$
1	KDD	KDD	KDD
2	CIKM	WSDM	CIKM
3	DASFAA	SIGMOD	ICDM
4	SIGMOD	PVLDB	PAKDD
5	ICDM	WWW	WWW
6	WSDM	ICDM	WWWC
7	ICDE	EDBT	TKDE
8	PVLDB	PAKDD	KAIS
9	WWW	GRC	DASFAA
10	EDBT	FPGA	WSDM

Besides, CARL's online update module can efficiently learn representation of each new node without semantic content. To answer **RQ3** and show the effectiveness of learned new node representations, we employ the Tensorflow embedding projector to visualize the new paper representations inferred by paper semantic encoder and new author representations learned by online update module in sequence. Figure 7 shows the results of new paper and author nodes (appearing after T) of four selected research categories, i.e., Data Mining (DM), Computer Vision (CV), Human Computer Interaction (HCI) and Theory, on AMiner-*II* (T = 2013). Specifically, we choose three top venues[2] for each area. Each new paper is assigned according to venue's area and the category of each author



(a) Visualization of New Papers

(b) Visualization of New Authors

Figure 7: Representation visualizations of new paper and author nodes in four selected research categories.

is assigned to the area with the majority of his/her publications. We randomly sample 100 new papers/authors of each area. According to Figure 7: (a) The representations of new papers in the same category cluster closely and can be well discriminated from others for both 2D and 3D visualizations, indicating that the semantic

[2] DM: KDD, WSDM, ICDM. CV: CVPR, ICCV, ECCV. HCI: CHI, CSCW, UIST. T: SODA, STOC, FOCS

encoder achieves satisfactory performance in inferring semantic representations of new papers. Notice that, papers belong to DM have few intersections with the other three since semantic content of few DM papers are quite similar to CV, HCI and Theory papers w.r.t. model, application and theoretical basis, respectively. (b) The representations of new authors in the same category are clearly discriminated from others without intersection for both 2D and 3D visualizations, which demonstrates the effectiveness of online update module in learning new author representations.

5 RELATED WORK

In the past decade, many works have been devoted to mining HetNets for different applications, such as relevance search [2, 7, 16, 28], node clustering [17, 18], personalized recommendation [14, 26, 27].

The network representation learning has gained a lot of attention in the last few years. Some walk sampling based models [4, 5, 12] have been proposed to learn vectorized node representations that can be further utilized in various tasks in network. Specifically, inspired by word2vec [10] for learning distributed representations of words in text corpus, Perozzi et al. developed the innovative Deepwalk [12] which introduces node-context concept in network (analogy to word-context) and feeds a set of random walks over network (analogy to "sentences") to SkipGram for learning node representations. In order to deal with neighborhood diversity, Grover & Leskovec suggested taking biased random walks (a mixture of BFS and DFS) as the input of SkipGram. More recently, we argued that those models are not able to truly tackle network heterogeneity and proposed metapath2vec [4] for heterogeneous network representation learning by feeding meta-path walks to SkipGram. In addition, many other models have been proposed [1, 11, 13, 19, 20, 22, 24, 25], such as PTE [19] for text data embedding, HNE [1] for image-text data embedding and NetMF [13] for unifying network representation models as matrix factorization.

Our work furthers the investigation of network representation learning by developing a content-aware representation learning model CARL for HetNets. Unlike previous models, CARL leverages both structural closeness and unstructured semantic relations to learn node representations, and contains an online update module for learning representations of new nodes.

6 CONCLUSION

In this paper, we formalize the problem of content-aware representation learning in HetNets and propose a novel model CARL to solve the problem. CARL performs joint optimization of heterogeneous SkipGram and deep semantic encoding for capturing both structural closeness and unstructured semantic relations in a HetNet. Furthermore, an online update module is designed to efficiently learn representations of new nodes. Extensive experiments demonstrate that CARL outperforms state-of-the-art baselines in various HetNet mining tasks, such as link prediction, document retrieval, node recommendation and relevance search. Besides, the online update module achieves satisfactory performance, as reflected by category visualization of new nodes. In the future, we plan to design the dynamic heterogeneous network representation learning models by using time series information of nodes and links.

REFERENCES

[1] Shiyu Chang, Wei Han, Jiliang Tang, Guo-Jun Qi, Charu C Aggarwal, and Thomas S Huang. 2015. Heterogeneous network embedding via deep architectures. In *KDD*. 119–128.

[2] Ting Chen and Yizhou Sun. 2017. Task-Guided and Path-Augmented Heterogeneous Network Embedding for Author Identification. In *WSDM*. 295–304.

[3] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv:1406.1078* (2014).

[4] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *KDD*. 135–144.

[5] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *KDD*. 855–864.

[6] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *SIGIR*. 549–558.

[7] Zhipeng Huang, Yudian Zheng, Reynold Cheng, Yizhou Sun, Nikos Mamoulis, and Xiang Li. 2016. Meta structure: Computing relevance in large heterogeneous information networks. In *KDD*. 1595–1604.

[8] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[9] Xiaozhong Liu, Yingying Yu, Chun Guo, and Yizhou Sun. 2014. Meta-path-based ranking with pseudo relevance feedback on heterogeneous graph for citation recommendation. In *CIKM*. 121–130.

[10] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*. 3111–3119.

[11] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. 2016. Asymmetric transitivity preserving graph embedding. In *KDD*. 1105–1114.

[12] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *KDD*. 701–710.

[13] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. 2018. Network Embedding as Matrix Factorization: UnifyingDeepWalk, LINE, PTE, and node2vec. In *WSDM*.

[14] Xiang Ren, Jialu Liu, Xiao Yu, Urvashi Khandelwal, Quanquan Gu, Lidan Wang, and Jiawei Han. 2014. Cluscite: Effective citation recommendation by information network-based clustering. In *KDD*. 821–830.

[15] Yizhou Sun, Jiawei Han, Charu C Aggarwal, and Nitesh V Chawla. 2012. When will it happen?: relationship prediction in heterogeneous information networks. In *WSDM*. 663–672.

[16] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. 2011. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *VLDB* 4, 11 (2011), 992–1003.

[17] Yizhou Sun, Brandon Norick, Jaiwei Han, Xifeng Yan, Philip Yu, and Xiao Yu. 2012. PathSelClus: Integrating Meta-Path Selection with User-Guided Object Clustering in Heterogeneous Information Networks. In *KDD*. 1348–1356.

[18] Yizhou Sun, Yintao Yu, and Jiawei Han. 2009. Ranking-based clustering of heterogeneous information networks with star network schema. In *KDD*. 797–806.

[19] Jian Tang, Meng Qu, and Qiaozhu Mei. 2015. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *KDD*. 1165–1174.

[20] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *WWW*. 1067–1077.

[21] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. Arnetminer: extraction and mining of academic social networks. In *KDD*. 990–998.

[22] Ke Tu, Peng Cui, Xiao Wang, Fei Wang, and Wenwu Zhu. 2018. Structural Deep Embedding for Hyper-Networks. In *AAAI*.

[23] Chi Wang, Jiawei Han, Yuntao Jia, Jie Tang, Duo Zhang, Yintao Yu, and Jingyi Guo. 2010. Mining advisor-advisee relationships from research publication networks. In *KDD*. 203–212.

[24] Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In *KDD*. 1225–1234.

[25] Hongwei Wang, Jia Wang, Jialin Wang, Miao Zhao, Weinan Zhang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. GraphGAN: Graph Representation Learning with Generative Adversarial Nets. In *AAAI*.

[26] Carl Yang, Lanxiao Bai, Chao Zhang, Quan Yuan, and Jiawei Han. 2017. Bridging Collaborative Filtering and Semi-Supervised Learning: A Neural Approach for POI Recommendation. In *KDD*. 1245–1254.

[27] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. 2014. Personalized entity recommendation: A heterogeneous information network approach. In *WSDM*. 283–292.

[28] Chuxu Zhang, Chao Huang, Lu Yu, Xiangliang Zhang, and Nitesh Chawla. 2018. Camel: Content-Aware and Meta-path Augmented Metric Learning for Author Identification. In *WWW*.