# metapath2vec: Scalable Representation Learning for Heterogeneous Networks

Yuxiao Dong*
Microsoft Research
Redmond, WA 98052
yuxdong@microsoft.com

Nitesh V. Chawla
University of Notre Dame
Notre Dame, IN 46556
nchawla@nd.edu

Ananthram Swami
Army Research Laboratory
Adelphi, MD 20783
ananthram.swami.civ@mail.mil

## ABSTRACT

We study the problem of representation learning in heterogeneous networks. Its unique challenges come from the existence of multiple types of nodes and links, which limit the feasibility of the conventional network embedding techniques. We develop two scalable representation learning models, namely *metapath2vec* and *metapath2vec++*. The *metapath2vec* model formalizes meta-path-based random walks to construct the heterogeneous neighborhood of a node and then leverages a heterogeneous skip-gram model to perform node embeddings. The *metapath2vec++* model further enables the simultaneous modeling of structural and semantic correlations in heterogeneous networks. Extensive experiments show that *metapath2vec* and *metapath2vec++* are able to not only outperform state-of-the-art embedding models in various heterogeneous network mining tasks, such as node classification, clustering, and similarity search, but also discern the structural and semantic correlations between diverse network objects.

## CCS CONCEPTS

**•Information systems →Social networks; •Computing methodologies →Unsupervised learning; Learning latent representations;** *Knowledge representation and reasoning;*

## KEYWORDS

Network Embedding; Heterogeneous Representation Learning; Latent Representations; Feature Learning; Heterogeneous Information Networks

**ACM Reference format:**
Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable Representation Learning for Heterogeneous Networks. In *Proceedings of KDD '17, August 13-17, 2017, Halifax, NS, Canada, ,* 10 pages.
DOI: http://dx.doi.org/10.1145/3097983.3098036

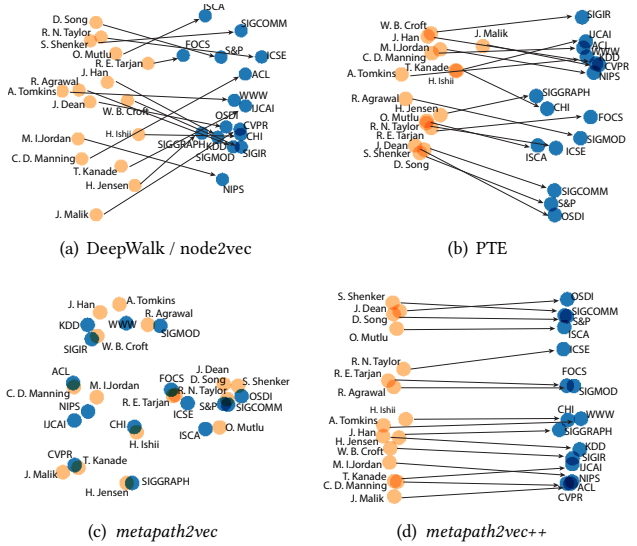



**Figure 1: 2D PCA projections of the 128D embeddings of 16 top CS conferences and corresponding high-profile authors.**

## 1 INTRODUCTION

Neural network-based learning models can represent latent embeddings that capture the internal relations of rich, complex data across various modalities, such as image, audio, and language [15]. Social and information networks are similarly rich and complex data that encode the dynamics and types of human interactions, and are similarly amenable to representation learning using neural networks. In particular, by mapping the way that people choose friends and maintain connections as a "social language," recent advances in natural language processing (NLP) [3] can be naturally applied to network representation learning, most notably the group of NLP models known as word2vec [17, 18]. A number of recent research publications have proposed word2vec-based network representation learning frameworks, such as DeepWalk [22], LINE [30], and node2vec [8]. Instead of handcrafted network feature design, these representation learning methods enable the automatic discovery of useful and meaningful (latent) features from the "raw networks."

However, these work has thus far focused on representation learning for homogeneous networks—representative of singular type of nodes and relationships. Yet a large number of social and information networks are heterogeneous in nature, involving diversity of node types and/or relationships between nodes [25]. These heterogeneous networks present unique challenges that cannot be handled by representation learning models that are specifically designed for homogeneous networks. Take, for example, a heterogeneous academic network: How do we effectively preserve the concept of "word-context" among multiple types of nodes, e.g., authors, papers, venues, organizations, etc.? Can random walks, such those used in DeepWalk and node2vec, be applied to networks

*This work was done when Yuxiao was a Ph.D. student at University of Notre Dame.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
*KDD '17, August 13-17, 2017, Halifax, NS, Canada*
© 2017 ACM. 978-1-4503-4887-4/17/08…$15.00
DOI: http://dx.doi.org/10.1145/3097983.3098036

**Table 1: Case study of similarity search in the heterogeneous DBIS data used in [26].**

| Method | PathSim [26] | | DeepWalk / node2vec [8, 22] | | LINE (1st+2nd) [30] | | PTE [29] | | *metapath2vec* | | *metapath2vec++* | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Input | meta-paths | | heterogeneous random walk paths | | heterogeneous edges | | heterogeneous edges | | probabilistic meta-paths | | probabilistic meta-paths | |
| Query | PKDD | C. Faloutsos | PKDD | C. Faloutsos | PKDD | C. Faloutsos | PKDD | C. Faloutsos | PKDD | C. Faloutsos | PKDD | C. Faloutsos |
| 1 | ICDM | J. Han | R. S. | J. Pan | W. K. | C. Aggarwal | KDD | C. Aggarwal | A. S. | C. Aggarwal | KDD | R. Agrawal |
| 2 | SDM | R. Agrawal | M. N. | H. Tong | S. A. | P. Yu | ICDM | P. Yu | M. B. | J. Pei | PAKDD | J. Han |
| 3 | PAKDD | J. Pei | R. P. | H. Yang | A. B. | D. Gunopulos | SDM | Y. Tao | P. B. | P. Yu | ICDM | J. Pei |
| 4 | KDD | C. Aggarwal | G. G. | R. Filho | M. S. | N. Koudas | DMKD | N. Koudas | M. S. | H. Cheng | DMKD | C. Aggarwal |
| 5 | DMKD | H. Jagadish | F. J. | R. Chan | S. A. | M. Vlachos | PAKDD | R. Rastogi | M. K. | V. Ganti | SDM | P. Yu |

of multiple types of nodes? Can we directly apply homogeneous network-oriented embedding architectures (e.g., skip-gram) to heterogeneous networks?

By solving these challenges, the latent heterogeneous network embeddings can be further applied to various network mining tasks, such as node classification [13], clustering [27, 28], and similarity search [26, 35]. In contrast to conventional meta-path-based methods [25], the advantage of latent-space representation learning lies in its ability to model similarities between nodes without connected meta-paths. For example, if authors have never published papers in the same venue—imagine one publishes 10 papers all in NIPS and the other has 10 publications all in ICML; their "APCPA"-based PathSim similarity [26] would be zero—this will be naturally overcome by network representation learning.

**Contributions.** We formalize the heterogeneous network representation learning problem, where the objective is to simultaneously learn the low-dimensional and latent embeddings for multiple types of nodes. We present the *metapath2vec* and its extension *metapath2vec++* frameworks. The goal of *metapath2vec* is to maximize the likelihood of preserving both the structures and semantics of a given heterogeneous network. In *metapath2vec*, we first propose meta-path [25] based random walks in heterogeneous networks to generate heterogeneous neighborhoods with network semantics for various types of nodes. Second, we extend the skip-gram model [18] to facilitate the modeling of geographically and semantically close nodes. Finally, we develop a heterogeneous negative sampling-based method, referred to as *metapath2vec++*, that enables the accurate and efficient prediction of a node's heterogeneous neighborhood.

The proposed *metapath2vec* and *metapath2vec++* models are different from conventional network embedding models, which focus on homogeneous networks [8, 22, 30]. Specifically, conventional models suffer from the identical treatment of different types of nodes and relations, leading to the production of indistinguishable representations for heterogeneous nodes—as evident through our evaluation. Further, the *metapath2vec* and *metapath2vec++* models also differ from the Predictive Text Embedding (PTE) model [29] in several ways. First, PTE is a semi-supervised learning model that incorporates label information for text data. Second, the heterogeneity in PTE comes from the text network wherein a link connects two words, a word and its document, and a word and its label. Essentially, the raw input of PTE is words and its output is the embedding of each word, rather than multiple types of objects.

We summarize the differences of these methods in Table 1, which lists their input to learning algorithms, as well as the top-five similarity search results in the DBIS network for the same two queries used in [26] (see Section 4 for details). By modeling the heterogeneous neighborhood and further leveraging the heterogeneous negative sampling technique, *metapath2vec++* is able to achieve the best top-five similar results for both types of queries. Figure 1 shows the visualization of the 2D projections of the learned embeddings for 16 CS conferences and corresponding high-profile researchers in each field. Remarkably, we find that *metapath2vec++* is capable of automatically organizing these two types of nodes and implicitly learning the internal relationships between them, suggested by the similar directions and distances of the arrows connecting each pair. For example, it learns J. Dean → OSDI and C. D. Manning → ACL. *metapath2vec* is also able to group each author-conference pair closely, such as R. E. Tarjan and FOCS. All of these properties are not discoverable from conventional network embedding models.

To summarize, our work makes the following contributions:

(1) Formalizes the problem of heterogeneous network representation learning and identifies its unique challenges resulting from network heterogeneity.
(2) Develops effective and efficient network embedding frameworks, *metapath2vec* & *metapath2vec++*, for preserving both structural and semantic correlations of heterogeneous networks.
(3) Through extensive experiments, demonstrates the efficacy and scalability of the presented methods in various heterogeneous network mining tasks, such as node classification (achieving relative improvements of 35–319% over benchmarks) and node clustering (achieving relative gains of 13–16% over baselines).
(4) Demonstrates the automatic discovery of internal semantic relationships between different types of nodes in heterogeneous networks by *metapath2vec* & *metapath2vec++*, not discoverable by existing work.

## 2 PROBLEM DEFINITION

We formalize the representation learning problem in heterogeneous networks, which was first briefly introduced in [21]. In specific, we leverage the definition of heterogeneous networks in [25, 27] and present the learning problem with its inputs and outputs.

*Definition 2.1.* **A Heterogeneous Network** is defined as a graph $G = (V, E, T)$ in which each node $v$ and each link $e$ are associated with their mapping functions $\phi(v) : V \rightarrow T_V$ and $\varphi(e) : E \rightarrow T_E$, respectively. $T_V$ and $T_E$ denote the sets of object and relation types, where $|T_V| + |T_E| > 2$.

For example, one can represent the academic network in Figure 2(a) with authors (A), papers (P), venues (V), organizations (O) as nodes, wherein edges indicate the coauthor (A–A), publish (A–P,

P–V), affiliation (O–A) relationships. By considering a heterogeneous network as input, we formalize the problem of heterogeneous network representation learning as follows.

Problem 1. ***Heterogeneous Network Representation Learning:*** *Given a heterogeneous network $G$, the task is to learn the $d$-dimensional latent representations $\mathbf{X} \in \mathbb{R}^{|V| \times d}$, $d \ll |V|$ that are able to capture the structural and semantic relations among them.*

The output of the problem is the low-dimensional matrix $\mathbf{X}$, with the $v^{th}$ row—a $d$-dimensional vector $X_v$—corresponding to the representation of node $v$. Notice that, although there are different types of nodes in $V$, their representations are mapped into the same latent space. The learned node representations can benefit various heterogeneous network mining tasks. For example, the embedding vector of each node can be used as the feature input of node classification, clustering, and similarity search tasks.

The main challenge of this problem comes from the network heterogeneity, wherein it is difficult to directly apply homogeneous language and network embedding methods. The premise of network embedding models is to preserve the proximity between a node and its neighborhood (context) [8, 22, 30]. In a heterogeneous environment, how do we define and model this 'node–neighborhood' concept? Furthermore, how do we optimize the embedding models that effectively maintain the structures and semantics of multiple types of nodes and relations?

## 3 THE *METAPATH2VEC* FRAMEWORK

We present a general framework, *metapath2vec*, which is capable of learning desirable node representations in heterogeneous networks. The objective of *metapath2vec* is to maximize the network probability in consideration of multiple types of nodes and edges.

### 3.1 Homogeneous Network Embedding

We, first, briefly introduce the word2vec model and its application to homogeneous network embedding tasks. Given a text corpus, Mikolov et al. proposed *word2vec* to learn the distributed representations of words in a corpus [17, 18]. Inspired by it, DeepWalk [22] and node2vec [8] aim to map the word-context concept in a text corpus into a network. Both methods leverage random walks to achieve this and utilize the skip-gram model to learn the representation of a node that facilitates the prediction of its structural context—local neighborhoods—in a homogeneous network. Usually, given a network $G = (V, E)$, the objective is to maximize the network probability in terms of local structures [8, 18, 22], that is:

$$\arg\max_{\theta} \prod_{v \in V} \prod_{c \in N(v)} p(c|v;\theta) \tag{1}$$

where $N(v)$ is the neighborhood of node $v$ in the network $G$, which can be defined in different ways such as $v$'s one-hop neighbors, and $p(c|v;\theta)$ defines the conditional probability of having a context node $c$ given a node $v$.

### 3.2 Heterogeneous Network Embedding: *metapath2vec*

To model the heterogeneous neighborhood of a node, *metapath2vec* introduces the heterogeneous skip-gram model. To incorporate the heterogeneous network structures into skip-gram, we propose meta-path-based random walks in heterogeneous networks.

**Heterogeneous Skip-Gram.** In *metapath2vec*, we enable skip-gram to learn effective node representations for a heterogeneous network $G = (V, E, T)$ with $|T_V| > 1$ by maximizing the probability of having the heterogeneous context $N_t(v), t \in T_V$ given a node $v$:

$$\arg\max_{\theta} \sum_{v \in V} \sum_{t \in T_V} \sum_{c_t \in N_t(v)} \log p(c_t|v;\theta) \tag{2}$$

where $N_t(v)$ denotes $v$'s neighborhood with the $t^{th}$ type of nodes and $p(c_t|v;\theta)$ is commonly defined as a softmax function [3, 7, 18, 24], that is: $p(c_t|v;\theta) = \frac{e^{X_{c_t} \cdot X_v}}{\sum_{u \in V} e^{X_u \cdot X_v}}$, where $X_v$ is the $v^{th}$ row of $\mathbf{X}$, representing the embedding vector for node $v$. For illustration, consider the academic network in Figure 2(a), the neighborhood of one author node $a_4$ can be structurally close to other authors (e.g., $a_2$, $a_3$ & $a_5$), venues (e.g., ACL & KDD), organizations (CMU & MIT), as well as papers (e.g., $p_2$ & $p_3$).

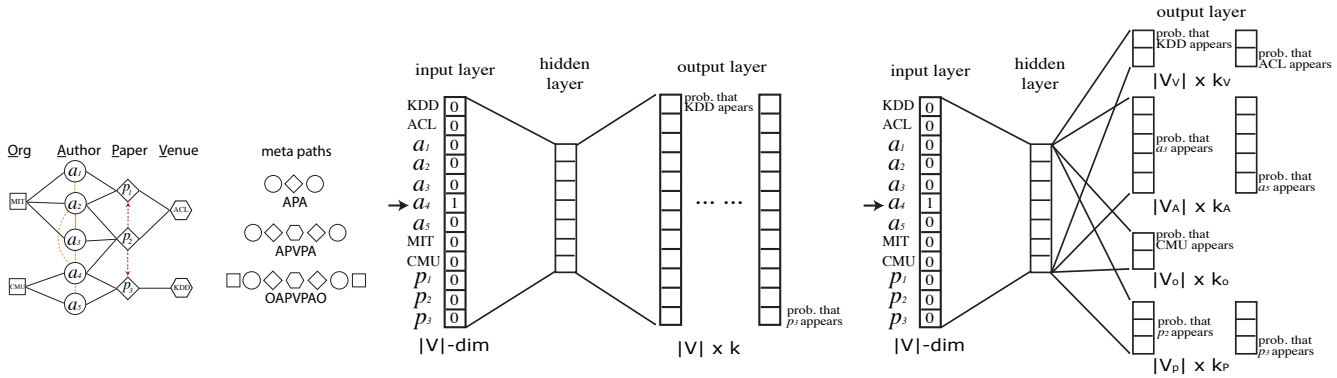
To achieve efficient optimization, Mikolov et al. introduced negative sampling [18], in which a relatively small set of words (nodes) are sampled from the corpus (network) for the construction of softmax. We leverage the same technique for *metapath2vec*. Given a negative sample size $M$, Eq. 2 is updated as follows: $\log \sigma(X_{c_t} \cdot X_v) + \sum_{m=1}^{M} \mathbb{E}_{u^m \sim P(u)}[\log \sigma(-X_{u^m} \cdot X_v)]$, where $\sigma(x) = \frac{1}{1+e^{-x}}$ and $P(u)$ is the pre-defined distribution from which a negative node $u^m$ is drew from for $M$ times. *metapath2vec* builds the the node frequency distribution by viewing different types of nodes homogeneously and draw (negative) nodes regardless of their types.

**Meta-Path-Based Random Walks.** How to effectively transform the structure of a network into skip-gram? In DeepWalk [22] and node2vec [8], this is achieved by incorporating the node paths traversed by random walkers over a network into the neighborhood function.

Naturally, we can put *random walkers in a heterogeneous network* to generate paths of multiple types of nodes. At step $i$, the transition probability $p(v^{i+1}|v^i)$ is denoted as the normalized probability distributed over the neighbors of $v^i$ by ignoring their node types. The generated paths can be then used as the input of node2vec and DeepWalk. *However, Sun et al. demonstrated that heterogeneous random walks are biased to highly visible types of nodes—those with a dominant number of paths—and concentrated nodes—those with a governing percentage of paths pointing to a small set of nodes [26].*

In light of these issues, we design meta-path-based random walks to generate paths that are able to capture both the semantic and structural correlations between different types of nodes, facilitating the transformation of heterogeneous network structures into *metapath2vec*'s skip-gram.

Formally, a meta-path scheme $\mathcal{P}$ is defined as a path that is denoted in the form of $V_1 \xrightarrow{R_1} V_2 \xrightarrow{R_2} \cdots V_t \xrightarrow{R_t} V_{t+1} \cdots \xrightarrow{R_{l-1}} V_l$, wherein $R = R_1 \circ R_2 \circ \cdots \circ R_{l-1}$ defines the composite relations between node types $V_1$ and $V_l$ [25]. Take Figure 2(a) as an example, a meta-path "APA" represents the coauthor relationships on a paper (P) between two authors (A), and "APVPA" represents two authors (A) publish papers (P) in the same venue (V). Previous work has shown that many data mining tasks in heterogeneous information networks can benefit from the modeling of meta-paths [6, 25, 27].




(a) An academic network (b) Skip-gram in *metapath2vec*, node2vec, & DeepWalk (c) Skip-gram in *metapath2vec++*

**Figure 2: An illustrative example of a heterogeneous academic network and skip-gram architectures of *metapath2vec* and *metapath2vec++* for embedding this network.** (a). Yellow dotted lines denote coauthor relationships and red dotted lines denote citation relationships. (b) The skip-gram architecture used in *metapath2vec* when predicting for $a_4$, which is the same with the one in node2vec if node types are ignored. $|V|$=12 denotes the number of nodes in the heterogeneous academic network in (a) and $a_4$'s neighborhood is set to include CMU, $a_2$, $a_3$, $a_5$, $p_2$, $p_3$, ACL, & KDD, making $k$ = 8. (c) The heterogeneous skip-gram used in *metapath2vec++*. Instead of one set of multinomial distributions for all types of neighborhood nodes in the output layer, it specifies one set of multinomial distributions for each type of nodes in $a_4$'s neighborhood. $V_t$ denotes one specific $t$-type nodes and $V = V_V \cup V_A \cup V_O \cup V_P$. $k_t$ specifies the size of a particular type of one's neighborhood and $k = k_V + k_A + k_O + k_P$.

Here we show how to use meta-paths to guide heterogeneous random walkers. Given a heterogeneous network $G = (V, E, T)$ and a meta-path scheme $\mathcal{P}$: $V_1 \xrightarrow{R_1} V_2 \xrightarrow{R_2} \cdots V_t \xrightarrow{R_t} V_{t+1} \cdots \xrightarrow{R_{l-1}} V_l$, the transition probability at step $i$ is defined as follows:

$$p(v^{i+1}|v_t^i, \mathcal{P}) = \begin{cases} \frac{1}{|N_{t+1}(v_t^i)|} & (v^{i+1}, v_t^i) \in E, \phi(v^{i+1}) = t{+}1 \\ 0 & (v^{i+1}, v_t^i) \in E, \phi(v^{i+1}) \neq t{+}1 \\ 0 & (v^{i+1}, v_t^i) \notin E \end{cases} \tag{3}$$

where $v_t^i \in V_t$ and $N_{t+1}(v_t^i)$ denote the $V_{t+1}$ type of neighborhood of node $v_t^i$. In other words, $v^{i+1} \in V_{t+1}$, that is, the flow of the walker is conditioned on the pre-defined meta-path $\mathcal{P}$. In addition, meta-paths are commonly used in a symmetric way, that is, its first node type $V_1$ is the same with the last one $V_l$ [25, 26, 28], facilitating its recursive guidance for random walkers, i.e.,

$$p(v^{i+1}|v_t^i) = p(v^{i+1}|v_1^i), \text{ if } t = l \tag{4}$$

The meta-path-based random walk strategy ensures that the semantic relationships between different types of nodes can be properly incorporated into skip-gram. For example, in a traditional random walk procedure, in Figure 2(a), the next step of a walker on node $a_4$ transitioned from node CMU can be all types of nodes surrounding it—$a_2$, $a_3$, $a_5$, $p_2$, $p_3$, and CMU. However, under the meta-path scheme 'OAPVPAO', for example, the walker is biased towards paper nodes (P) given its previous step on an organization node CMU (O), following the semantics of this path.

### 3.3 *metapath2vec++*

*metapath2vec* distinguishes the context nodes of node $v$ conditioned on their types when constructing its neighborhood function $N_t(v)$ in Eq. 2. However, it ignores the node type information in softmax. In other words, in order to infer the specific type of context $c_t$ in $N_t(v)$ given a node $v$, *metapath2vec* actually encourages all types of negative samples, including nodes of the same type $t$ as well as the other types in the heterogeneous network.

**Heterogeneous negative sampling.** We further propose the *metapath2vec++* framework, in which the softmax function is normalized with respect to the node type of the context $c_t$. Specifically, $p(c_t|v;\theta)$ is adjusted to the specific node type $t$, that is,

$$p(c_t|v;\theta) = \frac{e^{X_{c_t} \cdot X_v}}{\sum_{u_t \in V_t} e^{X_{u_t} \cdot X_v}} \tag{5}$$

where $V_t$ is the node set of type $t$ in the network. In doing so, *metapath2vec++* specifies one set of multinomial distributions for each type of neighborhood in the output layer of the skip-gram model. Recall that in *metapath2vec* and node2vec / DeepWalk, the dimension of the output multinomial distributions is equal to the number of nodes in the network. However, in *metapath2vec++*'s skip-gram, the multinomial distribution dimension for type $t$ nodes is determined by the number of $t$-type nodes. A clear illustration can be seen in Figure 2(c). For example, given the target node $a_4$ in the input layer, *metapath2vec++* outputs four sets of multinomial distributions, each corresponding to one type of neighbors—venues $V$, authors $A$, organizations $O$, and papers $P$.

Inspired by PTE [29], the sampling distribution is also specified by the node type of the neighbor $c_t$ that is targeted to predict, i.e., $P_t(\cdot)$. Therefore, we have the following objective:

$$O(\mathbf{X}) = \log \sigma(X_{c_t} \cdot X_v) + \sum_{m=1}^{M} \mathbb{E}_{u_t^m \sim P_t(u_t)} [\log \sigma(-X_{u_t^m} \cdot X_v)] \tag{6}$$

**Input:** The heterogeneous information network $G = (V, E, T)$, a meta-path scheme $\mathcal{P}$, #walks per node $w$, walk length $l$, embedding dimension $d$, neighborhood size $k$
**Output:** The latent node embeddings $\mathbf{X} \in \mathbb{R}^{|V| \times d}$

initialize $\mathbf{X}$ ;
**for** $i = 1 \rightarrow w$ **do**
  **for** $v \in V$ **do**
    $MP$ = MetaPathRandomWalk($G$, $\mathcal{P}$, $v$, $l$) ;
    $\mathbf{X}$ = HeterogeneousSkipGram($\mathbf{X}$, $k$, $MP$) ;
  **end**
**end**
return $\mathbf{X}$ ;

**MetaPathRandomWalk**($G$, $\mathcal{P}$, $v$, $l$)
$MP[1] = v$ ;
**for** $i = 1 \rightarrow l-1$ **do**
  draw u according to Eq. 3 ;
  $MP$[i+1] = $u$ ;
**end**
return $MP$ ;

**HeterogeneousSkipGram**($\mathbf{X}$, $k$, $MP$)
**for** $i = 1 \rightarrow l$ **do**
  $v = MP$[i] ;
  **for** $j = max(0, i\text{-}k) \rightarrow min(i+k, l)$ & $j \neq i$ **do**
    $c_t = MP$[j] ;
    $X^{new} = X^{old} - \eta \cdot \frac{\partial O(\mathbf{X})}{\partial X}$ (Eq. 7) ;
  **end**
**end**

**ALGORITHM 1:** The *metapath2vec++* Algorithm.

whose gradients are derived as follows:

$$\frac{\partial O(\mathbf{X})}{\partial X_{u_t^m}} = (\sigma(X_{u_t^m} \cdot X_v - \mathbb{I}_{c_t}[u_t^m]))X_v \tag{7}$$

$$\frac{\partial O(\mathbf{X})}{\partial X_v} = \sum_{m=0}^{M} (\sigma(X_{u_t^m} \cdot X_v - \mathbb{I}_{c_t}[u_t^m]))X_{u_t^m}$$

where $\mathbb{I}_{c_t}[u_t^m]$ is an indicator function to indicate whether $u_t^m$ is the neighborhood context node $c_t$ and when $m = 0$, $u_t^0 = c_t$. The model is optimized by using stochastic gradient descent algorithm. The pseudo code of *metapath2vec++* is listed in Algorithm 1.

## 4 EXPERIMENTS

In this section, we demonstrate the efficacy and efficiency of the presented *metapath2vec* and *metapath2vec++* frameworks for heterogeneous network representation learning.

**Data.** We use two heterogeneous networks, including the AMiner Computer Science (CS) dataset [31] and the Database and Information Systems (DBIS) dataset [26]. Both datasets and code are publicly available[1]. This AMiner CS dataset consists of 9,323,739 computer scientists and 3,194,405 papers from 3,883 computer science venues—both conferences and journals—held until 2016. We construct a heterogeneous collaboration network, in which there are three types of nodes: authors, papers, and venues. The links represent different types of relationships among three sets of nodes—such as collaboration relationships on a paper.

The DBIS dataset was constructed and used by Sun et al. [26]. It covers 464 venues, their top-5000 authors, and corresponding 72,902 publications. We also construct the heterogeneous collaboration networks from DBIS wherein a link may connect two authors, one author and one paper, as well as one paper and one venue.

### 4.1 Experimental Setup

We compare *metapath2vec* and *metapath2vec++* with several recent network representation learning methods:

(1) DeepWalk [22] / node2vec [8]: With the same random walk path input ($p$=1 & $q$=1 in node2vec), we find that the choice between hierarchical softmax (DeepWalk) and negative sampling (node2vec) techniques does not yield significant differences. Therefore we use $p$=1 and $q$=1 [8] in node2vec for comparison.
(2) LINE [30]: We use the advanced version of LINE by considering both the 1st- and 2nd-order of node proximity;
(3) PTE [29]: We construct three bipartite heterogeneous networks (author–author, author–venue, venue–venue) and restrain it as an unsupervised embedding method;
(4) Spectral Clustering [33] / Graph Factorization [2]: With the same treatment to these methods in node2vec [8], we exclude them from our comparison, as previous studies have demonstrated that they are outperformed by DeepWalk and LINE.

For all embedding methods, we use the same parameters listed below. In addition, we also vary each of them and fix the others for examining the parameter sensitivity of the proposed methods.

(1) The number of walks per node $w$: 1000;
(2) The walk length $l$: 100;
(3) The vector dimension $d$: 128 (LINE: 128 for each order);
(4) The neighborhood size $k$: 7;
(5) The size of negative samples: 5.

For *metapath2vec* and *metapath2vec++*, we also need to specify the meta-path scheme to guide random walks. We surveyed most of the meta-path-based work and found that the most commonly and effectively used meta-path schemes in heterogeneous academic networks are "APA" and "APVPA" [12, 25–27]. Notice that "APA" denotes the coauthor semantic, that is, the traditional (homogeneous) collaboration links / relationships. "APVPA" represents the heterogeneous semantic of authors publishing papers at the same venues. Our empirical results also show that this simple meta-path scheme "APVPA" can lead to node embeddings that can be generalized to diverse heterogeneous academic mining tasks, suggesting its applicability to potential applications for academic search services.

We evaluate the quality of the latent representations learned by different methods over three classical heterogeneous network mining tasks, including multi-class node classification [13], node clustering [27], and similarity search [26]. In addition, we also use the embedding projector in TensorFlow [1] to visualize the node embeddings learned from the heterogeneous academic networks.

[1]The network data, learned latent representations, labeled ground truth data, and source code can be found at https://ericdongyx.github.io/metapath2vec/m2v.html

**Table 2: Multi-class venue node classification results in AMiner data.**

| Metric | Method | 5% | 10% | 20% | 30% | 40% | 50% | 60% | 70% | 80% | 90% |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Macro-F1 | DeepWalk/node2vec | 0.0723 | 0.1396 | 0.1905 | 0.2795 | 0.3427 | 0.3911 | 0.4424 | 0.4774 | 0.4955 | 0.4457 |
| | LINE (1st+2nd) | 0.2245 | 0.4629 | 0.7011 | 0.8473 | 0.8953 | 0.9203 | 0.9308 | 0.9466 | 0.9410 | 0.9466 |
| | PTE | 0.1702 | 0.3388 | 0.6535 | 0.8304 | 0.8936 | 0.9210 | 0.9352 | 0.9505 | 0.9525 | 0.9489 |
| | *metapath2vec* | 0.3033 | 0.5247 | 0.8033 | 0.8971 | 0.9406 | 0.9532 | 0.9529 | 0.9701 | 0.9683 | 0.9670 |
| | *metapath2vec++* | 0.3090 | 0.5444 | 0.8049 | 0.8995 | 0.9468 | 0.9580 | 0.9561 | 0.9675 | 0.9533 | 0.9503 |
| Micro-F1 | DeepWalk/node2vec | 0.1701 | 0.2142 | 0.2486 | 0.3266 | 0.3788 | 0.4090 | 0.4630 | 0.4975 | 0.5259 | 0.5286 |
| | LINE (1st+2nd) | 0.3000 | 0.5167 | 0.7159 | 0.8457 | 0.8950 | 0.9209 | 0.9333 | 0.9500 | 0.9556 | 0.9571 |
| | PTE | 0.2512 | 0.4267 | 0.6879 | 0.8372 | 0.8950 | 0.9239 | 0.9352 | 0.9550 | 0.9667 | 0.9571 |
| | *metapath2vec* | 0.4173 | 0.5975 | 0.8327 | 0.9011 | 0.9400 | 0.9522 | 0.9537 | 0.9725 | 0.9815 | 0.9857 |
| | *metapath2vec++* | 0.4331 | 0.6192 | 0.8336 | 0.9032 | 0.9463 | 0.9582 | 0.9574 | 0.9700 | 0.9741 | 0.9786 |

**Table 3: Multi-class author node classification results in AMiner data.**

| Metric | Method | 5% | 10% | 20% | 30% | 40% | 50% | 60% | 70% | 80% | 90% |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Macro-F1 | DeepWalk/node2vec | 0.7153 | 0.7222 | 0.7256 | 0.7270 | 0.7273 | 0.7274 | 0.7273 | 0.7271 | 0.7275 | 0.7275 |
| | LINE (1st+2nd) | 0.8849 | 0.8886 | 0.8911 | 0.8921 | 0.8926 | 0.8929 | 0.8934 | 0.8936 | 0.8938 | 0.8934 |
| | PTE | 0.8898 | 0.8940 | 0.897 | 0.8982 | 0.8987 | 0.8990 | 0.8997 | 0.8999 | 0.9002 | 0.9005 |
| | *metapath2vec* | 0.9216 | 0.9262 | 0.9292 | 0.9303 | 0.9309 | 0.9314 | 0.9315 | 0.9316 | 0.9319 | 0.9320 |
| | *metapath2vec++* | 0.9107 | 0.9156 | 0.9186 | 0.9199 | 0.9204 | 0.9207 | 0.9207 | 0.9208 | 0.9211 | 0.9212 |
| Micro-F1 | DeepWalk/node2vec | 0.7312 | 0.7372 | 0.7402 | 0.7414 | 0.7418 | 0.7420 | 0.7419 | 0.7420 | 0.7425 | 0.7425 |
| | LINE (1st+2nd) | 0.8936 | 0.8969 | 0.8993 | 0.9002 | 0.9007 | 0.9010 | 0.9015 | 0.9016 | 0.9018 | 0.9017 |
| | PTE | 0.8986 | 0.9023 | 0.9051 | 0.9061 | 0.9066 | 0.9068 | 0.9075 | 0.9077 | 0.9079 | 0.9082 |
| | *metapath2vec* | 0.9279 | 0.9319 | 0.9346 | 0.9356 | 0.9361 | 0.9365 | 0.9365 | 0.9365 | 0.9367 | 0.9369 |
| | *metapath2vec++* | 0.9173 | 0.9217 | 0.9243 | 0.9254 | 0.9259 | 0.9261 | 0.9261 | 0.9262 | 0.9264 | 0.9266 |

## 4.2 Multi-Class Classification

For the classification task, we use third-party labels to determine the class of each node. First, we match the eight categories[2] of venues in Google Scholar[3] with those in AMiner data. Among all of the 160 venues (20 per category × 8 categories), 133 of them are successfully matched and labeled correspondingly (Most of unmatched venues are pre-print venues, such as arXiv). Second, for each author who published in these 133 venues, his / her label is assigned to the category with the majority of his / her publications, and a tie is resolved by random selection among the possible categories; 246,678 authors are labeled with research category.

Note that the node representations are learned from the full dataset. The embeddings of above labeled nodes are then used as the input to a logistic regression classifier. In the classification experiments, we vary the size of the training set from 5% to 90% and the remaining nodes for testing. We repeat each prediction experiment ten times and report the average performance in terms of both Macro-F1 and Micro-F1 scores.

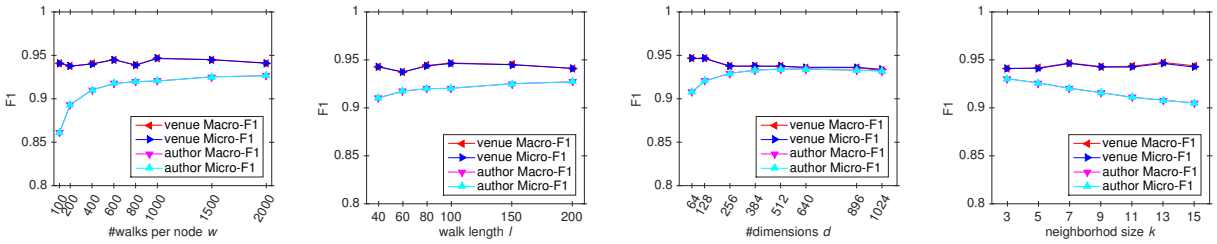
[2]1. Computational Linguistics, 2. Computer Graphics, 3. Computer Networks & Wireless Communication, 4. Computer Vision & Pattern Recognition, 5. Computing Systems, 6. Databases & Information Systems, 7. Human Computer Interaction, and 8. Theoretical Computer Science.

[3]https://scholar.google.com/citations?view_op=top_venues&hl=en&vq=eng. Accessed on February, 2017.

**Results.** Tables 2 and 3 list the eight-class classification results. Overall, the proposed *metapath2vec* and *metapath2vec++* models consistently and significantly outperform all baselines in terms of both metrics. When predicting for the venue category, the advantage of both *metapath2vec* and *metapath2vec++* are particular strong given a small size of training data. Given 5% of nodes as training data, for example, *metapath2vec* and *metapath2vec++* achieve 0.08–0.23 (relatively 35–319%) improvements in terms of Macro-F1 and 0.13–0.26 (relatively 39–145%) gains in terms of Micro-F1 over DeepWalk / node2vec, LINE, and PTE. When predicting for authors' categories, the performance of each method is relatively stable when varying the train-test split. The constant gain achieved by the proposed methods is around 2-3% over LINE and PTE, and ~20% over DeepWalk / node2vec.

In summary, *metapath2vec* and *metapath2vec++* learn significantly better heterogeneous node embeddings than current state-of-the-art methods, as measured by multi-class classification performance. The advantage of the proposed methods lies in their proper consideration and accommodation of the network heterogeneity challenge—the existence of multiple types of nodes and relations.

**Parameter sensitivity.** In skip-gram-based representation learning models, there exist several common parameters (see Section 4.1). We conduct a sensitivity analysis of *metapath2vec++* to these parameters. Figure 3 shows the classification results as a function




(a) #walks per node $w$ (b) walk length $l$ (c) #dimensions $d$ (d) neighborhood size $k$

**Figure 3: Parameter sensitivity in multi-class node classification.** 50% as training data and the remaining as test data.

**Table 4: Node clustering results (NMI) in AMiner data.**

| methods | venue | author |
|---|---|---|
| DeepWalk/node2vec | 0.1952 | 0.2941 |
| LINE (1st+2nd) | 0.8967 | 0.6423 |
| PTE | 0.9060 | 0.6483 |
| *metapath2vec* | 0.9274 | 0.7470 |
| *metapath2vec++* | 0.9261 | 0.7354 |

of one chosen parameter when the others are controlled for. In general, we find that in Figures 3(a) and 3(b) the number of walks $w$ rooting from each node and the length $l$ of each walk are positive to the author classification performance, while they are surprisingly inconsequential for inferring venue nodes' categories as measured by Macro-F1 and Micro-F1 scores. The increase of author classification performance converges as $w$ and $l$ reach around 1000 and 100, respectively. Similarly, Figures 3(c) and 3(d) suggest that the number of embedding dimensions $d$ and neighborhood size $k$ are again of relatively little relevance to the predictive task for venues, and $k$ on the other hand is positively crucial to determine the class of a venue. However, the descending lines as the increase of $k$ for author classifications imply that a smaller neighborhood size actually produces the best embeddings for separating authors. This finding differs from those in a homogeneous environment [8], wherein the neighborhood size generally shows a positive effect on node classification.

According to the analysis, *metapath2vec++* is not strictly sensitive to these parameters and is able to reach high performance under a cost-effective parameter choice (the smaller, the more efficient). In addition, our results also indicate that those common parameters show different functions for heterogeneous network embedding with those in homogeneous network cases, demonstrating the request of different ideas and solutions for heterogeneous network representation learning.

## 4.3 Node Clustering

We illustrate how the latent representations learned by embedding methods can help the node clustering task in heterogeneous networks. We employ the same eight-category author and venue nodes used in the classification task above. The learned embeddings by each method is input to a clustering model. Here we leverage the $k$-means algorithm to cluster the data and evaluate the clustering results in terms of normalized mutual information (NMI) [26]. In addition, we also report *metapath2vec++*'s sensitivity with respect to different parameter choices. All clustering experiments are conducted 10 times and the average performance is reported.

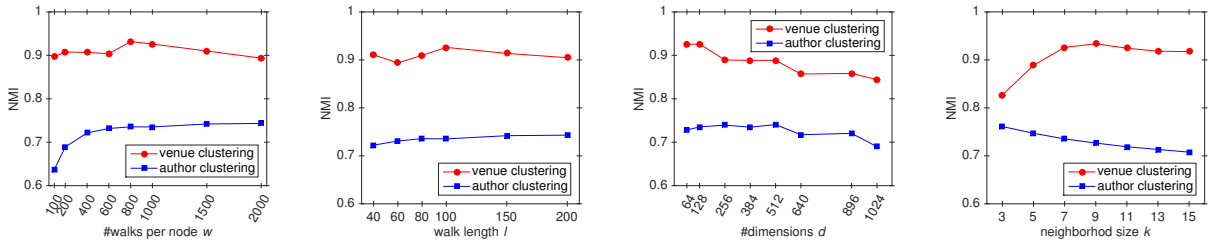
**Results.** Table 4 shows the node clustering results as measured by NMI in the AMiner CS data. Overall, the table demonstrates that *metapath2vec* and *metapath2vec++* outperform all the comparative methods. When clustering for venues, the task is trivial as evident from the high NMI scores produced by most of the methods: *metapath2vec*, *metapath2vec++*, LINE, and PTE. Nevertheless, the proposed two methods outperform LINE and PTE by 2–3%. The author clustering task is more challenging than the venue case, and the gain obtained by *metapath2vec* and *metapath2vec++* over the best baselines (LINE and PTE) is more significant—around 13–16%.

In summary, *metapath2vec* and *metapath2vec++* generate more appropriate embeddings for different types of nodes in the network than comparative baselines, suggesting their ability to capture and incorporate the underlying structural and semantic relationships between various types of nodes in heterogeneous networks.

**Parameter sensitivity.** Following the same experimental procedure in classification, we study the parameter sensitivity of *metapath2vec++* as measured by the clustering performance. Figure 4 shows the clustering performance as a function of each of the four parameters when fixing the other three. From Figures 4(a) and 4(b), we can observe that the balance between computational cost (a small $w$ and $l$ in $x$-axis) and efficacy (a high NMI in $y$-axis) can be achieved at around $w = 800{\sim}1000$ and $l = 100$ for the clustering of both authors and venues. Further, different from the positive effect of increasing $w$ and $l$ on author clustering, $d$ and $k$ are negatively correlated with the author clustering performance, as observed from Figures 4(c) and 4(d). Similarly, the venue clustering performance also shows an descending trend with an increasing $d$, while on the other hand, we observe a first-increasing and then-decreasing NMI line when $k$ is increased. Both figures together imply that $d = 128$ and $k = 7$ are capable of embedding heterogeneous nodes into latent space for promising clustering outcome.

## 4.4 Case Study: Similarity Search

We conduct two case studies to demonstrate the efficacy of our methods. We select 16 top CS conferences from the corresponding




(a) #walks per node $w$ (b) walk length $l$ (c) #dimensions $d$ (d) neighborhood size $k$

**Figure 4: Parameter sensitivity in clustering.**

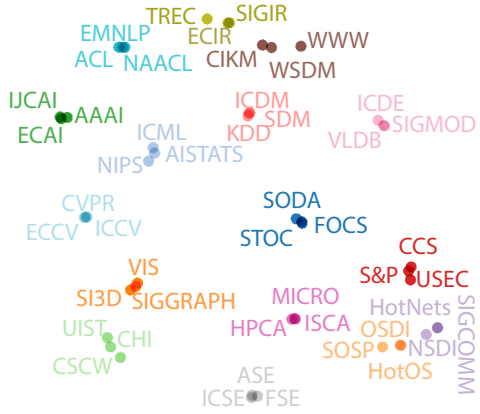



**Figure 5: 2D t-SNE projections of the 128D embeddings of 48 CS venues, three each from 16 sub-fields.**

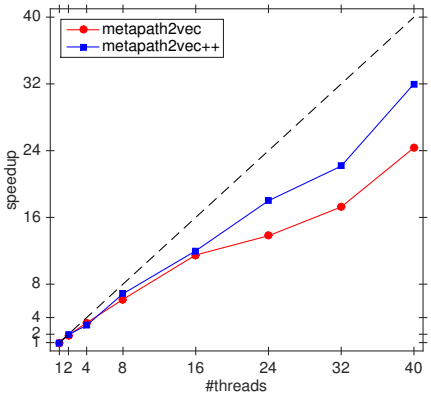



**Figure 6: Scalability of *metapath2vec* and *metapath2vec++*.**

sub-fields in the AMiner CS data and another 5 from the DBIS data. This results in a total of 21 query nodes. We use cosine similarity to determine the distance (similarity) between the query node and the remaining others.

Table 5 lists the top ten similar results for querying the 16 leading conferences in corresponding computer science sub-fields. One can observe that for the query "ACL", for example, *metapath2vec++* returns venues with the same focus—natural language processing, such as EMNLP ($1^{st}$), NAACL ($2^{nd}$), Computational Linguistics ($3^{rd}$), CoNLL ($4^{th}$), COLING ($5^{th}$), and so on. Similar performance can be also achieved when querying the other conferences from various fields. More surprisingly, we find that in most cases, the top three results cover venues with similar prestige to the query one, such as STOC to FOCS in theory, OSDI to SOSP in system, HPCA to ISCA in architecture, CCS to S&P in security, CSCW to CHI in human-computer interaction, EMNLP to ACL in NLP, ICML to NIPS in machine learning, WSDM to WWW in Web, AAAI to IJCAI in artificial intelligence, PVLDB to SIGMOD in database, etc. Similar results can also be observed in Tables 6 and 1, which show the similarity search results for the DBIS network.

## 4.5 Case Study: Visualization

We employ the TensorFlow embedding projector to further visualize the low-dimensional node representations learned by embedding models. First, we project multiple types of nodes—16 top CS conferences and corresponding top-profile authors—into the same space in Figure 1. From Figure 1(d), we can clearly see that *metapath2vec++* is able to automatically organize these two types of nodes and implicitly learn the internal relationships between them, indicated by the similar directions and distances of the arrows connecting each pair of them, such as J. Dean → OSDI, C. D. Manning → ACL, R. E. Tarjan → FOCS, M. I. Jordan → NIPS, and so on. In addition, these two types of nodes are clearly located in two separate and straight columns. Neither of these two results can be made by the recent network embedding models in Figures 1(a) and 1(b).

As to *metapath2vec*, instead of separating the two types of nodes into two columns, it is capable of grouping each pair of one venue and its corresponding author closely, such as R. E. Tarjan and FOCS, H. Jensen and SIGGRAPH, H. Ishli and CHI, R. Agrawal and SIGMOD, etc. Together, both models arrange nodes from similar fields close to each other and dissimilar ones distant from each other, such as the "Core CS" cluster of systems (OSDI), networking (SIGCOMM), security (S&P), and architecture (ISCA), as well as the "Big AI" cluster of data mining (KDD), information retrieval (SIGIR), artificial intelligence (AI), machine learning (NIPS), NLP (ACL), and vision (CVPR). These groupings are also reflected by their corresponding author nodes.

Second, Figure 5 visualizes the latent vectors—learned by *metapath2vec++*—of 48 venues used in similarity search of Section 4.4,

**Table 5: Case study of similarity search in AMiner Data**

| Rank | ACL | NIPS | IJCAI | CVPR | FOCS | SOSP | ISCA | S&P | ICSE | SIGGRAPH | SIGCOMM | CHI | KDD | SIGMOD | SIGIR | WWW |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | ACL | NIPS | IJCAI | CVPR | FOCS | SOSP | ISCA | S&P | ICSE | SIGGRAPH | SIGCOMM | CHI | KDD | SIGMOD | SIGIR | WWW |
| 1 | EMNLP | ICML | AAAI | ECCV | STOC | TOCS | HPCA | CCS | TOSEM | TOG | CCR | CSCW | SDM | PVLDB | ECIR | WSDM |
| 2 | NAACL | AISTATS | AI | ICCV | SICOMP | OSDI | MICRO | NDSS | FSE | SI3D | HotNets | TOCHI | TKDD | ICDE | CIKM | CIKM |
| 3 | CL | JMLR | JAIR | IJCV | SODA | HotOS | ASPLOS | USENIX S | ASE | RT | NSDI | UIST | ICDM | DE Bull | IR J | TWEB |
| 4 | CoNLL | NC | ECAI | ACCV | A-R | SIGOPS E | PACT | ACSAC | ISSTA | CGF | CoNEXT | DIS | DMKD | VLDBJ | TREC | ICWSM |
| 5 | COLING | MLJ | KR | CVIU | TALG | ATC | ICS | JCS | E SE | NPAR | IMC | HCI | KDD E | EDBT | SIGIR F | HT |
| 6 | IJCNLP | COLT | AI Mag | BMVC | ICALP | NSDI | HiPEAC | ESORICS | MSR | Vis | TON | MobileHCI | WSDM | TODS | ICTIR | SIGIR |
| 7 | NLE | UAI | ICAPS | ICPR | ECCC | OSR | PPOPP | TISS | ESEM | JGT | INFOCOM | INTERACT | CIKM | CIDR | WSDM | KDD |
| 8 | ANLP | KDD | CI | EMMCVPR | TOC | ASPLOS | ICCD | ASIACCS | A SE | VisComp | PAM | GROUP | PKDD | SIGMOD R | TOIS | TIT |
| 9 | LREC | CVPR | AIPS | T on IP | JAIG | EuroSys | CGO | RAID | ICPC | GI | MobiCom | NordiCHI | ICML | WebDB | IPM | WISE |
| 10 | EACL | ECML | UAI | WACV | ITCS | SIGCOMM | ISLPED | CSFW | WICSA | CG | IPTPS | UbiComp | PAKDD | PODS | AIRS | WebSci |

**Table 6: Case study of similarity search in DBIS Data**

| Rank | KDD | SIGMOD | SIGIR | WWW | WSDM |
|---|---|---|---|---|---|
| 0 | KDD | SIGMOD | SIGIR | WWW | WSDM |
| 1 | SDM | PVLDB | TREC | CIKM | WWW |
| 2 | ICDM | ICDE | CIKM | SIGIR | SIGIR |
| 3 | DMKD | TODS | IPM | KDD | KDD |
| 4 | KDD E | VLDBJ | IRJ | ICDE | AIRWeb |
| 5 | PKDD | PODS | ECIR | TKDE | CIKM |
| 6 | PAKDD | EDBT | TOIS | VLDB | WebDB |
| 7 | TKDE | CIDR | WWW | TOTT | ICDM |
| 8 | CIKM | TKDE | JASIST | SIGMOD | VLDB |
| 9 | ICDE | ICDT | JASIS | WebDB | VLDBJ |
| 10 | TKDD | DE Bull | SIGIR F | WISE | SDM |

three each from 16 sub-fields. We can see that conferences from the same domain are geographically grouped to each other and each group is well separated from others, further demonstrating the embedding ability of *metapath2vec++*. In addition, similar to the observation in Figure 1, we can also notice that the heterogeneous embeddings are able to unveil the similarities across different domains, including the "Core CS" sub-field cluster at the bottom right and the "Big AI" sub-field cluster at the top right.

Thus, Figures 1 and 5 intuitively demonstrate *metapath2vec++*'s novel capability to discover, model, and capture the underlying structural and semantic relationships between multiple types of nodes in heterogeneous networks.

## 4.6 Scalability

In the era of big (network) data, it is necessary to demonstrate the scalability of the proposed network embedding models. The *metapath2vec* and *metapath2vec++* methods can be parallelized by using the same mechanism as word2vec and node2vec [8, 18]. All codes are implemented in C and C++ and our experiments are conducted in a computing server with Quad 12 (48) core 2.3 GHz Intel Xeon CPUs E7-4850. We run experiments on the AMiner CS data with the default parameters with different number of threads, i.e., 1, 2, 4, 8, 16, 24, 32, 40, each of them utilizing one CPU core.

Figure 6 shows the speedup of *metapath2vec* & *metapath2vec++* over the single-threaded case. Optimal speedup performance is denoted by the dashed $y = x$ line, which represents perfect distribution and execution of computation across all CPU cores. In general, we find that both methods achieve acceptable sublinear speedups as both lines are close to the optimal line. In specific, they can reach 11–12× speedup with 16 cores and 24–32× speedup with 40 cores used. By using 40 cores, *metapath2vec++*'s learning process costs only 9 minutes for embedding the full AMiner CS network, which is composed of over 9 million authors with 3 million papers published in more than 3800 venues. Overall, the proposed *metapath2vec* and *metapath2vec++* models are efficient and scalable for large-scale heterogeneous networks with millions of nodes.

# 5 RELATED WORK

Network representation learning can be traced back to the usage of latent factor models for network analysis and graph mining tasks [10, 34], such as the application of factorization models for recommendation systems [14, 16], node classification [32], relational mining [19], and role discovery [9]. This rich line of research focuses on factorizing the matrix/tensor format (e.g., the adjacency matrix) of a network, generating latent-dimension features for nodes or edges in this network. However, the computational cost of decomposing a large-scale matrix/tensor is usually very expensive, and also suffers from its statistical performance drawback [8], making it neither practical nor effective for addressing tasks in big networks.

With the advent of deep learning techniques, significant effort has been devoted to designing neural network-based representation learning models. For example, Mikolov et al. proposed the *word2vec* framework—a two-layer neural network—to learn the distributed representations of words in natural language [17, 18]. Building on word2vec, Perozzi et al. suggested that the "context" of a node can be denoted by their co-occurrence in a random walk path [22]. Formally, they put random walkers over networks to record their walking paths, each of which is composed of a chain of nodes that could be considered as a "sentence" of words in a text corpus. More recently, in order to diversify the neighborhood of a node, Grover & Leskovec presented biased random walkers—a mixture of breadth-first and width-first search procedures—over networks to produce paths of nodes [8]. With node paths generated, both works leveraged the skip-gram architecture in word2vec to model the structural correlations between nodes in a path. In addition, several other methods have been proposed for learning representations in networks [4, 5, 11, 20, 23]. In particular, to learn network embeddings, Tang et al. decomposed a node's context into first-order (friends) and second-order (friends' friends) proximity [30], which was further developed into a semi-supervised model PTE for embedding text data [29].

Our work furthers this direction of investigation by designing the *metapath2vec* and *metapath2vec++* models to capture heterogeneous structural and semantic correlations exhibited from large-scale networks with multiple types of nodes, which can not be handled by previous models, and applying these models to a variety of network mining tasks.

## 6 CONCLUSION

In this work, we formally define the representation learning problem in heterogeneous networks in which there exist diverse types of nodes and links. To address the network heterogeneity challenge, we propose the *metapath2vec* and *metapath2vec++* methods. We develop the meta-path-guided random walk strategy in a heterogeneous network, which is capable of capturing both the structural and semantic correlations of differently typed nodes and relations. To leverage this method, we formalize the heterogeneous neighborhood function of a node, enabling the skip-gram-based maximization of the network probability in the context of multiple types of nodes. Finally, we achieve effective and efficient optimization by presenting a heterogeneous negative sampling technique. Extensive experiments demonstrate that the latent feature representations learned by *metapath2vec* and *metapath2vec++* are able to improve various heterogeneous network mining tasks, such as similarity search, node classification, and clustering. Our results can be naturally applied to real-world applications in heterogeneous academic networks, such as author, venue, and paper search in academic search services.

Future work includes various optimizations and improvements. For example, 1) the *metapath2vec* and *metapath2vec++* models, as is also the case with DeepWalk and node2vec, face the challenge of large intermediate output data when sampling a network into a huge pile of paths, and thus identifying and optimizing the sampling space is an important direction; 2) as is also the case with all meta-path-based heterogeneous network mining methods, *metapath2vec* and *metapath2vec++* can be further improved by the automatic learning of meaningful meta-paths; 3) extending the models to incorporate the dynamics of evolving heterogeneous networks; and 4) generalizing the models for different genres of heterogeneous networks.

**Acknowledgments.** We would like to thank Reid Johnson for discussions and suggestions. This work is supported by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053 and the National Science Foundation (NSF) grants CNS-1629914 and IIS-1447795.

## REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, and others. 2016. TensorFlow: A system for large-scale machine learning. In *OSDI '16*.

[2] Amr Ahmed, Nino Shervashidze, Shravan Narayanamurthy, Vanja Josifovski, and Alexander J. Smola. 2013. Distributed Large-scale Natural Graph Factorization. In *WWW '13*. ACM, 37–48.

[3] Yoshua Bengio, Aaron Courville, and Pierre Vincent. 2013. Representation learning: A review and new perspectives. *IEEE TPAMI* 35, 8 (2013), 1798–1828.

[4] 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 '15*. ACM, 119–128.

[5] Ting Chen and Yizhou Sun. 2017. Task-Guided and Path-Augmented Heterogeneous Network Embedding for Author Identification. In *WSDM '17*. ACM.

[6] Yuxiao Dong, Jing Zhang, Jie Tang, Nitesh V. Chawla, and Bai Wang. 2015. CoupledLP: Link Prediction in Coupled Networks. In *KDD '15*. ACM, 199–208.

[7] Yoav Goldberg and Omer Levy. 2014. word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. *CoRR* abs/1402.3722 (2014).

[8] Aditya Grover and Jure Leskovec. 2016. Node2Vec: Scalable Feature Learning for Networks. In *KDD '16*. ACM, 855–864.

[9] Keith Henderson, Brian Gallagher, Tina Eliassi-Rad, Hanghang Tong, Sugato Basu, Leman Akoglu, Danai Koutra, Christos Faloutsos, and Lei Li. 2012. Rolx: structural role extraction & mining in large graphs. In *KDD '12*. ACM, 1231–1239.

[10] Peter D Hoff, Adrian E Raftery, and Mark S Handcock. 2002. Latent space approaches to social network analysis. *Journal of the American Statistical association* 97, 460 (2002), 1090–1098.

[11] Xiao Huang, Jundong Li, and Xia Hu. 2017. Label Informed Attributed Network Embedding. In *WSDM '17*. na.

[12] 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 '16*. ACM, 1595–1604.

[13] Ming Ji, Jiawei Han, and Marina Danilevsky. 2011. Ranking-based classification of heterogeneous information networks. In *KDD '11*. ACM, 1298–1306.

[14] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD '08*. ACM, 426–434.

[15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.

[16] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. 2011. Recommender systems with social regularization. In *WSDM '11*. 287–296.

[17] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR* abs/1301.3781 (2013). http://arxiv.org/abs/1301.3781

[18] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS '13*. 3111–3119.

[19] Jennifer Neville and David Jensen. 2005. Leveraging relational autocorrelation with latent group models. In *Proceedings of the 4th international workshop on Multi-relational mining*. ACM, 49–55.

[20] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. 2016. Asymmetric Transitivity Preserving Graph Embedding. In *KDD '16*. ACM, 1105–1114.

[21] Siddharth Pal, Yuxiao Dong, Bishal Thapa, Nitesh V Chawla, Ananthram Swami, and Ram Ramanathan. 2016. Deep learning for network analysis: Problems, approaches and challenges. In *Military Communications Conference, MILCOM 2016-2016*. IEEE, 588–593.

[22] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: Online Learning of Social Representations. In *KDD '14*. ACM, 701–710.

[23] Xiang Ren, Wenqi He, Meng Qu, Clare R Voss, Heng Ji, and Jiawei Han. 2016. Label noise reduction in entity typing by heterogeneous partial-label embedding. In *KDD '16*. ACM.

[24] Xin Rong. 2014. word2vec Parameter Learning Explained. *CoRR* abs/1411.2738 (2014). http://arxiv.org/abs/1411.2738

[25] Yizhou Sun and Jiawei Han. 2012. *Mining Heterogeneous Information Networks: Principles and Methodologies*. Morgan & Claypool Publishers.

[26] 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. In *VLDB '11*. 992–1003.

[27] Yizhou Sun, Brandon Norick, Jiawei Han, Xifeng Yan, Philip S. Yu, and Xiao Yu. 2012. Integrating Meta-path Selection with User-guided Object Clustering in Heterogeneous Information Networks. In *KDD '12*. ACM, 1348–1356.

[28] Yizhou Sun, Yintao Yu, and Jiawei Han. 2009. Ranking-based Clustering of Heterogeneous Information Networks with Star Network Schema. In *KDD '09*. ACM, 797–806.

[29] Jian Tang, Meng Qu, and Qiaozhu Mei. 2015. PTE: Predictive Text Embedding Through Large-scale Heterogeneous Text Networks. In *KDD '15*. ACM, 1165–1174.

[30] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale Information Network Embedding.. In *WWW '15*. ACM.

[31] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. ArnetMiner: Extraction and Mining of Academic Social Networks. In *KDD '08*. 990–998.

[32] Lei Tang and Huan Liu. 2009. Relational learning via latent social dimensions. In *KDD '09*. 817–826.

[33] Lei Tang and Huan Liu. 2011. Leveraging social media networks for classification. *DMKD* 23, 3 (2011), 447–478.

[34] Shuicheng Yan, Dong Xu, Benyu Zhang, Hong-Jiang Zhang, Qiang Yang, and Stephen Lin. 2007. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE TPAMI* 29, 1 (2007).

[35] Jing Zhang, Jie Tang, Cong Ma, Hanghang Tong, Yu Jing, and Juanzi Li. 2015. Panther: Fast top-k similarity search on large networks. In *KDD '15*. ACM, 1445–1454.