Inductive Contextual Relation Learning for Personalization

CHUXU ZHANG, Brandeis University, USA
HUAXIU YAO, Pennsylvania State University, USA
LU YU, Ant Financial Services Group, China
CHAO HUANG, JD Finance America Corporation, USA
DONGJIN SONG, University of Connecticut, USA
HAIFENG CHEN, NEC Laboratories America Inc, USA
MENG JIANG and NITESH V. CHAWLA, University of Notre Dame, USA

Web personalization, e.g., recommendation or relevance search, tailoring a service/product to accommodate specific online users, is becoming increasingly important. Inductive personalization aims to infer the relations between existing entities and unseen new ones, e.g., searching relevant authors for new papers or recommending new items to users. This problem, however, is challenging since most of recent studies focus on transductive problem for existing entities. In addition, despite some inductive learning approaches have been introduced recently, their performance is sub-optimal due to relatively simple and inflexible architectures for aggregating entity's content. To this end, we propose the inductive contextual personalization (ICP) framework through contextual relation learning. Specifically, we first formulate the pairwise relations between entities with a ranking optimization scheme that employs neural aggregator to fuse entity's heterogeneous contents. Next, we introduce a node embedding term to capture entity's contextual relations, as a smoothness constraint over the prior ranking objective. Finally, the gradient descent procedure with adaptive negative sampling is employed to learn the model parameters. The learned model is capable of inferring the relations between existing entities and inductive ones. Thorough experiments demonstrate that ICP outperforms numerous baseline methods for two different applications, i.e., relevant author search and new item recommendation.

CCS Concepts: • **Information systems → Personalization**; • **Computing methodologies → Neural networks**;

Additional Key Words and Phrases: Personalization, content-based recommendation, relation learning, node embedding

35

Authors' addresses: C. Zhang, Brandeis University, Waltham, MA; email: chuxuzhang@brandeis.edu; H. Yao, Pennsylvania State University, State College, PA; email: huaxiuyao@psu.edu; L. Yu, Ant Financial Services Group, Hangzhou, China; email: lu.yu@kaust.edu.sa; C. Huang, JD Finance America Corporation, Mountain View, CA; email: chaohuang75@gmail.com; D. Song, University of Connecticut, Storrs, CT; email: dongjin.song@uconn.edu; H. Chen, NEC Laboratories America Inc, Princeton, NJ; email: haifeng@nec-labs.com; M. Jiang and N. V. Chawla, University of Notre Dame, Notre Dame, IN; emails: {mjiang2, nchawla}@nd.edu.

1046-8188/2021/05-ART35 $15.00
https://doi.org/10.1145/3450353

ACM Reference format:
Chuxu Zhang, Huaxiu Yao, Lu Yu, Chao Huang, Dongjin Song, Haifeng Chen, Meng Jiang, and Nitesh V. Chawla. 2021. Inductive Contextual Relation Learning for Personalization. *ACM Trans. Inf. Syst.* 39, 3, Article 35 (May 2021), 22 pages.
https://doi.org/10.1145/3450353

1 INTRODUCTION

With the rapid growth of various web services such as Amazon e-commerce platform or Google Scholar, web personalization systems are becoming increasingly important, which facilitate relation learning study in the past decade. The related applications include item recommendation [19, 27, 39, 40, 52], relevant entity search [10, 30, 44, 51], and so on. Since entities in these systems may carry different types of content information, e.g., attribute, text, or image, many content-based approaches [8, 16, 17, 34, 50, 63] have been developed to perform relation inference by utilizing heterogeneous content information. Some of these methods can be adapted to inductive relation inference that aims to infer the relations between existing entities and inductive ones (i.e., those do not currently present in the system). However, their performances are sub-optimal due to relatively simple architecture to aggregate content information of inductive entities.

In this article, we study the problem of inductive contextual personalization by leveraging the existing relations among entities and heterogeneous contents carried by each entity. Note that the term "inductive" refers to the setting that there exist entities that are not currently present in the system. This is in contrast to content-based recommendation approaches with transductive setting that expects all test entities to appear in the system for training. In addition, it differs to the recent inductive node embedding techniques, e.g., GraphSAGE [15], since it does not require inductive entities to be aware of their neighbor information and thus is relatively more practical for real-world applications. Figure 1 gives two illustrative applications of this problem. In web academic systems such as AMiner or Google Scholar, we can leverage existing author–paper relations and paper content information (e.g., title, abstract, venue, or references) to develop a relation learning model so as to identify correlated authors for new papers by using the learned model. The model is also applicable for reviewer search or paper recommendation. In this case, new papers in test period are inductive entities. In e-commerce platform such as Amazon or Walmart online shopping systems, we can employ existing useritem relations and item content information (e.g., title, description, or image) to build a relation learning model to recommend new items to users. In this case, new items are inductive entities. To develop an effective model for inductive relation inference, there are two main challenges:

- **Inductive Relation.** Since we aim to infer the inductive relations between existing entities and unseen ones, transductive content-based recommendation methods [6, 17, 50] and unsupervised node embedding models [9, 13, 36] are initially designed for transductive inference. Their performances will be impaired when they are generalized/adapted to inductive relation inference. In addition, the recent inductive node embedding techniques, e.g., GraphSAGE [15, 58], are also not suitable for this task as they need neighbor information of each inductive entity as prior knowledge. This information, however, is often unavailable in practice. Therefore, the first challenge is how to design a machine learning model that can infer the inductive relations without any prior knowledge about neighbor information of inductive entities.
- **Heterogeneous Contents.** An inductive entity can carry heterogeneous contents in many real-world applications. The recent studies [8, 57] for inductive relation learning are

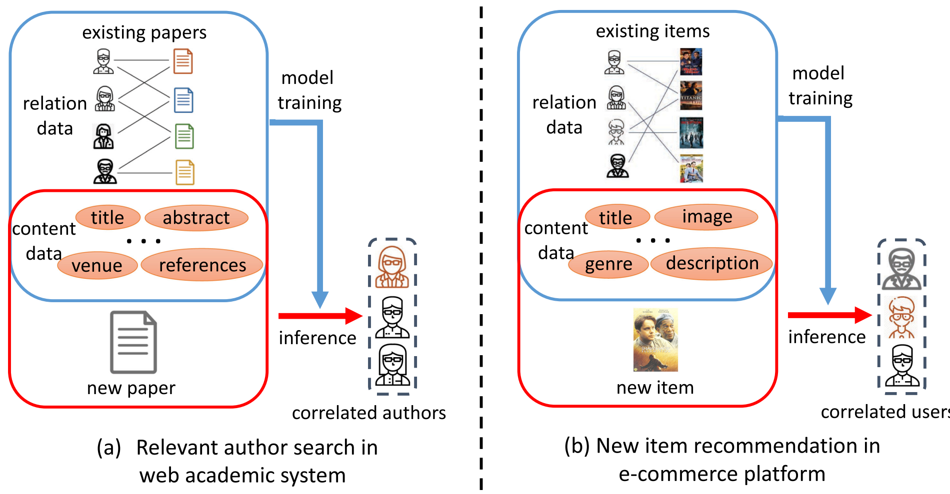

Fig. 1. Applications of inductive contextual personalization problem: (a) relevant author search for new paper in web academic system and (b) new item recommendation for user in e-commerce platform.

designed for a specific task. In addition, they use relatively simple and inflexible architecture to aggregate heterogeneous contents, which limits the model capability and make inference performance inferior. Different content information represent different aspects of an entity. For example, the text description of an item (in e-commerce data) has semantic meaning while the picture of an item expresses its visual nature. In addition, intuitively, different content features have different contributions to the entity representation and accumulate the different content features could lead to large expression capability. For instance, the abstract feature should be more important for representing a paper than the venue feature (in academic data) and aggregating different content features helps generate useful paper representation. Due to this reason, the second challenge is how to develop an effective neural network aggregator to fuse heterogeneous content features for representing each inductive entity.

To address the above challenges and solve the proposed problem, we develop a novel model for **inductive contextual personalization (ICP)** through contextual relation learning. Specifically, we first formulate the pairwise relations between entity pairs (e.g., user-item or author–paper) with a ranking optimization scheme that employs neural network aggregator to fuse entity's heterogeneous content features. The proposed neural network aggregator can be easily implemented and is flexible for extension. Next, we introduce a node embedding term to capture implicit contextual relations between entity pairs, which is further incorporated into the former ranking objective as a smoothness constraint. Finally, we design a batch gradient descent procedure with adaptive negative sampling to train model and optimize parameters. The learned model is able to generate inductive entity's representation and further infer relations between existing entities and inductive ones.

To summarize, the main contributions of this article are as follows:

- We propose the problem of inductive contextual personalization where entities' heterogeneous content information are available. It is different from previous transductive relation learning tasks and recent inductive node representation learning work. It can handle the

entities that do not present in the current system and does not require any prior knowledge about neighbor information of each inductive entity.
- We develop a novel machine learning model, i.e., ICP, to solve the problem. ICP employs inductive pairwise ranking optimization with a neural content aggregator to model direct relations between entity pairs, and is further augmented with a node embedding term that encodes implicit contextual relations among these entities.
- We conduct extensive experiments on two datasets derived from AMiner and Amazon systems to justify the effectiveness of ICP. Our results demonstrate that the proposed model outperforms numerous baseline methods for two real-world applications.

2 RELATED WORK

This work is closely related to several lines of research including relation learning, content-based recommendation, and node embedding learning.

Relation Learning. Recently, many studies have been devoted to relation learning for different applications such as personalized recommendation [17, 18, 22, 24, 31, 38, 39] and relevant entity search [8, 23, 30, 35, 44, 51, 57, 61, 62]. One typical approach is using pairwise ranking optimization. The traditional models [38, 39] denote node (entity) in the system with latent feature, and discriminate between the set of correlated items and the set of remaining uncorrelated items by a pairwise ranking loss. In addition, some recent extended studies [14, 16, 17, 24, 54] have been presented by using content information. For example, He et al. [16] utilized image content and Wang et al. [24] employed heterogeneous item relations to enhance latent feature-based pairwise ranking models. Besides content information, node/entity can be associated with contextual neighbor information in the system. Accordingly, some task-guided and contextual relation learning models [8, 57] have been developed for inductive relation inference by using both node's content and contextual neighbor information.

Content-based Recommendation. Different from the traditional collaborative filtering approaches [19, 40, 52] that purely utilize user preference and behavior data for recommendation, the majority of content-based recommendation approaches [32] either employ side information or develop sophisticated algorithms to process available information. Different types of side information have been employed to build content-based recommendation models including attributes/features of items [4, 12, 43], user generated content [7], visual and multimedia features [11, 17], and others [2, 34]. From the algorithmic perspective, there are many different types of contextual/content-based recommendation models such as meta-path-based approaches [21, 56], new metadata encoding methods [48], or deep learning models [34, 63, 64]. Content-based recommendation models have been proved to be more effective in real-world applications than their pure collaborative filtering-based counterparts in many application domains. In addition, many cold-start recommendation approaches [3, 31, 41, 42, 50, 60] have been proposed to leverage content information to alleviate the sparsity issue.

Node Embedding Learning. The node embedding learning in graph have attracted a lot of attention in recent years. The purpose is to automate the discovery of meaningful vectorized embedding for nodes in the graph so as to facilitate various downstream applications. There are two major groups of approaches. The first group is proximity-preserving methods (or network embedding) [9, 13, 36, 45, 59] that capture graph structure information and learn node embeddings by preserving proximity between nodes. The second group is message-passing models (or graph neural networks) [5, 15, 26, 49, 53, 58] that learn node embeddings by aggregating the neighbors' information through neural network. For both groups of approaches, the learned node embeddings can be further utilized to relation inference (e.g., link prediction) or other tasks such as node classification.

Here we summarize the connection and difference between our study and the above related work. The previous relation learning work usually focus on transductive inference or are designed for a specific task while we study inductive relation inference that involves new nodes to the system and can benefit a variety of applications. Despite some task-guided and content-aware relation learning models are applicable for inductive relation inference, they are either too simple to aggregate heterogeneous content features or not extensible for incorporating more content information, which limits the model capability. Like content-based recommendation models, we utilize node/entity content information to build learning model. However, most of previous methods are initially developed for transductive inference. In addition, they cannot fully explore both node content information and contextual neighbor information that we well incorporate into our model. The current node embedding learning methods either target learning transductive node embedding or their performances will be impaired when they are generalized/adapted to relation inference for inductive nodes. In this work, we address this issue by developing a neural network-based content aggregator to fuse node/entity content information and learn inductive node embedding.

3 PROBLEM DEFINITION

In this section, we first introduce the concept of content-associated bipartite graph that will be used throughout the article. Then, we formally define the problem of inductive contextual personalization.

Definition 3.1 (Content-associated Bipartite Graph). A content associated bipartite graph is defined as a graph $G = (U, V, E_{UV}, C)$ with two types of nodes U and V. Each edge $e_{uv} \in E_{UV}$ can only exist between two nodes of different types, and each node v (or u) can be associated with content $C_v \in C$, e.g., numerical attribute, text, or image.

We use the above content-associated bipartite graph to represent connected relations in the systems. For example, an author–paper graph (see Figure 1(a)) represents the relations between authors and papers in web academic system. In this graph, each paper node is associated with various contents such as abstract, references, and venue. Besides, a user-item graph (see Figure 1(b)) denotes the interactions between online users and items in e-commerce platform, and each item node is associated with different contents such as title, description, and picture.

Before giving the problem definition, we clarify two types of nodes in bipartite graph. In this work, there are two disjoint sets of nodes, i.e., U and V. The nodes in U and V that appear in the graph for training are defined as transductive nodes while the nodes that do not present in the training period are regarded as inductive nodes. For example, new papers in an academic system are inductive nodes since they are unseen during model training. Moreover, inductive nodes have no neighbor information as they are incoming new nodes in the graph. For example, new items in an e-commerce platform have no interaction with the existing users. This setting is different from the recent inductive node embedding study, e.g., GraphSAGE [15], and more realistic for real-world applications. Based on the above definitions, the problem is formally defined as:

PROBLEM 1 [INDUCTIVE CONTEXTUAL PERSONALIZATION]. *Given the historical relations between two types of nodes (from U and V) in the system (represented as a bipartite graph) before time T_s, accompanying with node content information C, the problem is to design a machine learning model to infer relation between existing nodes and new inductive nodes appearing after T_s.*

Note that in this work we study inductive relation inference that involves one type of inductive node (e.g., new paper in web academic system or new item in e-commerce platform), which is applicable for many real-world applications, such as reviewer/author searching for new papers or new item recommendation for users. This study is extensible or modifiable for the scenario of two types of inductive nodes.

4 MODEL FRAMEWORK

In this section, we elaborate the details of proposed model by three parts: (1) inductive relation modeling relying on pairwise ranking optimization and neural network-based content aggregator; (2) implicit contextual relation augmentation based on node embedding learning; (3) model training procedure design using adaptive negative sampling strategy.

4.1 Inductive Pairwise Relation Modeling

4.1.1 Pairwise Ranking Optimization. Given a content-associated bipartite graph $G = (U, V, E_{UV}, C)$ and assume that each node $v \in V$ (or $u \in U$) is associated with content $C_v \in C$, we first model the direct relations between two types of nodes. Specifically, in the training period, for a given node $u \in U$, the correlated node $v \in \{\mathcal{EN}(u)\}$ should be ranked higher than the uncorrelated node $v' \in \{V \backslash \mathcal{EN}(u)\}$, where $\mathcal{EN}(u)$ represents the set of nodes that ever interacted with u. Thus we formulate the objective function as follows:

$$\mathcal{L}_{dir} = \sum_{u \in U} \sum_{v \in \mathcal{EN}(u)} \sum_{v' \notin \mathcal{EN}(u)} \left[\log \sigma(s_{u,v} - s_{u,v'}) \right], \tag{1}$$

where $\sigma(\cdot)$ is the sigmoid function and $s_{u,v}$ denotes the correlation score between u and v. The current techniques, e.g., BPR [39], use the inner product of two nodes' latent features to measure $s_{u,v}$ and employ the optimized features to infer potential interactions among existing nodes in the graph. However, such transductive setting is not designed for the inductive inference task in this work as the inductive nodes (e.g., new items) are unseen (without latent feature) during the model training period. To solve this challenge, we encode the heterogeneous contents of each node by the neural network aggregator f, which will be trained and further utilized to infer the inductive node representation. That is, the embedding representation of each $v \in V$ (or $u \in U$) is formulated as $f(C_v) \in \mathbb{R}^{d \times 1}$ (d: the embedding dimension) and C_v is the content set of v, which can be numerical attribute, text, image, and so on. Therefore, f can be used to infer representations of inductive nodes. The inner product of two nodes' content embeddings measures the correlation score, i.e., $s_{u,v} = f(C_u)^T f(C_v)$. In addition, $f(C_v)$ (or $f(C_u)$) will be degenerated to a latent feature $Q_v \in \mathbb{R}^{d \times 1}$ if v has no content information. Next, we introduce neural network content aggregator f in detail.

4.1.2 Neural Network Content Aggregator. To encode and fuse the heterogeneous content information of each node (e.g., $v \in V$) as a fixed size embedding, we design three independent aggregators based on different neural network architectures. For the ease of presentation, we denote the feature of ith content in C_v as $\mathbf{x}_i \in \mathbb{R}^{d_f \times 1}$ (d_f: content feature dimension). Note that $\mathbf{x}_i$ can be pre-trained using different techniques w.r.t. different content meanings.

Attention Aggregator. This aggregator is motivated by the intuition that different contents contribute differently to the final representation of v. For instance, in web academic system, the abstract content of a paper should be more important than venue information. Thus we propose the first aggregator, i.e., attention aggregator, based on the attention mechanism [1], which is formulated as follows:

$$f(C_v) = \sum_{i \in C_v} \alpha_i \mathbf{x}_i,\ \alpha_i = \frac{\exp\{\langle \mathcal{FC}_{\theta_x}(\mathbf{x}_i), \mathbf{z} \rangle\}}{\sum_{j \in C_v} \exp\left\{\left\langle \mathcal{FC}_{\theta_x}(\mathbf{x}_j), \mathbf{z} \right\rangle\right\}}, \tag{2}$$

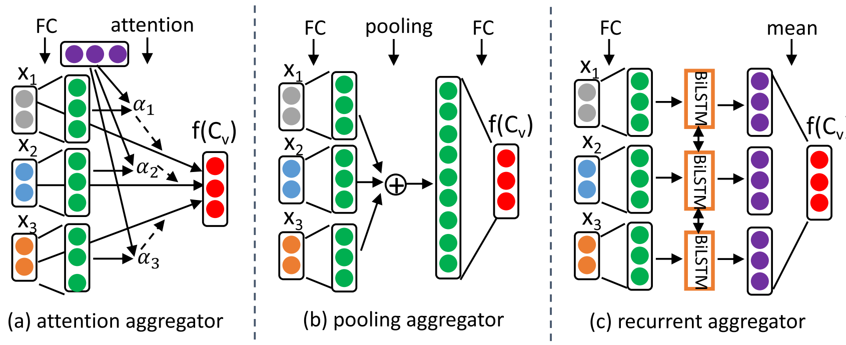

Fig. 2. Neural network content aggregators proposed in this work: (a) attention aggregator based on attention mechanism (FC: fully connected neural network), (b) pooling aggregator using different pooling operators, and (c) recurrent aggregator built on recurrent neural network.

where $\mathbf{z} \in \mathbb{R}^{d\times1}$ is the attention parameter, $\mathcal{FC}_{\theta_x}$ denotes the **fully connected neural network (FC)** with parameter $\theta_x = \{\mathbf{W}, \mathbf{b}\}$ ($\mathbf{W} \in \mathbb{R}^{d\times d_f}$, $\mathbf{b} \in \mathbb{R}^{d\times1}$), i.e., $\mathcal{FC}_{\theta_x}(\mathbf{x}_i) = \mathbf{W}_i\mathbf{x}_i + \mathbf{b}_i$, and $\langle\cdot,\cdot\rangle$ is the vector dot product operator. The attention aggregator learns the weight α_i of $\mathbf{x}_i$ and combines all content features to obtain the final representation of v, as illustrated in Figure 2(a).

Pooling Aggregator. Motivated by the fact that modeling feature interaction appropriately can improve relation learning model [37, 46], we propose the second aggregator, i.e., pooling aggregator, and formulate it as follows:

$$f(C_v) = \mathcal{FC}_{\theta_C}\left\{\bigoplus_{i\in C_v}\mathcal{FC}_{\theta_x}(\mathbf{x}_i)\right\}, \tag{3}$$

where $\bigoplus$ is the pooling operator that can be average, summation, or concatenation operation over all content features. $\mathcal{FC}_{\theta_C}$ and $\mathcal{FC}_{\theta_x}$ are two independent FC layers that have different functions: $\mathcal{FC}_{\theta_x}$ maps different content features into the same feature space and $\mathcal{FC}_{\theta_C}$ captures feature interactions. Therefore, the pooling aggregator jointly considers feature transformation and interaction to generate the final representation of v, as illustrated in Figure 2(b).

Recurrent Aggregator. Besides the previous two aggregators, we further design a more complex architecture and propose the third aggregator, i.e., recurrent aggregator, based on the recurrent neural network, which is formulated as follows:

$$f(C_v) = \frac{\sum_{i\in C_v}\mathcal{RNN}_{\theta_{rnn}}\{\mathcal{FC}_{\theta_x}(\mathbf{x}_i)\}}{|C_v|}, \tag{4}$$

where $\mathcal{RNN}_{\theta_{rnn}}\{\mathcal{FC}_{\theta_x}(\mathbf{x}_i)\} \in \mathbb{R}^{d\times1}$ is the concatenation output of the **bi-directional LSTM (Bi-LSTM)**:

$$\mathcal{RNN}_{\theta_{rnn}}\{\mathcal{FC}_{\theta_x}(\mathbf{x}_i)\} = \left[\overrightarrow{LSTM}\{\mathcal{FC}_{\theta_x}(\mathbf{x}_i)\}, \overleftarrow{LSTM}\{\mathcal{FC}_{\theta_x}(\mathbf{x}_i)\}\right], \tag{5}$$

where LSTM is defined in Reference [20]. To be more specific, the recurrent aggregator first uses a FC layer to map all content features to the same feature space, then employs the Bi-LSTM to capture in-depth feature interaction and accumulate expression capability of all content features, and finally utilizes a mean pooling layer[1] over all hidden states to obtain the final representation

[1]We also experimented with max pooling or taking the last hidden state but did not find significant difference in performance.

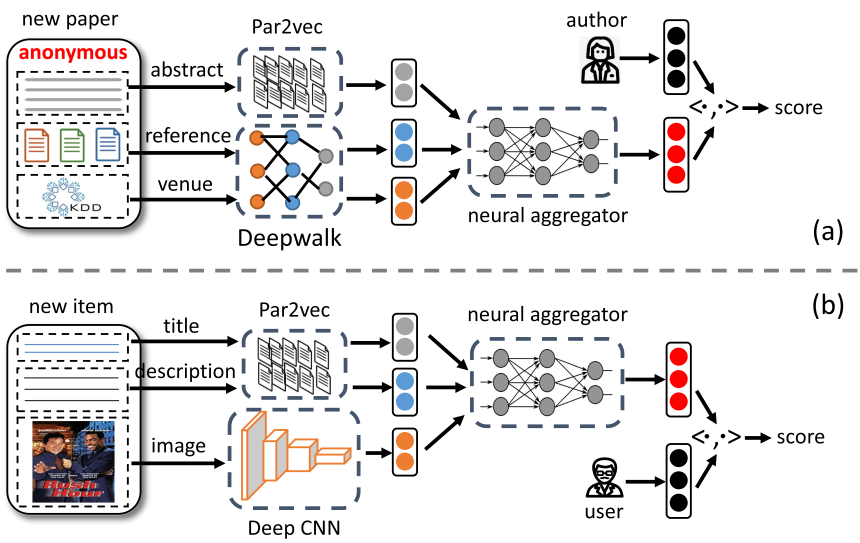

Fig. 3. Illustrations of inductive pairwise relation modeling for (a) relevant author search in web academic system and (b) new item recommendation in e-commerce platform.

of v, as illustrated in Figure 2(c). Note that we adapt Bi-LSTM to operate on an unordered set C_v, which is inspired by graph neural network model [15] for aggregating unordered neighbor information.

In summary, there are several advantages of the proposed content aggregators: (1) they have concise structures with low complexity (small parameter size), making the model implementation and tuning relatively easy; (2) they are able to fuse the heterogeneous content information, leading to a strong expression capability; and (3) they are flexible to add extra content features, making the model extension and improvement convenient.

Figure 3 illustrates inductive pairwise relation modeling for paper-author in web academic system and useritem in e-commerce platform. Specifically, we first extract different contents of each paper/item; then, we employ different techniques to pretrain these content features (e.g., the language model Par2Vec [29] for text, the node embedding model DeepWalk [36]) for existing nodes, the vision model CNN [28] for image); subsequently, we feed these features to the proposed aggregator and obtain the refined representation of each paper/item; finally, we denote each author/user (without content information) as a latent feature and compute the correlation score between paper/item aggregated content embedding and author/user latent feature.

4.2 Contextual Relation Augmentation

$\mathcal{L}_{dir}$ explicitly models the direct interactions between two types of nodes yet ignores implicit contextual relations that can be inferred from existing relations and beneficial for the model enhancement. To address this issue, we further extend the model by formulating the implicit contextual relations. One way to formulate such contextual relations is using another pairwise ranking function. However, it cannot distinguish direct interact relations with indirect contextual relations. Thus, we aim at formulating indirect contextual relations as a smoothness constraint over the objective of direct relations, i.e., $\mathcal{L}_{dir}$. Specifically, inspired by node embedding technique (e.g., DeepWalk [36]), we first employ random walk node sequences to extract these contextual relations. Figure 4 gives illustrative examples of this process. Specifically, in the web academic system (see author-paper graph in Figure 4(a)), a random walk node sequence $\mathbf{w}_1 \equiv \{1 \rightarrow a \rightarrow 2 \rightarrow c \rightarrow 4 \rightarrow d\}$ is generated.

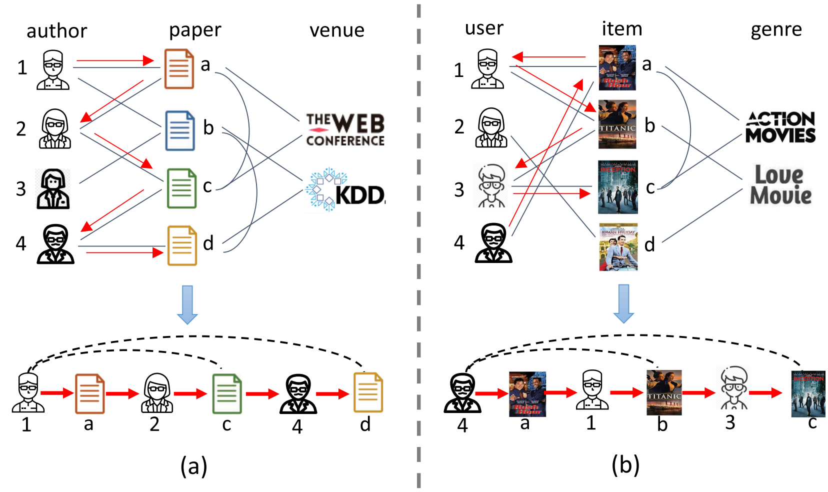

Fig. 4. Examples of indirect contextual relation extraction on (a) academic graph and (b) e-commerce graph. Red arrow represents random walk sequence and black dash line indicates indirect correlation between two nodes.

Besides direct relations, e.g., author 1 writes paper a or author 2 writes paper c, $\mathbf{w}_1$ can also capture implicit contextual relations, e.g., author 1 has potential correlation with paper c and d since they are written by co-author and co-author of co-author, respectively. In the e-commerce platform (see user-item graph in Figure 4(b)), random walk $\mathbf{w}_4 \equiv \{4 \rightarrow a \rightarrow 1 \rightarrow b \rightarrow 3 \rightarrow c\}$ captures both direct relations, e.g., user 4 buys item a or user 1 buys item b, and implicit contextual relations, e.g., user 4 has potential correlation with item b and c since they are bought buy co-purchaser or co-purchaser of co-purchaser.

Suppose a set of node sequences W are generated, we then model the implicit contextual relations over W with the Skip-gram model [29], which is formulated as:

$$\mathcal{L}_{indir} = \sum_{\mathbf{w} \in W} \sum_{u \in \mathbf{w}} \sum_{v \in c_u} \log p(v|u), \tag{6}$$

where c_u is the set of contextual neighbors (V-type neighboring nodes within the given window size τ) of u in $\mathbf{w}$. The probability $p(v|u)$ is computed with Softmax function in terms of $s_{u,v}$, i.e., $p(v|u) = \frac{exp(s_{u,v})}{\sum_{v' \in V} exp(s_{u,v'})}$ (V: set of all V-type nodes in the graph) and $s_{u,v} = f(C_u)^T f(C_v)$. Obviously, the above objective function is to maximize the co-occurrence likelihood of contextual neighbors for each target node in short random walks. By applying the negative sampling strategy [33], $p(v|u)$ can be approximated as:

$$\log p(v|u) \approx \log \sigma(s_{u,v}) + \sum_{i=1}^{k} \mathbb{E}_{v' \sim p_V(v')} \{\log \sigma(-s_{u,v'})\}, \tag{7}$$

where $p_V(v')$ is the pre-defined noise distribution and k is the negative sample size. In this model, we set $k = 1$ since $k > 1$ makes no much difference in performance. In addition, let $\mathcal{IN}(u)$ denote the set of implicit contextual neighbors of u extracted in W. Thus we can rewrite the objective $\mathcal{L}_{indir}$ as:

$$\mathcal{L}_{indir} = \sum_{u \in U} \sum_{v \in \mathcal{IN}(u)} \sum_{\substack{v' \notin \mathcal{IN}(u) \\ v' \notin \mathcal{EN}(u)}} [\log \sigma(s_{u,v}) + \log \sigma(-s_{u,v'})]. \tag{8}$$

In other words, the correlation score between u and contextual neighbor v should be larger than those of uncorrelated ones that are not implicit contextual neighbors or not direct neighbors.

To summarize, two main advantages of contextual relation augmentation are as follows: (1) It can model indirect contextual correlations among different nodes that cannot be extracted from the bipartite graph directly, enhancing the model capability, and (2) the random walk-based strategy can be easily extended on heterogeneous graph [9] in the system for extracting more heterogeneous contextual relations and improving model performance.

4.3 Model Inference

4.3.1 Training Procedure. Finally, we introduce the model training procedure. By incorporating contextual relation modeling objective $\mathcal{L}_{indir}$ into $\mathcal{L}_{dir}$ as an augmentation term, the objective of proposed model is formulated as follows:

$$\mathcal{L}_{ICP} = -(\mathcal{L}_{dir} + \gamma \mathcal{L}_{indir}) + \lambda \mathcal{L}_{reg}, \tag{9}$$

where $\mathcal{L}_{reg}$ is L_2 regularization term and λ controls regularization penalty, the tradeoff factor γ balances $\mathcal{L}_{dir}$ and $\mathcal{L}_{indir}$. We denote all model parameters as Φ, and use $\mathcal{T}_{dir}$ and $\mathcal{T}_{indir}$ to denote the sets of all triplets (u, v, v') in $\mathcal{L}_{dir}$ and $\mathcal{L}_{indir}$, respectively. Thus we can rewrite $\mathcal{L}_{ICP}$ as following:

$$\mathcal{L}_{ICP} = \sum_{(u,v,v') \in \mathcal{T}_{dir}} [-\log \sigma(s_{u,v} - s_{u,v'})] + \sum_{(u,v,v') \in \mathcal{T}_{indir}} [-\log \sigma(s_{u,v}) - \log \sigma(-s_{u,v'})] + \lambda \left\|\Phi\right\|_2^2 . \tag{10}$$

To minimize $\mathcal{L}_{ICP}$ and optimize the model parameters, we design a training procedure based on batch sampling and the Adam optimizer [25]. Specifically, at each iteration, we sample a mini-batch of triplets in $\mathcal{L}_{dir}$ and $\mathcal{L}_{indir}$, then accumulate the objective according to Equation (10), and next update the parameters via Adam. We repeat the training iterations until the change between two consecutive iterations is sufficiently small. The pseudocode of ICP is described in Algorithm 1.

4.3.2 Adaptive Negative Sampling. The uniform negative sampling method used in training procedure, i.e., v' is randomly sampled, results slow convergence and gradient vanishes in training, as demonstrated by previous work [38]. To address this problem, we propose an adaptive negative sampling strategy to improve the training procedure of ICP. Specifically, in training period of pairwise ranking model, we need to select negative sample (i.e., v' in Equation (1)) from the whole candidate set. The gradient magnitude is measured by $(1 - \sigma(s_{u,v} - s_{u,v'}))$ [38]. Therefore, we need to obtain larger $s_{u,v'}$ for larger gradient magnitude when doing negative sampling. A simple way is to rank $s_{u,v'}$ for a batch of candidates and v' with higher ranking (larger score) should be selected with larger probability. Motivated by this, we introduce a ranking position generator: $\hat{r} = min\,\{K, int(\psi)\}$, where ψ is generated from a pre-defined distribution: $\psi = -\frac{\ln p}{\mu}$ (p: random value from uniform distribution, μ: rate parameter) and K is the candidate batch size. When we set p as $(0, 1]$ uniform distribution and $\mu = 1$, ψ will be a small value for most cases. In other words, there is a large probability that $\hat{r}$ will be small integer and gradient magnitude will be large. Such probability could be adjusted by setting different μ. Figure 5 gives an illustrative example of this strategy for the author–paper relation learning. Specifically, we first randomly sample a batch $\mathcal{S}$ (with size K) of uncorrelated authors of the given paper and employ the content aggregator to infer representation of this paper, then rank sampled authors according to the correlation scores with the paper, finally select the author ranked at position $\hat{r}$. To train ICP with adpative negative sampling strategy, we only need to replace random sampling step (line 16) in Algorithm 1 by adaptive sampling step (line 17 in comment).

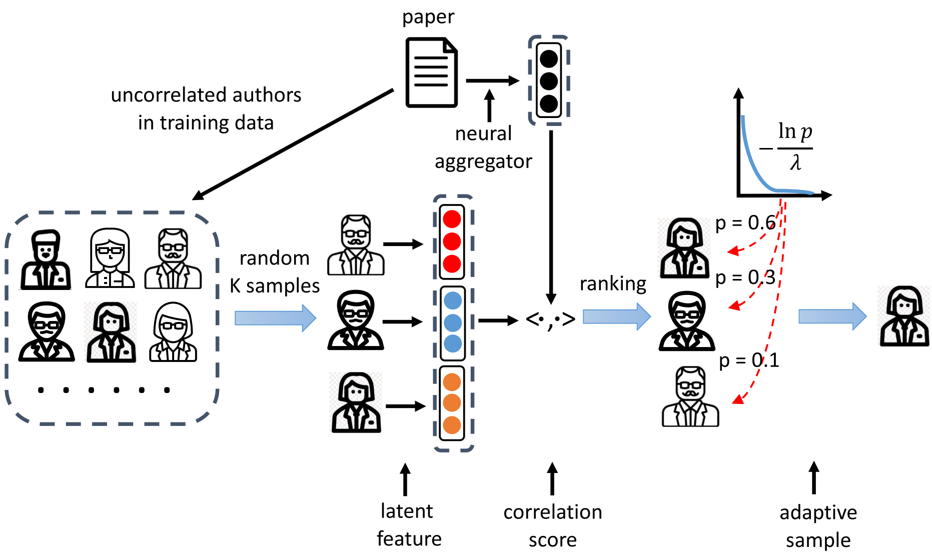

Fig. 5. Illustration of the proposed adaptive negative sampling strategy for relevant author search in web academic system.

There are two advantages of this sampling strategy: (1) uncorrelated authors with relative large correlation scores are more likely to be selected since $int(\psi)$ is usually small integer, which can avoid the slow convergence and gradient vanishes in training; (2) by introducing the batch size K, we only need to compute correlation scores between u and a small set of v', which largely reduces the sampling time.

4.3.3 Relation Inference (Model Test). After sufficient model training, we infer correlation score between a new inductive node v^* (or u^*) and an existing node u (or v) by:

$$s_{u,v^*} = f^*(C_u)^T f^*(C_{v^*}), \tag{11}$$

where f^* represent the learned content aggregator. It will be optimized latent feature Q^* for nodes without content information. For example, in the test period of relevant author search task, we use the learned content aggregator to generate representation of each new paper, then compute its correlation scores with the trained latent features of authors. Finally, we rank all authors for the query paper according to the correlation scores.

5 EXPERIMENTS

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

- **RQ1:** Whether ICP can outperform various baseline methods for different applications of inductive relation inference?
- **RQ2:** Whether different components, i.e., content aggregators, contextual relation augmentation, and adaptive negative sampling, are effective for improving the model performance?
- **RQ3:** Whether each content feature, e.g., text or image, contributes to improving the model capability?
- **RQ4:** How do hyper-parameters, i.e., tradeoff factor, impact the model performance?

ALGORITHM 1: Training Procedure of ICP

input: initialized latent features of nodes (without content), pre-train content features of nodes (with content), sets of direct neighbors $\mathcal{EN}$ and implicit contextual neighbors $\mathcal{IN}$
output: optimized latent features, optimized neural content aggregator
while *not done* **do**
  $\mathcal{T}_{dir} \leftarrow GenExpTriplet(\mathcal{EN})$
  $\mathcal{T}_{indir} \leftarrow GenConTriplet(\mathcal{IN})$
  for $i = 1 : batch\_n$ **do**
    sample a batch of (u, v, v') in $\mathcal{T}_{dir}$
    sample a batch of (u, v, v') in $\mathcal{T}_{indir}$
    accumulate the loss by Equation~(10)
    update the parameters by Adam Optimizer
  end
end
return latent features, neural content aggregator
Function $GenExpTriplet(\mathcal{EN})$ ($\underline{GenConTriplet(\mathcal{IN}}$))
$\mathcal{T} \leftarrow \{\}$
for $u \in U$ **do**
  for $v \in \mathcal{EN}(u)$ ($\underline{v \in \mathcal{IN}(u)}$) **do**
    $v' \leftarrow$ random $v' \notin \mathcal{EN}(u)$ ($\underline{v' \notin \{\mathcal{EN}(u) \bigcup \mathcal{IN}(u)\}}$)
    //$*$ $v' \leftarrow AdaptiveSample(u)$: adaptive sampling strategy described in Section 4.3.2 $*$//
    $\mathcal{T} \leftarrow \{\mathcal{T} \bigcup (u, v, v')\}$
  end
end
return $\mathcal{T}$
Function $AdaptiveSample(u)$
$\mathcal{S} \leftarrow K$ random $v' \notin \mathcal{EN}(u)$ ($\underline{v' \notin \{\mathcal{EN}(u) \bigcup \mathcal{IN}(u)\}}$)
$\hat{R} \leftarrow$ ranking of $\mathcal{S}$ according to correlation score $s_{uv'}$
$p \leftarrow$ random value from uniform distribution
$\hat{r} \leftarrow \min\left\{K, int(-\frac{\ln p}{\mu})\right\}$
$v' \leftarrow \hat{R}(\hat{r})$
return v'

5.1 Experiment Design

5.1.1 Datasets. We use two kinds of data, i.e., academic publication data and e-commerce data, from web academic system and e-commerce platform respectively, for model evaluation.

Academic publication data. We extract one dataset from the AMiner [47] data.[2] The dataset contains high quality publications of 10 years from 2006 to 2015 in six domains, i.e., **Artificial Intelligence (AI)**, **Computer Vision (CV)**, **Natural Language Processing (NLP)**, **Data Mining (DM)**, **Databases (DB)**, and **Web/Information Retrieval (W/IR)**. In each domain, we select three venues[3] that are considered to have influential papers. Each paper has bibliographic content information: title, abstract, authors, references, year, venue. Overall, the dataset contains 28,645 authors, 21,043 papers, 18 venues, and 69,311 author–paper writing relations.

[2] https://aminer.org/.
[3] AI: ICML, AAAI, IJCAI. CV: CVPR, ICCV, ECCV. NLP: ACL, EMNLP, NAACL. DM: KDD, WSDM, ICDM. DB: SIGMOD, VLDB, ICDE. W/IR: WWW, SIGIR, CIKM.

E-commerce data. We extract a dataset from the Amazon [16] data (Movies and TV category).[4] The dataset contains user reviews and item metadata from Amazon from 05/1996 to 07/2014. Each item has various content information: title, text description, price, genre, and image. Overall, the dataset contains 18,340 users, 56,361 items, and 629,125 user-item review relations.

5.1.2 Applications. We propose two applications of inductive relation inference, i.e., relevant author search and new item recommendation, using the above two datasets respectively.

Relevant author search. In academic publication data, given the set of papers $U_{<T_s}$ published before a given time T_s, accompanying with various bibliographic contents (e.g., abstract content, references, and venue), the task is to rank all potential authors $v \in V$ for each new anonymous paper $u^* \in U_{\geq T_s}$ ($U_{\geq T_s}$: set of papers published in or after T_s), such that its top ranked authors are true authors of u^*.

New item recommendation. In e-commerce data, given the historical reviews between users and items before time T_s, accompanying with item content (i.e., title, text description, image), the task is to rank all new items $v^* \in V_{\geq T_s}$ for each user $u \in U$ ($V_{\geq T_s}$: set of items appear in or after T_s), such that its top ranked items (for recommendation) are reviewed/bought by u after T_s.

It is worth to mention that we consider inductive scenario for model design and experimental evaluation, where inductive term refers to new nodes in the system. Specifically, for the recommendation task, we assume there are some new items put on sale and we target recommending them to users (who are interested in these items). Therefore, it is userwise model design and evaluation. Similarly, for the relevant author search task, we assume there are some new papers and we aim at finding the relevant authors for these papers. It is paperwise model design and evaluation. Both tasks involve inductive nodes that are new to the system and can carry content information. However, they vary from each other in area (academic service vs. e-commerce), task (relevant author search vs. new item recommendation), and data (scientific publication data vs. online user behavior data).

5.1.3 Baseline Methods. We consider numerous baseline methods that are illustrated as follows.

- **Unsupervised node embedding.** This is unsupervised node embedding model. Note that this type of method is only applicable for academic publication data since new items in e-commerce data have no context information like venue or reference of new paper in academic data. It first learns embeddings of different types of nodes (author, paper, venue) in the graph constructed based on paper information in training data, then obtains each new paper's representation by averaging node embeddings of references and venue of this paper, finally computes correlations between new paper embedding and author embeddings, and identifies relevant authors according to the ranking of correlation scores. Both homogeneous graph model **DeepWalk (DW)** [36] and heterogeneous **graph model metapath2vec (MP2V)** [9] are utilized to learn node embeddings.
- **Unsupervised inductive node embedding.** This is graph neural network model for learning inductive node embedding. We use latent feature to represent each existing node and employ **GSAGE** [15] to aggregate neighbors' latent features of each inductive node. To be more specific, paper's neighbors are its authors in academic data and item's neighbors are who review it. Note that each new inductive node has no neighbor information in this work, we take neighbors of the most similar existing node of inductive node as its neighbors. The similarity between inductive node and existing node is computed based on the content

[4] http://jmcauley.ucsd.edu/data/amazon/index.html.

similarity, i.e., paper content similarity in academic publication data and item description content similarity in e-commerce data.

- **Zero-shot relation learning.** This is zero-shot relation learning model. We employ the same idea in **ZSRL** [31] for cold-start recommendation. Specifically, we first introduce an encoder to map node behavior (i.e., paper-author writing, user-item reviewing) into content attributes, then utilize a symmetric decoder to reconstruct node behavior from content attributes. Finally, the loss between reconstructed and true node behavior is further used to train the encoder-decoder framework. The optimized decoder is further employed to infer correlations between existing nodes and new inductive nodes by using content attributes of inductive nodes.
- **Content-based recommendation.** This is content-based recommendation model. We utilize **content-based pairwise ranking method (C-BPR)**. It first employs a pooling operator over the content features to obtain inductive node representations, then applies pairwise ranking technique [39] to learn latent features of existing nodes, next computes correlations between each new inductive node and existing nodes, finally identifies potential correlated nodes for inductive node according to the ranking of correlation scores. We try three content feature pooling operators, i.e., average, summation, and concatenation, to compute inductive node embedding and report the best of them, which is concatenation in this work. For review data, we consider one more baseline, **VBPR,** that only uses image feature to represent item. It can be seen as a variant of VBPR [17] since there is no latent feature of inductive item nodes. In addition, we employ one more content-based model called **CNM** [50], which combines various content features through neural network for recommendation.
- **Task-guided relation learning.** This is the task-guided relation learning model. We employ **TGR** [8] and **Camel** [8] for comparison. TGR uses the learnable weights to combine content features of inductive nodes, and captures both direct relations and implicit contextual relations between node pairs. Further, the node embedding model guided by pairwise relation is utilized to learn weights of content features of inductive nodes and latent features of existing nodes. Different from TGR, Camel utilizes distance metric learning to model pairwise relations and employs node embedding model to learn embeddings of inductive nodes and existing nodes.
- **Collaborative similarity embedding.** This is a unified framework that exploits both direct relations and indirect contextual relations. We employ the idea in **CSE** [6] and makes it adapt to inductive nodes. It first encodes node content information through neural network, then extracts several relations available in the bipartite graph, finally learn node embedding for relation inference through pairwise ranking optimization over these relations.

To summarize, we can group all of the baseline methods into three groups based on the information used in different models. The first group is the model that uses both direct relations and indirect contextual relations, which includes DW, MP2V, and GSAGE. The second group represents method that uses both direct relations and node content information, which contains ZSRL, C-BPR, VBPR, and CNM. The last group denotes the model that utilizes direct relations and indirect contextual relations as well as node content information, which includes TGR, Camel, and CSE.

5.1.4 Evaluation Metrics. We use four popular metrics, i.e., **Recall@k (Rec@k)**, **Precision@k (Pre@k)**, **Mean Reciprocal Rank (MRR)**, and **Normalized Discounted Cumulative Gain (NDCG)**, to evaluate performance of each method. The definitions of all metrics are illustrated as follows.

- **Recall@k.** It shows the ratio of true authors/items being retrieved in the top-k return list, which can be computed by:

$$Rec@k = \frac{1}{|U_{test}|} \sum_{u \in U_{test}} \frac{|\hat{\mathcal{EN}}(u) \bigcap \mathcal{EN}(v)|}{|\mathcal{EN}(u)|}, \tag{12}$$

where U_{test} is the set of test papers/users in test data for evaluation, $\mathcal{EN}(u)$ and $\hat{\mathcal{EN}}(u)$ denote the sets of true authors/items of paper/user u and top-k ranked authors/items by a specific method, respectively.
- **Precision@k.** It reflects the precision of top-k ranked authors/items by a specific method, which is defined as:

$$Pre@k = \frac{1}{|U_{test}|} \sum_{u \in U_{test}} \frac{|\hat{\mathcal{EN}}(u) \bigcap \mathcal{EN}(u)|}{k}. \tag{13}$$

- **MRR.** It measures the ranking quality of true authors/items, which is defined as:

$$MRR = \frac{1}{|U_{test}|} \sum_{u \in U_{test}} \frac{1}{|\mathcal{EN}(u)|} \sum_{u \in \mathcal{EN}(u)} \frac{1}{\hat{r}(v)}, \tag{14}$$

where $\hat{r}(v)$ represents the ranking position of author/item v in the whole list.
- **NDCG.** It measures the ranking quality of true authors/items by the following equation:

$$NDCG = \frac{1}{|U_{test}|} \sum_{u \in U_{test}} \frac{\sum_{v \in \mathcal{EN}(u)} \frac{2^{rel_v} - 1}{\log_2(i+1)}}{\sum_{i=1}^{k'} \frac{1}{\log_2(i+1)}}, \tag{15}$$

where rel_v equals 1 if author/item v is ranked in top-k′ (k′ = $|\mathcal{EN}(u)|$) list, otherwise 0.

5.1.5 *Reproducibility Settings.* There are several key settings of experiments for reproducibility, which are illustrated as following.

- **Train/test split.** In academic data, papers published before the given time T_s are used as training data and papers in or after T_s are left for evaluation. We design two train/test splits by setting T_s = 2012 and 2013. In e-commerce data, we split train/test data sequentially based on item's first appearing time. Experiments of two splits (in terms of item number), i.e., 8/2 and 7/3, are conducted.
- **Hyper-parameters.** For ICP, the feature dimension d is set to 128, the regularization parameter λ equals 0.05. The tradeoff factor γ of objective function (Equation (9)) equals 0.1. In contextual relations modeling, the number of walks rooted at each node equals 10, the length of each walk is set to 20, and the window size τ equals 3. In addition, the sample size K equals 10 and rate parameter μ equals 1.0 for the adaptive negative sampling strategy. The grid search of feature dimension is performed for baseline methods and most of them are set to 128.
- **Evaluation candidates.** We adopt the same setting in previous work [8, 55] that randomly samples a set of 100/50 negative false authors/items and combines it with the set of true authors/items to form a candidate set, which makes the results large enough for clear performance comparison of different methods. The reported scores are averaged over 10 repeated experiments of such setting.
- **Pre-trained content features.** For the academic data, we use the Par2Vec [29] to pre-train paper abstract feature. Besides, we construct the author–paper–venue graph [9] based on papers information in training data and employ DeepWalk [36] to learn node embeddings.

Table 1. Performance Comparisons of All Methods for Relevant Author Search in Web Academic System

Metric	T_s	DR + IDR			DR + C			DR + IDR + C			
		DW	MP2V	GSAGE	ZSRL	CNM	C-BPR	TGR	Camel	CSE	ICP
Rec@3	2013	0.101	0.107	0.206	0.316	0.369	0.381	0.410	0.443	0.436	**0.459**
	2012	0.097	0.084	0.173	0.261	0.299	0.370	0.375	0.404	0.398	**0.420**
Pre@3	2013	0.078	0.090	0.174	0.253	0.302	0.311	0.336	0.352	0.355	**0.368**
	2012	0.075	0.074	0.135	0.180	0.216	0.270	0.277	0.301	0.296	**0.311**
Rec@5	2013	0.152	0.146	0.289	0.403	0.451	0.471	0.501	0.547	0.540	**0.555**
	2012	0.155	0.153	0.254	0.356	0.403	0.463	0.466	0.511	0.496	**0.531**
Pre@5	2013	0.073	0.075	0.146	0.196	0.226	0.234	0.253	0.276	0.273	**0.277**
	2012	0.070	0.072	0.118	0.151	0.178	0.206	0.211	0.232	0.226	**0.239**
MRR	2013	0.116	0.146	0.194	0.284	0.313	0.328	0.344	0.359	0.358	**0.381**
	2012	0.108	0.112	0.171	0.235	0.267	0.322	0.324	0.341	0.340	**0.355**
NDCG	2013	0.076	0.110	0.157	0.250	0.338	0.329	0.371	0.367	0.381	**0.405**
	2012	0.071	0.076	0.126	0.189	0.226	0.294	0.318	0.327	0.321	**0.352**

Differences between ICP and each baseline method are significant with $p < 0.05$. DR, IDR, C denote direct relation, indirect contextual relation, and content information used in different models.

The reference and venue features of each paper are obtained from the learned node embeddings. For the e-commerce data, we use Par2Vec to pre-train item title and description features and employ CNN [28] to obtain item image feature. The dimension d_f of all content features is set to 128.

5.2 Results—Relevant Author Search

5.2.1 Comparison with Baselines (RQ1). The performances of all methods for relevant author search are reported in Table 1. The best results are highlighted in bold. DR, IDR, C denote direct relation, indirect contextual relation, and content information used in different models. Note that the reported ICP is the best proposed model with recurrent content aggregator, contextual relation augmentation, and adaptive negative sampling strategy. According to this table:

- The pairwise ranking models with paper content features (C-BPR, CNM) and ZSRL have better performances than the node embedding baseline methods. It demonstrates that utilizing content information generates better node representation than using node latent embedding.
- TGR, Camel, and CSE are better than the pairwise ranking models, showing that content-aware relation learning model extended by contextual relations generates better embeddings of author and paper for relevant author search.
- The proposed ICP performs best in all experimental settings. The average improvement of ICP over the best baseline method is over 4%, demonstrating the power of our model in learning and inferring inductive relations between author and paper in web academic system.

5.2.2 Comparison with Model Variants (RQ2). ICP is a joint learning model of several components. How different content aggregators impact the model performance? Whether implicit contextual relations modeling and adaptive negative sampling strategy improve the model capability? To answer these questions, we conduct experiments (T_s = 2013) to evaluate performances of model variants that include: (1) ICP^{Att}, (ICP$^{Pool}_{Ave}$, ICP$^{Pool}_{Sum}$, ICP$^{Pool}_{Cat}$), and ICP^{RNN} that use attention,

Table 2. Performance Comparisons of Different Model Variants in Relevant Author Search

Metric	ICP^{Att}	ICP^{Pool}_{Ave}	ICP^{Pool}_{Sum}	ICP^{Pool}_{Cat}	ICP^{RNN}	ICP^{RNN}_{R}	ICP
Rec@3	0.375	0.396	0.405	0.407	0.432	0.440	**0.459**
Pre@3	0.298	0.317	0.329	0.328	0.348	0.355	**0.368**
Rec@5	0.494	0.510	0.512	0.523	0.547	0.551	**0.555**
Pre@5	0.241	0.251	0.254	0.258	0.271	0.274	**0.277**
MRR	0.323	0.336	0.343	0.347	0.361	0.364	**0.381**
NDCG	0.312	0.332	0.343	0.346	0.369	0.389	**0.405**

Differences between ICP and each variant model are significant with $p < 0.05$.

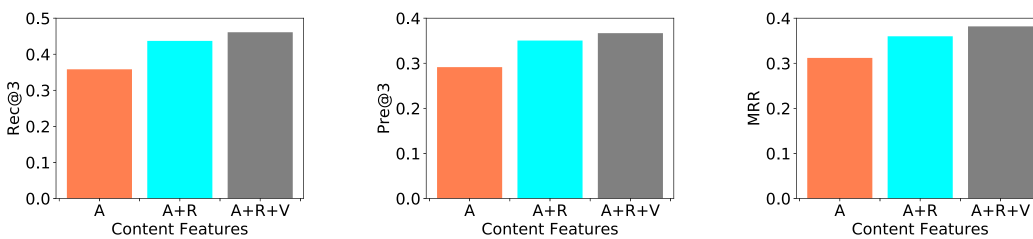

Fig. 6. The performances of proposed model with different content features for relevant author search task (A: abstract feature, R: reference feature, V: venue feature).

(average, summation, concatenation) pooling, and recurrent aggregator, respectively. In addition, all of them use $\mathcal{L}_{dir}$ objective and train model by uniform negative sampling strategy; (2) ICP^{RNN}_{R}, which is based on ICP^{RNN} and further augmented by implicit contextual relations, i.e., $\mathcal{L}_{indir}$; (3) the best proposed model, i.e., ICP, which is based on ICP^{RNN}_{R} and trains model by adaptive negative sampling strategy. The results are reported in Table 2. From this table:

- ICP^{Pool} is better than ICP^{Att} and ICP^{RNN} outperforms ICP^{Pool}. It means that pooling aggregator explicitly models feature interaction and generates better node representation than attention aggregator. In addition, the recurrent aggregator has larger expression capability than the pooling aggregator.
- ICP^{RNN}_{R} has better result than ICP^{RNN}, demonstrating that the contextual relation augmentation is effective for improving model performance.
- ICP outperforms all model variants including ICP^{RNN}_{R}, indicating that it is better to train model by adaptive negative sampling strategy, which further elevates model performance.

5.2.3 Impact of Different Content Features (RQ3). In this task, we use features of three content information: abstract, references, and venue, as the input of neural content aggregator. To study the impacts of different features on model performance, we conduct experiments (T_s = 2013) to evaluate the performances (in terms of Rec@3, Pre@3, MRR) of ICP with different combinations of content features, i.e., A, A+R, and A+R+V, which represent abstract feature, abstract + reference features, abstract + reference + venue features, respectively. The performances of ICP with different features are reported in Figure 6. It is easy to see that model with A+R features is better than model with only A feature and model with A+R+V features has the best performance, demonstrating that incorporating different content features elevates model capability and improves performance.

5.2.4 Impact of Tradeoff Factor γ (RQ4). ICP inherently models implicit contextual relations by incorporating $\mathcal{L}_{indir}$ into $\mathcal{L}_{dir}$ with tradeoff factor γ. To investigate the impact of γ, we conduct

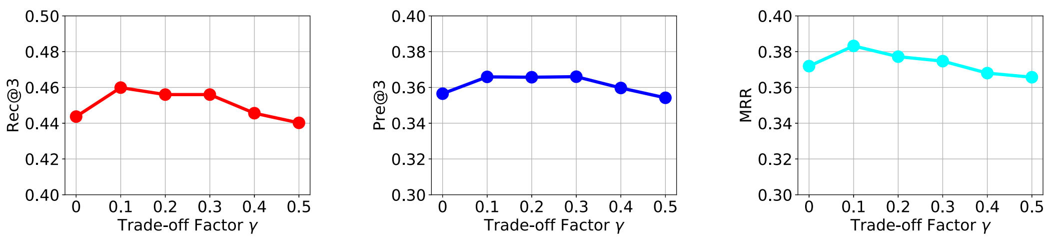

Fig. 7. The impact of tradeoff factor γ of the joint objective function on model performance for relevant author search task.

Table 3. Performance Comparisons of All Methods for New Item Recommendation in e-commerce Platform

Metric	Train/Test	DR + IDR	DR + C				DR + IDR + C			
		GSAGE	ZSRL	VBPR	CNM	C-BPR	TGR	Camel	CSE	ICP
Rec@3	8/2	0.095	0.115	0.127	0.172	0.168	0.195	0.214	0.187	**0.232**
	7/3	0.103	0.116	0.119	0.171	0.176	0.187	0.195	0.185	**0.226**
Pre@3	8/2	0.089	0.104	0.112	0.144	0.144	0.165	0.179	0.159	**0.189**
	7/3	0.133	0.146	0.151	0.205	0.209	0.220	0.224	0.217	**0.249**
Rec@5	8/2	0.151	0.172	0.196	0.250	0.267	0.277	0.298	0.263	**0.313**
	7/3	0.164	0.182	0.188	0.246	0.251	0.265	0.273	0.264	**0.310**
Pre@5	8/2	0.088	0.096	0.107	0.131	0.138	0.146	0.156	0.138	**0.164**
	7/3	0.125	0.140	0.145	0.184	0.187	0.196	0.201	0.195	**0.219**
MRR	8/2	0.121	0.136	0.147	0.178	0.187	0.195	0.211	0.188	**0.223**
	7/3	0.122	0.138	0.141	0.175	0.179	0.187	0.191	0.185	**0.214**
NDCG	8/2	0.089	0.104	0.114	0.144	0.153	0.162	0.181	0.160	**0.193**
	7/3	0.124	0.143	0.145	0.189	0.192	0.203	0.206	0.199	**0.235**

Differences between ICP and each baseline method are significant with $p < 0.05$. DR, IDR, C denote direct relation, indirect contextual relation, and content information used in different models.

experiments ($T_s = 2013$) to evaluate model performance under different γ. The performance curves (in terms of Rec@3, Pre@3, and MRR) are shown in Figure 7. With the increment of γ, performance increase as contextual relations information are incorporated into model. However, result decreases with the further increment of γ due to the bias on implicit contextual relations. Therefore, incorporating the suitable amount of contextual relations information leads to the best performance of ICP.

5.3 Results - New Item Recommendation

5.3.1 Comparison with Baselines (RQ1). The results of all methods for new item recommendation are reported in Table 3. The best results are highlighted in bold. DR, IDR, C denote direct relation, indirect contextual relation, and content information used in different models. According to this table:

- The pairwise ranking models with content features (CNM, VBPR, C-BPR) and ZSRL have better performances than GSAGE as they utilize content information and generate better node representation. In addition, C-BPR is better than VBPR because the former uses various content features while the latter only uses image feature.

Table 4. Performance Comparisons of Different Model Variants for New Item Recommendation

Metric	ICP^{Att}	ICP^{Pool}_{Ave}	ICP^{Pool}_{Sum}	ICP^{Pool}_{Cat}	ICP^{RNN}	ICP^{RNN}_{R}	ICP
Rec@3	0.195	0.203	0.209	0.214	0.219	0.223	**0.232**
Pre@3	0.160	0.170	0.169	0.173	0.178	0.182	**0.189**
Rec@5	0.275	0.282	0.288	0.297	0.303	0.306	**0.313**
Pre@5	0.142	0.147	0.146	0.151	0.154	0.157	**0.164**
MRR	0.196	0.198	0.203	0.209	0.212	0.212	**0.223**
NDCG	0.165	0.169	0.170	0.176	0.179	0.183	**0.193**

Differences between ICP and each variant model are significant with $p < 0.05$.

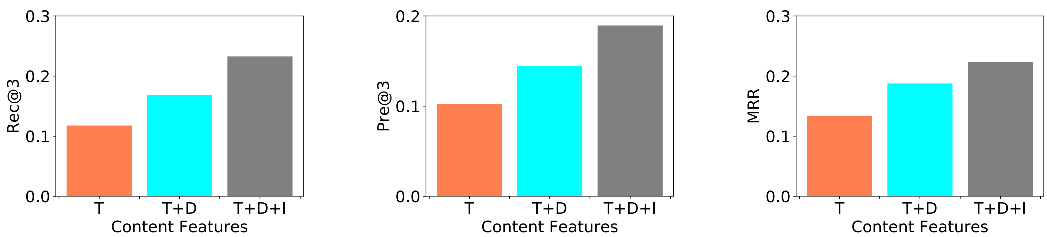

Fig. 8. The performances of proposed model with different content features for new item recommendation (T: title feature, D: description feature, I: image feature).

- TGR, Camel, and CSE outperform the pairwise ranking models, indicating that content-aware relation learning model with contextual relations augmentation generates better representations of user and item for new item recommendation.
- ICP performs best in all experimental settings. The average improvement of ICP over the best baseline method is over 9%, demonstrating the effectiveness of our model in learning and inferring inductive relations between users and items in e-commerce platform.

5.3.2 Comparison with Model Variants (RQ2). We conduct experiments (train/test data size split = 8/2) to evaluate the performances of different model variants to show impacts of different neural content aggregators, implicit contextual relation augmentation, and adaptive negative sampling strategy for the new item recommendation task. The results of different model variants (as described in the first task) are reported in Table 4. The main takeaways from this table are similar to the Table 2. That is, the recurrent aggregator has the best capability, the contextual relations further improve the model performance, and the adaptive negative sampling strategy benefits model training.

5.3.3 Impact of Different Content Features (RQ3). We design experiments (train/test data size split = 8/2) to evaluate the performances of ICP with different combinations of content features, i.e., T, T+D, and T+D+I, which represent title feature, title + description features, title + description + image features, respectively. The performances of ICP with different features are reported in Figure 8. Similarly to the first task, the model with all three content features has the best result and the model with only one content feature has the worst result, demonstrating that incorporating different content features elevates model capability and improves performance.

5.3.4 Impact of Trade-off Factor γ (RQ4). Following the same step in the first task, we conduct experiments (train/test data size split = 8/2) to study the impact of tradeoff factor γ on model performance, as shown by Figure 9. Accordingly, the certain value of γ leads to the best result of

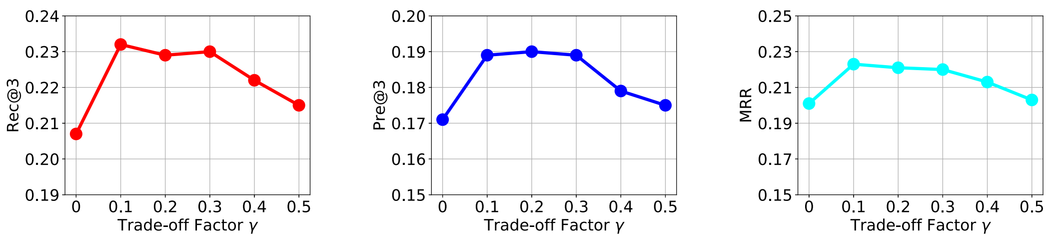

Fig. 9. The impact of tradeoff factor γ of the joint objective function on model performance for new item recommendation task.

ICP, indicating that suitable amount of implicit contextual relations information are incorporated into the model for best performance.

5.3.5 Additional Analysis. Besides γ, we also study impacts of other hyper-parameters, e.g., embedding dimension d, for both tasks and obtain similar results. That is, the suitable settings of hyper-parameters lead to the best performance of proposed model. Note that the best of proposed model employs recurrent neural network to operate an unordered set of different contents. To study the impact of content order in recurrent aggregator, we randomly shuffle content order to obtain performances at different content orders. We find that results with different content orders are close to each other, demonstrating that the content order makes little impact on model performance.

6 CONCLUSIONS

In this paper, we presented the problem of inductive contextual personalization, and proposed a novel model called ICP to solve the problem. ICP jointly considered inductive pairwise ranking optimization scheme, content aggregator, and node embedding smoothness term. In the training stage, a batch gradient descent procedure with adaptive negative data sampling was employed to optimize the model parameters. Extensive experiments on two datasets derived from different real-world systems, i.e., AMiner and Amazon, demonstrated that ICP can outperform numerous baseline approaches on two inductive relation inference applications, i.e., relevant author search and new item recommendation. The initial success of this work suggests several avenues for future research. Future work might consider extending the current model to more applications such as problem with more than one type of edges or with one single type of nodes. It is also possible to incorporate sequential information and modify pairwise ranking model as sequential pairwise ranking model for sequential recommendation problem. Moreover, this work investigated inductive scenario that involves one types of inductive node while our model is extensible or modifiable for the case of more than one types of inductive nodes.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

[2] Iman Barjasteh, Rana Forsati, Farzan Masrour, Abdol-Hossein Esfahanian, and Hayder Radha. 2015. Cold-start item and user recommendation with decoupled completion and transduction. In *RecSys*. 91–98.

[3] Oren Barkan, Noam Koenigstein, Eylon Yogev, and Ori Katz. 2019. CB2CF: A neural multiview content-to-collaborative filtering model for completely cold item recommendations. In *RecSys*. 228–236.

[4] Leonardo Cella, Stefano Cereda, Massimo Quadrana, and Paolo Cremonesi. 2017. Deriving item features relevance from past user interactions. In *UMAP*. 275–279.

[5] Yukuo Cen, Xu Zou, Jianwei Zhang, Hongxia Yang, Jingren Zhou, and Jie Tang. 2019. Representation learning for attributed multiplex heterogeneous network. In *KDD*. 1358–1368.

[6] Chih-Ming Chen, Chuan-Ju Wang, Ming-Feng Tsai, and Yi-Hsuan Yang. 2019. Collaborative similarity embedding for recommender systems. In *WWW*. 2637–2643.

[7] Li Chen, Guanliang Chen, and Feng Wang. 2015. Recommender systems based on user reviews: The state of the art. *User Model. User-Adapt. Interact.* 25, 2 (2015), 99–154.

[8] Ting Chen and Yizhou Sun. 2017. Task-guided and path-augmented heterogeneous network embedding for author identification. In *WSDM*. 295–304.

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

[10] Dmitry Efimov, Lucas Silva, and Benjamin Solecki. 2013. Kdd cup 2013-author-paper identification challenge: Second place team. In *KDD Cup Workshop*.

[11] Mehdi Elahi, Yashar Deldjoo, Farshad Bakhshandegan Moghaddam, Leonardo Cella, Stefano Cereda, and Paolo Cremonesi. 2017. Exploring the semantic gap for movie recommendations. In *RecSys*. 326–330.

[12] Zeno Gantner, Lucas Drumond, Christoph Freudenthaler, Steffen Rendle, and Lars Schmidt-Thieme. 2010. Learning attribute-to-feature mappings for cold-start recommendations. In *ICDM*. 176–185.

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

[14] Xinyu Guan, Zhiyong Cheng, Xiangnan He, Yongfeng Zhang, Zhibo Zhu, Qinke Peng, and Tat-Seng Chua. 2019. Attentive aspect modeling for review-aware recommendation. *ACM Trans. Inf. Syst.* 37, 3 (2019), 1–27.

[15] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NeurIPS*. 1024–1034.

[16] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *WWW*. 507–517.

[17] Ruining He and Julian McAuley. 2016. VBPR: Visual Bayesian personalized ranking from implicit feedback. In *AAAI*. 144–150.

[18] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*. 173–182.

[19] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. 1999. An algorithmic framework for performing collaborative filtering. In *SIGIR*. 230–237.

[20] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.* 9, 8 (1997), 1735–1780.

[21] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S. Yu. 2018. Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In *KDD*. 1531–1540.

[22] Chao Huang, Xian Wu, Xuchao Zhang, Chuxu Zhang, Jiashu Zhao, Dawei Yin, and Nitesh V Chawla. 2019. Online purchase prediction via multi-scale modeling of behavior dynamics. In *KDD*. 2613–2622.

[23] 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.

[24] Wang-Cheng Kang, Mengting Wan, and Julian McAuley. 2018. Recommendation through mixtures of heterogeneous item relationships. In *CIKM*. 1143–1152.

[25] Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

[26] Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

[27] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer*8 (2009), 30–37.

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *NeurIPS*. 1097–1105.

[29] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*. 1188–1196.

[30] Chun-Liang Li, Yu-Chuan Su, Ting-Wei Lin, Cheng-Hao Tsai, Wei-Cheng Chang, Kuan-Hao Huang, Tzu-Ming Kuo, Shan-Wei Lin, Young-San Lin, Yu-Chen Lu, et al. 2015. Combination of feature engineering and ranking models for paper-author identification in KDD Cup 2013. *J. Mach. Learn. Res.* 16, 1 (2015), 2921–2947.

[31] Jingjing Li, Mengmeng Jing, Ke Lu, Lei Zhu, Yang Yang, and Zi Huang. 2019. From zero-shot learning to cold-start recommendation. In *AAAI*. 4189–4196.

[32] Pasquale Lops, Dietmar Jannach, Cataldo Musto, Toine Bogers, and Marijn Koolen. 2019. Trends in content-based recommendation. *User Model. User-Adapt. Interact.* 29, 2 (2019), 239–249.

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

[34] Cataldo Musto, Tiziano Franza, Giovanni Semeraro, Marco de Gemmis, and Pasquale Lops. 2018. Deep content-based recommender systems exploiting recurrent neural networks and linked open data. In *UMAP*. 239–244.

[35] Chanyoung Park, Donghyun Kim, Qi Zhu, Jiawei Han, and Hwanjo Yu. 2019. Task-guided pair embedding in heterogeneous network. In *CIKM*. 489–498.

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

[37] Steffen Rendle. 2010. Factorization machines. In *ICDM*. 995–1000.
[38] Steffen Rendle and Christoph Freudenthaler. 2014. Improving pairwise learning for item recommendation from implicit feedback. In *WSDM*. 273–282.
[39] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*. 452–461.
[40] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *WWW*. 285–295.
[41] Martin Saveski and Amin Mantrach. 2014. Item cold-start recommendations: Learning local collective embeddings. In *RecSys*. 89–96.
[42] Suvash Sedhain, Scott Sanner, Darius Braziunas, Lexing Xie, and Jordan Christensen. 2014. Social collaborative filtering for cold-start recommendations. In *RecSys*. 345–348.
[43] Mohit Sharma, Jiayu Zhou, Junling Hu, and George Karypis. 2015. Feature-based factorized bilinear similarity model for cold-start top-n item recommendation. In *SDM*. 190–198.
[44] 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*. 992–1003.
[45] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *WWW*. 1067–1077.
[46] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *WSDM*. 565–573.
[47] 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.
[48] Flavian Vasile, Elena Smirnova, and Alexis Conneau. 2016. Meta-prod2vec: Product embeddings using side-information for recommendation. In *RecSys*. 225–232.
[49] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *ICLR*.
[50] Maksims Volkovs, Guang Wei Yu, and Tomi Poutanen. 2017. Content-based neighbor models for cold start in recommender systems. In *RecSys*. 1–6.
[51] 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.
[52] Jun Wang, Arjen P. De Vries, and Marcel J. T. Reinders. 2006. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *SIGIR*. 501–508.
[53] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S. Yu. 2019. Heterogeneous graph attention network. In *WWW*. 2022–2032.
[54] Xin Xin, Xiangnan He, Yongfeng Zhang, Yongdong Zhang, and Joemon Jose. 2019. Relational collaborative filtering: Modeling multiple item relations for recommendation. In *SIGIR*. 125–134.
[55] 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.
[56] 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.
[57] Chuxu Zhang, Chao Huang, Lu Yu, Xiangliang Zhang, and Nitesh V. Chawla. 2018. Camel: Content-aware and meta-path augmented metric learning for author identification. In *WWW*. 709–718.
[58] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. 2019. Heterogeneous graph neural network. In *KDD*. 793–803.
[59] Chuxu Zhang, Ananthram Swami, and Nitesh V. Chawla. 2019. Shne: Representation learning for semantic-associated heterogeneous networks. In *WSDM*. 690–698.
[60] Chuxu Zhang, Lu Yu, Yan Wang, Yan Wang, and Xiangliang Zhang. 2017. Collaborative user network embedding for social recommender systems. In *SDM*.
[61] Chuxu Zhang, Lu Yu, Xiangliang Zhang, and Nitesh V. Chawla. 2018. Task-guided and semantic-aware ranking for academic author-paper correlation inference. In *IJCAI*. 3641–3647.
[62] Dong Zhang, Shu Zhao, Zhen Duan, Jie Chen, Yanping Zhang, and Jie Tang. 2020. A multi-label classification method using a hierarchical and transparent representation for paper-reviewer recommendation. *ACM Trans. Inf. Syst.* 38, 1 (2020), 1–20.
[63] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *Comput. Surv.* 52, 1 (2019), 1–38.
[64] Yongfeng Zhang, Qingyao Ai, Xu Chen, and W. Bruce Croft. 2017. Joint representation learning for top-n recommendation with heterogeneous information sources. In *CIKM*. 1449–1458.

Received May 2020; revised January 2021; accepted February 2021