Chapter 7
Heterogeneous Graph Representation for Recommendation

Abstract With the rapid development of web services, various kinds of useful auxiliary data (a.k.a., side information) become available in recommender systems. To characterize these complex and heterogeneous auxiliary data, heterogeneous graph (HG) representation methods have been widely adopted due to the flexibility in modeling data heterogeneity. In this chapter, we introduce three HG representation based recommendation systems solving the unique challenges existing in diverse real-world scenarios, including Top-N recommendation (MCRec), cold-start recommendation (MetaHIN), bibliographic recommendation (ASI). In the field of HG representation for recommendation, methods mainly contain three key components: HG constructions, HG representation learning and recommendation based on the HG representation.

7.1 Introduction

In recent years, recommender systems, which help users discover items of interest from a large resource collection, have been playing an increasingly important role in various online services [15], such as item recommendation and collaborator recommendation. Traditional recommendation methods (*e.g.*, matrix factorization) mainly aim to learn an effective prediction function for recovering and completing interaction matrix. With the rapid development of web services, various kinds of auxiliary data (*e.g.*, side information) becomes available in recommender systems. Although auxiliary data is likely to contain useful information for recommendation, it is difficult to model and utilize these heterogeneous and complex information in recommender systems.

As a promising direction, heterogeneous graph has been proposed as a powerful information modeling method [30, 27, 25]. Due to its flexibility in modeling data heterogeneity, heterogeneous graph (HG) has been adopted in recommender systems to characterize rich auxiliary data. Under the HG based representation, the recommendation problem can be considered as a similarity search task over the HG

[30]. Such a recommendation setting is called as HG based recommendation. HG based recommendation has been widely adopted in recommender systems due to its excellence in modeling complex context information [8, 39, 28]. Although existing HG based recommendation methods have achieved performance improvement to some extent, they still meet unique challenges existing in diverse applications.

In this chapter, we introduce three HG representation based recommendation systems solving the challenges in diverse real-world scenarios, including Top-N recommendation, cold-start recommendation and bibliographic recommendation. First, to leverage rich meta-path based context for top-*N* recommendation, **M**etapath based **C**ontext for **Rec**ommendation (named **MCRec**) is designed as a novel deep neural network with the co-attention mechanism, explicitly learning the representation of meta-path and their complex relationships. Second, to better address the cold-start problem in recommendation, a **Meta**-learning approach to cold-start recommendation on **H**eterogeneous **I**nformation **N**etworks (named **MetaHIN**) is proposed to capture richer semantics, by exploiting the power of meta-learning at the model level and HINs at the data level simultaneously. Third, to capture relationships between authors in bibliographic recommendation, **A**uthor **S**et **I**dentification model (named **ASI**) first studies the problem of author set identification, which is to identify an author set related to an anonymous paper.

7.2 Top-*N* Recommendation

7.2.1 Overview

Existing heterogeneous graph based recommendation methods can be categorized into two types. The first type leverages path based semantic relatedness as direct features for recommendation relevance [8, 39, 28], and the second type performs some transformation on path based similarities for learning effective transformed features [39, 42]. These two types of methods both extract meta-path based features for improving the characterization of two-way user-item interactions, as illustrated in Fig. 7.1. While these existing methods have two major shortcomings. First, these models seldom learn an explicit representation for path or meta-path in the recommendation task. Second, they do not consider the mutual effect between the meta-path and the involved user-item pair in an interaction.

A basic idea for the problems is to leverage rich meta-path information from heterogeneous graph for top-*N* recommendation in a more principled way. Our main idea is to: (1) learn explicit representations for meta-path based context tailored for the recommendation task. (2) characterize a three-way interaction of the form: $\langle$ user, meta-path, item$\rangle$. However, the solution is challenging. We have to consider three key problems: (1) how to design the base architecture that is suitable for the complicated heterogeneous graph based interaction scenarios. (2) how to generate meaningful path instances for constructing high-quality meta-path based context. (3)

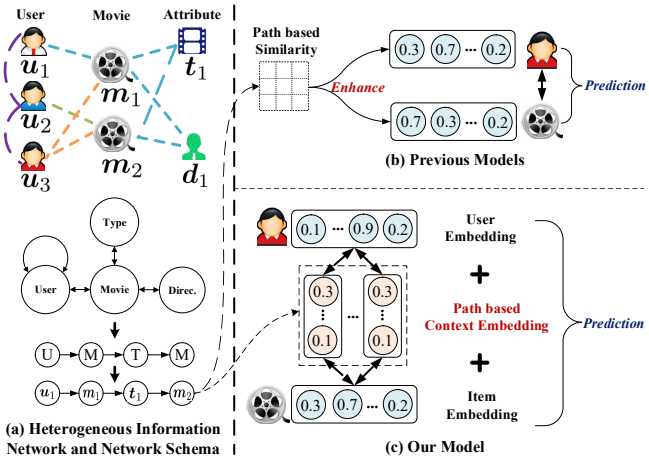

Fig. 7.1 The illustration for heterogeneous graph based recommendation setting (network schema, meta-path, path instance) and the comparison between our model and previous methods (two-way interaction v.s. three-way meta-path based interaction).

how to capture the mutual effect between the involved user-item pair and meta-path based context in an interaction.

In this section, we introduce a novel deep neural network with the co-attention mechanism by leveraging rich meta-path based context, which is able to learn interaction-specific representations for users, items and meta-path context. We present the proposed model that leverages **M**eta-path based **C**ontext for **Rec**ommendation, called **MCRec**. To our knowledge, it is the first time that meta-path based context has been explicitly modeled in a three-way neural interaction model for top-N recommendation in heterogeneous graph. More details about MCRec are given in the next section.

7.2.2 The MCRec Model

7.2.2.1 Model Framework

Differing existing heterogeneous graph based recommendation models, which only learn the representations for users and items, we explicitly incorporate meta-paths as the context in an interaction between a user and an item. Instead of modeling the two-way interaction $\langle user, item \rangle$, we aim to characterize a three-way interaction $\langle user$, meta-paths, $item \rangle$. We present the overall architecture for the proposed model in Fig. 7.2. As we can see, for learning a better interaction function that generates the recommendations, we learn the representations (*i.e.*, embedding) for users, items and their interaction contexts. Besides the components for learning user and item embeddings, the most important part lies in the embedding of meta-path

based context. We firstly use a priority based sampling technique to select high-quality path instances. Hence, the meta-path based context is first modeled into a low-dimensional embedding using a hierarchical neural network. With the initially learned embeddings for users, items and meta-path based context, the co-attention mechanism further improves the three representations through alternative enhancement. Due to the incorporation of meta-path based context, our model is expected to yield a better performance and also improve the interpretability for recommendation results.

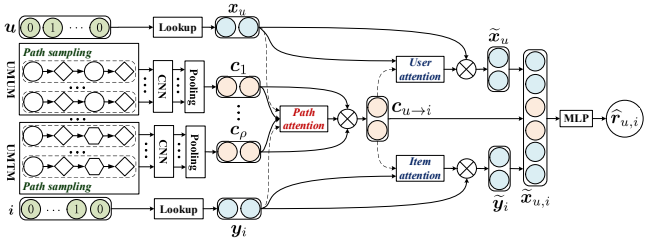

Fig. 7.2 The overall architecture of the proposed model.

7.2.2.2 Characterizing Meta-path based Context for Interaction

Sampling Path Instances via Priority based Random Walk. Existing heterogeneous graph embedding models mainly adopt a meta-path guided random walk strategy to generate path instances [7], relying on a uniform sampling over the outgoing nodes. Intuitively, at each step, the walker should wander to a neighbor of a higher "priority" score with a larger probability, since such an outgoing node can reflect more reliable semantics by forming a closer link. Hence, we propose to use a similar pretrain technique to measure the priority degree of each candidate out-going nodes. Firstly, we train the feature based matrix factorization framework SVDFeature [5] on all the available historical interaction records to learn a potential vector for each node that has a history of user-item interactions. We can incorporate the entities from heterogeneous graph related to an interaction as the context of a training instance. With the learned latent factors, we can compute the pairwise similarities between two consecutive nodes along a path instance, and then average these similarities for ranking the candidate path instances. Finally, given a meta-path, we only keep top K path instances with the highest average similarities.

Meta-path based Context Embedding. After obtaining path instances from multiple meta-paths, we focus on how to model these meta-path based context as an informative embedding. Our method naturally follows a hierarchical structure: embedding a single path instance → embedding a single meta-path → embedding the aggregated meta-paths.

For path instance embedding, formally, given a path p from some meta-path ρ, let $\mathbf{X}^p \in \mathbb{R}^{L \times d}$ denote the embedding matrix formed by concatenating node embeddings, where L is the length of the path instance and d is the embedding dimension for entities. We adopt the commonly used Convolution Neural Network (CNN) to deal with sequences of variable lengths. The structure of CNN consists of a convolution layer and a max pooling layer. We learn the embedding of a path instance p using CNN as follows:

$$\mathbf{h}_p = CNN(\mathbf{X}^p; \theta). \tag{7.1}$$

where $\mathbf{X}^p$ denotes the matrix of the path instance p and θ denotes all the related parameters in CNNs.

For meta-path embedding, since a meta-path can produce multiple path instances, we further apply the max pooling operation to derive the embedding for a meta-path. Let $\{\mathbf{h}_p\}_{p=1}^K$ denote the embeddings for the K selected path instances from meta-path ρ. The embedding $\mathbf{c}_\rho$ for meta-path ρ can be given

$$\mathbf{c}_\rho = \text{max-pooling}(\{\mathbf{h}_p\}_{p=1}^K). \tag{7.2}$$

Our max pooling operation is carried out over K path instance embeddings, which aims to capture the important dimension features from multiple path instances.

For simple average embedding for meta-path based context, we apply the average pooling operation to derive the embedding for modeling the aggregate meta-path based context

$$\mathbf{c}_{u \to i} = \frac{1}{|\mathcal{M}_{u \to i}|} \sum_{\rho \in \mathcal{M}_{u \to i}} \mathbf{c}_\rho, \tag{7.3}$$

where $\mathbf{c}_{u \to i}$ is the embedding for meta-path based context and $\mathcal{M}_{u \to i}$ is the set of the considered meta-paths for the current interaction. In this naive embedding method, each meta-path indeed receives equal attention, and the representation of meta-path based context fully depends on the generated path instances. It fails to take the involved user and item into consideration, which lacks the ability of capturing varying semantics from meta-paths in different interaction scenarios.

7.2.2.3 Improving Embeddings for Interaction via Co-Attention Mechanism

Inspired by the recent progress of attention mechanism made in computer vision and natural language processing [21, 38], we propose a novel co-attention mechanism to improve the embeddings of users, items and meta-paths.

Attention for Meta-path based Context. Since distinct meta-paths may have different semantics in an interaction, we learn the interaction-specific attention weights over meta-paths conditioned on the involved user and item. Given the user embedding $\mathbf{x}_u$, item embedding $\mathbf{y}_i$, the context embedding $\mathbf{c}_\rho$ for a meta-path ρ, we adopt a two-layer architecture to implement the attention

$$\boldsymbol{\alpha}^{(1)}_{u,i,\rho} = f(\mathbf{W}^{(1)}_{u}\mathbf{x}_u + \mathbf{W}^{(1)}_{i}\mathbf{y}_i + \mathbf{W}^{(1)}_{\rho}\mathbf{c}_\rho + \mathbf{b}^{(1)}), \tag{7.4}$$

$$\alpha^{(2)}_{u,i,\rho} = f(\mathbf{w}^{(2)\top}\boldsymbol{\alpha}^{(1)}_{u,i,\rho} + b^{(2)}), \tag{7.5}$$

where $\mathbf{W}^{(1)}_{*}$ and $\mathbf{b}^{(1)}$ denote the weight matrix and the bias vector for the first layer, and the $\mathbf{w}^{(2)}$ and $b^{(2)}$ denote the weight vector and the bias for the second layer. $f(\cdot)$ is set to the ReLU function.

The final meta-path weights are obtained by normalizing the above attentive scores over all the meta-paths using the softmax function,

$$\alpha_{u,i,\rho} = \frac{\exp(\alpha^{(2)}_{u,i,\rho})}{\sum_{\rho' \in \mathcal{M}_{u\to i}} \exp(\alpha^{(2)}_{u,i,\rho'})}, \tag{7.6}$$

which can be interpreted as the contribution of the meta-path ρ to the interaction between u and i. After we obtain the meta-path attention scores $\alpha_{u,i,\rho}$, the new embedding for aggregate meta-path context can be given as the following weighted sum:

$$\mathbf{c}_{u\to i} = \sum_{\rho \in \mathcal{M}_{u\to i}} \alpha_{u,i,\rho} \cdot \mathbf{c}_\rho, \tag{7.7}$$

where $\mathbf{c}_\rho$ the learned embedding for the meta-path ρ in Eq. 7.2. Since the attention weights $\{\alpha_{u,i,\rho}\}$ are generated for each interaction, they are interaction-specific and able to capture varying interaction context.

Attention for Users and Items. Given a user and an item, the meta-path connecting them provide important interaction context, which is likely to affect the original representations of users and items. Giving original user and item latent embeddings $\mathbf{x}_u$ and $\mathbf{y}_i$, and the meta-path based context embedding $\mathbf{c}_{u\to i}$ for the interaction between u and i, we use a single-layer network to compute the attention vectors $\boldsymbol{\beta}_u$ and $\boldsymbol{\beta}_i$ for user u and item i as,

$$\boldsymbol{\beta}_u = f(\mathbf{W}_u\mathbf{x}_u + \mathbf{W}_{u\to i}\mathbf{c}_{u\to i} + \mathbf{b}_u), \tag{7.8}$$

$$\boldsymbol{\beta}_i = f(\mathbf{W}'_i\mathbf{y}_i + \mathbf{W}'_{u\to i}\mathbf{c}_{u\to i} + \mathbf{b}'_i), \tag{7.9}$$

where $\mathbf{W}_*$ and $\mathbf{b}_u$ denote the weight matrix and bias vector for user attention layer, $\mathbf{W}'_*$ and $\mathbf{b}'_i$ denote the weight matrix and bias vector for item attention layer. Similarly, $f(\cdot)$ is set to the ReLU function. Then, the final representations of user and item are computed by using an element-wise product "$\odot$" with the attention vectors:

$$\tilde{\mathbf{x}}_u = \boldsymbol{\beta}_u \odot \mathbf{x}_u, \tag{7.10}$$

$$\tilde{\mathbf{y}}_i = \boldsymbol{\beta}_i \odot \mathbf{y}_i. \tag{7.11}$$

The attention vectors $\boldsymbol{\beta}_u$ and $\boldsymbol{\beta}_i$ are used for improving the original user and item embeddings conditioned on the calibrated meta-path based context $\mathbf{c}_{u\to i}$ (Eq. 7.7).

By combining the two parts of attention components, our model improves the original representations for users, items and meta-path based context in a mutual en-

hancement way. We call such an attention mechanism Co-Attention. To our knowledge, few heterogeneous graph based recommendation methods are able to learn explicit representations for meta-paths, especially in an interaction-specific way.

7.2.2.4 Overall Architecture

Until now, given an interaction between user u and item i, we have the embeddings for user u, item i and the meta-path connecting them. We combine the three embedding vectors into a unified representation of the current interaction as below:

$$\widetilde{\mathbf{x}}_{u,i} = \tilde{\mathbf{x}}_u \oplus \mathbf{c}_{u \to i} \oplus \tilde{\mathbf{y}}_i, \tag{7.12}$$

where "$\oplus$" denotes the vector concatenation operation, $\mathbf{c}_{u \to i}$ (Eq. 7.7) denotes the embedding of the meta-path based context for $\langle u, i \rangle$, $\tilde{\mathbf{x}}_u$ (Eq. 7.10) and $\tilde{\mathbf{y}}_i$ (Eq. 7.11) denote the improved embeddings of user u and item i respectively. $\widetilde{\mathbf{x}}_{u,i}$ encodes the information of an interaction from three aspects: the involved user, the involved item and the corresponding meta-path based context. Following [11], we feed $\widetilde{\mathbf{x}}_{u,i}$ into a MLP component in order to implement a nonlinear function for modeling complicated interactions:

$$\hat{r}_{u,i} = \text{MLP}(\widetilde{\mathbf{x}}_{u,i}). \tag{7.13}$$

MLP component involves two hidden layers with ReLU as the activation function and an output layer with the sigmoid function. With the premise that neural network models can learn more abstractive features of data via using a small number of hidden units for higher layers [10], we empirically implement a tower structure for the MLP component, halving the layer size for each successive higher layer.

Defining a proper objective function for model optimization is a key step for learning a good recommendation model. Traditional point-wise recommendation models for the rating prediction task usually adopt the squared error loss [16]. However, in our task, we only have implicit feedback available. Following [11, 31], we learn the parameters of our model with negative sampling and the objective for an interaction $\langle u, i \rangle$ can be formulated as follows:

$$\ell_{u,i} = -\log \hat{r}_{u,i} - E_{j \sim P_{neg}}[\log(1 - \hat{r}_{u,j})], \tag{7.14}$$

where the first term models the observed interaction, and the second term models the negative feedback drawn from the noise distribution P_{neg}. In MERec, we set the distribution P_{neg} as uniform distribution, which is flexible to extend to other biased distributions, *i.e.*, popularity based distribution.

Table 7.1 Statistics of the three datasets. The first row of each dataset corresponds to the number of users, items and interactions.

Datasets	Relations (A-B)	#A	#B	#A-B
Movielens	User-Movie	943	1,682	100,000
	User-User	943	943	47,150
	Movie-Movie	1,682	1,682	82,798
	Movie-Genre	1,682	18	2861
LastFM	User-Artist	1,892	17,632	92,834
	User-User	1,892	1,892	18,802
	Artist-Artist	17,632	17,632	153,399
	Artist-Tag	17,632	11,945	184,941
Yelp	User-Business	16,239	14,284	198,397
	User-User	16,239	16,239	158,590
	Business-City (Ci)	14,267	47	14,267
	Business-Category (Ca)	14,180	511	40,009

Table 7.2 The selected meta-paths used in each dataset.

Dataset	Meta-paths
Movielens	UMUM, UMGM, UUUM, UMMM
LastFM	UATA, UAUA, UUUA, UUA
Yelp	UBUB, UBCaB, UUB, UBCiB

7.2.3 Experiments

7.2.3.1 Experimental Settings

Datasets. In experiments, we employe three real datasets from different domains, namely Movielens [1] movie dataset, LastFM [2] music dataset and Yelp [3] business dataset. The detailed descriptions of the three datasets are shown in Table 7.1. The selected meta-paths for each dataset are reported in Table 7.2.

Baselines. In this section, we consider two kinds of representative recommendation methods: CF based methods (ItemKNN[24], BPR[22], MF[16], and NeuMF[11]) only utilizing implicit feedback, and HIN based methods utilizing rich heterogeneous information (SVDFeature$_{hete}$[5], SVDFeature$_{mp}$, HeteRS[20] and FMG$_{rank}$[42]). To examine the effectiveness of our priority based sampling strategy and co-attention mechanism, we prepare three variants of MCRec (MCRec$_{rand}$, MCRec$_{avg}$ and MCRec$_{mp}$). MCRec$_{rand}$ employs the random meta-path guided sampling strategy for path generation. MCRec$_{avg}$ employs the naive context embedding strategy for meta-paths. MCRec$_{mp}$ reserves the attention components for meta-paths and removes the attention component for users and items.

[1] https://grouplens.org/datasets/movielens/

[2] https://www.last.fm

[3] http://www.yelp.com/dataset-challenge

Parameter Settings. For our method MCRec, we set the batch size to 256, the learning rate to 0.001, the regularization parameter to 0.0001, the CNN filter size to 3, the dimension of user and item embeddings to 128, the dimension of predictive factors to 32, and the number of sampled path instances is 5. For MF and NeuMF, we follow the optimal configuration in [11]. Moreover, we use 10% training data as the validation set to optimize the parameters for the other methods.

Evaluation Metrics. The top-N recommendation task usually adopts similar evaluation metrics. Following [39, 11], we use Precision at rank K (Prec@K), Recall at rank K (Recall@K) and Normalized Discounted Cumulative Gain at rank K (NDCG@K) as the evaluation metrics. The final results are first averaged over all the test items of a user and then averaged over all the users. For stability, we perform ten runs using different random-splitting training/test sets and report the average results.

7.2.3.2 Comparisons and Analysis

Table 7.3 Results of effectiveness experiments on three datasets. We use "*" to mark the best performance from the baselines for each comparison. We use "#" to indicate the improvement of MCRec over the best performance from the baselines is significant based on paired t-test at the significance level of 0.01.Here we simplify Prec@10 (%) to P@10, Recall@10 (%) to R@10 and NDCGG@10 (%) to N@10.

Model	Movielens			LastFM			Yelp		
	P@10	R@10	N@10	P@10	R@10	N@10	P@10	R@10	N@10
ItemKNN	25.8	15.4	56.9	41.6	45.1	79.8	13.9	54.2	53.8
BRP	30.1	19.5	64.6	41.3	44.9	81.0	14.7	55.0	55.5
MF	32.5	20.5	65.1	43.6	46.3	79.2	15.0	53.5	53.2
NeuMF	32.9*	20.9	65.9	45.4	46.8	81.0	15.0	58.6	57.1
SVDFeature$_{hete}$	31.7	20.2	64.5	45.8	48.4	82.9*	14.0	56.1	52.9
SVDFeature$_{mp}$	31.1	19.3	65.4	43.9	46.5	81.2	15.2	59.3	59.7*
HeteRS	24.9	16.7	59.7	42.8	44.9	80.3	14.2	56.1	56.0
FMG$_{rank}$	32.6	21.7*	66.8*	46.3*	49.2*	82.6	15.4*	59.5*	58.6
MCRec$_{rand}$	32.2	21.0	66.5	45.4	48.0	80.0	15.1	58.4	57.2
MCRec$_{avg}$	32.7	21.1	66.3	46.5	49.1	83.1	16.0	59.3	60.2
MCRec$_{mp}$	34.0	22.0	68.3	46.6	49.2	84.3	16.6	63.0	62.3
MCRec	**34.5**$^{\#}$	**22.6**$^{\#}$	**69.0**$^{\#}$	**48.1**$^{\#}$	**50.7**$^{\#}$	**85.3**$^{\#}$	**16.9**$^{\#}$	**63.3**$^{\#}$	**63.0**$^{\#}$

To evaluate the performance, we randomly split the entire user implicit feedback records of each dataset into training and test set, *i.e.*, we use 80% feedback records to predict the remaining 20% feedback records [4]. We randomly sample 50 negative samples that have no interaction records with the target user. Then, we rank the list consisting of the positive item and 50 negative items.

[4] We hold out 10% training data as the validation set for parameter tuning.

The comparison results of our proposed model and baselines on three datasets are reported in Table 7.3. There are some observations and analysis. (1) Our complete model MCRec is consistently better than all the baselines on the three datasets. The results indicate the effectiveness of MCRec on the task of top-*N* recommendation, which has adopted a more principled way to leverage heterogeneous context information for improving recommendation performance. (2) Considering the three variants of MCRec, we can find that the overall performance order is as follows: MCRec > MCRec$_{mp}$ > MCRec$_{avg}$ > MCRec$_{rand}$. The results show that the co-attention mechanism is able to better utilize the meta-path based context for recommendation. First, the importance of each meta-path should depend on a specific interaction instead of being treated equal (*i.e.*, MCRec$_{avg}$). Second, meta-paths provide important context for the interaction between users and items, which has a potential influence on the learned representations of users and items. Ignoring such influence may not be able to achieve the optimal performance for utilizing meta-path based context information (*i.e.*, MCRec$_{mp}$). In addition, although MCRec$_{rand}$ achieves competitive performance compared to baselines, it is worse than the complete MCRec. Our complete model adopts the priority based sampling strategy to generate path instances, while MCRec$_{rand}$ adopts a random sampling strategy.

The more detailed method description and experiment validation can be seen in [13].

7.3 Cold-start Recommendation

7.3.1 Overview

In recommender systems, the interaction data of new users or new items are often of high sparsity, leading to the so-called cold-start issue [44] in which it becomes challenging to learn effective user or item representations. To alleviate this problem, at the data-level, heterogeneous information network (HIN) [27] have been leveraged to enrich user-item interactions with complementary heterogeneous information. As shown in Fig. 7.3a, a toy HIN can be constructed for movie recommendation, which captures how the movies are related with each other via actors and directors, in addition to the existing user-movie interactions. On the HIN, higher-order graph structures like meta-paths [30], a relation sequence connecting two objects, can effectively capture semantic contexts. For instance, the meta-path User–Movie–Actor–Movie or UMAM encodes the semantic context of "movies starring the same actor as a movie rated by the user". Together with the content-based methods, HIN-based methods [41, 13] also assume a data-level strategy to alleviate the cold-start problem, as illustrated in Fig. 7.3b.

On another line, at the model level, the recent episodic meta-learning paradigm [9] has offered insights into modeling new users or items with scarce interaction data [34]. Meta-learning focuses on deriving general knowledge (i.e., a prior) across

different learning tasks, so as to rapidly adapt to a new learning task with the prior and a small amount of training data. To some extent, cold-start recommendation can be formulated as a meta-learning problem, where each task is to learn the preferences of one user. From the tasks of existing users, the meta-learner learns a prior with strong generalization capacity during meta-training, such that it can be easily and quickly adapted to the new tasks of cold-start users with scarce interaction data during meta-testing. As illustrated in Fig. 7.3c, the cold-start user u_3 (with only one movie rating) can be adapted from the prior θ in meta-testing, where the prior is derived by learning how to adapt to existing users u_1 and u_2 in meta-training.

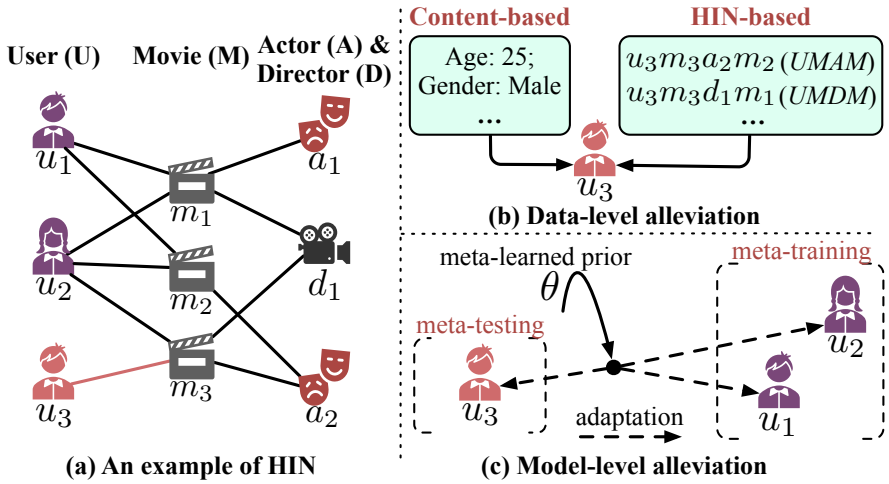

Fig. 7.3 An example of HIN and existing data or model-level alleviation for cold-start recommendation.

In this section, we propose to address the cold-start recommendation at both data and model levels, in which learning the preference of each user is regarded as a task in meta-learning, and a HIN is exploited to augment data. One is to augment the task for each user with multifaceted semantic contexts. That is, in a task of a specific user, besides considering the items directly interacted with the user, we also introduce items that are semantically related to the user via higher-order graph structures, i.e., meta-paths. These related items form the semantic contexts of each task, which can be further differentiated into multiple facets as implied by different meta-paths. The other is to propose a co-adaptation meta-learner, which is equipped with both semantic-wise adaptation and task-wise adaptation. Specifically, the semantic-wise adaptation learns a unique semantic prior for each facet. While the semantic priors are derived from different semantic spaces, they are regulated by a global prior to capture the general knowledge of encoding contexts on a HIN. Furthermore, the task-wise adaptation is designed for each task (i.e., user), which updates the preference of each user from the various semantic priors, such that tasks sharing the same facet of semantic contexts can hinge on a common semantic prior.

7.3.2 The MetaHIN Model

Before we introduce the framework of MetaHIN, we first define the cold-start problem on HINs as follows [27].

Definition 1. Cold-start Recommendation. Given a HIN $G = \{V, E, O, R\}$, let $V_U, V_I \subset V$ denote the set of user and item objects, respectively. Given a set of ratings between users and items, i.e., $\mathcal{R} = \{r_{u,i} \geq 0 : u \in V_U, i \in V_I, \langle u, i \rangle \in E\}$, we aim to predict the unknown rating $r_{u,i} \notin \mathcal{R}$ between user u and item i. In particular, if u is a new user with only a handful of existing ratings, i.e., $|\{r_{u',i} \in \mathcal{R} : u' = u\}|$ is small, it is known as user cold-start (UC); correspondingly, if i is a new item, it is known as item cold-start (IC); if both u and i are new, it is known as user-item cold-start (UIC).

7.3.2.1 Model Framework

As illustrated in Fig. 7.4, the proposed MetaHIN consists of two components: semantic-enhanced task constructor in Fig. 7.4a and co-adaptation meta-learner in Fig. 7.4b. First, we design a semantic-enhanced task constructor to augment the support and query sets of user tasks with heterogeneous semantic contexts, which comprise of items related to the user through meta-paths on a HIN. The semantic contexts are multifaceted in nature, such that each meta-path represents a different facet of heterogeneous semantics. Second, compared to task-wise adaptation, we perform semantic-wise adaptation, in order to adapt the global prior θ to finer-grained semantic priors for different facets (i.e., meta-paths) in a task. The global prior θ captures the general knowledge of encoding contexts for recommendation, and can be materialized in the form of a base model f_θ. Thus, our co-adaptation meta-learner performs both semantic- and task-wise adaptions on the support set, and further optimizes the global prior on the query set.

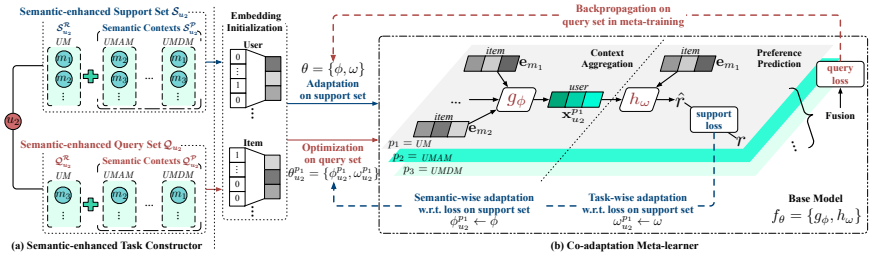

Fig. 7.4 Illustration of the meta-training procedure of a task in MetaHIN. (a) Semantic-enhanced task constructor, where the support and query sets are augmented with meta-path based heterogeneous semantic contexts. (b) Co-adaptation meta-learner, with semantic- and task-wise adaptations on the support set, while the global prior θ is optimized on the query set. During meta-testing, each task follows the same procedure except updating the global prior.

7.3.2.2 Semantic-enhanced Task Constructor

Given a user u with task $\mathcal{T}_u = (\mathcal{S}_u, \mathcal{Q}_u)$, the semantic-enhanced support set is defined as

$$\mathcal{S}_u = (\mathcal{S}_u^{\mathcal{R}}, \mathcal{S}_u^{\mathcal{P}}), \tag{7.15}$$

where $\mathcal{S}_u^{\mathcal{R}}$ is a set of items that has been rated by user u, and $\mathcal{S}_u^{\mathcal{P}}$ represents the semantic contexts based on a set of meta-paths $\mathcal{P}$.

For new users in cold-start scenarios, the set of rated items $\mathcal{S}_u^{\mathcal{R}}$ is usually small, i.e., a new user only has a few ratings. For meta-training tasks, we follow previous work [17] to construct $\mathcal{S}_u^{\mathcal{R}}$ by sampling a small subset of items rated by u, i.e., $\{i \in V_I : r_{u,i} \in \mathcal{R}\}$, in order to simulate new users.

On the other hand, the semantic contexts $\mathcal{S}_u^{\mathcal{P}}$ are employed to encode multifaceted semantics into the task. Specifically, assume a set of meta-paths $\mathcal{P}$, such that each path $p \in \mathcal{P}$ starts with User–Item and ends with Item with a length up to l. For example, in Fig. 7.3a, $\mathcal{P} = \{UM, UMAM, UMDM, UMUM\}$ if we set $l = 3$. For each user-item interaction $\langle u, i \rangle$, we define the semantic context of $\langle u, i \rangle$ induced by meta-path p as follows:

$$C_{u,i}^{p} = \{j : j \in \text{items reachable along } p \text{ starting from } u\text{–}i\}. \tag{7.16}$$

For instance, the semantic context of $\langle u_2, m_2 \rangle$ induced by UMAM is $\{m_2, m_3, \ldots\}$. Since in each task u may interact with multiple items, we build the p-induced semantic context for the task $\mathcal{T}_u$ as

$$\mathcal{S}_u^{p} = \bigcup_{i \in \mathcal{S}_u^{\mathcal{R}}} C_{u,i}^{p}. \tag{7.17}$$

Finally, accounting for all meta-paths in $\mathcal{P} = \{p_1, p_2, ..., p_n\}$, the semantic contexts $\mathcal{S}_u^{\mathcal{P}}$ of task $\mathcal{T}_u$ are formulated as

$$\mathcal{S}_u^{\mathcal{P}} = (\mathcal{S}_u^{p_1}, \mathcal{S}_u^{p_2}, \ldots, \mathcal{S}_u^{p_n}). \tag{7.18}$$

In essence, $\mathcal{S}_u^{\mathcal{P}}$ is the set of items that are reachable from user u via all items he/she has rated along the meta-paths, which incorporates multifaceted semantic contexts such that each meta-path represents one facet. As shown in Fig. 7.4a, following the meta-path UMAM, the reachable items of user u_2 are $\{m_2, m_3, \ldots\}$, which are the movies starring the same actor of movies that u_2 has rated in the past. That is, the semantic context induced by UMAM incorporates movies starring the same actor as a facet of user preferences, which makes sense since the user might be a fan of an actor and prefers most movies played by the actor.

Likewise, we can construct the semantic-enhanced query set $\mathcal{Q}_u = (\mathcal{Q}_u^{\mathcal{R}}, \mathcal{Q}_u^{\mathcal{P}})$. In particular, $\mathcal{Q}_u^{\mathcal{R}}$ contains items rated by u for calculating the task loss in meta-training, or items with hidden rating for making predictions in meta-testing; $\mathcal{Q}_u^{\mathcal{P}}$ captures the semantic contexts induced by meta-paths $\mathcal{P}$. Note that in a task $\mathcal{T}_u$, the items with ratings in the support and query sets are mutually exclusive, i.e., $\mathcal{S}_u^{\mathcal{R}} \cap \mathcal{Q}_u^{\mathcal{R}} = \emptyset$.

7.3.2.3 Co-adaptation Meta-learner

Given the semantic-enhanced tasks, the co-adaptation meta-learner with both semantic- and task-wise adaptations in order to learn fine-grained prior knowledge. The global prior can be abstracted as a base model to encode the general knowledge of how to learn with contexts on HINs, which can be further adapted to different semantic facets within a task.

Base Model. As shown in Fig. 7.4b, the base model f_θ involves context aggregation g_ϕ to derive user embeddings, and preference prediction h_ω to estimate the rating score, i.e., $f_\theta = (h_\omega, g_\phi)$.

In context aggregation, the user embeddings are aggregated from his/her contexts, which are his/her related items via direct interactions or meta-paths (i.e., semantic contexts), since user preferences are reflected in items. Following [17], we initialize the user and item embeddings based their features (or an embedding look up if there are no features), say $\mathbf{e}_u \in \mathbb{R}^{d_U}$ for user u and $\mathbf{e}_i \in \mathbb{R}^{d_I}$ for item i where d_U, d_I are the embedding dimensions. Subsequently, we obtain user u's embedding $\mathbf{x}_u$ as follows:

$$\mathbf{x}_u = g_\phi(u, \mathcal{C}_u) = \sigma\left(\text{MEAN}(\{\mathbf{W}\mathbf{e}_j + \mathbf{b} : j \in \mathcal{C}_u\})\right), \tag{7.19}$$

where $\mathcal{C}_u$ denotes the set of items related to user u via direct interactions (i.e., the rated items) or meta-paths (i.e., their induced semantic contexts), $\text{MEAN}(\cdot)$ is mean pooling, and σ is the activation function (we use LeaklyReLU). Here g_ϕ is the context aggregation function parameterized by $\phi = \{\mathbf{W} \in \mathbb{R}^{d \times d_I}, \mathbf{b} \in \mathbb{R}^d\}$, which are trainable to distill semantic information for user preferences. $\mathbf{x}_u$ can be further concatenated with u's initial embedding $\mathbf{e}_u$, when user features are available.

In preference prediction, given user u's embedding $\mathbf{x}_u$ and item i's embedding $\mathbf{e}_i$, we estimate the rating of user u on the item i as:

$$\hat{r}_{ui} = h_\omega(\mathbf{x}_u, \mathbf{e}_i) = \text{MLP}(\mathbf{x}_u \oplus \mathbf{e}_i), \tag{7.20}$$

where MLP is a two-layer multilayer perceptron, and $\oplus$ denotes concatenation. Here h_ω is the rating prediction function parameterized by ω, which contains the weights and biases in MLP. Finally, we minimize the following loss for user u to learn his/her preferences:

$$\mathcal{L}_u = \frac{1}{|\mathcal{R}_u|} \sum_{i \in \mathcal{R}_u} (r_{ui} - \hat{r}_{ui})^2, \tag{7.21}$$

where $\mathcal{R}_u = \{i : r_{ui} \in \mathcal{R}\}$ denotes the set of items rated by u, and r_{ui} is the actual rating of u on item i.

Note that the base model $f_\theta = (g_\phi, h_\omega)$ is a supervised model for recommendation, which typically requires a large number of example ratings to achieve reasonable performance, which is not upheld in the cold-start scenario. As motivated, we recast the cold-start recommendation as a meta-learning problem. Specifically, we abstract the base model $f_\theta = \{g_\phi, h_\omega\}$ as encoding the prior knowledge $\theta = \{\phi, \omega\}$ of how to learn user preferences from contexts on HINs. Next, we detail the proposed co-adaptation meta-learner to learn the prior knowledge.

Co-adaptation. The goal of the co-adaptation meta-learner is to learn the prior knowledge $\theta = (\phi, \omega)$, which can quickly adapt to a new user task with just a few example ratings. As discussed in Fig. 7.4a, each task is augmented with multifaceted semantic contexts. Thus, the prior should not only encode the global knowledge shared across tasks, but also become capable of generalizing to different semantic facets within each task. To this end, we enhance the meta-learner with semantic- and task-wise adaptations.

For semantic-wise adaptation, the semantic-enhanced support set $\mathcal{S}_u$ of the task $\mathcal{T}_u$ is associated with semantic contexts induced by different meta-paths (e.g., UMAM and UMDM in Fig. 7.4), where each meta-path represents one semantic facet. The semantic-wise adaptation evaluates the loss based on the semantic context induced by a meta-path p (i.e., $\mathcal{S}_u^p$). With one (or a few) gradient descent step w.r.t. the p-specific loss, the global context prior ϕ, which encodes how to learn with contexts on a HIN, is adapted to the semantic space induced by the meta-path p.

Formally, given a task $\mathcal{T}_u$ of user u, the support set $\mathcal{S}_u = (\mathcal{S}_u^{\mathcal{R}}, \mathcal{S}_u^{\mathcal{P}})$ is augmented with semantic contexts $\mathcal{S}_u^{\mathcal{P}}$, comprising various facets $\mathcal{S}_u^{p_i}$ induced by different meta-paths p_i as in Eq. 7.18. Given a meta-path $p \in \mathcal{P}$, user u's embedding in the semantic space of p is

$$\mathbf{x}_u^p = g_\phi(u, \mathcal{S}_u^p). \tag{7.22}$$

In this semantic space of p, we can further calculate the loss on the support set of rated items $\mathcal{S}_u^{\mathcal{R}}$ in task $\mathcal{T}_u$ as

$$\mathcal{L}_{\mathcal{T}_u}(\omega, \mathbf{x}_u^p, \mathcal{S}_u^{\mathcal{R}}) = \frac{1}{|\mathcal{S}_u^{\mathcal{R}}|} \sum_{i \in \mathcal{S}_u^{\mathcal{R}}} (r_{ui} - h_\omega(\mathbf{x}_u^p, \mathbf{e}_i))^2, \tag{7.23}$$

where $h_\omega(\mathbf{x}_u^p, \mathbf{e}_i)$ represents the predicted rating of user u on item i in the meta-path p-induced semantic space.

Next, we adapt the global context prior ϕ w.r.t. the loss in each semantic space of p in task $\mathcal{T}_u$ with one gradient descent step, to obtain the semantic prior ϕ_u^p. Thus, the meta-learner learns more fine-grained prior knowledge for various semantic facets:

$$\phi_u^p = \phi - \alpha \frac{\partial \mathcal{L}_{\mathcal{T}_u}(\omega, \mathbf{x}_u^p, \mathcal{S}_u^{\mathcal{R}})}{\partial \phi} = \phi - \alpha \frac{\partial \mathcal{L}_{\mathcal{T}_u}(\omega, \mathbf{x}_u^p, \mathcal{S}_u^{\mathcal{R}})}{\partial \mathbf{x}_u^p} \frac{\partial \mathbf{x}_u^p}{\partial \phi}, \tag{7.24}$$

where α is the semantic-wise learning rate, and $\mathbf{x}_u^p = g_\phi(u, \mathcal{S}_u^p)$ is a function of ϕ.

For task-wise adaptation, in the semantic space of meta-path p with adapted semantic prior ϕ_u^p, the task-wise adaptation further adapts the global prior ω, which encodes how to learn rating predictions of u, to the task $\mathcal{T}_u$ with one (or a few) gradient descent step.

The semantic prior ϕ_u^p subsequently updates user u' embeddings in the semantic space of p on the support set to $\mathbf{x}_u^{p\langle S\rangle} = g_{\phi_u^p}(u, \mathcal{S}_u^p)$, which further transforms the global prior ω to the same space:

$$\omega^p = \omega \odot \kappa(\mathbf{x}_u^{p\langle S\rangle}), \tag{7.25}$$

where $\odot$ is the element-wise product and $\kappa(\cdot)$ serves as a transformation function realized with a fully connected layer. Intuitively, ω is gated into the current p-induced semantic space. We then adapt ω^p to the task $\mathcal{T}_u$ with one gradient descent step:

$$\omega_u^p = \omega^p - \beta \frac{\partial \mathcal{L}_{\mathcal{T}_u}(\omega^p, \mathbf{x}_u^{p\langle S\rangle}, \mathcal{S}_u^{\mathcal{R}})}{\partial \omega^p}, \tag{7.26}$$

where β is the task-wise learning rate.

With the semantic- and task-wise adaptations, we have adapted the global prior θ to the semantic- and task-specific parameters $\theta_u^p = \{\phi_u^p, \omega_u^p\}$ in the p-induced semantic space of task $\mathcal{T}_u$. Given a set of meta-paths $\mathcal{P}$, the meta-learner is trained by optimizing the performance of the adapted parameters θ_u^p on the query set $\mathcal{Q}_u$ in all semantic spaces of $\mathcal{P}$ across all meta-training tasks. That is, as shown in Fig. 7.4b, the global prior $\theta = (\phi, \omega)$ will be optimized through backpropgation of the query loss:

$$\min_{\theta} \sum_{\mathcal{T}_u \in \mathcal{T}^{\text{tr}}} \mathcal{L}_{\mathcal{T}_u}(\omega_u, \mathbf{x}_u, \mathcal{Q}_u^{\mathcal{R}}), \tag{7.27}$$

where ω_u and $\mathbf{x}_u$ are fused from multiple semantic spaces (i.e., meta-paths in $\mathcal{P}$). Specifically,

$$\omega_u = \textstyle\sum_{p\in\mathcal{P}} a_p \omega_u^p, \quad \mathbf{x}_u = \textstyle\sum_{p\in\mathcal{P}} a_p \mathbf{x}_u^{p\langle Q\rangle}, \tag{7.28}$$

where $a_p = \text{softmax}(-\mathcal{L}_{\mathcal{T}_u}(\omega_u^p, \mathbf{x}_u^{p\langle Q\rangle}, \mathcal{Q}_u^{\mathcal{R}}))$ is the weight of the p-induced semantic space, and $\mathbf{x}_u^{p\langle Q\rangle} = g_{\phi_u^p}(u, \mathcal{Q}_u^p)$ is u's embedding aggregated on the query set. Since the loss value reflects the model performance [3], it is intuitive that the larger the loss value in a semantic space, the smaller the corresponding weight should be.

In summary, the co-adaption meta-learner aims to optimize the global prior θ across several tasks, in such a way that the query loss of each meta-training task $\mathcal{T}_u$ using the adapted parameters $\{\theta_u^p : p \in \mathcal{P}\}$ can be minimized (i.e., "learning to learn"); it does not directly update the global prior using task data. It particular, with the co-adaption mechanism, we adapt the parameters not only to each task, but also to each semantic facet within a task.

7.3.3 Experiments

7.3.3.1 Experimental Settings

Datasets. We conduct experiments on three benchmark datasets, namely, DBook[5], MovieLens[6], and Yelp[7], from publicly accessible repositories.

[5] https://book.douban.com

[6] https://grouplens.org/datasets/movielens/

[7] https://www.yelp.com/dataset/challenge

Baselines. We compare our proposed MetaHIN with three categories of methods. (1) Traditional methods, including FM [23], NeuMF [11] and GC-MC [1]. As they cannot handle HINs, we take the heterogeneous information (e.g., actor) as the features of users or items. (2) HIN-based methods, including mp2vec [7] and HERec [26]. Both methods are based on meta-paths, and we utilize the same set of meta-paths as in our method. (3) Cold-start methods, including content-based DropoutNet [35], as well as meta-learning-based MeteEmb [19] and MeLU [17]. Since they do not handle HINs either, we input the heterogeneous information as user or item features following the original papers. We follow [17] to train the non-meta-learning baselines with the union of rated items in all support and query sets from meta-training tasks. To handle new users or items, we fine-tune the trained models with support sets and evaluate on query sets in meta-testing tasks.

Evaluation Metrics. We adopt three widely-used evaluation protocols [26, 36, 17], namely, mean absolute error (MAE), root mean square error (RMSE), and normalized discounted cumulative gain at rank K (nDCG@K). Here we use $K = 5$.

7.3.3.2 Comparisons and Analysis

In this experiment, we empirically compare MetaHIN to several state-of-the-art baselines, in three cold-start scenarios and the traditional non-cold start scenario. Table 7.4 demonstrates the performance comparison between all methods w.r.t. four recommendation scenarios.

Cold-start Scenarios. The first three parts of Table 7.4 present three cold-start scenarios (UC, IC and UIC). Overall, our MetaHIN consistently yields the best performance among all methods on three datasets. For instance, MetaHIN improves over the best baseline w.r.t. MAE by 3.05-5.26%, 2.89-5.55%, and 2.22-5.19% on three datasets, respectively. Among different baselines, traditional methods (e.g., MF, NeuMF and GC-MC) are least competitive despite incorporating heterogeneous information as content features. Such treatment of heterogeneous information is not ideal as higher-order graph structures are lost. HIN-based methods perform better due to the incorporation of such structures (i.e., meta-paths). Nevertheless, supervised learning methods generally cannot perform effectively given limited training data for new users and items.

On the other hand, meta-learning methods typically cope better in such cases. In particular, the best baseline is consistently MeLU or MeteEmb. However, they still underperform our MetaHIN in all scenarios. The reason might be that both of them only integrate heterogeneous information as content features, without capturing multifaceted semantics derived from higher-order structures like meta-paths. In contrast, in MetaHIN, we perform semantic- and task-wise co-adaptions, to effectively adapt to not only tasks, but also different semantic facets within a task.

Non-cold-start Scenario. In the last part of Table 7.4, we investigate the traditional recommendation scenario. Our MetaHIN is still robust, outperforming all the baselines. While this is a traditional scenario, the datasets are still very sparse in general. Thus, incorporating the semantic-rich HINs can often alleviate the sparsity

challenge at the data level. MetaHIN further addresses the problem at the model level with the co-adaptation meta-learner, and thus can better deal with sparse data. Of course, compared to cold-start scenarios, MetaHIN's performance lift over the baselines tend to be smaller as the sparsity issue is not as severe.

Table 7.4 Experimental results in four recommendation scenarios and on three datasets. A smaller MAE or RMSE value, and a larger nDCG@5 value indicate a better performance. The best method is bolded, and second best is underlined.

Scenario	Model	DBook			MovieLens			Yelp		
		MAE ↓	RMSE ↓	nDCG@5 ↑	MAE ↓	RMSE ↓	nDCG@5 ↑	MAE ↓	RMSE↓	nDCG@5 ↑
Existing items for new users (User Cold-start or UC)	FM	0.7027	0.9158	0.8032	1.0421	1.3236	0.7303	0.9581	1.2177	0.8075
	NeuMF	0.6541	0.8058	0.8225	0.8569	1.0508	0.7708	0.9413	1.1546	0.7689
	GC-MC	0.9061	0.9767	0.7821	1.1513	1.3742	0.7213	0.9321	1.1104	0.8034
	mp2vec	0.6669	0.8391	0.8144	0.8793	1.0968	0.8233	0.8972	1.1613	0.8235
	HERec	0.6518	0.8192	0.8233	0.8691	0.9916	0.8389	0.8894	1.0998	0.8265
	DropoutNet	0.8311	0.9016	0.8114	0.9291	1.1721	0.7705	0.8557	1.0369	0.7959
	MeteEmb	0.6782	0.8553	0.8527	0.8261	1.0308	0.7795	0.8988	1.0496	0.7875
	MeLU	0.6353	0.7733	0.8793	0.8104	0.9756	0.8415	0.8341	1.0017	0.8275
	MetaHIN	**0.6019**	**0.7261**	**0.8893**	**0.7869**	**0.9593**	**0.8492**	**0.7915**	**0.9445**	**0.8385**
New items for existing users (Item Cold-start or IC)	FM	0.7186	0.9211	0.8342	1.3488	1.8503	0.7218	0.8293	1.1032	0.8122
	NeuMF	0.7063	0.8188	0.7396	0.9822	1.2042	0.6063	0.9273	1.1009	0.7722
	GC-MC	0.9081	0.9702	0.7634	1.0433	1.2753	0.7062	0.8998	1.1043	0.8023
	mp2vec	0.7371	0.9294	0.8231	1.0615	1.3004	0.6367	0.7979	1.0304	0.8337
	HERec	0.7481	0.9412	0.7827	0.9959	1.1782	0.7312	0.8107	1.0476	0.8291
	DropoutNet	0.7122	0.8021	0.8229	0.9604	1.1755	0.7547	0.8116	1.0301	0.7943
	MeteEmb	0.6741	0.7993	0.8537	0.9084	1.0874	0.8133	0.8055	0.9407	0.8092
	MeLU	0.6518	0.7738	0.8882	0.9196	1.0941	0.8041	0.7567	0.9169	0.8451
	MetaHIN	**0.6252**	**0.7469**	**0.8902**	**0.8675**	**1.0462**	**0.8341**	**0.7174**	**0.8696**	**0.8551**
New items for new users (User-Item Cold-start or UIC)	FM	0.8326	0.9587	0.8201	1.3001	1.7351	0.7015	0.8363	1.1176	0.8278
	NeuMF	0.6949	0.8217	0.8566	0.9686	1.2832	0.8063	0.9860	1.1402	0.7836
	GC-MC	0.7813	0.8908	0.8003	1.0295	1.2635	0.7302	0.8894	1.1109	0.7923
	mp2vec	0.7987	1.0135	0.8527	1.0548	1.2895	0.6687	0.8381	1.0993	0.8137
	HERec	0.7859	0.9813	0.8545	0.9974	1.1012	0.7389	0.8274	0.9887	0.8034
	DropoutNet	0.8316	0.8489	0.8012	0.9635	1.1791	0.7617	0.8225	0.9736	0.8059
	MeteEmb	0.7733	0.9901	0.8541	0.9122	1.1088	0.8087	0.8285	0.9476	0.8188
	MeLU	0.6517	0.7752	0.8891	0.9091	1.0792	0.8106	0.7358	0.8921	0.8452
	MetaHIN	**0.6318**	**0.7589**	**0.8934**	**0.8586**	**1.0286**	**0.8374**	**0.7195**	**0.8695**	**0.8521**
Existing items for existing users (Non-cold-start)	FM	0.7358	0.9763	0.8086	1.0043	1.1628	0.6493	0.8642	1.0655	0.7986
	NeuMF	0.6904	0.8373	0.7924	0.9249	1.1388	0.7335	0.7611	0.9731	0.8069
	GC-MC	0.8056	0.9249	0.8032	0.9863	1.2238	0.7147	0.8518	1.0327	0.8023
	mp2vec	0.6897	0.8471	0.8342	0.8788	1.1006	0.7091	0.7924	1.0191	0.8005
	HERec	0.6794	0.8409	0.8411	0.8652	1.0007	0.7182	0.7911	0.9897	0.8101
	DropoutNet	0.7108	0.7991	0.8268	0.9595	1.1731	0.7231	0.8219	1.0333	0.7394
	MeteEmb	0.7095	0.8218	0.7967	0.8086	1.0149	0.8077	0.7677	0.9789	0.7740
	MeLU	0.6519	0.7834	0.8697	0.8084	0.9978	0.8433	0.7382	0.9028	0.8356
	MetaHIN	**0.6393**	**0.7704**	**0.8859**	**0.7997**	**0.9491**	**0.8499**	**0.6952**	**0.8445**	**0.8477**

The more detailed method description and experiment validation can be seen in [18].

7.4 Author Set Recommendation

7.4.1 Overview

Heterogeneous bibliographic network [29] has also received more and more attention in recent years. As an important related task, the problem of author identification

has been extensively studied, which aims to rank potential authors for an anonymous paper based on public information. The existing studies mainly employ the network structure or semantic content of the paper to predict the correlation between the paper and author, while they usually ignore relationships among authors. Generally, in many scenarios such as finding a potential author group for a given paper, the relationships among authors are very significant. Therefore, in this section, we propose to study a new problem called author set identification. We illustrate the problem setting in Fig.7.5, in which the heterogeneous bibliographic network and network schema are given as the input. The goal is to learn a model that can identify the optimal author set for a new anonymous paper. The problem of author set identification is to acquire an author set with a strong relationship, while the traditional problem only get an author ranking for the target node of anonymous paper.

A basic idea for the problem is to find a set of closely connected authors that are related to an anonymous paper. Therefore, we need to characterize the relationship between anonymous paper and authors, as well as that between authors. However, it is non-trivial to take both relationships into account simultaneously. Moreover, the number of subsets of authors is enormous especially when the number of authors is large. Hence, it is also very difficult to select the optimal one from all subsets. There are two challenges in this problem. (1) How can we model interactions between anonymous paper and authors, meanwhile preserving rich inherent structural information among authors in heterogeneous bibliographic network. (2) How can we find an optimal set of closely connected authors that are related to the anonymous paper.

In this section, we propose a novel **A**uthor **S**et **I**dentification approach called **ASI**. In order to tackle the first challenge, we propose to only emphasize on two types of nodes including anonymous paper and candidate authors. Therefore, ASI first constructs a paper-author interactive network denoted by weighted paper-ego-network, which only contains the mentioned two types of nodes and corresponding relations (paper-author and author-author). Then in order to preserve rich inherent structural information in heterogeneous bibliographic network, the task-guided embedding method called TaskGE is presented to learn the low-dimensional representations of nodes, which can be further used to determine the weights of edges in the constructed network. For the sake of solving the second challenge, we introduce the concept of quasi-clique in dense subgraph and convert the optimal author set identification to the quasi-clique discovery in the weighted paper-ego-network. Specifically, we design the local-search heuristic method under the guidance of a novel density function to find the optimal quasi-clique (author set). Meanwhile, we regard the anonymous paper as a constraint and claim the discovered set of closely connected authors must be related to the anonymous paper.

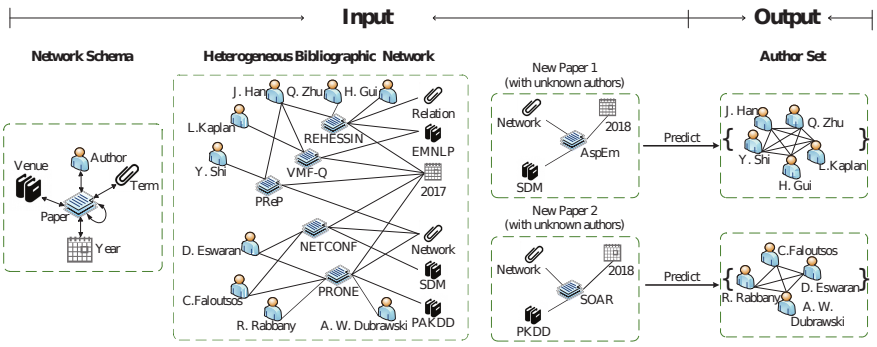

Fig. 7.5 The problem of author set identification in heterogeneous bibliographic networks.

7.4.2 The ASI Model

In this section, we study the novel problem of author set identification in bibliographic network, which can be defined as follows.

Definition 2. Author Set Identification Problem. Given a bibliographic network $G = (V, E)$, which includes a set of papers and papers' relevant information (i.e., authors, venues, terms and year), the goal is to design a method to acquire an author set S'_A from C_A for a new anonymous paper p, such that S'_A is the optimal set to collaborate on the paper p among all subsets of C_A, where $C_A = \{a_1, a_2, \cdots, a_m\}$ denotes the set of all candidate authors.

In order to find the optimal author set, we present the proposed method that leverages quasi-clique for Author Set Identification, called **ASI**. In order to find the optimal author set, we introduce the concept of quasi-clique, which can be defined as follows.

Definition 3. Quasi-Clique [33]. A set of nodes S is an α-quasi-clique if $e[S] \geq \alpha\binom{|S|}{2}$, i.e., if the edge density of the subgraph induced by S exceeds a threshold parameter $\alpha \in (0, 1)$. The edge density is defined as $e[S]/\binom{|S|}{2}$, where $e[S]$ is the size of edges in the subgraph induced by S.

7.4.2.1 Model Framework

The overall architecture of ASI is shown in Fig.7.6. Given a heterogeneous bibliographic network and an anonymous paper (Fig.7.6a), we first construct a weighted paper-ego-network for each anonymous paper (Fig.7.6b), and then find the optimal quasi-clique with constraint (OQCC) in the weighted paper-ego-network (Fig.7.6c). In the following, we will clarify the basic idea and specific details about these two phases. We aim to find a set of closely connected authors that are related to

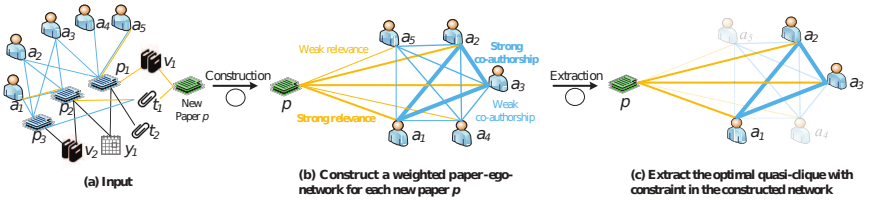

Fig. 7.6 The overall architecture of proposed method ASI (weighted paper-ego-network construction and optimal quasi-clique with constraint extraction).

the anonymous paper. However, it is challenging to incorporate interactions between anonymous paper and authors, meanwhile preserve rich inherent structural information among authors in heterogeneous bibliographic network. We consider to construct a weighted paper-ego-network only containing the anonymous paper and authors. Because there are no direct links between anonymous paper and authors, as well as between authors in bibliographic network, we need to devise an approach to determine the weight of these two kinds of edges. Hence, we propose the task-guided embedding method to learn vector representations of nodes, which can be further used to determine the weights of edges through proper distance function for constructing the weighted paper-ego-network. Then we transform the author set identification into the problem of quasi-clique extraction with constraint. The constraint condition means that the discovered optimal quasi-clique should contain the node of anonymous paper, which also implies the close relationship between author set and anonymous paper. Finally, we propose an approach of local-search heuristic under the guidance of designed novel density function, so as to discover the optimal quasi-clique in the constructed network.

7.4.2.2 Weighted Paper-Ego-Network Construction

Because we aim to find a set of closely connected authors that are related to the anonymous paper, so we just need to focus on two kinds of relationships, including that between the anonymous paper and author, as well as that between authors. Therefore, we consider to construct a weighted paper-ego-network, which naturally should only contain nodes of anonymous paper p and candidate authors except for the other types of redundant nodes (V, T and Y). The key of constructing the network is to determine the weights of edges between anonymous paper and authors, and that between authors. For edges between authors, we propose the task-guided embedding (TaskGE) to learn the low-dimensional representations of nodes. Since the feature representation for the anonymous paper is unknown, we first employ the weighted combination of feature vectors of its observed neighbors in the network to calculate its vector. Then we can easily determine the edges between anonymous paper and authors based on the computed representation of the anonymous paper and the vectors of authors obtained in TaskGE.

Specifically, for each anonymous paper p, we denote the constructed weighted paper-ego-network by $G_p = (V, E, W)$, where V is a set of nodes, E is a set of edges and W is a set of the weight on each edge. V includes two types of nodes, that is, anonymous paper p and candidate authors. Correspondingly, E contains two types of edges, namely, the edge between paper p and any candidate author a, and the edge between any two candidate authors a_1, a_2. We denote the weights of these two types of edges by w_{pa} and $w_{a_1a_2}$, respectively. Different from existing general-purpose embedding, our embedding method is totally dependent of specific-task. We exploit two unique characteristics or significant aspects of author set identification task. One is the proximity between anonymous paper and authors, we model it as paper-author-aware embedding. The other is the strong relationship between authors, we model it as author-author-aware embedding.

Paper-Author-Aware Embedding. Intuitively, for a given paper p, the relevance score of p and any one a of its true authors should be ξ larger than that of p and other author a' who is not the author of p. Otherwise, a loss penalty will incur. Here, we employ the hinge loss [40] to define a general function to model the relationship between paper and author as follows:

$$\mathcal{L}_{R_{P\frown A}} = \sum_{r \in \mathcal{P}_{P\frown A}} \mathbb{E}_{<p,a,a'>|r} [\xi + f(p,a) - f(p,a')]_+, \tag{7.29}$$

where $[x]_+ = max(x, 0)$ is the standard hinge loss, ξ is the safety margin size [2]. $< p, a, a' >$ denotes the triples <paper, positive author, negative author>. r and $\mathcal{P}_{P\frown A}$ denote any meta path and the set of meta paths between paper and author, respectively. Generally, we can add any proper meta paths between paper and author to $\mathcal{P}_{P\frown A}$ for leveraging multiple information. Actually, there exist multiple indirect relations besides the direct relation between paper and author. For example, $\mathcal{P}_{P\frown A} = \{PA, PTPA\}$ denotes we not only consider the direct author but also take the potential authors into account. Correspondingly, $\mathcal{P}_{P\frown A} = \{PA\}$ means we only consider the direct author of paper. $f(p,a)$ stands for the metric between paper p and author a. As demonstrated by CML [12], distance metric [37] satisfies better triangle inequality and transition property than inner-product, we use the euclidean distance to define the metric:

$$f(p,a) = \|\mathbf{X}_p - \mathbf{X}_a\|_2^2, \tag{7.30}$$

where $\mathbf{X}_p$ and $\mathbf{X}_a$ are the embedding vectors of p and a, respectively. For a new anonymous paper, we adopt similar approach to Chen et al. [4] to calculate its vector representation. That is, the embedding of a paper is represented as the weighted combination of the vectors of observed different types of neighbors in the network as follows:

$$\mathbf{X}_p = \sum_{t=1}^{n} w_t \mathbf{X}_p^t, \tag{7.31}$$

where n is the number of neighbors' types of paper p, $\mathbf{X}_p^t$ is the mean of vectors of the t-th node type, $\mathbf{X}_p^t = \sum_{i \in N_p^t} \frac{\mathbf{X}_i}{|N_p^{(t)}|}$, $N_p^{(t)}$ denotes the set of nodes of the t-th

type. In this section, we do not employ the reference type of nodes due to the lack of citation data.

Author-Author-Aware Embedding. $\mathcal{L}_{R_{P\frown A}}$ models the relationship between paper and author, in this subsection, we will consider how to model the relationship between authors. It is reasonable that there should be strong relationships between co-authors. In other words, the relevance score between co-authors should be larger than that of authors who have never collaborated with each other. Correspondingly, there might exist some potential co-authorship between authors implicitly indicated by meta paths like $APTPA$. Therefore, we also define a general function to formulate the triple relation $< a^*, a^+, a^- >$.

$$\mathcal{L}_{R_{A\frown A}} = \sum_{r \in \mathcal{P}_{A\frown A}} \mathbb{E}_{(a^*,a^+,a^-)|r} [\xi + f(a^*, a^+) - f(a^*, a^-)]_+, \tag{7.32}$$

where a^+ means any co-author of a^*, a^- denotes any author who has never cooperated with a^*. f is the metric function which has been introduced in Equation 7.30. r denotes any meta path between authors. $\mathcal{P}_{A\frown A}$ is the set of meta path between authors. $\mathcal{P}_{A\frown A} = \{APA\}$ means we only consider existing co-authors.

Regularization. Recently, Cogswell et al. [6] propose a new regularization technique called covariance regularization, which is initially used to reduce the correlation between activations in a deep neural network. Afterwards, Hsieh et al. [12] find that it is useful in de-correlating the dimensions. As covariances can be seen as a measure of linear redundancy between dimensions, this loss of covariance regularization essentially tries to prevent each dimension from being redundant. Therefore, we employ loss of covariance regularization as follows:

$$\mathcal{L}_{reg} = \frac{1}{N}(\|C\|_f - \|diag(C)\|_2^2), \tag{7.33}$$

where $\|\cdot\|_f$ is the Froeninus norm, C is covariance matrix between all pairs of dimensions i and j, $C_{ij} = \frac{1}{N}\sum_{k=1}^{N}(\mathbf{X}_k^{(i)} - u_i)(\mathbf{X}_k^{(j)} - u_j)$, $u_i = \frac{1}{N}\sum_{k=1}^{N}\mathbf{X}_k^{(i)}$, $\mathbf{X}_k^{(i)}$ denotes the i-th dimension of embedding vector of node k.

Finally, we combine three parts above to get the unified objective function for task-guided embedding as follows:

$$\mathcal{L} = \mathcal{L}_{R_{P\frown A}} + \gamma\mathcal{L}_{R_{A\frown A}} + \lambda\mathcal{L}_{reg}, \tag{7.34}$$

where $\mathcal{L}_{reg}$ is the regularization term for avoiding over-fitting, λ controls penalty of regularization, γ is a harmonic factor to balance two components. In this section, we only consider the direct relation PA in $\mathcal{L}_{R_{P\frown A}}$ and APA in $\mathcal{L}_{R_{A\frown A}}$.

To minimize $\mathcal{L}$, we design a sampling based mini-batch Adam optimizer [14]. To get the training triples $< p, a, a' >$ and $< a^*, a^+, a^- >$, we draw positive samples according to the proportion of path instances of different meta paths. This sampling strategy can avoid the problem of under-sampling for relations with a large number of links or over-sampling for those with a small number of links. For each sampled positive example $< p, a >$, we first fix vertex p and the corresponding relation. Then

we randomly generate negative vertex a' which has not the same relation with p to construct training triples $< p, a, a' >$. Similarly, we can fix a^* and corresponding relation to acquire training triples $< a^*, a^+, a^- >$. Given the low-dimension representation learned above, we can easily calculate w_{pa} and $w_{a_1 a_2}$ using distance function such as cosine.

7.4.2.3 Optimal Quasi-Clique with Constraint Extraction in Weighted Paper-Ego-Network

For each new paper p, we construct a weighted paper-ego-network $G_p = (V, E, W)$. In order to find the optimal author set for the given paper p in G_p, we propose a new method called OQCCE which is an adaptation of the local-search heuristic by Tsourakakis et al. [33]. The algorithm selects p as initial set. Then under the guidance of designed novel density function, algorithm iterates two phases of adding or removing the designated nodes until the quasi-clique with maximum density function is discovered. What's more, the novel density function considers two kinds of heterogeneous relationships, including the close relationship between the anonymous paper and author, as well as that between authors.

In specific, we regard the node p as constraint, which means that the extracted subgraph must contain node p. In [33], there is only one type of edge. However, there exist two types of edges in weighted paper ego network. The simplest method is to assign equal significance to two types of edges. In fact, the importance may vary. Therefore, we introduce a variable β to adjust the importance of two types of edges. Meanwhile, we also adapt the density function to accommodate the weighted network. Accordingly, the proposed novel density function can be defined as follows:

$$g_{\alpha,\beta}(S) = \beta \sum_{(i,j)\in D_{PA}} w_{ij} + \sum_{(k,l)\in D_{AA}} w_{kl} - \alpha \binom{|S|}{2}, \tag{7.35}$$

where S represents a subset of vertices of network G_p having $S \subseteq V$, $|S|$ denote the number of nodes in the subgraph induced by S, w_{ij} is the weight of edge between nodes i and j in the subgraph induced by S. D_{PA} represents the set of edges between given paper p and candidate authors in the subgraph induced by S. Likewise, D_{AA} represents the set of edges between authors in the subgraph induced by S. β controls the importance of paper-author edge. α is a constant. The first two parts in Eq. 7.35 favors subgraphs with abundant edges while the third part penalizes large subgraphs.

Based on the proposed density function above, next we will describe how to find the optimal quasi-clique with constraint in G_p. The algorithm firstly selects constrained node p as the initial set. Then it traverses all nodes one by one and adds u to S if $g_{\alpha,\beta}(S \cup \{u\})$ improves. Afterward, the algorithm traverses every vertex v in S and remove v if $g_{\alpha,\beta}(S \setminus \{v\})$ enhances. Note that we cannot remove constrained node p during the period of removal. The algorithm repeats these two phases of addition and removal until an optimum is reached or the number of iterations exceeds I_{max}.

7.4.3 Experiments

7.4.3.1 Experimental Settings

Datasets. AMiner [32] is a classical academic network. Specifically, we extract two subsets with different scale, denoted by AMiner-I and AMiner-II. AMiner-I is a small subset data of some important venues in data mining area, which includes 5 venues, namely KDD, ICDM, SDM, CIKM and PKDD. AMiner-II is a large subset of four areas, including Artificial Intelligence (AI), Data Mining (DM), Databases (DB), and Information System (IS). For each area, we choose some important venues[8] which have influential publications.

Baselines. In order to examine the effectiveness of our approach, we compare against the following three kinds of representative methods. (1) Similarity measure. We design two kinds of similarity measure methods based on meta paths $PTPA$ and $PCPA$, which can indirectly connect the new paper and candidate authors with term or venue. Then we rank candidate authors according to the similarity scores (i.e., the number of path instances) between candidate authors and the new paper. (2) Feature method. Following the work of Chen et al. [4], we extracted 17 features for each paper-author pair. We choose LR, SVM and Bayes as learning algorithms. (3) HetNetE. HetNetE is recently proposed in [4] for author identification problem. It first learns the low-dimensional feature vectors of nodes to predict author of the given paper.

Parameter Settings. For our method ASI, we set the embedding dimension d to 128, the size of negative samples to 2, the margin ξ to 2, the learning rate to 0.00001, the batch size to 200, the regularization penalty λ to 10, the trade-off factor γ to 1.0, α to 0.01, β to 0.1. For HetNetE and Feature method, we choose the optimal parameter. Three meta paths APC, APW, APP are jointly used in HetNetE. In addition, for fairness comparison, we do not adopt the reference types of nodes when computing the embedding vectors of papers due to the lack of most citations in HetNetE and ASI.

Evaluation Metrics. We adopt $Precision$ (P), $Recall$ (R), $F1$ score, $Jaccard$ index (J), MAP (mean average precision) and $RMSE$ as evaluation metrics. (1) P. It reflects the accuracy of returned author set, which can be defined as the ratio of the true authors in the returned author set. $P = \frac{|S'_A \cap S_A|}{|S'_A|}$, where S'_A denotes the returned author set or the returned top-k author set in $P@k$. S_A means the true author set. (2) R. It shows the ratio of returned true authors in the whole true author set. It can be computed as follows: $R = \frac{|S'_A \cap S_A|}{|S_A|}$, where S'_A and S_A have the same meanings introduced above. (3) $F1$. It is the harmony average of P and R, which is defined as: $F1 = \frac{2*P*R}{P+R}$. (4) $Jaccard$ index. It measures similarity between two sets and is formulated as: $J = \frac{|S'_A \cap S_A|}{|S'_A \cup S_A|}$, which means the ratio of the intersection and the union of two sets. (5) MAP. It is computed as mean of AP at different k for a paper.

[8] AI: ICML, AAAI, IJCAI, NIPS. DM: KDD, WSDM, ICDM, PKDD. DB: SIGMOD, VLDB, ICDE. IS: SIGIR, CIKM.

$AP = \frac{\sum_{i=1}^{k} \text{p@}i \times rel_i}{\text{\# of correct author}}$, where rel_i equals 1 if the result at rank i is correct author and 0 otherwise. (6) $RMSE$. It is a measure of difference between the number of authors returned by model and the number of true authors. $RMSE = \sqrt{\frac{\sum(|S'_A| - |S_A|)^2}{|m|}}$, where m is the number of test papers, S'_A and S_A are the number of returned author and true author, respectively.

Table 7.5 Results of effectiveness experiments on AMiner-I. We use bold to mark the best performance for each comparison. ↑ indicates higher is better, ↓ indicates lower is better. "Avg." means the average rank of different methods.

Methods			Evaluation					
			P (↑)	R (↑)	J (↑)	F1 (↑)	MAP (↑)	RMSE (↓)
Top-5	Similarity measure	PTPA	0.2716 (2)	0.5007 (7)	0.2310 (2)	0.3356 (2)	**0.6109** (1)	0.1714 (2)
		PCPA	0.2098 (7)	0.3937 (11)	0.1680 (7)	0.2614 (7)	0.4718 (9)	0.1714 (2)
	Feature method	LR	0.2160 (5)	0.3915 (12)	0.1827 (6)	0.2657 (4)	0.4834 (7)	0.1714 (2)
		SVM	0.2493 (3)	0.4562 (9)	0.2154 (4)	0.3081 (3)	0.5451 (3)	0.1714 (2)
		Bayes	0.2209 (4)	0.4075 (10)	0.1888 (5)	0.2733 (5)	0.4951 (6)	0.1714 (2)
	HetNetE		0.2123 (6)	0.3870 (13)	0.1669 (8)	0.2616 (6)	0.4571 (11)	0.1714 (2)
Top-10	Similarity measure	PTPA	0.1555 (9)	0.5779 (2)	0.1454 (10)	0.2365 (9)	0.5897 (2)	0.5023 (3)
		PCPA	0.1388 (11)	0.5066 (5)	0.1257 (13)	0.2110 (11)	0.4517 (12)	0.5023 (3)
	Feature method	LR	0.1358 (13)	0.5005 (8)	0.1270 (12)	0.2059 (13)	0.4664 (10)	0.5023 (3)
		SVM	0.1629 (8)	**0.5988** (1)	0.1538 (9)	0.2477 (8)	0.5296 (4)	0.5023 (3)
		Bayes	0.1364 (12)	0.5010 (6)	0.1277 (11)	0.2069 (12)	0.4767 (8)	0.5023 (3)
	HetNetE		0.1506 (10)	0.5347 (3)	0.2269 (3)	0.2275 (10)	0.4435 (13)	0.5023 (3)
ASI			**0.4589** (1)	0.5284 (4)	**0.4009** (1)	**0.4712** (1)	0.5295 (5)	**0.1123** (1)

Table 7.6 Results of effectiveness experiments on AMiner-II. We use bold to mark the best performance for each comparison. ↑ indicates higher is better, ↓ indicates lower is better. "Avg." means the average rank of different methods.

Methods			Evaluation					
			P (↑)	R (↑)	J (↑)	F1 (↑)	MAP (↑)	RMSE (↓)
Top5	Similarity measure	PTPA	0.3391 (2)	0.5899 (6)	0.2886 (2)	0.4108 (2)	0.7165 (3)	0.2880 (2)
		PCPA	0.3287 (3)	0.5743 (8)	0.2776 (4)	0.3986 (3)	0.6595 (6)	0.2880 (2)
	Feature method	LR	0.3113 (4)	0.5400 (9)	0.2645 (5)	0.3769 (4)	0.6605 (5)	0.2880 (2)
		SVM	0.2202 (7)	0.4553 (12)	0.1674 (11)	0.2803 (9)	**0.9948** (1)	0.2880 (2)
		Bayes	0.2964 (5)	0.5144 (10)	0.2491 (6)	0.3587 (5)	0.6458 (8)	0.2880 (2)
	HetNetE		0.2645 (6)	0.4561 (11)	0.2078 (7)	0.3191 (6)	0.6021 (12)	0.2880 (2)
Top10	Similarity measure	PTPA	0.1927 (8)	**0.6624** (1)	0.1795 (8)	0.2884 (7)	0.6913 (4)	0.8536 (3)
		PCPA	0.1913 (9)	0.6531 (2)	0.1778 (9)	0.2860 (8)	0.6363 (10)	0.8536 (3)
	Feature method	LR	0.1857 (10)	0.5779 (7)	0.1729 (10)	0.2775 (10)	0.6382 (9)	0.8536 (3)
		SVM	0.1101 (13)	0.4553 (12)	0.0943 (13)	0.1702 (13)	0.9948 (1)	0.8536 (3)
		Bayes	0.1786 (11)	0.6157 (4)	0.1661 (12)	0.2673 (11)	0.6227 (11)	0.8536 (3)
	HetNetE		0.1720 (12)	0.6350 (3)	0.2858 (3)	0.2564 (12)	0.5602 (13)	0.8536 (3)
ASI			**0.5981** (1)	0.6019 (5)	**0.4943** (1)	**0.5720** (1)	0.6566 (7)	**0.2058** (1)

7.4.3.2 Comparisons and Analysis

To evaluate the performance, we regard papers published before 2014 as training set and papers published in 2014 and 2015 as test set. Since it is time consuming to rank all candidate authors for each anonymous paper in the evaluation procedure, following the strategy in [4], for each paper in the test set, we randomly sample some negative authors and obtain 100 candidate authors in all. Then, we rank the 100 candidate authors consisting of the positive and sampled negative authors for each paper. For our method ASI, we also select the same 100 candidate authors to construct the weighted paper ego network for each test paper. The final results are averaged over all the test papers for each evaluation metric.

We report the results of performance comparison in Tables 7.5, 7.6. There are some observations and analysis. (1) Our method ASI achieves better performance than all baselines on all measures except R and MAP. It improves the performance by more than 15% on P, J and $F1$ averagely. Although ASI does not achieve the best performance on R, it is also near the best value. (2) ASI can automatically confirm the appropriate number of authors for a given paper, which can be clearly demonstrated by the lowest value on metric $RMSE$. In a word, ASI not only can discover a set of authors with strong relationship but also can determine the proper number of authors for an anonymous paper. (3) To our surprise, the similarity measure method based on PTPA has very good performance, which indicates that the term has a significant role in finding author set for a given paper.

The more detailed method description and experiment validation can be seen in [43].

7.5 Conclusions

In recent years, to characterize the complex and heterogeneous auxiliary data in web services, HG representation techniques have become a very popular approach for recommender systems. In this chapter, we present three HG representation based recommendation systems respectively, solving the unique challenges existing in diverse real-world scenarios. Particularly, we study the Top-N recommendation scenario and propose the MCRec framework, which is a three-way neural interaction model based HG representation method. In addition, we study the cold-start problem in the recommendation and propose a meta-learning based method for HG representation, named MetaHIN. At Last, we study the author set identification problem in the bibliographic recommendation and propose ASI method to solve this problem. The experiments demonstrate the effectiveness of HG representation in each application.

In future work, we will consider how to combine more auxiliary multi-modal information to improve performance. In addition, we will extend the HG representation approach to other more challenging applications.

References

1. Berg, R.v.d., Kipf, T.N., Welling, M.: Graph convolutional matrix completion. arXiv preprint arXiv:1706.02263 (2017)
2. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: NeurIPS, pp. 2787–2795 (2013)
3. Chai, T., Draxler, R.R.: Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. Geoscientific model development **7**(3), 1247–1250 (2014)
4. Chen, T., Sun, Y.: Task-guided and path-augmented heterogeneous network embedding for author identification. In: WSDM, pp. 295–304. ACM (2017)
5. Chen, T., Zhang, W., Lu, Q., Chen, K., Zheng, Z., Yu, Y.: Svdfeature: a toolkit for feature-based collaborative filtering. Journal of Machine Learning Research **13**, 3619–3622 (2012)
6. Cogswell, M., Ahmed, F., Girshick, R., Zitnick, L., Batra, D.: Reducing overfitting in deep networks by decorrelating representations. arXiv preprint arXiv:1511.06068 (2015)
7. Dong, Y., Chawla, N.V., Swami, A.: metapath2vec: Scalable representation learning for heterogeneous networks. In: KDD, pp. 135–144 (2017)
8. Feng, W., Wang, J.: Incorporating heterogeneous information for personalized tag recommendation in social tagging systems. In: KDD, pp. 1276–1284 (2012)
9. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: ICML, pp. 1126–1135 (2017)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR, pp. 770–778 (2016)
11. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: WWW, pp. 173–182 (2017)
12. Hsieh, C.K., Yang, L., Cui, Y., Lin, T.Y., Belongie, S., Estrin, D.: Collaborative metric learning. In: WWW, pp. 193–201 (2017)
13. Hu, B., Shi, C., Zhao, W.X., Yu, P.S.: Leveraging meta-path based context for top-n recommendation with a neural co-attention model. In: KDD, pp. 1531–1540 (2018)
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. Third International Conference on Learning Representations (2015)
15. Koren, Y., Bell, R.: Advances in collaborative filtering. In: Recommender systems handbook, pp. 77–118 (2015)
16. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42** (2009)
17. Lee, H., Im, J., Jang, S., Cho, H., Chung, S.: Melu: Meta-learned user preference estimator for cold-start recommendation. In: KDD, pp. 1073–1082 (2019)
18. Lu, Y., Fang, Y., Shi, C.: Meta-learning on heterogeneous information networks for cold-start recommendation. In: SIGKDD, pp. 1563–1573 (2020)
19. Pan, F., Li, S., Ao, X., Tang, P., He, Q.: Warm up cold-start advertisements: Improving ctr predictions via learning to learn id embeddings. In: SIGIR, pp. 695–704 (2019)
20. Pham, T.A.N., Li, X., Cong, G., Zhang, Z.: A general recommendation model for heterogeneous networks. IEEE Transactions on Knowledge and Data Engineering **28**, 3140–3153 (2016)
21. Phan, M.C., Sun, A., Tay, Y., Han, J., Li, C.: Neupl: Attention-based semantic matching and pair-linking for entity disambiguation. In: CIKM, pp. 1667–1676 (2017)
22. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. In: UAI (2009)
23. Rendle, S., Gantner, Z., Freudenthaler, C., Schmidt-Thieme, L.: Fast context-aware recommendations with factorization machines. In: SIGIR, pp. 635–644 (2011)
24. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: WWW, pp. 285–295 (2001)
25. Shi, C., Hu, B., Zhao, W.X., Yu, P.S.: Heterogeneous information network embedding for recommendation. IEEE Transactions on Knowledge and Data Engineering (2018)

26. Shi, C., Hu, B., Zhao, X., Yu, P.: Heterogeneous information network embedding for recommendation. IEEE Transactions on Knowledge and Data Engineering (2018)
27. Shi, C., Li, Y., Zhang, J., Sun, Y., Philip, S.Y.: A survey of heterogeneous information network analysis. IEEE Transactions on Knowledge and Data Engineering **29**, 17–37 (2017)
28. Shi, C., Zhang, Z., Luo, P., Yu, P.S., Yue, Y., Wu, B.: Semantic path based personalized recommendation on weighted heterogeneous information networks. In: CIKM, pp. 453–462 (2015)
29. Sun, Y., Han, J.: Mining heterogeneous information networks: principles and methodologies. Synthesis Lectures on Data Mining and Knowledge Discovery **3**(2), 1–159 (2012)
30. Sun, Y., Han, J., Yan, X., Yu, P.S., Wu, T.: Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. Very Large Data Base Endowment **4**, 992–1003 (2011)
31. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: Large-scale information network embedding. In: WWW, pp. 1067–1077 (2015)
32. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: Arnetminer: extraction and mining of academic social networks. In: SIGKDD, pp. 990–998. ACM (2008)
33. Tsourakakis, C., Bonchi, F., Gionis, A., Gullo, F., Tsiarli, M.: Denser than the densest subgraph: extracting optimal quasi-cliques with quality guarantees. In: SIGKDD, pp. 104–112. ACM (2013)
34. Vartak, M., Thiagarajan, A., Miranda, C., Bratman, J., Larochelle, H.: A meta-learning perspective on cold-start recommendations for items. In: NeurIPS, pp. 6904–6914 (2017)
35. Volkovs, M., Yu, G., Poutanen, T.: Dropoutnet: Addressing cold start in recommender systems. In: NeurIPS, pp. 4957–4966 (2017)
36. Wang, X., He, X., Wang, M., Feng, F., Chua, T.: Neural graph collaborative filtering. In: SIGIR, pp. 165–174 (2019)
37. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. Journal of Machine Learning Research **10**(Feb), 207–244 (2009)
38. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: ICML, pp. 2048–2057 (2015)
39. Yu, X., Ren, X., Sun, Y., Gu, Q., Sturt, B., Khandelwal, U., Norick, B., Han, J.: Personalized entity recommendation: A heterogeneous information network approach. In: WSDM, pp. 283–292 (2014)
40. Zhang, C., Huang, C., Yu, L., Zhang, X., Chawla, N.V.: Camel: Content-aware and meta-path augmented metric learning for author identification. In: WWW, pp. 709–718 (2018)
41. Zhang, Y., Ai, Q., Chen, X., Croft, W.B.: Joint representation learning for top-n recommendation with heterogeneous information sources. In: CIKM, pp. 1449–1458 (2017)
42. Zhao, H., Yao, Q., Li, J., Song, Y., Lee, D.L.: Meta-graph based recommendation fusion over heterogeneous information networks. In: KDD, pp. 635–644 (2017)
43. Zheng, Y., Shi, C., Kong, X., Ye, Y.: Author set identification via quasi-clique discovery. In: CIKM, pp. 771–780 (2019)
44. Zhu, Y., Lin, J., He, S., Wang, B., Guan, Z., Liu, H., Cai, D.: Addressing the item cold-start problem by attribute-driven active learning. IEEE Transactions on Knowledge and Data Engineering (2019)