Meta-learning on Heterogeneous Information Networks for Cold-start Recommendation

Yuanfu Lu[1,2†], Yuan Fang[3], Chuan Shi[1,4‡]
[1]Beijing University of Posts and Telecommunications
[2]WeChat Search Application Department, Tencent Inc. China
[3]Singapore Management University
[4]Peng Cheng Laboratory, Shenzhen, China
{luyuanfu,shichuan}@bupt.edu.cn,yfang@smu.edu.sg

ABSTRACT

Cold-start recommendation has been a challenging problem due to sparse user-item interactions for new users or items. Existing efforts have alleviated the cold-start issue to some extent, most of which approach the problem at the data level. Earlier methods often incorporate auxiliary data as user or item features, while more recent methods leverage heterogeneous information networks (HIN) to capture richer semantics via higher-order graph structures. On the other hand, recent meta-learning paradigm sheds light on addressing cold-start recommendation at the model level, given its ability to rapidly adapt to new tasks with scarce labeled data, or in the context of cold-start recommendation, new users and items with very few interactions. Thus, we are inspired to develop a novel meta-learning approach named MetaHIN to address cold-start recommendation on HINs, to exploit the power of meta-learning at the model level and HINs at the data level simultaneously. The solution is non-trivial, for how to capture HIN-based semantics in the meta-learning setting, and how to learn the general knowledge that can be easily adapted to multifaceted semantics, remain open questions. In MetaHIN, we propose a novel semantic-enhanced tasks constructor and a co-adaptation meta-learner to address the two questions. Extensive experiments demonstrate that MetaHIN significantly outperforms the state of the arts in various cold-start scenarios. (Code and dataset are available at https://github.com/rootlu/MetaHIN.)

CCS CONCEPTS

• **Information systems** → **Data mining**; **Recommender systems**;

KEYWORDS

Heterogeneous Information Network, Meta-learning, Cold-start Recommendation

† Work done while a visiting research student at Singapore Management University.
‡ The corresponding author.

KDD '20, August 23–27, 2020, Virtual Event, CA, USA

ACM ISBN 978-1-4503-7998-4/20/08…$15.00
https://doi.org/10.1145/3394486.3403207

ACM Reference Format:
Yuanfu Lu[1,2†], Yuan Fang[3], Chuan Shi[1,4‡]. 2020. Meta-learning on Heterogeneous Information Networks for Cold-start Recommendation. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3394486.3403207

1 INTRODUCTION

Recommender systems [3, 11, 31] have been widely deployed in various online services, such as E-commerce platforms and news portals, to address the issue of information overload for users. At their core, they typically adopt *collaborative filtering*, aiming to estimate the likelihood of a user adopting an item based on the interaction history like past purchases and clicks. However, the interaction data of new users or new items are often very sparse, leading to the so-called *cold-start* scenarios [35] in which it becomes challenging to learn effective user or item representations.

To alleviate the cold-start problem, a common approach is to integrate auxiliary data to enhance the representations of new users or items, where user or item contents (e.g., age and gender of users) are often exploited [17, 35]. More recently, heterogeneous information networks (HIN) [23] have been leveraged to enrich user-item interactions with complementary heterogeneous information. As shown in Fig. 1(a), 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 [25], 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 [11, 34] also assume a *data-level* strategy to alleviate the cold-start problem, as illustrated in Fig. 1(b).

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 [27]. 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

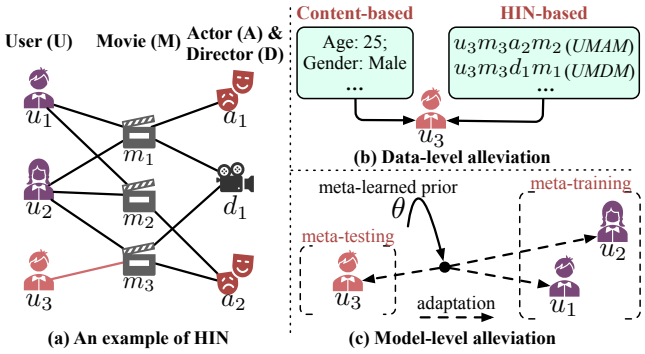


Figure 1: An example of HIN and existing data or model-level alleviation for cold-start recommendation.

easily and quickly adapted to the new tasks of cold-start users with scarce interaction data during meta-testing. As illustrated in Fig. 1(c), 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. As such, a limited number of recent studies [6, 16, 19] have leveraged meta-learning for the cold-start problem and achieved promising results. However, they typically involve a direct adoption of meta-learning frameworks (e.g., MAML [9]), but neglect to explore the unique heterogeneous graph structures and semantics on HINs for cold-start recommendation.

Challenges and present work. 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. However, meta-learning on HINs is non-trivial, presenting us with two key challenges. (1) *How to capture the semantics on HINs in the meta-learning setting?* Existing methods either model HINs under traditional supervised learning settings [11, 22], or ignore the rich structures and semantic contexts in meta-learning settings [16, 19]. Hence, it is vital to re-examine the design of user-based tasks, to enrich user-item interaction data with higher-order semantics. (2) *How to learn the general knowledge across tasks, particularly in a way that can be easily generalized to work with multifaceted heterogeneous semantics?* In existing meta-learning methods for cold-start recommendation [6, 16], they perform adaptations for new tasks (e.g., new users) from a globally shared prior. In other words, the prior is designed for generalization to different tasks. However, there also exist multi-faceted semantics (e.g., movies directed by the same director, or starring the same actor) brought by HINs. Hence, it is crucial for the meta-learned prior to be capable of generalizing to different semantic facets within each task too.

The above challenges motivate us to develop a **Meta**-learning approach to cold-start recommendation on **H**eterogeneous **I**nformation **N**etworks, named **MetaHIN**. To address the first challenge, we propose 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. For the second challenge, we 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.

Contributions. To summarize, this work makes the following major contributions. (1) This is the first attempt to exploit meta-learning on HINs for cold-start recommendation, which alleviates the cold-start problem at both data and model levels. (2) We propose a novel method MetaHIN, which leverages multifaceted semantic contexts and a co-adaption meta-learner in order to learn finer-grained semantic priors for new tasks in both semantic and task-wise manners. (3) We conduct extensive empirical studies on three real-world datasets on different cold-start scenarios, and demonstrate that MetaHIN consistently and significantly outperforms various state of the arts.

2 RELATED WORK

Cold-start Recommendation. While collaborative filtering [17, 31] has achieved considerable success in recommendation systems, difficulty often arises in dealing with new users or items with sparse user-item interactions, known as cold-start recommendation. Traditional cold-start solutions rely on data augmentation, by incorporating user or item side information [15, 29, 35]. Beyond these content-based features and user-item interactions, richer heterogeneous data that captures the interactions between items (e.g., movies) and other objects (e.g., actors) has been exploited in the form of a heterogeneous information network (HIN). On HINs, one major line of work leverages higher-order graph structures such as meta-paths [25] or meta-graphs [7, 8] to explore heterogeneous semantics in recommendation settings [11, 22]. Some methods also integrate review texts, images or knowledge graphs to further enhance user and item representations [30, 34]. Additionally, some transfer learning-based methods use features from a source domain to apply to the target domain [12, 13], with the assumption that a source domain is available and users or items can be aligned in both domains. Although these methods achieve promising performances, they only alleviate the cold-start problem at the data level, which heavily relies on the availability and quality of auxiliary data.

Meta-learning. Also known as learning to learn, meta-learning intends to learn the general knowledge across similar learning tasks, so as to rapidly adapt to new tasks based on a few examples [28]. Among previous work on meta-learning, metric-based methods [24, 26] learn a metric or distance function over tasks, while model-based methods [18, 21] aim to design a architecture or training process for rapid generalization across tasks. Lastly, optimization-based methods directly adjust the optimization algorithm to enable quick adaptation with just a few examples [9, 32].

The success of meta-learning in few-shot settings (i.e., each task only has a few labeled examples) has shed light on the problem of cold-start recommendation [6, 16, 27]. Specifically, Vartak et. al. [27] present a metric-based approach to address the item cold-start problem. The work of Lee et. al. [16] applies the MAML framework [9], which rapidly adapts to new users or items based on sparse interaction history. Meta-learning with MAML has also been applied to scenario-based cold-start problems, which formulates the recommendation scenario (e.g., baby and outdoor products are two different scenarios) as a learning task [6]. Moreover, some studies focus on specific applications, including CTR prediction [19] and clinical risk prediction [33]. Unfortunately, these methods do not consider the unique multifaceted semantic contexts enabled by HINs for cold-start recommendation.

3 PRELIMINARIES

In this section, we formalize the problem of HIN-based cold-start recommendation, and introduce the meta-learning perspective for recommendation.

3.1 Problem Formulation

Our study focuses on HIN-based cold-start recommendation, wherein a HIN can be defined as follows [23].

Definition 1. ***Heterogeneous Information Network (HIN).*** *A HIN is defined as a graph $G = \{V, E, O, R\}$ with nodes V and edges E, which belong to multiple types. Each node and edge are respectively associated with a type mapping function $\varphi_O : V \to O$ and $\varphi_R : E \to R$, where O and R represent the set of object and relation types, respectively. G is a HIN if $|O| + |R| > 2$.*

On a HIN, meta-path [25] can capture complex semantics between objects via composite relations, as defined in the following.

Definition 2. ***Meta-path.*** *Given a HIN with node types O and relation types R, a meta-path of length l is defined as a composite relation $P = o_1 \xrightarrow{r_1} o_2 \xrightarrow{r_2} \cdots \xrightarrow{r_l} o_{l+1}$, where each $o_i \in O$ and $r_i \in R$. If there is only a single type of relation between two types, we omit the relations and write a meta-path simply as $P = o_1 o_2 \cdots o_{l+1}$.*

As a form of higher-order graph structure, meta-paths are widely used to explore the semantics in a HIN [5]. Fig. 1(a) shows an example of HIN for a movie recommender system, which consists of four types of nodes, i.e., O = {*User (U)*, *Movie (M)*, *Actor (A)* and *Director (D)*}. Meta-paths such as *UMAM* and *UMDM* define different semantic relations: movies starring the same actor, or directed by the same director, respectively. Moreover, $u_3 m_3 a_2 m_2$ is an *instance* of *UMAM*, and $u_3 m_3 d_1 m_1$ is an instance of *UMDM*.

Definition 3. ***Cold-start Recommendation.*** *On 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)**; similarly, 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)**.*

3.2 Meta-learning for Recommendation

Our work is inspired by the optimization-based meta-learning [9, 32], which optimizes globally shared parameters (i.e., prior knowledge) over several tasks, so as to rapidly adapt to a new task with just one or a few gradient steps based on a small number of examples. In recommendation [16], a task $\mathcal{T}_u = (\mathcal{S}_u, \mathcal{Q}_u)$ involves one user u, consisting of a support set $\mathcal{S}_u$ and a query set $\mathcal{Q}_u$. We learn the prior shared across a set of meta-training tasks $\mathcal{T}^{\text{tr}}$, and adapt the prior to new tasks, known as meta-testing tasks $\mathcal{T}^{\text{te}}$, in order to predict item ratings.

Specifically, during meta-training, for each task $\mathcal{T}_u \in \mathcal{T}^{\text{tr}}$, its support and query sets contain items sampled from the set of items rated by u, such that the support and query items are mutually exclusive. Typically the support set only contains a few items. The meta-learner adapts the global prior θ to task-specific parameters w.r.t. the loss on the support set $\mathcal{S}_u$. Next, on the query set $\mathcal{Q}_u$, the loss under the task-specific parameters is calculated and backward propagated to update the global θ. Formally,

$$\min_{\theta} \sum_{\mathcal{T}_u \in \mathcal{T}^{\text{tr}}} \mathcal{L}\left(\theta - \eta \nabla_\theta \mathcal{L}\left(\theta, \mathcal{S}_u\right), \mathcal{Q}_u\right), \tag{1}$$

where $\mathcal{L}$ is the loss function, ∇ denotes the gradient, and η is the meta-learning rate. Here $\theta - \eta \nabla_\theta \mathcal{L}(\theta, \mathcal{S}_u)$ is the task-specific parameters adapted to $\mathcal{T}_u$ after one gradient step from the global θ.

During meta-testing, for each task $\mathcal{T}_u \in \mathcal{T}^{\text{te}}$, the support set still contains a small number of items rated by u, but the query set only contains items whose ratings are to be predicted. The meta-learner adapts the prior θ learned during meta-training to $\mathcal{T}_u$ via one or a few gradient steps w.r.t. its support set $\mathcal{S}_u$. The adapted parameters are then applied to predict the ratings of items in the query set, i.e., $\{\hat{r}_{u,i} : i \in \mathcal{Q}_u\}$. Depending on how a meta-testing task is defined, we address different cold-start scenarios: (1) **UC**, if the task is about an new user not seen in meta-training; (2) **IC**, if it is an existing user but the items in the support and query sets are new items; (3) **UIC**, if both the user and items are not observed in meta-training.

4 METHODOLOGY

In this section, we present a novel method MetaHIN for cold-start recommendation, based on meta-learning on HINs.

4.1 Overview of MetaHIN

As illustrated in Fig. 2, the proposed MetaHIN consists of two components: *semantic-enhanced* task constructor in Fig. 2(a) and *co-adaptation* meta-learner in Fig. 2(b).

First, existing meta-learning methods for recommendation [6, 16] only consider user-item interactions, but a HIN often carries valuable semantic information. Thus, 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 items related via each meta-path represents a different facet of heterogeneous semantics.

Second, existing methods only adopt task-wise adaptation from a global prior θ. However, as the semantic contexts are multifaceted, it is also crucial to perform semantic-wise adaptation, in order to adapt the global prior to finer-grained semantic priors for different

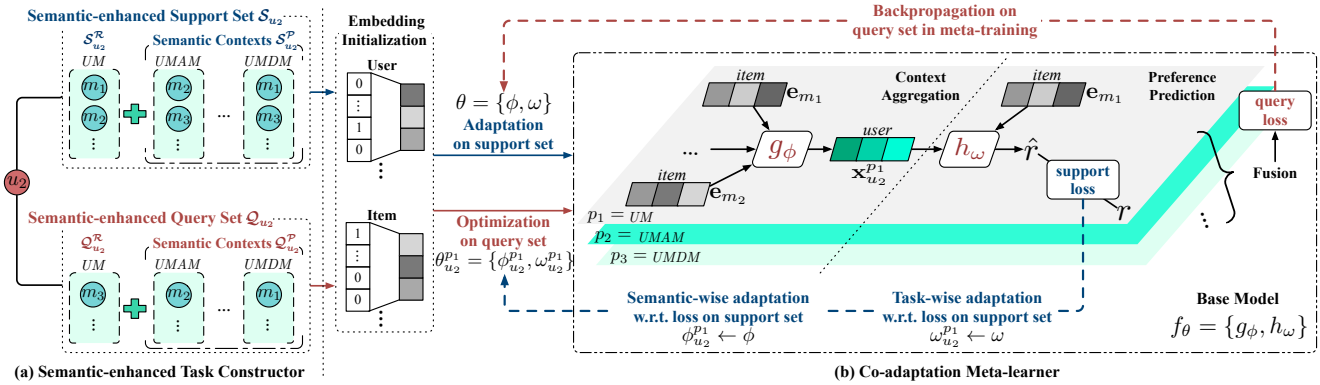


Figure 2: Illustration of the meta-training procedure of a task in MetaHIN. (a) Semantic-enhanced task constructor, where the support and query sets are augmented with meta-path based heterogeneous semantic contexts. (b) Co-adaptation meta-learner, with semantic- and task-wise adaptations on the support 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.

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.

4.2 Semantic-enhanced Task Constructor

As motivated, towards effective meta-learning on HINs, it is important to incorporate multifaceted semantic contexts into tasks. Given a user u with task $\mathcal{T}_u = (\mathcal{S}_u, \mathcal{Q}_u)$, the semantic-enhanced support set is defined as

$$\mathcal{S}_u = (\mathcal{S}^{\mathcal{R}}_u, \mathcal{S}^{\mathcal{P}}_u), \tag{2}$$

where $\mathcal{S}^{\mathcal{R}}_u$ is a set of items that has been rated by user u, and $\mathcal{S}^{\mathcal{P}}_u$ 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 $\mathcal{S}^{\mathcal{R}}_u$ is usually small, i.e., a new user only has a few ratings. For meta-training tasks, we follow previous work [16] to construct $\mathcal{S}^{\mathcal{R}}_u$ 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 $\mathcal{S}^{\mathcal{P}}_u$ is 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. 1(a), $\mathcal{P} = \{UM, UMAM, UMDM, UMUM\}$ 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:

$$C^p_{u,i} = \{j : j \in \text{items reachable along } p \text{ starting from } u\text{–}i\}. \tag{3}$$

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

$$\mathcal{S}^p_u = \bigcup_{i \in \mathcal{S}^{\mathcal{R}}_u} C^p_{u,i}. \tag{4}$$

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

$$\mathcal{S}^{\mathcal{P}}_u = (\mathcal{S}^{p_1}_u, \mathcal{S}^{p_2}_u, \ldots, \mathcal{S}^{p_n}_u). \tag{5}$$

In essence, $\mathcal{S}^{\mathcal{P}}_u$ 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. 2(a), 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 $\mathcal{Q}_u = (\mathcal{Q}^{\mathcal{R}}_u, \mathcal{Q}^{\mathcal{P}}_u)$. In particular, $\mathcal{Q}^{\mathcal{R}}_u$ contains items rated by u for calculating the task loss in meta-training, or items with hidden rating for making predictions in meta-testing; $\mathcal{Q}^{\mathcal{P}}_u$ 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., $\mathcal{S}^{\mathcal{R}}_u \cap \mathcal{Q}^{\mathcal{R}}_u = \emptyset$.

4.3 Co-adaptation Meta-learner

Given the semantic-enhanced tasks, we design a 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.

4.3.1 ***Base Model.*** As shown in Fig. 2(b), 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 [16], 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. Details of the embedding initialization can be found in the supplement A.1.1. Subsequently, we obtain user u's embedding $\mathbf{x}_u$ as follows:

$$\mathbf{x}_u = g_\phi(u, C_u) = \sigma\left(\textsc{mean}(\{\mathbf{W}\mathbf{e}_j + \mathbf{b} : j \in C_u\})\right), \tag{6}$$

where 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), $\textsc{mean}(\cdot)$ 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:

$$\hat{r}_{ui} = h_\omega(\mathbf{x}_u, \mathbf{e}_i) = \textsc{mlp}(\mathbf{x}_u \oplus \mathbf{e}_i), \tag{7}$$

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:

$$\mathcal{L}_u = \frac{1}{|\mathcal{R}_u|} \sum_{i \in \mathcal{R}_u} (r_{ui} - \hat{r}_{ui})^2, \tag{8}$$

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 require 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.

4.3.2 ***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. 2(a), 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 equip the meta-learner with semantic- and task-wise adaptations.

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. 2), 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. (5). Given a meta-path $p \in \mathcal{P}$, user u's embedding in the semantic space of p is

$$\mathbf{x}_u^p = g_\phi(u, \mathcal{S}_u^p). \tag{9}$$

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

$$\mathcal{L}_{\mathcal{T}_u}(\omega, \mathbf{x}_u^p, \mathcal{S}_u^\mathcal{R}) = \frac{1}{|\mathcal{S}_u^\mathcal{R}|} \sum_{i \in \mathcal{S}_u^\mathcal{R}} (r_{ui} - h_\omega(\mathbf{x}_u^p, \mathbf{e}_i))^2, \tag{10}$$

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, as follows.

$$\phi_u^p = \phi - \alpha \frac{\partial \mathcal{L}_{\mathcal{T}_u}(\omega, \mathbf{x}_u^p, \mathcal{S}_u^\mathcal{R})}{\partial \phi} = \phi - \alpha \frac{\partial \mathcal{L}_{\mathcal{T}_u}(\omega, \mathbf{x}_u^p, \mathcal{S}_u^\mathcal{R})}{\partial \mathbf{x}_u^p} \frac{\partial \mathbf{x}_u^p}{\partial \phi}, \tag{11}$$

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.

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:

$$\omega^p = \omega \odot \kappa(\mathbf{x}_u^{p\langle S\rangle}), \tag{12}$$

where $\odot$ is the element-wise product and $\kappa(\cdot)$ serves as a transformation function realized with a fully connected layer (see its detailed form in the supplement A.1.2). 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:

$$\omega_u^p = \omega^p - \beta \frac{\partial \mathcal{L}_{\mathcal{T}_u}(\omega^p, \mathbf{x}_u^{p\langle S\rangle}, \mathcal{S}_u^\mathcal{R})}{\partial \omega^p}, \tag{13}$$

where β is the task-wise learning rate.

Optimization. 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. 2(b), the global prior $\theta = (\phi, \omega)$ will be optimized through backpropgation of the query loss:

$$\min_\theta \sum_{\mathcal{T}_u \in \mathcal{T}^{\text{tr}}} \mathcal{L}_{\mathcal{T}_u}(\omega_u, \mathbf{x}_u, \mathcal{Q}_u^\mathcal{R}), \tag{14}$$

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

$$\omega_u = \sum_{p \in \mathcal{P}} a_p \omega_u^p, \quad \mathbf{x}_u = \sum_{p \in \mathcal{P}} a_p \mathbf{x}_u^{p\langle Q\rangle}, \tag{15}$$

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 [2], 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.

5 EXPERIMENTS

In this section, we conduct extensive experiments to answer three research questions: (**RQ1**) How does MetaHIN perform compared to state-of-the-art approaches? (**RQ2**) How does MetaHIN benefit from the multifaceted semantic contexts and co-adaptation meta-learner? (**RQ3**) How is MetaHIN impacted by its hyper-parameters?

5.1 Experimental Setup

Dataset. We conduct experiments on three benchmark datasets, namely, DBook[1], MovieLens[2], and Yelp[3], from publicly accessible repositories. Their statistics are summarized in Table 1.

For each dataset, we divide users and items into two groups: *existing* and *new*, according to user joining time (or first user action time) and item releasing time. Then, we split each dataset into meta-training and meta-testing data. (1) The meta-training data only contains existing user ratings for existing items. We randomly select 10% of them as the validation set. (2) The rest are meta-testing data, which are divided into three partitions corresponding to three cold-start scenarios: (**UC**) User Cold-start, i.e., recommendation of existing items for new users; (**IC**) Item Cold-start, i.e., recommendation of new items for existing users; (**UIC**) User-Item Cold-start, i.e., recommendation of new items for new users.

To construct the support and query sets of rated items (i.e., $\mathcal{S}_u^{\mathcal{R}}$ and $\mathcal{Q}_u^{\mathcal{R}}$), we follow previous work [16]. Specifically, we include only users who rated between 13 and 100 items. Among the items rated by a user u, we randomly select 10 items as the query set (i.e., $|\mathcal{Q}_u^{\mathcal{R}}| = 10$), and the remaining items are used as the support set (i.e., $\mathcal{S}_u^{\mathcal{R}}$). We will also study how the size of the support set affect the performance in Sect. 5.2. Furthermore, to construct the semantic contexts for the support and query sets, we take all meta-paths $\mathcal{P}$ starting with *User–Item* and ending with *Item* with a length up to 3, as discussed in Sect. 4.2. For each meta-path $p \in \mathcal{P}$, we construct the p-induced semantic contexts (i.e., $\mathcal{S}_u^p$ and $\mathcal{Q}_u^p$).

More detailed description of the datasets and our preprocessing are included in the supplement A.3.

Evaluation metrics. We adopt three widely-used evaluation protocols [16, 22, 31], namely, mean absolute error (MAE), root mean square error (RMSE), and normalized discounted cumulative gain at rank K (nDCG@K). Here we use $K = 5$.

Baselines. We compare our proposed MetaHIN with three categories of methods: **(1) Traditional methods**, including FM [20], NeuMF [10] 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 [5] and HERec [22]. 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 [29], as well as meta-learning-based MeteEmb [19] and MeLU [16]. Since they do not handle HINs either, we input the heterogeneous information as user or item features following the original papers. We follow [16] 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. More implementation details of baselines are included in the supplement A.4.

Environment and Parameter Settings. Experimental environment and hyper-parameter settings are discussed in the supplement A.5 and A.6, respectively. We will also study the impact of hyper-parameters in MetaHIN in Sect. 5.4.

Table 1: Statistics of the three datasets. The underlined node type refers to the target item type for recommendation.

Datasets	Node type	#Nodes	Edge type	#Edges	Sparsity
DBook	User (U)	10,592	UB	649,381	99.71%
	Book (B)	20,934	BA	20,934	
	Author (A)	10,544	UU	169,150	
MovieLens	User (U)	6,040	UM	1,000,209	95.73%
	Movie (M)	3,881	MA	15,398	
	Actor (A)	8,030	MD	4,210	
	Director (D)	2,186			
Yelp	User (U)	51,624	UB	1,301,869	92.63%
	Business (B)	34,199	BC	34,199	
	City (C)	510	BT	103,150	
	Category (T)	541			

5.2 Performance Comparison (RQ1)

In this section, we empirically compare MetaHIN to several state-of-the-art baselines, in three cold-start scenarios and the traditional non-cold start scenario. Table 2 demonstrates the performance comparison between all methods w.r.t. four recommendation scenarios. Figs. 3 and 4 further showcase performance analyses of MetaHIN.

Cold-start Scenarios. The first three parts of Table 2 presents 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

[1] https://book.douban.com
[2] https://grouplens.org/datasets/movielens/
[3] https://www.yelp.com/dataset/challenge

Table 2: 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**

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 2, 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 (see Table 1). 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.

Performance Analysis. First, we compare MetaHIN to the non-cold-start baselines (i.e., FM, NeuMF, GC-MC, mp2vec and HERec) in the four recommendation scenarios. We calculate the improvement of our MetaHIN over the baselines, and report the average improvement with standard deviations in Fig. 3. Generally, MetaHIN achieves more significant performance improvements in harder scenarios: non-cold-start $\prec$ UC $\sim$ IC $\prec$ UIC, which is intuitive.

Second, we vary the size constraint on the support set of rated items, i.e., $|\mathcal{S}_u^{\mathcal{R}}|$, between 5 and 90. In Fig. 4, we showcase the hardest scenario UIC, i.e., recommending new items for new users, among the traditional method NeuMF, the HIN-based method HERec, the meta-learning approach MeLU, and our MetaHIN. Overall, with a larger support set (i.e., more training data), all methods tend to achieve better MAE performances. However, when the support set becomes smaller, the decrease in performance is the smallest on MetaHIN among all methods. This implies that MetaHIN is robust and can cope with the harder situation with very limited data.

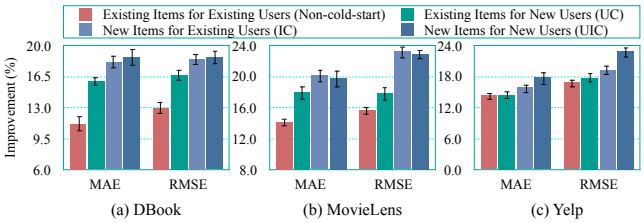


Figure 3: Performance improvement of MetaHIN.

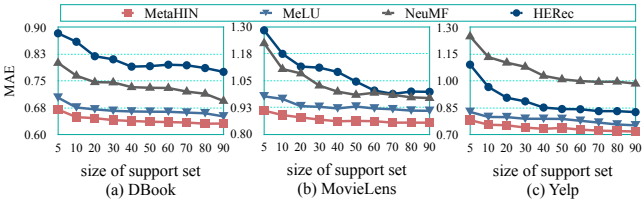


Figure 4: Impact of the size of support sets in UIC scenario.

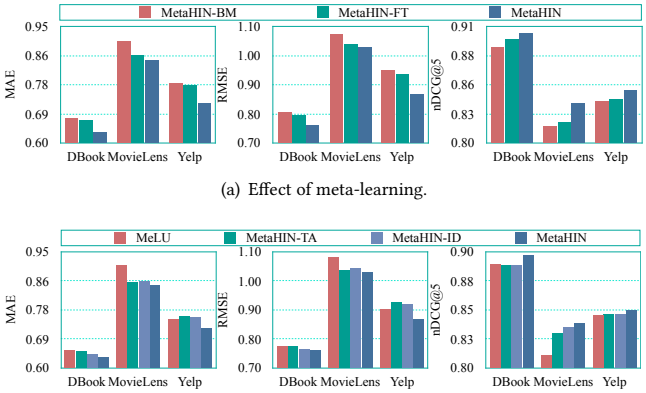
5.3 Model Analysis (RQ2)

Next, we investigate the underlying mechanism of MetaHIN: the effect of meta-learning on cold-start recommendation, as well as the impacts of semantic contexts and co-adaptation meta-learner.

Effect of Meta-learning. We attempt to understand how meta-learning facilitates the learning for cold-start recommendation. Towards this end, we conduct an ablation study, and consider two ablated variants of MetaHIN: (1) the base model (Sect. 4.3.1) without meta-learning, named **MetaHIN-BM**; (2) in order to warm up new users or items, following [4] we further *fine-tune* the base model with the union of support sets from the meta-testing data, which derives another variant named **MetaHIN-FT**. Since similar trends are observed in the three cold-start scenarios, here we only report the performance in the hardest scenario UIC.

As presented in Fig. 5(a), from the base model to fine-tuning we observe a performance gain on all datasets, although the gain can be marginal on DBook and Yelp. The reason might be that example ratings of new users and items are often scarce, which is inadequate for the fine-tuning process. On the other hand, the full model MetaHIN achieves significant improvements over fine-tuning, across all metrics and datasets. We attribute the improvement to the effective learning of prior knowledge by the meta-learning framework.

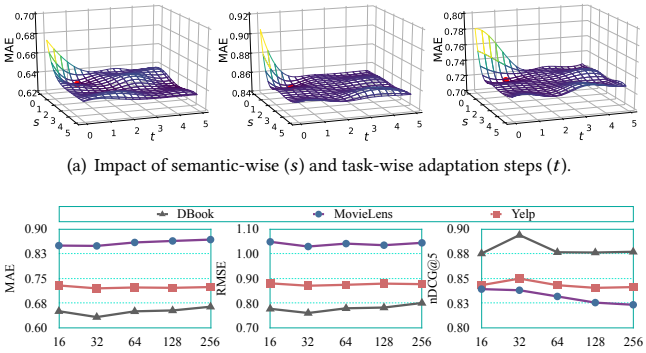
Effect of Semantic Contexts and Co-adaptation. As the semantic contexts and co-adaptation play pivotal roles in MetaHIN, we compare to two more ablated variants, namely **MetaHIN-TA** and **MetaHIN-ID**. MetaHIN-TA only includes the task-wise adaptation also used in existing meta-learning approaches, without semantic-wise adaptation. Thus, it aggregates all semantic contexts without differentiating their facets. On the other hand, MetaHIN-ID independently adopts task-wise adaptation in different semantic spaces (i.e., based on different meta-paths), without a global prior shared across semantic facets to encode the general knowledge of how to learn from contexts on a HIN. Finally, we also include the comparison to the baseline MeLU, which only utilize heterogeneous information as user or item features. As in the previous experiment, here we report the results in the UIC scenario in Fig. 5(b).



(a) Effect of meta-learning.

(b) Effect of semantic contexts and co-adaptation.

Figure 5: Analysis of MetaHIN using various ablated models.



(a) Impact of semantic-wise (s) and task-wise adaptation steps (t).

(b) Impact of user embedding dimensions.

Figure 6: Analysis of parameters in MetaHIN.

Overall, our MetaHIN is consistently superior to all variant models and MeLU. Compared with MeLU, MetaHIN-TA achieves better performance on three datasets, which demonstrates that HIN-based semantic contexts can alleviate the cold-start problem to some extent. However, MetaHIN-TA still underperforms MetaHIN, illustrating the limitation in simply bringing semantic contexts without fine-grained differentiation of the facets. The results also imply that the semantic-wise adaptation is a crucial component of MetaHIN. Similarly, the performance of MetaHIN-ID is often better than MeLU but worse than MetaHIN. The reason might be that MetaHIN-ID does not learn a global prior that connects different semantic facets. In particular, it learns a prior for each facet independently, and thus loses the general knowledge of learning with contexts on a HIN, which would be shared across all semantic facets.

5.4 Parameter Analysis (RQ3)

Lastly, we investigate the impact of parameters on the recommendation performance. As with previous analysis, we showcase the

UIC scenario since the observations are similar in other cold-start scenarios. Specifically, we investigate how the number of semantic- and task-wise adaptation steps affect the performance. Then, we analyze the influence of the dimension of user representations.

Number of Co-adaptation Steps. Let s and t be the number of semantic- and task-wise adaption steps, respectively. We plot the performance of MetaHIN under combinations of $0 \leq s \leq 5$ and $0 \leq t \leq 5$ in Fig. 6(a), in terms of MAE. We observe that the performance of MetaHIN is generally stable for different values of s and t, except when they are zero (i.e., no adaptation at all). In particular, our MetaHIN can adapt quickly with only one gradient update in both adaptions (i.e., $s = t = 1$), which makes it possible for efficient online recommendations.

Impact of Embedding Dimensions. We also explore how the dimensions of user embeddings d would affect the performance. The results are summarized in Fig. 6(b). We observe that our MetaHIN achieves optimal performance when the dimension is set to 32. Our model is generally stable around the optimal setting, indicating that MetaHIN is robust w.r.t. the embedding dimensions.

6 CONCLUSION

In this paper, we proposed a novel meta-learning approach called MetaHIN for cold-start recommendation on HINs, which alleviates the cold-start problem at both data and model levels. Specifically, we designed a semantic-enhanced task constructor to explore rich semantics on HINs in the meta-learning setting, and a co-adaptation meta-learner with semantic- and task-wise adaptions to cope with different semantic facets within each task. Extensive experiments on three datasets demonstrated that MetaHIN significantly outperforms state-of-the-art baselines in various scenarios.

ACKNOWLEDGMENTS

This research/project is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-RP-2018-001), the National Natural Science Foundation of China (No. U1936220, 61772082, 61806020, 61702296), the National Key Research and Development Program of China (2018YFB1402600), and the Tencent WeChat Rhino-Bird Focused Research Program. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore. Yuanfu Lu is also supported by 2019 Tencent Rhino-Bird Elite Training Program.

REFERENCES

[1] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263* (2017).
[2] Tianfeng Chai and Roland R Draxler. 2014. Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature. *Geoscientific model development* 7, 3 (2014), 1247–1250.
[3] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. 2019. Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In *RecSys*. 101–109.
[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). 4171–4186.
[5] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *SIGKDD*. 135–144.
[6] Zhengxiao Du, Xiaowei Wang, Hongxia Yang, Jingren Zhou, and Jie Tang. 2019. Sequential Scenario-Specific Meta Learner for Online Recommendation. In *SIGKDD*. 2895–2904.
[7] Yuan Fang, Wenqing Lin, Vincent W Zheng, Min Wu, Kevin Chen-Chuan Chang, and Xiao-Li Li. 2016. Semantic proximity search on graphs with metagraph-based learning. In *ICDE*. IEEE, 277–288.
[8] Yuan Fang, Wenqing Lin, Vincent W Zheng, Min Wu, Jiaqi Shi, Kevin Chang, and Xiaoli Li. 2019. Metagraph-based Learning on Heterogeneous Graphs. *IEEE TKDE* (2019).
[9] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*. 1126–1135.
[10] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*. 173–182.
[11] 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 *SIGKDD*. 1531–1540.
[12] Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. Conet: Collaborative cross networks for cross-domain recommendation. In *CIKM*. 667–676.
[13] SeongKu Kang, Junyoung Hwang, Dongha Lee, and Hwanjo Yu. 2019. Semi-Supervised Learning for Cross-Domain Recommendation to Cold-Start Users. In *CIKM*. 1563–1572.
[14] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.
[15] Duc-Trong Le, Yuan Fang, and Hady W Lauw. 2016. Modeling sequential preferences with dynamic user and context factors. In *ECML-PKDD*. Springer, 145–161.
[16] Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. 2019. MeLU: Meta-Learned User Preference Estimator for Cold-Start Recommendation. In *SIGKDD*. 1073–1082.
[17] Xiaopeng Li and James She. 2017. Collaborative variational autoencoder for recommender systems. In *SIGKDD*. 305–314.
[18] Tsendsuren Munkhdalai and Hong Yu. 2017. Meta networks. In *ICML*. 2554–2563.
[19] Feiyang Pan, Shuokai Li, Xiang Ao, Pingzhong Tang, and Qing He. 2019. Warm Up Cold-start Advertisements: Improving CTR Predictions via Learning to Learn ID Embeddings. In *SIGIR*. 695–704.
[20] Steffen Rendle, Zeno Gantner, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2011. Fast context-aware recommendations with factorization machines. In *SIGIR*. 635–644.
[21] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. 2016. Meta-learning with memory-augmented neural networks. In *ICML*. 1842–1850.
[22] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and Philip S. Yu. 2018. Heterogeneous Information Network Embedding for Recommendation. *IEEE TKDE* 31, 2 (2018), 357–370.
[23] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and S Yu Philip. 2017. A survey of heterogeneous information network analysis. *IEEE TKDE* 29, 1 (2017), 17–37.
[24] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *NeurIPS*. 4077–4087.
[25] 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. *PVLDB* 4, 11 (2011), 992–1003.
[26] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *CVPR*. 1199–1208.
[27] Manasi Vartak, Arvind Thiagarajan, Conrado Miranda, Jeshua Bratman, and Hugo Larochelle. 2017. A meta-learning perspective on cold-start recommendations for items. In *NeurIPS*. 6904–6914.
[28] Ricardo Vilalta and Youssef Drissi. 2002. A perspective view and survey of meta-learning. *Artificial intelligence review* 18, 2 (2002), 77–95.
[29] Maksims Volkovs, Guangwei Yu, and Tomi Poutanen. 2017. Dropoutnet: Addressing cold start in recommender systems. In *NeurIPS*. 4957–4966.
[30] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *SIGKDD*. 950–958.
[31] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *SIGIR*. 165–174.
[32] Huaxiu Yao, Ying Wei, Junzhou Huang, and Zhenhui Li. 2019. Hierarchically Structured Meta-learning. In *ICML*. 7045–7054.
[33] Xi Sheryl Zhang, Fengyi Tang, Hiroko H Dodge, Jiayu Zhou, and Fei Wang. 2019. Metapred: Meta-learning for clinical risk prediction with limited patient electronic health records. In *SIGKDD*. 2487–2495.
[34] 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.
[35] Yu Zhu, Jinghao Lin, Shibi He, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. 2019. Addressing the item cold-start problem by attribute-driven active learning. *IEEE TKDE* (2019).

A REPRODUCIBILITY SUPPLEMENT

We have omitted some details in the main paper and present them here to enhance the reproducibility. We first give additional implementation notes and the pseudo code of our training procedure. Then, we describe the datasets in more details including any processing we have done, as well as implementation of the baselines. Finally, we introduce the experimental environment and hyper-parameters settings. The code and datasets will be also publicly available after the review.

A.1 Additional Implementation Notes

A.1.1 Embedding Initialization. In the base model (Sect. 4.3.1), following [16], we initialize the representations of users and items with the concatenation of their feature embeddings. In detail, given a user u with m features, we represent each feature with a one-hot (e.g., gender) or multi-hot vector (e.g., hobbies). Then the i-th feature of the user is projected into a d-dimensional dense vector $\mathbf{p}_i \in \mathbb{R}^d$ by the embedding lookup as following:

$$\mathbf{p}_i = \mathbf{Z}^\top \mathbf{z}_i, \tag{16}$$

where $\mathbf{z}_i \in \mathbb{R}^{d_i}$ is the d_i-dimensional one-hot or multi-hot vector for the i-th feature, and $\mathbf{Z} \in \mathbb{R}^{d \times d_i}$ is the corresponding feature embedding matrix. Accounting for all features of the user, the initial embeddings of user u is given as:

$$\mathbf{e}_u = \mathbf{p}_1 \oplus \mathbf{p}_2 \oplus \cdots \oplus \mathbf{p}_m, \tag{17}$$

where $\mathbf{e}_u \in \mathbf{R}^{d_U}$, and $d_U = md$. Similarly, we can get the initial embeddings for items.

A.1.2 Transformation Function. In task-wise adaptation (Sect. 4.3.2), we apply a transformation function $\kappa(\cdot)$ to adapt ω to the semantic-wise initial parameters ω_p. The detailed form of $\kappa(\cdot)$ is:

$$\kappa(\mathbf{x}_u^{p\langle S\rangle}) = \text{sigmoid}(\mathbf{W}_\kappa \mathbf{x}_u^{p\langle S\rangle} + \mathbf{b}_\kappa), \tag{18}$$

where $\{\mathbf{W}_\kappa, \mathbf{b}_\kappa\}$ are learnable parameters of the function $\kappa(\cdot)$.

A.2 Pseudocode and Complexity Analysis

Pseudocode. The pseudocode of the training procedure for MetaHIN is detailed in Algorithm 1. The training of MetaHIN involves the initialization of user and item embeddings, co-adaptations and rating prediction. At the beginning (Line 1), we randomly initialize all learnable parameters (denoted as Θ) in our MetaHIN, including the meta-learner parameters $\theta = \{\phi, \omega\}$ and other global parameters (e.g., embedding lookup tables). Then, we construct the semantic-enhanced tasks for meta-training, and each task includes a support set and a query set (Line 2). In each training iteration, we sample a batch of tasks, i.e., user tasks from the meta-training set $\mathcal{T}^{\text{tr}}$ (Line 4). For each task $\mathcal{T}_u \in \mathcal{T}^{\text{tr}}$, we perform semantic- and task-wise adaptations on the support set in each semantic space (Line 5-12). Furthermore, we fuse the adaptations in all semantic spaces (Line 13). At last, we update all learnable parameters in MetaHIN (Line 15). The process stops when the model converges.

Complexity analysis. We nexr conduct a complexity analysis of our training procedure, which involves the initialization of user and item embeddings, co-adaptations and rating prediction. Thus, the time cost is $O(e \cdot |\mathcal{T}^{\text{tr}}| \cdot |\mathcal{P}| \cdot d \cdot d_I)$, where e is the number of epochs, $|\mathcal{T}^{\text{tr}}|$ and $|\mathcal{P}|$ are the number of meta-training tasks and meta-paths. d and d_I are the dimension of user embeddings and item initial embedding. As $|\mathcal{P}|$, d and d_I are usually small, the complexity of MetaHIN is linear with the number of tasks, i.e., $|\mathcal{T}^{\text{tr}}|$.

Algorithm 1 Meta-training of MetaHIN

Require: a HIN G; a set of meta-path $\mathcal{P}$; a set of meta-training tasks $\mathcal{T}^{\text{tr}}$; semantic and task-wise update steps: s and t; semantic-wise, task-wise and meta-learning rates: α, β and η;

1: Randomly initialize meta-learner parameters $\theta = \{\phi, \omega\}$ and other global parameters (e.g., embedding lookup tables)
2: Construct the semantic-enhanced tasks for meta-training $\mathcal{T}^{\text{tr}}$, each task $\mathcal{T}_u \in \mathcal{T}^{\text{tr}}$ consisting of a support $\mathcal{S}_u$ and a query set $\mathcal{Q}_u$, and $\mathcal{S}_u = \mathcal{S}_u^{\mathcal{R}} \cup \mathcal{S}_u^{\mathcal{P}}, \mathcal{Q}_u = \mathcal{Q}_u^{\mathcal{R}} \cup \mathcal{Q}_u^{\mathcal{P}}$
3: **while** not done **do**
4: Sample a batch of tasks $\mathcal{T}_u \in \mathcal{T}^{\text{tr}}$
5: **for all** task $\mathcal{T}_u$ w.r.t. user u **do**
6: **for all** meta-path $p \in \mathcal{P}$ **do**
7: Compute $\mathbf{x}_u^p$ using $\mathcal{S}_u^p \subset \mathcal{S}_u^{\mathcal{P}}$ by Eq. (9)
8: Evaluate $\mathcal{L}_{\mathcal{T}_u}(\omega, \mathbf{x}_u^p, \mathcal{S}_u^{\mathcal{R}})$
9: Semantic-wise adaptation by Eq. (11) with s updates
10: Evaluate $\mathcal{L}_{\mathcal{T}_u}(\omega^p, \mathbf{x}_u^{p\langle S\rangle}, \mathcal{S}_u^{\mathcal{R}})$
11: Task-wise adaptation by Eq. (13) with t updates
12: **end for**
13: Calculate ω_u and $\mathbf{x}_u$ with adaptation fusion as in Eq. (15)
14: **end for**
15: Update all learnable parameters Θ in MetaHIN as: $\Theta \leftarrow \Theta - \eta \nabla_\Theta \sum_{\mathcal{T}_u \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_u}(\omega_u, \mathbf{x}_u, \mathcal{Q}_u^{\mathcal{R}})$
16: **end while**

A.3 Data Processing

We construct a HIN with each dataset, as follows.

- **DBook** is a widely used book-rating dataset obtained from Douban [22], where users rate books from 1 to 5. The features of users and items are {*Location*} and {*Publish Year*, *Publisher*}. We divide books into existing and new items according to the publication year, with a ratio of approximately 8:2. Since we have no time information about users, we randomly selected eighty percent of users as the existing users and the rests as the new users.
- **MovieLens** is a stable benchmark published by GroupLens, where movies rated on a scale of 1 to 5 were released from 1919 to 2000. To introduce movie contents, we collect additional information from IMDB. Each user and item is associated with the feature set {*Age, Gender, ZipCode, Occupation*} and {*Rate, Year, Genre*}, respectively. According to the released year, we divide movies into existing items (released before 1998) and new items (released from 1998 to 2000), with a ratio of about 8:2. To define the new users in MovieLens dataset, we sort the first rating timestamp of users, and the most recent 20% of users are regarded to be new to MovieLens.
- **Yelp** is a widely used datasets for recommendation [31]. Wherein, each user and business is associated with the features {*Fans, Year jointed Yelp, Avg. Rating*} and {*Stars, PostalCode*}, respectively. The rating of a business ranges from 1 to 5. We take users who joined

Yelp before May 1, 2014 as existing users and the rests as news users. Similarly, we define the existing businesses and the new businesses based on when they were first rated. The ratio of new users (businesses) to existing users (businesses) is about 8:2.

A.4 Implementation of Baselines

We compare our proposed MetaHIN with three categories of methods: **(1) Traditional methods**, i.e., FM, NeuMF and GC-MC. **(2) HIN-based methods**, i.e., mp2vec and HERec. **(3) Cold-start methods**, i.e., DropoutNet, MeteEmb and MeLU.

- **FM** [20] is a feature-based baseline which is able to utilize various kinds of auxiliary information. In addition to the existing contents, here we also incorporate heterogeneous information of both datasets as additional input features.
- **NeuMF** [10] is a state-of-the-art neural network CF model which consists of a generalized matrix factorization component and a MLP component. Here we redefine the output unit as a linear layer for rating prediction.
- **GC-MC** [1] adopts GCN [14] to generate the embeddings for users and items and then predict the ratings of users to items.
- **mp2vec** [5] is a classic HIN embedding method, which samples meta-path based random walks for learning node embeddings.
- **HERec** [22] is a HIN-based model for rating prediction, which exploits heterogeneous information with matrix factorization.
- **DropoutNet** [29] is a neural network based model for cold-start problem, which explicitly train neural networks through dropout.
- **MeteEmb** [19] is a meta-learning based methods for CTR prediction, which generates initial embeddings for new ad IDs with ad contents and attributes. We employ it to generate new user/item embedding and then predict ratings.
- **MeLU** [16] applies MAML [9] to address cold-star problem, where only user-item interactions and features are considered.

A.5 Experiment Environment

All experiments are conducted on a Linux server with one GPU (GeForce RTX) and CPU (Intel Xeon W-2133), and its operating system is Ubuntu 16.04.1. We implement the proposed MetaHIN with deep learning library PyTorch. The Python and PyTorch versions are 3.6.9 and 1.3.1, respectively. The code and datasets will be publicly available after the review.

A.6 Parameter Settings

We adopt Adaptive Moment Estimation (Adam) to optimize our MetaHIN. For all dataset, we use a batch size of 32 and set the meta-learning rate to 0.0005 (i.e., $\eta = 0.0005$). We perform one step gradient descent update in both semantic-wise and task-wise adaptations. We set both the semantic-wise and task-wise learning rate to 0.005 (i.e., $\alpha = \beta = 0.005$) for DBook and MovieLens datasets, while $\alpha = \beta = 0.001$ for Yelp dataset. The maximum number of epochs are set to 50,100 and 20 for DBook, MovieLens and Yelp, respectively. Note that we also study the impact of hyper-parameters in MetaHIN in Sect. 5.4.

For baselines, we optimize their parameters empirically under the guidance of literature. Specifically, for FM, we set the rank of the factorization used for the second order interactions as 8 and utilize L2 regularization with coefficients 0.1. For NeuMF, we set the layers to (64, 32, 16, 8) and learning rate to 0.001. For GC-MC, the number of hidden units in the first and second layer are set to 500 and 75, respectively. The dropout fraction is set to 0.7. For mp2vec, we set the length of random walk, the number of walks and the size of windows to 40, 10 and 3, respectively. The tuning coefficients in HERec (i.e., α and β) are set to 1.0, and the random walk settings are same as in mp2vec. As suggested in the original paper, the learning rate in DropoutNet is set to 0.9 and the dropout rate is set to 0.5. We leverage the architectures of the embedding generator suggested by the authors in MetaEmb, and set the coefficient α to 0.1. For MeLU, we utilize the suggested two layers for decision-making layers with 64 nodes each, and set the local update step as 1. Other baseline parameters either adopt the original optimal settings or are optimized by the validation set. For all methods (including MetaHIN), the embedding dimensions is fixed to 32.