Improving Bandit Learning Via Heterogeneous Information Networks: Algorithms and Applications

XIAOYING ZHANG, The Chinese University of Hong Kong
HONG XIE, Chongqing University
JOHN C. S. LUI, The Chinese University of Hong Kong

Contextual bandit serves as an invaluable tool to balance the *exploration vs. exploitation* tradeoff in various applications such as online recommendation. In many applications, *heterogeneous information networks* (HINs) provide rich side information for contextual bandits, such as different types of attributes and relationships among users and items. In this article, we propose the first HIN-assisted contextual bandit framework, which utilizes a given HIN to assist contextual bandit learning. The proposed framework uses meta-paths in HIN to extract rich relations among users and items for the contextual bandit. The main challenge is how to leverage these relations, since users' preference over items, the target of our online learning, are closely related to users' preference over meta-paths. However, it is unknown which meta-path a user prefers more. Thus, both preferences are needed to be learned in an online fashion with exploration vs. exploitation tradeoff balanced. We propose the HIN-assisted upper confidence bound (HUCB) algorithm to address such a challenge. For each meta-path, the HUCB algorithm employs an independent base bandit algorithm to handle online item recommendations by leveraging the relationship captured in this meta-path. A bandit master is then employed to learn users' preference over meta-paths to dynamically combine base bandit algorithms with a balance of exploration vs. exploitation tradeoff. We theoretically prove that the HUCB algorithm can achieve similar performance compared with the optimal algorithm where each user is served according to his true preference over meta-paths (assuming the optimal algorithm knows the preference). Moreover, we prove that the HUCB algorithm benefits from leveraging HIN in achieving a smaller regret upper bound than the baseline algorithm without leveraging HIN. Experimental results on a synthetic dataset, as well as real datasets from LastFM and Yelp demonstrate the fast learning speed of the HUCB algorithm.

CCS Concepts: • **Information systems** → **Data mining**; **Data mining**; *Web applications*; • **Theory of computation** → **Online learning algorithms;**

Additional Key Words and Phrases: Contextual bandit, heterogeneous information network, online recommendation

The work of Hong Xie was supported in part National Nature Science Foundation of China (61902042), Chongqing Natural Science Foundation (cstc2020jcyj-msxmX0652) and Chongqing Talents: Exceptional Young Talents Project (cstc2021ycjh-bgzxm0195). The work of John C. S. Lui was supported in part by the GRF 14201819.
Authors' addresses: X. Zhang and J. C. S. Lui, Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, NT, Hong Kong SAR, China, 000000; emails: {xyzhang, cslui}@cse.cuhk.edu.hk; H. Xie (corresponding author), Chongqing Key Laboratory of Software Theory and Technology, Chongqing University, No. 174 Shazhengjie, Shapingba, Chongqing, China, 400044; email: xiehong2018@cqu.edu.cn.

1556-4681/2022/07-ART111 $15.00
https://doi.org/10.1145/3522590

ACM Reference format:
Xiaoying Zhang, Hong Xie, and John C. S. Lui. 2022. Improving Bandit Learning Via Heterogeneous Information Networks: Algorithms and Applications. *ACM Trans. Knowl. Discov. Data.* 16, 6, Article 111 (July 2022), 25 pages.
https://doi.org/10.1145/3522590

1 INTRODUCTION

Contextual bandit provides a principled online method to optimize the performance of various systems, e.g., online recommender systems, through learning from interactions with the user. For the contextual bandit based online recommendation algorithms [1, 8, 15, 27], each item is mapped as an arm in the contextual bandit, the observed information of an item with respect to a given user is mapped as its contextual vector, and the user's feedback to that item (e.g., click action) is mapped as a reward. The algorithm sequentially recommends items to the user, and acquires the user's feedback on the recommended item. The goal of the algorithm is to discover an item recommendation (arm selection) strategy on the fly, so that the user's feedback in the long run can be optimized, i.e., the cumulative reward is maximized. In general, the algorithm needs to make a tradeoff between exploitation (i.e., leveraging users' known preference) and exploration (i.e., revealing users' unknown preference). Contextual bandit algorithm is drawing increasing attention in online recommendation problems and one can refer to [8, 27] for a thorough survey of works in this research line.

In many applications, heterogeneous information, such as different types of attributes and relationships of users and items, is usually available. For example, On Yelp,[1] a social network exists since users can follow other users. The location-based businesses have categories, and users can write reviews to businesses as well. Such heterogeneous information captures rich relations among users and items, and thus has a high potential to improve bandit learning, since knowledge gathered about a user or an item can be used to assist the parameter learning of other users or items. However, previous contextual bandit algorithms either do not consider any relationships among users and arms [15, 28], or leverage only one single relationship, e.g., users' friendships [6, 19, 29, 30]. This article is the first to utilize the rich heterogeneous information to assist bandit learning.

This article proposes a new contextual bandit framework called HIN-assisted contextual bandit, where a **heterogeneous information network** (**HIN**) and a set of selected meta-paths in the HIN are given (please refer to Definition 1 for a precise description). Formally, the HIN [25] is a framework to represent many types of entities and relations in a unified manner. For example, Figure 1 shows a simple example of HIN built from Yelp, which contains relations between users, categorical and geographical attributes of businesses (i.e., arms), and so on. A meta-path is a path over node types of HIN and each meta-path defines a new composite relation on HIN (please refer to Definition 2 for a precise description). For example, the meta-path "user→business→category→business" in Figure 1 depicts how users prefer businesses with similar categories. In a HIN-assisted contextual bandit, the objective is still to learn an arm selection (or item recommendation) strategy, by utilizing the given HIN and selected relations, so that users' overall satisfaction (cumulative reward) can be maximized.

The main challenge of designing arm selection strategy while utilizing given relations is that users' preference over relations (or meta-paths) as well as over items are correlated, and both these preferences are unknown. In other words, we need to learn both preferences in an online

[1] https://www.yelp.com/.

manner while balancing the exploration vs. exploitation tradeoff. To address the challenge, we design the HUCB algorithm. In the HUCB algorithm, the user's preference over arms under different meta-paths are learned online by a group of independent base bandit algorithms, which handle the exploitation vs. exploration tradeoff. Meanwhile, the user's preference over different meta-paths are updated dynamically based on the performance of base bandit algorithms. Namely, if one base bandit algorithm can predict the user's preference over arms more accurately in previous rounds, the user's preference on this meta-path will be enlarged. However, inferring the user's preference over meta-paths solely based on the historical performance of base bandit algorithms will lead to suboptimal solutions, i.e., trapped by suboptimal base bandits. For example, a base bandit algorithm, which is exploratory initially (i.e., bad performance) but can excel later or may not even be selected. Thus, we also develop a bandit master to dynamically ensemble base bandit algorithms while balancing the explore/exploit tradeoff in learning user's preference over meta-paths. We theoretically prove that the **HIN-assisted upper condence bound** (**HUCB**) algorithm achieves the same-order of upper regret bound as the optimal algorithm where each user is served according to his true preference over meta-paths (assuming the optimal algorithm knows the true preference). Moreover, by leveraging HIN, we prove that the HUCB algorithm can achieve a smaller regret upper bound (i.e., improved performance) than the baseline algorithm without leveraging HIN. Experimental results on synthetic datasets, and real datasets from LastFM and Yelp, show that the HUCB algorithm significantly outperforms the baseline algorithms.

In summary, our contributions are as follows:

— We formulate the first HIN-assisted contextual bandit to leverage rich relations on a given heterogeneous information network (Section 2).
— We design the HUCB algorithm for HIN-assisted contextual bandit, and theoretically show the benefits of leveraging HIN, which has similar performance as the optimal algorithm where each user is served according to his true preference over meta-paths (assuming the optimal algorithm knows this information). (Section 3).
— We conduct extensive experiments on synthetic datasets and real datasets from Yelp and LastFM, and demonstrate the fast learning speed of the HUCB algorithm (Sections 4 and 5).

The reminder of this article is organized as follows. Section 2 presents the HIN-assisted contextual bandit model and problem formulation. Section 3 presents the design and analysis of HUCB algorithm, which solves the HIN-assisted contextual bandit problem. Section 4 presents the experimental results on synthetic data. Section 5 presents the experimental results on real-world data. Section 6 presents technical proofs to theorems and lemmas. Section 7 presents the related work and Section 8 concludes the article.

2 PROBLEM FORMULATION

In this section, we first briefly introduce the traditional contextual bandit. Then, we generalize it to leverage heterogeneous information represented by a *heterogeneous information network*.

2.1 Contextual Bandit

In contextual bandit, given a finite set of $N \in \mathbb{N}_+$ arms denoted by $\mathcal{A}$, an agent aims at maximizing cumulative reward in $T \in \mathbb{N}_+$ decision rounds through interacting with users. In recommendation application, the agent can be mapped as the recommender system, and each arm $a \in \mathcal{A}$ can be mapped as an item. At each round $t = 1, \ldots, T \in \mathbb{N}_+$, a subset of arms $\mathcal{A}_t \subseteq \mathcal{A}$ is shown to the agent. Each arm $a \in \mathcal{A}_t$ is associated with a d-dimensional contextual vector $\mathbf{x}_{a,t} \in \mathbb{R}^d$, which describes the observable information of arm a and a given user u at round t, where $d \in \mathbb{N}_+$. Based on the contextual information $\{\mathbf{x}_{a,t}\}_{a \in \mathcal{A}_t}$, the selected arms and received rewards in previous

rounds, the agent chooses an arm a_t from $\mathcal{A}_t$, shows the arm a_t to the user u, and receives a new reward or feedback denoted by $r_{u,a_t,t} \in \mathcal{F}$. For example, $\mathcal{F} = \{0, 1\}$ models a binary reward, while $\mathcal{F} = \mathbb{R}$ models a continuous reward.

The goal of the agent is to maximize the expected cumulative reward in T rounds. Let $\sum_{t=1}^{T} \mathbb{E}[r_{u,a_t^*,t}]$ denote the maximum expected cumulative reward in T rounds, where $a_t^* \in \mathcal{A}_t$ is the optimal arm at round t for user u, i.e.,

$$\mathbb{E}[r_{u,a_t^*,t}] \geq \mathbb{E}[r_{u,a,t}], \qquad \forall a \in \mathcal{A}_t.$$

The goal of contextual bandit is formally defined as minimizing the cumulative regret in T rounds:

$$R(T) \triangleq \sum_{t=1}^{T} \left(\mathbb{E}[r_{u,a_t^*,t}] - \mathbb{E}[r_{u,a_t,t}] \right). \tag{1}$$

A smaller regret $R(T)$ implies that the cumulative reward is close to the optimal cumulative reward. The agent needs to make a tradeoff between exploitation (i.e., choose the best arm estimated from the reward history) and exploration (i.e., enquire arms to reveal users' unknown preference).

In the standard contextual bandit problem, the reward $r_{u,a_t,t}$ is a function related to the contextual vector $\mathbf{x}_{a_t,t}$ and an unknown parameter vector θ_u. The parameter vector θ_u can be mapped as user u's preference, and it is what the agent wants to learn. Let ϵ_t denote a random variable representing the random noise in the reward. The noise captures uncertainty in reward and it can be caused by human factors such as bias. The LinUCB algorithm [15] considers a reward function

$$r_{u,a_t,t} = \mathbf{x}_{a_t,t}^T \theta_u + \epsilon_t,$$

while hLinUCB algorithm [28] considers a reward function

$$r_{u,a_t,t} = (\mathbf{x}_{a_t,t}, \mathbf{v}_{a_t})^T \theta_u + \epsilon_t,$$

where $\mathbf{v}_{a_t} \in \mathbb{R}^l$ denotes the unknown hidden features associated with arm a_t that the agent also needs to learn.

2.2 HIN-assisted Contextual Bandit

Previous works estimate $\{\theta_u\}$ (and $\{\mathbf{v}_a\}$ if applicable) either independently for each user (for each arm) [15, 28], or considering a single relationship, for example, users' friendship [29]. However, in many cases, additional information regarding to users and arms, e.g., users' friendships, categorical and geographical attributes of arms, can be obtained. Such information is beneficial to bandit learning, as they reveal the *dependency* between users and arms. Thus, the knowledge gathered about a user or an arm can be leveraged to improve the parameter learning of other users or arms. The heterogeneous information network, whose nodes are of different types and links among nodes represent different relations, has been shown as an effective way to represent all these information in a unified framework [22, 31]. Moreover, different types of relations among users and arms can be obtained in the heterogeneous information network and we aim at leveraging those relations to assist bandit learning.

Heterogeneous information network. We first give a formal definition of heterogeneous information network.

Definition 1 (HIN). A heterogeneous information network is defined as a directed graph $G = (\mathcal{V}, \mathcal{E}, \mathcal{K}, \mathcal{R}, \phi, \psi)$, where each element of the graph is defined as follows:

- $\mathcal{V}$ denotes a finite set of $V \in \mathbb{N}_+$ nodes representing users, arms, and so on.;
- $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes a finite set of directed edges, with $[v_1, v_2] \in \mathcal{E}$ indicating a directed edge from $v_1 \in \mathcal{V}$ to $v_2 \in \mathcal{V}$;

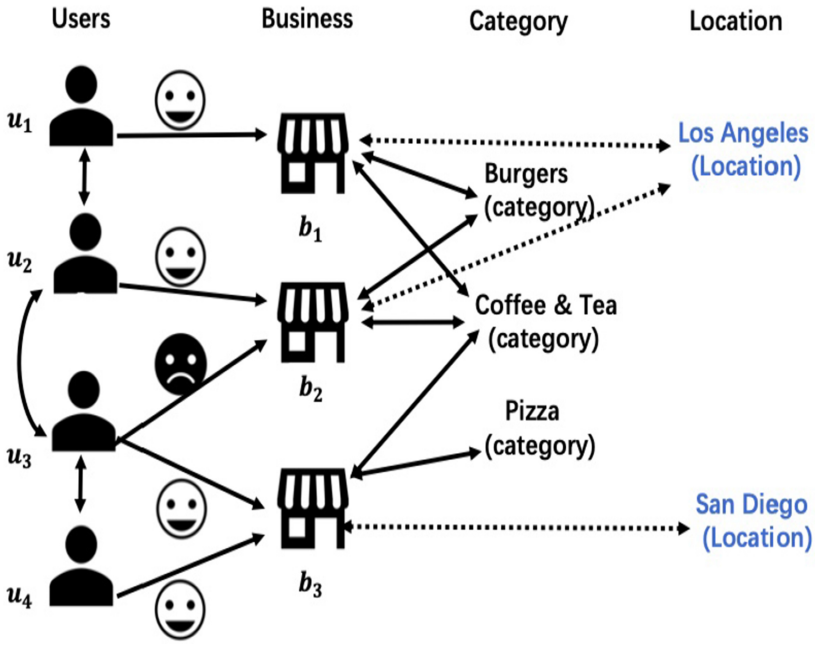


Fig. 1. A single example is HIN from Yelp.

— $\mathcal{K}$ denotes a set of all possible types associated with nodes;
— $\mathcal{R}$ denotes a set of all possible types associated with edges;
— $\phi : \mathcal{V} \to \mathcal{K}$ denotes a node type mapping function, which prescribes a type $\phi(v)$ for each node $v \in \mathcal{V}$;
— $\psi : \mathcal{E} \to \mathcal{R}$ denotes an edge type mapping function, which prescribes a type $\psi([v_1, v_2])$ for each edge $[v_1, v_2] \in \mathcal{E}$.

Figure 1 shows an example of a heterogeneous information network built on Yelp. It contains four types of nodes, i.e.,

$$\mathcal{K} = \{\text{“user”, “business”, “location”,“category”}\},$$

and four types of links, i.e.,

$$\mathcal{R} = \{\text{ “user}\to\text{business”, “business}\leftrightarrow\text{location”, “business}\leftrightarrow\text{category”, “user}\leftrightarrow\text{user” }\}.$$

One can observe that $\phi(u_1)$=“user”, $\phi(b_1)$=“business” and $\psi(u_1 \to b_1)$ = “user→business”.

In this article, we emphasize that the HIN is allowed to be time-varying. Denote the HIN at round t as

$$G_t = (\mathcal{V}_t, \mathcal{E}_t, \mathcal{K}, \mathcal{R}, \phi, \psi).$$

Here, the node-set $\mathcal{V}_t$ and edge set $\mathcal{E}_t$ may vary over round t, capturing that outdated items may be deleted or the new item may be added. We consider a class of HIN G_t satisfying that the type of each edge is uniquely determined by the corresponding starting node type and ending node type. Formally, the following holds:

$$[\phi(v_1), \phi(v_2)] = [\phi(v_3), \phi(v_4)] \Rightarrow \psi(v_1 \to v_2) = \psi(v_3 \to v_4), \tag{2}$$

where $v_1, v_2, v_3, v_4 \in \mathcal{V}_t$. For example, Figure 1 satisfies above property and $\psi(u_1 \to b_1) = \psi(u_3 \to b_2)$ = “user→business”. To simplify the presentation, define a relation function $R : \mathcal{K} \times \mathcal{K} \to \mathcal{R}$

to capture the property of the Equation (2), which satisfies that

$$\psi(v_1 \to v_2) = R(\phi(v_1), \phi(v_2)).$$

To extract rich relations from HIN, one can use the meta-path technique [22, 31, 33]. A meta-path summarizes the semantics of a path in a HIN, which can be utilized to quantify similarity between arms and users. Formally, a meta-path is defined as follows.

Definition 2 (Meta-path). A meta-path of length $m \in \mathbb{N}_+$ is defined as a path over node types, and is denoted by

$$p \triangleq (K_0 \to K_1 \to \cdots \to K_m),$$

where $K_0, K_1, \ldots, K_m \in \mathcal{K}$ denote $m+1$ node types. This meta-path defines a new composite relation $R(K_0, K_1)R(K_1, K_2) \ldots R(K_{m-1}, K_m)$ between node type K_0 and K_m.

For example, "user→business→category→business" is a meta-path in Figure 1. It characterizes users' preferences on the business with similar categories. The semantics of a path $(v_0 \to v_1 \to \cdots \to v_m)$ in a HIN, where $v_0, v_1, \ldots, v_m \in \mathcal{V}_t$, can be summarized by a meta-path $p = (\phi(v_0) \to \phi(v_1) \to \cdots \to \phi(v_m))$. For example, the semantics of the path "$u_1 \to b_1 \to$ Coffee&Tea $\to b_2$" and the path "$u_2 \to b_2 \to$ Coffee&Tea $\to b_3$" are summarized by the meta-path "user→business→category→business". The meta-path carries rich similarity information among users or items (details are in the next section), which can be utilized to speed up the bandit learning. We next present our problem formulation so to make this point clearer.

Problem Formulation. In the HIN-assisted contextual bandit, the agent learns to maximize the cumulative reward in T rounds through interacting with users. In each round t, besides a finite set of arms denoted by $\mathcal{A}_t$ and their associated contextual vectors $\{\mathbf{x}_{a,t} | a \in \mathcal{A}_t\}$, a heterogeneous information network G_t, and a finite set of selected meta-paths denoted by $\mathcal{P}$ are given. Without loss of generality, we normalize the contextual vector such that $\|\mathbf{x}_{a,t}\|_2 = 1$. Based on the interactions in the previous $t-1$ rounds, i.e., $\{(a_\tau, r_{u,a_\tau,\tau})\}_{\tau=1}^{t-1}$, and the relations defined by the given meta-paths $\mathcal{P}$ in the HIN G_t, the agent selects an arm $a_t \in \mathcal{A}_t$, receiving the reward $r_{u,a_t,t}$. The problem in HIN-assisted contextual bandit is to find an arm selection (or item recommendation) strategy that can effectively leverage the given relations, so that the cumulative regret in Equation (1) is minimized.

Remark. The environment of the HIN-assisted contextual bandit is static, i.e., the reference vector θ_u is fixed and it does not evolve with round t.

3 ALGORITHM & THEORETICAL ANALYSIS

In this section, we propose the HUCB algorithm for HIN-assisted contextual bandit. We first present the learning of users' preference over arms/items under each meta-path via the independent base bandit algorithm, then we describe how to ensemble these base bandit algorithms via the bandit master. Finally, we gives a rigorous proof on the upper regret bound of the HUCB algorithm.

3.1 Base Bandit Algorithm under Meta-path p

We first quantify similarities among users and items under the *user-centric meta-path* and *arm-centric meta-path*. We present two base bandit algorithms to utilize these similarities for learning. We would like to remark that in this article we develop the base bandit algorithm by extending the hLinUCB algorithm [28], one can also easily extend other algorithms [18].

Similarities induced by a meta-path. This article mainly focuses on two classes of meta-paths characterized by the format "user→ $\cdots$ →user→arm type" or "user→arm type→ $\cdots$ →arm type". Formally:

— *User-centric meta-path:*

$$p = (K_0 \to K_1 \to \cdots \to K_{m-1} \to K_m),$$

where $K_0 = K_1 = \cdots = K_{m-1}$ ="user" and K_m ="arm type".

— *Arm-centric meta-path:*

$$p = (K_0 \to K_1 \to \cdots \to K_m),$$

where K_0 ="user" and $K_1 = \cdots = K_m$ ="arm type".

For example, in Yelp, each business corresponds to an arm, and in Figure 1, "user→user→business" is a user-centric meta-path, while "user→business→category→business" is an arm-centric meta-path. The intuition of using user-centric and arm-centric meta-paths are to find arms that similar users like, and to diffuse the observed users' preference to similar arms respectively. Similar users are identified through friendship link, while similar arms are usually identified through attributes of arms (or arm types).

Given a user-centric (or arm-centric) meta-path, we apply the commonly-used approach, i.e., computing commuting matrices [25], to quantify similarities among users (or among arms). Let us first consider an arm-centric meta-path $p = (K_0 \to K_1 \to \cdots \to K_m)$ with K_0 ="user" and $K_1 = K_m$ ="arm type". Let

$$\mathbf{W}_t = [\mathrm{W}_t(v_1, v_2) | v_1 \in \mathcal{V}_t, v_2 \in \mathcal{V}_t]$$

denote the adjacency matrix of the HIN G_t:

$$\mathrm{W}_t(v_1, v_2) = \begin{cases} 1, & \text{if } (v_1, v_2) \in \mathcal{E}_t, \\ 0, & \text{otherwise.} \end{cases}$$

Let $\mathbf{W}_t|_{K_i, K_j}$ denote the adjacency matrix with restriction to row nodes having type K_i and column nodes having type K_j, formally

$$\mathbf{W}_t|_{K_i, K_j} = [\mathrm{W}_t(v_1, v_2); v_1 \in \mathcal{V}_{K_i}, v_2 \in \mathcal{V}_{K_j}],$$

where $\mathcal{V}_{K_i}$ and $\mathcal{V}_{K_j}$ denote the set of nodes having type K_i and K_j respectively. For example, if $K_i = K_j$ = "user", then $\mathbf{W}_t|_{K_i, K_j}$ represents the adjacency matrix among users. Then, we can compute the commuting matrix associated with the sub-meta-path $(K_1 \to \cdots \to K_m)$ of the meta-path $p = (K_0 \to K_1 \to \cdots \to K_m)$ by

$$\mathbf{C}_{p,t} = \mathbf{W}_t|_{K_1, K_2} \times \mathbf{W}_t|_{K_2, K_3} \times \cdots \times \mathbf{W}_t|_{K_{m-1}, K_m}.$$

For any $v_1 \in \mathcal{V}_{K_1}, v_2 \in \mathcal{V}_{K_m}$, $\mathrm{C}_{p,t}(v_1, v_2)$ represents the number of path instances in HIN G_t between v_1 and v_2 along the sub-meta-path $(K_1 \to \cdots \to K_m)$. For example, in Figure 1, if meta-path p = "user→business→category→business", $\mathrm{C}_{p,t}(b_1, b_2) = 2$ and $\mathrm{C}_{p,t}(b_2, b_3) = 1$. Then the similarity matrix between K_1 and K_m under meta-path p at time t, denoted as $\mathbf{S}_{p,t}$, can be calculated by

$$\mathrm{S}_{p,t}(v_1, v_2) = \frac{2\mathrm{C}_{p,t}(v_1, v_2)}{\mathrm{C}_{p,t}(v_1, v_1) + \mathrm{C}_{p,t}(v_2, v_2)}, v_1 \in \mathcal{V}_{K_1}, v_2 \in \mathcal{V}_{K_m}.$$

Take the meta-path "user→business→category→business" in Figure 1 as an example. One can have

$$\mathrm{S}_{p,t}(b_1, b_2) = \frac{2 * 2}{2 + 2} = 1, \quad \mathrm{S}_{p,t}(b_2, b_3) = \frac{2 * 1}{2 + 2} = 0.5.$$

This implies that business b_2 is more similar to business b_1 than business b_3 since business b_1 and b_2 have the same set of categories. We further normalize $\mathrm{S}_{p,t}$ so that

$$\sum_{v_2 \in \mathcal{V}_{Km}} \mathrm{S}_{p,t}(v_1, v_2) = 1,$$

holds for any $v_1 \in \mathcal{V}_{K_1}$. For the user-centric meta-path p, the similarities between users can be derived similarly and we denote it by $\widetilde{\mathrm{S}}_{p,t}$. The similarity metric $\widetilde{\mathrm{S}}_{p,t}$, follows the PathSim algorithm [25].

Base bandit algorithm for user-centric meta-path. Wang et al. [29] proposed the factorUCB algorithm to leverage users' friendships. Thus, for user-centric meta-paths, the factorUCB algorithm can be directly taken as the base bandit algorithm under meta-path p, where similarities among users are calculated as $\widetilde{\mathrm{S}}_{p,t}$.

Base bandit algorithm for arm-centric meta-path. We now develop a base bandit algorithm for arm-centric meta-paths. We generalize the reward model of hLinUCB [28] to consider similarity network of arms. Specifically, given the arms' similarity matrix $\mathrm{S}_{p,t}$, the reward of arm a with respect to user u at time t is modeled by

$$r^{p}_{u,a,t} = \beta(\mathbf{x}_{a,t}, \mathbf{v}_{p,a})^T \theta_{p,u} + (1-\beta) \sum_{j \in \mathcal{A}} \mathrm{S}_{p,t}(a,j)[\mathbf{x}_{j,t}, \mathbf{v}_{p,j}]^T \theta_{p,u}, \tag{3}$$

where β is a pre-set parameter controlling the relative importance of arm a's own reward and influence from similar arms; $\mathbf{v}_{p,a} \in \mathbb{R}^l$ and $\theta_{p,u} \in \mathbb{R}^{d+l}$ are arm a's hidden feature vector and user u's preference vector in the base bandit algorithm of meta-path p respectively. To make the presentation more compact, we define an extension of the notation $\mathrm{S}_{p,t}(a,j)$ as follows:

$$\mathrm{M}^{a,j}_{p,t} = (1-\beta)\mathrm{S}_{p,t}(a,j) + \beta \mathbb{1}_{\{a=j\}}.$$

Then $\{\mathbf{v}_{p,a}\}$ and $\theta_{p,u}$ can be inferred through the following optimization problem:

$$\min_{\theta_{p,u}, \{\mathbf{v}_{p,a}\}} \sum_{\tau=1}^{t} \left(\sum_{j \in \mathcal{A}} \mathrm{M}^{a_\tau, j}_{p,t} (\mathbf{x}_{j,\tau}, \mathbf{v}_{p,j})^T \theta_{p,u} - r_{u,a_\tau,\tau} \right)^2 + \lambda_1 \|\theta_{p,u}\|_2^2 + \lambda_2 \sum_{j \in \mathcal{A}} \|\mathbf{v}_{p,j}\|_2^2, \tag{4}$$

where $\lambda_1 \in \mathbb{R}_+$ and $\lambda_2 \in \mathbb{R}_+$ are the tradeoff parameters for the L2 regularization.

We use the coordinate decent algorithm to estimate $\theta_{p,u}$ and $\{\mathbf{v}_{p,a}\}$, then their closed-form formula at time t can be derived as

$$\hat{\theta}_{p,u,t} = \mathbf{A}^{-1}_{p,u,t} \mathbf{b}_{p,u,t}, \quad \hat{\mathbf{v}}_{p,a,t} = \mathbf{C}^{-1}_{p,a,t} \mathbf{d}_{p,a,t},$$

where $\mathbf{A}_{p,u,t}, \mathbf{b}_{p,u,t}, \mathbf{C}_{p,a,t}$ and $\mathbf{d}_{p,a,t}$ are derived as

$$\begin{aligned}
\mathbf{A}_{p,u,t} &= \lambda_1 \mathbf{I}_1 + \sum_{\tau=1}^{t-1} \left(\sum_j \mathrm{M}^{a_\tau, j}_{p,\tau} (\mathbf{x}_{j,\tau}, \hat{\mathbf{v}}_{p,j,\tau}) \right) \times \left(\sum_j \mathrm{M}^{a_\tau, j}_{p,\tau} (\mathbf{x}_{j,\tau}, \hat{\mathbf{v}}_{p,j,\tau}) \right)^T, \\
\mathbf{b}_{p,u,t} &= \sum_{\tau=1}^{t-1} \left(\sum_j \mathrm{M}^{a_\tau, j}_{p,\tau} (\mathbf{x}_{j,\tau}, \hat{\mathbf{v}}_{p,j,\tau}) \right) r_{u,a_\tau,\tau}, \\
\mathbf{C}_{p,a,t} &= \lambda_2 \mathbf{I}_2 + \sum_{\tau=1}^{t-1} (\mathrm{M}^{a_\tau, a}_{p,\tau})^2 \hat{\theta}^{v}_{p,u,\tau} (\hat{\theta}^{v}_{p,u,\tau})^T, \\
\mathbf{d}_{p,a,t} &= \sum_{\tau=1}^{t-1} \mathrm{M}^{a_\tau, a}_{p,\tau} \hat{\theta}^{v}_{p,u,\tau} \left(r_{u,a_\tau,\tau} - \sum_j \mathrm{M}^{a_\tau, j}_{p,\tau} \mathbf{x}^T_{j,\tau} \hat{\theta}^{x}_{p,u,\tau}, - \sum_{j \neq a} \mathrm{M}^{a_\tau, j}_{p,\tau} \hat{\mathbf{v}}_{p,j,\tau} \hat{\theta}^{v}_{p,u,\tau} \right).
\end{aligned} \tag{5}$$

In the above formulas, $\mathbf{I}_1$ and $\mathbf{I}_2$ are two identity matrices with dimensions of $(d+l)\times(d+l)$ and $l\times l$ respectively. We denote $\hat{\theta}_{p,u,t} = (\hat{\theta}^x_{p,u,t}, \hat{\theta}^v_{p,u,t})$, where $\hat{\theta}^x_{p,u,t} \in \mathbb{R}^d$ and $\hat{\theta}^v_{p,u,t} \in \mathbb{R}^l$ are the preference parameter regarding to the observed contextual features and hidden features respectively. Projection of the estimated $\hat{\theta}_{p,u,t}$ and $\hat{\mathbf{v}}_{p,a,t}$ is necessary to satisfy the constraint on their L2 norms, i.e., $\|\theta_{p,u}\|_2 \le S$ and $\|(\mathbf{x}_{a,t}, \mathbf{v}_{p,a})\|_2 \le L$.

To balance the exploitation-exploration tradeoff, we adopt the widely-used **upper confidence bound** (**UCB**) strategy for arm selection. In UCB algorithm, at time t, the agent selects arm with the largest upper confidence bound value, which is the sum of the estimated reward $\hat{r}^p_{u,a,t}$ and its confidence interval. The confidence interval of $\hat{r}^p_{u,a,t}$ measures the uncertainty of current estimation at round t, and it is related to estimation uncertainties of users' preference vector (i.e., $\|\hat{\theta}_{p,u,t} - \theta_{p,u,*}\|$) and arms' latent features (i.e., $\|\hat{\mathbf{v}}_{p,a,t} - \mathbf{v}_{p,a,*}\|$). Here $\theta_{p,u,*}$ and $\mathbf{v}_{p,a,*}$ are the ground-truth preference vector of user u and the ground-truth latent feature vector of arm a respectively. Based on Equation (5), we can derive confidence intervals of $\hat{\theta}_{p,u,t}$ and $\hat{\mathbf{v}}_{p,a,t}$ as shown in Lemma 3.

Let α^θ_t and α^v_t denote the upper bound of $\|\hat{\theta}_{p,u,t} - \theta_{p,u,*}\|_{\mathrm{A}_{p,u,t}}$ and $\|\hat{\mathbf{v}}_{p,a,t} - \mathbf{v}_{p,a,*}\|_{\mathrm{C}_{p,a,t}}$ respectively, then the base bandit algorithm under meta-path p selects arm a_t as follows:

$$a_t = \arg\max_{a\in\mathcal{A}_t}\left(\sum_{j\in\mathcal{A}} \mathrm{M}^{a,j}_{p,t}(\mathbf{x}_{j,t}, \hat{\mathbf{v}}_{p,j,t})^T\hat{\theta}_{p,u,t} + \alpha^\theta_t\left\|\sum_{j\in\mathcal{A}}\mathrm{M}^{a,j}_{p,t}(\mathbf{x}_{j,t}, \hat{\mathbf{v}}_{p,j,t})\right\|_{\mathrm{A}^{-1}_{p,u,t}} + \alpha^v_t\sum_{j\in\mathcal{A}}\mathrm{M}^{a,j}_{p,t}\|\hat{\theta}^v_{p,u,t}\|_{\mathrm{C}^{-1}_{p,j,t}}\right). \tag{6}$$

The first term in Equation (6) is the predicted reward of arm a to user u at time t (i.e., $\hat{r}^p_{u,a,t}$), while the second and third terms measure the estimation uncertainty of $\hat{\theta}_{p,u,t}$ and $\hat{\mathbf{v}}_{p,a,t}$. With more observations, the last two terms will be reduced, causing fewer explorations. In the following lemma, we derive the upper bound for $\|\hat{\theta}_{p,u,t} - \theta_{p,u,*}\|_{\mathrm{A}_{p,u,t}}$ and $\|\hat{\mathbf{v}}_{p,a,t} - \mathbf{v}_{p,a,*}\|_{\mathrm{C}_{p,a,t}}$.

Lemma 3. *Assume ϵ_t is conditionally 1-sub-Gaussian. If the Hessian matrix of the objective function defined in Equation (4) is positive definite at the optimizer $\theta_{p,u,*}$ and $\mathbf{v}_{p,a,*}$, with proper initialization, for any $\mu_1 \ge 0$, $\mu_2 \ge 0$, $\exists 0 \le q_1, q_2 \le 1$, with probability at least $1-\sigma$ it holds that*

$$\|\hat{\theta}_{p,u,t} - \theta_{p,u,*}\|_{\mathrm{A}_{p,u,t}} \le (d+l)\log\left(\left(1 + \frac{\sum_{\tau=1}^{t-1}\|\sum_{j\in\mathcal{A}} M^{a_\tau,j}_{p,\tau}(\mathbf{x}_{j,\tau}, \hat{\mathbf{v}}_{p,j,\tau})\|_2^2}{\lambda_1(d+l)}\right)/\sigma\right) + \sqrt{\lambda_1}S + \frac{L^2S}{\sqrt{\lambda_1}}\frac{(q_1+\mu_1)(1-(q_1+\mu_1)^t)}{1-(q_1+\mu_1)},$$

and

$$\|\hat{\mathbf{v}}_{p,a,t} - \mathbf{v}_{p,a,*}\|_{\mathrm{C}_{p,a,t}} \le \sqrt{\lambda_2}L + \frac{S^2L}{\sqrt{\lambda_2}}\left(\frac{(q_1+\mu_1)(1-(q_1+\mu_1)^t)}{1-(q_1+\mu_1)} + \frac{(q_2+\mu_2)(1-(q_2+\mu_2)^t)}{1-(q_2+\mu_2)}\right) + l\log\left(\left(1 + \frac{\sum_{\tau=1}^{t-1}(M^{a_\tau,a}_{p,\tau})^2S^2}{\lambda_2 l}\right)/\sigma\right).$$

3.2 A Dynamic Ensemble of Base Bandit Algorithms

Recall that we are given a set of meta-paths $\mathcal{P}$. For each meta-path $p \in \mathcal{P}$, a base bandit algorithm can be developed as described in Section 3.1 to leverage the relation under meta-path p.

Next, we consider how to learn users' preferences over different meta-paths so as to ensemble these base bandit algorithms. Observe that the user's preference to one specific meta-path is closely related to the performance of the base bandit algorithm under that meta-path. For example, if the user prefers items that his friends like, then the base bandit algorithm under the meta-path "user→user→item" may have better performance; while for the user who enjoys items of the same category as that they consumed, the base bandit algorithm under the meta-path "user→item→category→item" may be more effective. Thus we try to learn the user's preference over meta-paths based on the performance of base bandit algorithms.

Note that one cannot infer users' preference over meta-paths solely based on the historical performance of base bandit algorithms, since it will lead to a suboptimal solution, for example, a base bandit algorithm which is exploratory initially but excels, later on, might fall out of favor. Thus we employ another bandit algorithm, called *bandit master*, for each user, to learn users' preference over meta-paths with exploration-exploitation tradeoff balanced, so to dynamically ensemble base bandit algorithms.

More specifically, the bandit master uses the vector

$$\mathbf{w}_{u,t} = \left[w^1_{u,t}, \ldots, w^{|\mathcal{P}|}_{u,t}\right] \in \mathbb{R}^{|\mathcal{P}|},$$

to represent the user u's preference over different meta-paths, i.e., $w^p_{u,t}$ represents the user u's preference on meta-path p at time t. Note that user u's preference on meta-path p also denotes his preference on the base bandit algorithm under meta-path p. For simplicity, in the following, we describe $\mathbf{w}_{u,t}$ as user u's preference over different base bandit algorithms. At each round t, the bandit master samples a base bandit algorithm p_t according to $\mathbf{w}_{u,t}$, shows the arm selected by the base bandit algorithm p_t to the user, receives feedback, and updates $\mathbf{w}_{u,t}$ accordingly. The above process handles the exploration vs. exploitation tradeoff, since $\mathbf{w}_{u,t}$ is updated based on the historical performance of base bandit algorithms under different meta-paths (i.e., exploitation), while selecting arms by sampling a base bandit algorithm p_t (i.e., exploration).

The detailed steps of the HUCB algorithm are summarized in Algorithm 1, assuming that the given meta-paths are all arm-centric meta-paths. Specifically, the bandit master sets

$$w^i_{u,0} = \frac{1}{|\mathcal{P}|}, \forall i = 1, \ldots, |\mathcal{P}|,$$

at the beginning, implying each base bandit algorithm has equal probability to be selected. At each round t, the probability distribution $\hat{\mathbf{w}}_{u,t}$ is generated from $\mathbf{w}_{u,t}$ to sample a base bandit algorithm p_t (line 2). Here, the parameter γ represents the probability of uniformly exploring base bandit algorithms, and it prevents some base bandit algorithms from never being selected. Then the bandit master selects the arm with the largest upper confidence bound value under base bandit algorithm p_t to recommend to the user, and receives the feedback $r_{u,a_t,t}$ (lines 3–5). Then the bandit master updates *every* base bandit algorithm $p \in \mathcal{P}$ with the newly received feedback (lines 6–10). The updating process includes two parts: (1) updating the base bandit model (lines 7–8); (2) updating the weight vector $\mathbf{w}_{u,t}$ (lines 9–10): if the base bandit algorithm under meta-path p also selects the arm a_t, i.e., $a^p_t = a_t$, $w^p_{u,t}$ will be exponentially boosted by a factor $\eta \cdot r_{u,a_t,t}/(\sum_{p':a^{p'}_t = a_t} \hat{w}^{p'}_{u,t})$, otherwise $w^p_{u,t} = w^p_{u,t-1}$. Here a^p_t denotes the arm selected by the base bandit algorithm under meta-path p at time t, and the hyper-parameter η controls the extent of boosting. In fact, the bandit master adopts a similar algorithm as the Exp3 algorithm [5], we note that one can also use other algorithms that learns from experts [2, 14].

ALGORITHM 1: The HUCB algorithm

Input: HIN G, feature vector $\{\mathbf{x}_{a,t}\}_{a\in\mathcal{A}_t}$, $\lambda_1, \lambda_2 \in (0, +\infty)$, $\gamma, \beta \in (0, 1)$.
Init: $\mathbf{w}^i_{u,0} = \frac{1}{|\mathcal{P}|}, \forall i = 1, \dots |\mathcal{P}|$;
for $p = 1, 2, \dots, |\mathcal{P}|$ **do**
  for each user u, initialize
  $$\mathbf{A}_{p,u,t} \leftarrow \lambda_1 \mathbf{I}_1, \mathbf{b}_{p,u,t} \leftarrow \mathbf{0}, \hat{\theta}_{p,u,t} \leftarrow \mathbf{0}.$$
  for each arm a, initialize
  $$\mathbf{C}_{p,a,t} \leftarrow \lambda_2 \mathbf{I}_2, \mathbf{d}_{p,a,t} \leftarrow \mathbf{0}, \hat{\mathbf{v}}_{p,a,t} \leftarrow \mathbf{0}.$$
for $t = 1, 2, \dots, T$ **do**
  for each $p \in \mathcal{P}$, set
  $$\hat{w}^p_{u,t} = (1-\gamma)\frac{w^p_{u,t-1}}{\sum_j w^j_{u,t-1}} + \frac{\gamma}{|\mathcal{P}|}.$$
  sample a base bandit algorithm p_t according to $\hat{\mathbf{w}}_{u,t}$.
  select the arm a_t using base bandit algorithm p_t according to Equation (6).
  get the user's feedback $r_{u,a_t,t}$.
  for $p = 1, 2, \dots, |\mathcal{P}|$ **do**
    with the interaction record $(u, a_t, r_{u,a_t,t})$, update $\mathbf{A}_{p,u,t}, \mathbf{b}_{p,u,t}, \mathbf{C}_{p,a,t}, \mathbf{d}_{p,a,t}$ according to Equation (5).
    $\hat{\theta}_{p,u,t} = \mathbf{A}^{-1}_{p,u,t}\mathbf{b}_{p,u,t}$.
    $\hat{\mathbf{v}}_{p,a,t} = \mathbf{C}^{-1}_{p,a,t}\mathbf{d}_{p,a,t}$.
    take $l_{p,t} = r_{u,a_t,t} / \sum_{p':a^{p'}_t = a_t} \hat{w}^{p'}_{u,t}$ if $a^p_t = a_t$, otherwise $l_{p,t} = 0$.
    update $w^p_{u,t}$ by $w^p_{u,t} = w^p_{u,t-1}\exp(\eta l_{p,t})$.

3.3 Regret Analysis of HUCB

We first provide the regret upper bound for the base bandit algorithm under an arm-centric meta-path. One can refer to article [29] for the regret upper bound of the base bandit algorithm under a user-centric meta-path.

Theorem 4. *Assume the condition in Lemma 3 holds, and meta-path p is an arm-centric meta-path, then the cumulative regret of the base bandit algorithm under meta-path p is upperbounded as follows:*

$$\begin{aligned} R^p(T) \le & 2\alpha^v_T \sum_{j\in\mathcal{A}} \sqrt{2Tl\log\left(1 + \frac{\sum_{t=1}^T (M^{a_t,j}_{p,t})^2 S^2}{\lambda_2 l}\right)} \\ & + 2\alpha^\theta_T \sqrt{2T(d+l)\log\left(1 + \frac{\sum_{t=1}^T \| \sum_{j\in\mathcal{A}} M^{a_t,j}_{p,t}(\mathbf{x}_{j,t}, \hat{\mathbf{v}}_{p,j,t})\|_2^2}{\lambda_1(d+l)}\right)} \\ & + 2\frac{\alpha^\theta_T L}{\sqrt{\lambda_1}} \frac{(q_1+\mu_1)(1-(q_1+\mu_1)^T)}{1-(q_1+\mu_1)}. \end{aligned} \tag{7}$$

When no relation between arms are used, i.e., $M^{a_t,a}_{p,t} = \mathbb{1}_{\{a=a_t\}}$, the base bandit algorithm reduces to the hLinUCB algorithm [28]. From Theorem 4, we can observe that leveraging relations between arms can bring smaller regret upper bound than the hLinUCB algorithm, since $0 \le M^{a_t,j}_{p,t} \le 1$ and

$\sum_{j\in\mathcal{A}} \mathrm{M}_{p,t}^{a_t,j} = 1$. The same conclusion is also achieved for user-centric meta-path as shown in article [29].

Next we bound the regret of the proposed HUCB algorithm.

Theorem 5. *Let* $\mathbf{w}_{u,*}\in R^{|\mathcal{P}|}$ *denote the unknown true preference over meta-paths of user* u*. Note that* $\sum_p w_{u,*}^p = 1$*. Let* $\eta = \frac{k\gamma}{|\mathcal{P}|}, k \geq 0$ *and* $\gamma = \min(1, \sqrt{\frac{|\mathcal{P}|\ln(|\mathcal{P}|)}{Tk((e-2)k+1)}})$*, then the cumulative regret up to* T *of HUCB can be bounded by*

$$R(T) \leq \sum_p w_{u,*}^p R^p(T) + 2\sqrt{\frac{(e-2)k+1}{k}}\sqrt{T|\mathcal{P}|\ln(|\mathcal{P}|)}.$$

Here $R^p(T)$ is the regret upper bound of the base bandit algorithm for the meta-path p. If p is an arm-centric (or a user-centric) meta-path, the detailed formula of $R^p(T)$ can be found in Theorem 4 (or article [29]). Obviously, $R(T)$ in Theorem 5 is dominated by $R^p(T)$. In other words, the HUCB algorithm achieves a similar performance as compared with the optimal scenario, where each user is served according to his true preference over meta-paths, while the true preference is usually unknown beforehand. Moreover, as discussed before, $R^p(T)$ is smaller than the regret upper bound of the hLinUCB algorithm which does not leverage HIN. This implies that the HUCB algorithm achieves smaller regret upper bound by leveraging HIN. Note that smaller regret upper bound implies better performance or higher accumulated reward.

The computational complexity of HUCB is determined by the complexity of the base bandit algorithm. Compared to traditional contextual bandit algorithm such as hLinUCB algorithms [28], the base bandit algorithm only incurs additional complexity in incorporating the meta-paths to calculating matrix $\boldsymbol{A}_{p,u,t}, \boldsymbol{C}_{p,a,t}$,*and so on*. In calculating these matrices, the essential part is computing the commuting matrices, which can be addressed by [25].

4 EXPERIMENTS ON SYNTHETIC DATASET

In this section, we conduct experiments on synthetic datasets to evaluate the proposed HUCB algorithm.

4.1 Experimental Settings

We first present how we generate synthetic data, then describe baselines to compare with.

Synthetic data. We synthesize a heterogeneous information network $G = (\mathcal{V}, \mathcal{E}, \mathcal{K}, \mathcal{R}, \phi, \psi)$ with three types of nodes, i.e.,

$$\mathcal{K} = \{U = \text{“User”}, A = \text{“Arm”}, C = \text{“Category”}\}.$$

Thus we divide the node set into three subsets, i.e.,

$$\mathcal{V} = \mathcal{V}_C \cup \mathcal{V}_A \cup \mathcal{V}_U,$$

where $\mathcal{V}_C, \mathcal{V}_A$, and $\mathcal{V}_U$ denote subset of category nodes, arm nodes, and user nodes. The edge set $\mathcal{E}$ is partitioned into four subsets:

$$\mathcal{E} = \mathcal{E}_{A,C} \cup \mathcal{E}_{C,A} \cup \mathcal{E}_{U,A} \cup \mathcal{E}_{U,U},$$

where $\mathcal{E}_{A,C}, \mathcal{E}_{C,A}, \mathcal{E}_{U,A}$, and $\mathcal{E}_{U,U}$ denote a set of directed edges from arms to category nodes, from category nodes to arms, from users to arms and from users to users respectively. The edge set $\mathcal{E}_{A,C}, \mathcal{E}_{C,A}$, and $\mathcal{E}_{U,U}$ are fixed and are represented by adjacency matrix $\mathbf{W}|_{A,C}, \mathbf{W}|_{C,A}$ and $\mathbf{W}|_{U,U}$ respectively.

We generate the network G as follows. We first generate $N_C \in \mathbb{N}_+$ category nodes ($|\mathcal{V}_C| = N_C$), and $N_A \in \mathbb{N}_+$ arm nodes ($|\mathcal{V}_A| = N_A$), where each arm $a \in \mathcal{V}_A$ is associated with a $(d+l)$-dimensional feature vector $(\mathbf{x}_a, \mathbf{v}_a) \in \mathbb{R}^{d+l}$. Here d is the dimension of known contextual vectors (i.e., $\mathbf{x}_a \in \mathbb{R}^d$), while l is the dimension of latent features (i.e., $\mathbf{v}_a \in \mathbb{R}^l$). Each element of the vector $(\mathbf{x}_a, \mathbf{v}_a)$ is generated from the interval $(0, 1)$ uniformly at random. And we normalize the feature vector so that $\|(\mathbf{x}_a, \mathbf{v}_a)\|_2 = 1$. Note that all $(d+l)-$dimensional features are used to generate the true reward for each arm, while only revealing d dimension of features (i.e., $\mathbf{x}_a$) to the learning algorithm.

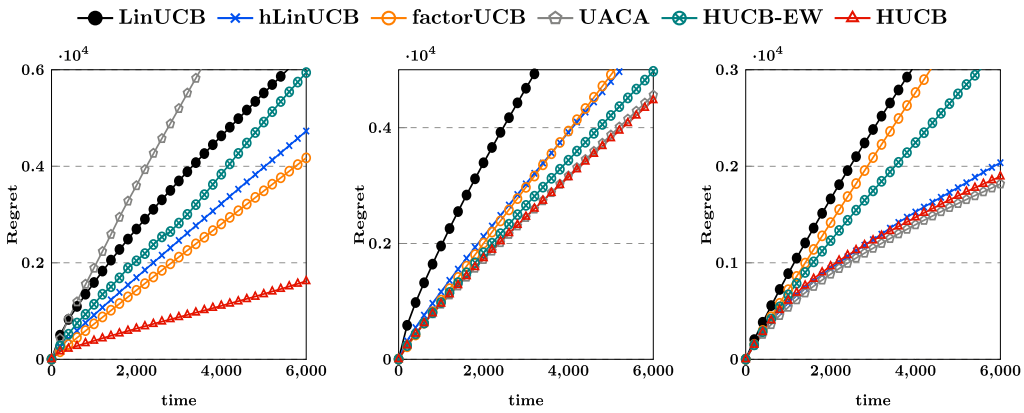
We then generate N_U users ($|\mathcal{V}_U| = N_U$), each of whom is associated with a $(d+l)$-dimensional vector $\theta_{u,*}$, representing the ground-truth preference parameter of user $u \in \mathcal{V}_U$. Each element of the vector $\theta_{u,*}$ is generated from the interval $(0, 1)$ uniformly at random. We also normalize it so that $\|\theta_{u,*}\|_2 = 1$. For each user u, we also need to generate his true preference over meta-paths, i.e., $\mathbf{w}_{u,*}$. To verify the correctness of the learned users' preference over meta-paths in an explicit way, we consider the case that each user only prefers one particular meta-path. Specifically, we randomly select one meta-path for each user u as his preferred meta-path. For example, if two meta-paths exist, then we select $\mathbf{w}_{u,*} = (1, 0)$ or $(0, 1)$ with equal probability. We want to remark that $\{\theta_{u,*}\}$ and $\{\mathbf{w}_{u,*}\}$ are only used for generating true reward of each arm, and will not reveal to the learning algorithm.

To generate $\mathbf{W}|_{A,C}$ and $\mathbf{W}|_{C,A}$, for each category node $c \in \mathcal{V}_C$, we generate a $(d+l)$-dimensional feature vector $(\tilde{\mathbf{x}}_c, \tilde{\mathbf{v}}_c)$ similar as above, and set $W|_{A,C}(a, c) = W|_{C,A}(c, a) = 1$, if $(\mathbf{x}_a, \mathbf{v}_a)^T(\tilde{\mathbf{x}}_c, \tilde{\mathbf{v}}_c) \geq \zeta_{ac}$, otherwise $W|_{A,C}(a, c) = W|_{C,A}(c, a) = 0$, where $\zeta_{ac} \in \mathbb{R}$. Note that feature vectors of category nodes are only used for generating $\mathbf{W}|_{A,C}$ and $\mathbf{W}|_{C,A}$, and will not be used in simulations. We generate $W|_{U,U}$ similarly, i.e., $W|_{U,U}(i, j) = 1$ if $\theta_{i,*}^T \theta_{j,*} \geq \zeta_{uu}$, otherwise $W_{u,u}(i, j) = 0$, where $\zeta_{uu} \in \mathbb{R}$.

Given above HIN, we consider the following two meta-paths, i.e., $\mathcal{P} = \{$"user→arm→category→arm", "user→user→arm" $\}$, whose reward models are expressed in Equation (3) in Section 3.1 and Equation (3) in [29] respectively. Here, we select one user-centric meta-path and one arm-centric meta-path for the purpose of simplifying the presentation. Our objective is to impact these two types of meta-paths on the learning speed. while still showing their impact on the learning speed. We denote the base bandit algorithm under above meta-paths as UACA and factorUCB respectively. In the synthetic dataset, the user u's feedback to arm a is simulated by the sum of true reward of arm a regarding to user u (generated according to the reward model) and a random noise $\epsilon_t \sim N(0, \kappa^2)$. We observe that the parameter β in Equation (3) determines the degree of mismatching between two base bandit algorithms. For example, if $\beta = 0$, UACA has a pretty poor performance when feedback are generated according to the reward model of factorUCB. This is because factorUCB assumes arm a's reward is determined by its own feature, i.e., $(\mathbf{x}_a, \mathbf{v}_a)$, while UACA with $\beta = 0$ assumes arm a's reward is determined by the weighted average of similar arms' rewards, as shown in Equation (3). And when β is larger, for example $\beta = 0.1$, UACA enables to leverage each arm's feature, thus its performance becomes better. Thus, we can evaluate algorithms under different relationship between base bandit algorithms. Specifically, we consider the following two cases:

- **Base bandits with high mismatch degree**: we set $\beta = 0$ in this case. As discussed before, UACA performs poorly when rewards are generated according to the reward model of factorUCB. However, for around half of users with $\mathbf{w}_{u,*} = (0, 1)$, feedback are generated based on the reward model of factorUCB.
- **Base bandits with low mismatch degree**: we set $\beta = 0.1$ in this case. With a larger β, the performance of UACA becomes better when feedback are generated according to the reward model of factorUCB.

In simulation, we set $\kappa = 0.1, d = 20, l = 5, N_U = 100, N_A = 100, N_C = 20, \zeta_{ac} = \zeta_{uu} = 0.5$.



(a) Cumulative regret with highly mismatched base bandits (b) Cumulative regret with lowly mismatched base bandits (c) Cumulative regret with $\mathbf{w}_{u,*} = (1, 0), \forall u$

Fig. 2. Experimental results on a synthetic dataset.

Baselines. We compare the proposed HUCB algorithm with the following algorithms.

- LinUCB [15]: the state-of-the-art contextual bandit algorithm. LinUCB only works with observed contextual features and does not consider hidden features and any other relations.
- hLinUCB [28]: it extends LinUCB to consider hidden features, but it does not leverage any other relations.
- factorUCB [29]: it builds from hLinUCB while considering users' friendships. It is the base bandit algorithm under the meta-path "user→user→arm".
- UACA: the base bandit algorithm under the meta-path "user→arm→category→arm".
- HUCB-EW: a variant of HUCB that randomly selects base bandit algorithms at each round t, i.e., $w^p_{u,t} = \frac{1}{|\mathcal{P}|}, \forall p \in \mathcal{P}$.

4.2 Evaluation Results

We evaluate all algorithms in terms of cumulative regret defined in Equation (1). At each round t, we randomly select 25 arms from $\mathcal{A}$ without replacement as $\mathcal{A}_t$, and only show $\mathcal{A}_t$ to the agent for arm selection. The same $\mathcal{A}_t$ and ϵ_t are presented to all algorithms. Under all cases, we run the experiment 10 times, and plot the average cumulative regret.

Cumulative regret with highly mismatched base bandits: Figure 2(a) shows the cumulative regret of six algorithms when base bandit algorithms are of a high degree of mismatch. One can observe that HUCB still achieves the smallest cumulative regret, even though one of its base bandit algorithm (UACA) performs worse than LinUCB. The poor performance of UACA, as discussed before, is because of its mismatch to the reward model of factorUCB, while feedback of around half of users are generated according to the reward model of factorUCB. Table 1 shows the average and standard deviation of learned users' preference over meta-paths (i.e., $\{\mathbf{w}_{u,t}\}$) after 6,000 iterations. From Table 1, one can observe that for user u with $\mathbf{w}_{u,*} = (0, 1)$, i.e., whose feedback are generated according to the reward model of factorUCB, $\mathbf{w}_{u,t}$ has converged to $\mathbf{w}_{u,*}$, due to the poor performance of UACA for these users. For user u with $\mathbf{w}_{u,*} = (1, 0)$, i.e., whose feedbackare generated according to the reward model of UACA, $\mathbf{w}_{u,t}$ has not converged to the ground truth. However, we can observe that HUCB has already assigned higher weight to UACA than factorUCB, since UACA is more accurate for these users. This also explains why HUCB outperforms factorUCB. HUCB-EW

Table 1. Statistics of Learned $\mathbf{w}_{u,t}$ with Highly Mismatched Base Bandits

	$\mathbf{w}^0_{u,t}$	$\mathbf{w}^1_{u,t}$
users with $\mathbf{w}_{u,*} = (1,0)$	$0.7608(\pm 0.347)$	$0.2391(\pm 0.347)$
users with $\mathbf{w}_{u,*} = (0,1)$	$0(\pm 0)$	$1(\pm 0)$

performs better than LinUCB, but it is worse than hLinUCB and factorUCB, since it treats UACA and factorUCB equally, thus distorted heavily by UACA. factorUCB performs better than hLinUCB because it leverages users' friendships, while hLinUCB achieves better performance compared to LinUCB via learning hidden features.

Cumulative regret with lowly mismatched base bandits: Figure 2(b) shows the cumulative regret of six algorithms when base bandit algorithms are of low degree of mismatch. As shown in Figure 2(b), HUCB still achieves the best performance, while UACA is only slightly worse than HUCB. This is because UACA is enabled to leverage each arm's features with $\beta = 0.1$, thus it performs well when feedback are generated according to the reward model of factorUCB. The performance of HUCB-EW is between that of UACA and factorUCB. Moreover, we can observe that the performance of factorUCB is slightly worse than hLinUCB. This is because the performance of factorUCB drops when generating feedback according to the reward model of UACA, since the reward of factorUCB is a weighted average of rewards from similar users, thus failing to capture the unique features of each user used in feedback generation in UACA. LinUCB performs the worst since it does not leverage hidden features and any relationship among users and items.

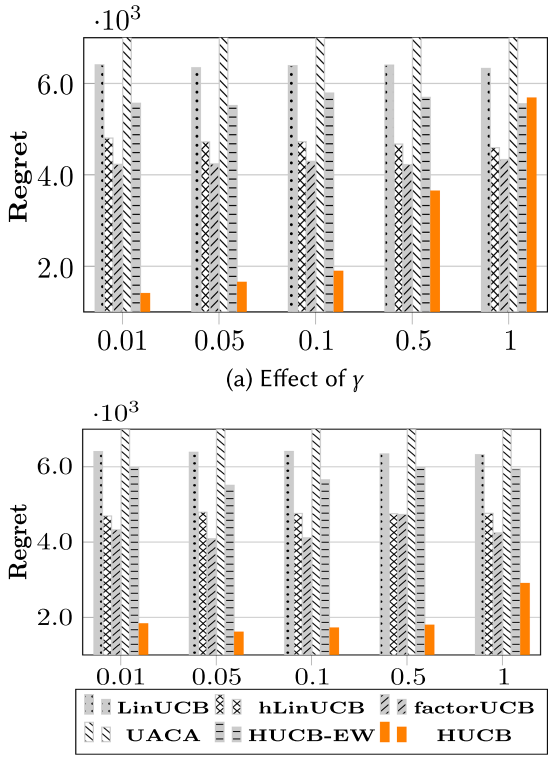
Cumulative regret with $\mathbf{w}_{u,*} = (1,0), \forall u$: We also conduct experiments to evaluate the performance of six algorithms when one of base bandit algorithm is the optimal algorithm. Figure 2(c) shows the cumulative regret when all users' feedback are generated according to the reward model of UACA, i.e., $\mathbf{w}_{u,*} = (1,0), \forall u$. One can observe that UACA achieves the smallest cumulative regret. However, the performance of HUCB is only slightly worse than UACA, followed by hLinUCB, HUCB-EW, factorUCB, and LinUCB. factorUCB performs poorly out of the same reason as discussed before. The performance of HUCB-EW is worse than hLinUCB, since it is distorted by factorUCB.

Lesson Learned. In summary, HUCB learns users' preference over meta-paths to dynamically ensemble base bandit algorithms. It can still work efficiently when there is a mismatch between base bandit algorithms. And when there is an unknown best base bandit algorithm, HUCB can still achieve a roughly similar performance.

The most important part of HUCB is to learn users' preference over meta-paths accurately, thus next we investigate the impact of γ and η in Algorithm 1. We conduct experiments with different γ and η under the same experimental setting when base bandit algorithms are of a high degree of mismatch. We also do experiments under other cases, and observe similar patterns.

Impact of γ: Figure 3(a) shows the cumulative regret under different γ after 6,000 iterations. Recall that with probability γ, we randomly select a base bandit algorithm (line 2 in Algorithm 1). Thus intuitively large γ hurts the performance of HUCB, since a large γ means a large probability to select sub-optimal base bandit algorithms. Figure 3(a) verifies the intuition: the cumulative regret of HUCB increases with γ increasing.

Impact of η: Figure 3(b) shows the cumulative regret under different η after 6,000 iterations. Recall that η determines the extent of boosting $\mathbf{w}_{u,t}$ with positive feedback (line 10 in Algorithm 1). As shown in Figure 3(b), if η is too small, i.e., $\eta = 0.01$, the cumulative regret of HUCB is larger



(a) Effect of γ

(b) Effect of η

Fig. 3. Effect of γ and η.

(compared to results with $\eta = 0.05$), since it needs longer time to make the probability of selecting sub-optimal base bandit algorithms small. However, HUCB performs badly with large η, e.g., $\eta = 1$, since it will be easily disturbed by noise.

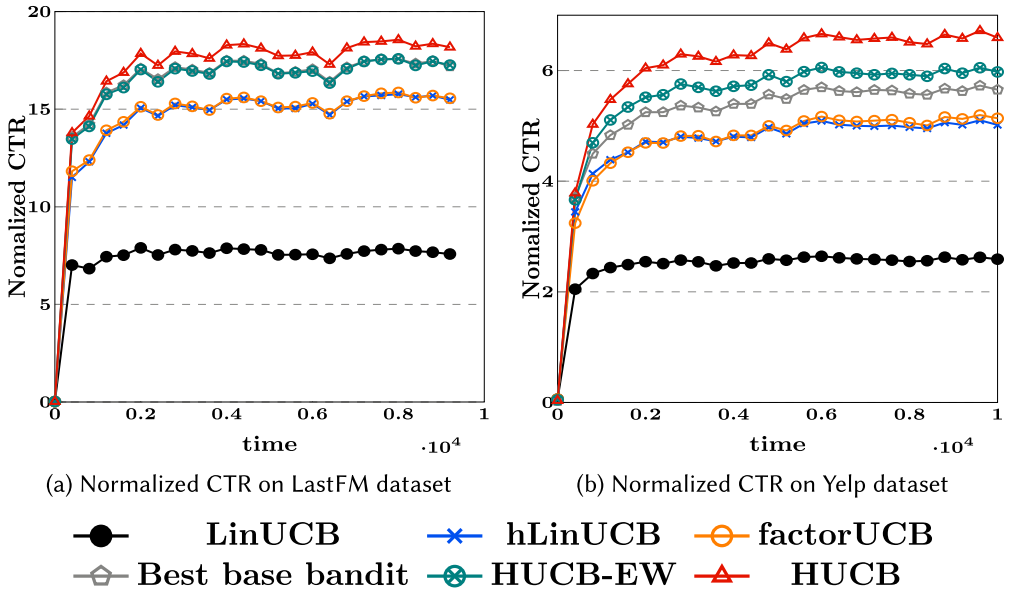
5 EXPERIMENTS ON REAL DATASETS

In this section, we evaluate the performance of the HUCB algorithm on two real-world datasets from LastFM and Yelp respectively.

5.1 Experiments on LastFM Dataset

The LastFM dataset is extracted from the online music streaming service Last.fm.[2] It contains three types of nodes: "user", "artist" and "tag", and four types of edges: "user↔user", "user→artist", "artist↔tag", "user→tag". The LastFM dataset contains 1,892 users and 17,362 artists. We take each artist as an arm. If the user listened to an artist at least once, the reward is 1, otherwise the reward is 0. We only keep those users with at least 50 interaction records. Following [28], we first generate each arm's TF-IDF feature vector with all tags associated with the arm. Then, PCA is applied to reduce the dimension of features and take the first 10 principle components as the arm's contextual vector, i.e., $d = 10$. We set the dimension of hidden features as 5. In LastFM dataset, we consider the following set of meta-paths, $\mathcal{P} = \{$ "user→user→artist", "user→artist→tag→artist", "user→artist→tag→artist→tag→artist"$\}$. We select this set of meta-paths because LastFM is a music recommendation system and for musics, user preferences are mainly tied to artist and tags of musics.

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



(a) Normalized CTR on LastFM dataset (b) Normalized CTR on Yelp dataset

Fig. 4. Experimental results on real datasets.

The unbiased offline evaluation protocol proposed in [16] is applied to evaluate algorithms. The baselines are similar to that in Section 4, except that we repace the baseline UACA with the best base bandit algorithm. The unbiased offline evaluation protocol only works when feedback are collected under a random policy. Hence, we simulate the *random* policy by generating a candidate pool as follows. At each time t, we store the arm presented to the user (a_t), and its received feedback. Then we create $\mathcal{A}_t$ by including the served arm along with 24 extra arms the user has interacted with (hence $|\mathcal{A}_t| = 25, \forall t$). The 24 extra arms are drawn uniformly at random so that for any arm a the user interacted with: If a occurs in some set $\mathcal{A}_t$, this arm will be served 1/25 of the times. The algorithms are evaluated by **Click through-rate** (**CTR**), which is the ratio between the number of positive rewards an algorithm receives and the number of recommendations it makes. In particular, we use the average CTR in *every* 400 iterations (not the cumulative CTR) as the evaluation metric. Following [15], we normalize the resulting CTR from different algorithms by the corresponding logged random strategy's CTR.

Evaluation results. Figure 4(a) shows the normalized CTRs of six algorithms. One can observe that the HUCB algorithm achieves the highest CTRs, while the LinUCB algorithm has the lowest CTRs. The HUCB-EW algorithm performs worse than HUCB algorithm, implying the effectiveness of dynamically ensembling base bandit algorithms. For LastFM dataset, the best base bandit algorithm is under the meta-path "user→artist→tag→artist", and its performance is similar with the HUCB-EW algorithm. Namely, user preference is more likely to be revealed by this meta-path. Although it may not be obvious in Figure 4(a) due to the scale of y-values, the factorUCB algorithm is slightly better than the hLinUCB algorithm, especially in the beginning phrase.

5.2 Experiments on Yelp Dataset

The public Yelp dataset[3] contains users' reviews on businesses on Yelp. Each business in the dataset is associated with a number of categories and its location. For example, one restaurant

[3]http://www.yelp.com/academic_dataset.

named "Filiberto's Mexican food" is located at "Avondale", and associated with the following categories: {"Mexican", "Restaurant"}. Thus, the dataset contains four types of nodes: "user", "business", "category" and "location", and four types of links: "user↔user", "user→business", "business↔category", and "business↔location". We take each business as an arm, and consider the following set of meta-paths $\mathcal{P}$ = {"user→business→category→business", "user→user→business", "user→business→location→business"}. We select these meta-paths because Yelp is a restaurant recommendation system and for restaurants, user preferences are mainly tied to businesses and categories of businesses. We construct the contextual vectors as follows: we first generate feature vectors from the business's raw attributes, including geographic features, categorical features, average rating, and total review count, as well as attributes. Then, we apply PCA on the feature vectors, and take the first 8 components as contextual vectors.[4] We also normalize each contextual vector, i.e., $||\mathbf{x}_a||_2 = 1, \forall a$, and set the dimension of hidden features as 3. The original 5-scale ratings are converted to a binary-valued feedback between businesses and users, i.e., high ratings (4 and 5) as positive(1) and low ratings (≤3) as negative(0). We only keep users with more 50 positive feedback.

Evaluation results. Following a similar procedure of experiments on LastFM dataset, we compare all algorithms with normalized CTRs. The results are shown in Figure 4(b). Similarly, we can observe that HUCB achieves the highest CTRs, followed by HUCB-EW, the best base bandit algorithm, factorUCB, hLinUCB, and LinUCB. For Yelp dataset, the best base bandit algorithm is the one under the meta-path "user→item→location→item". Namely, user preference is more likely to be revealed by this meta-path. Namely, user preference is more likely to revealed by this meta-path. It is reasonable, since location is pretty important when people choose where to consume. Moreover, the performance of HUCB-EW is better than the best base bandit algorithm. This is because selecting sub-optimal base bandit algorithms enables the bandit to explore from different aspects, thus contributing to better performance. The Yelp dataset contains more arms than LastFM dataset, thus the benefit will be larger.

6 PROOF TO THEOREMS AND LEMMAS

6.1 Proof of Theorem 5

The cumulative regret of HUCB up to time T is

$$\begin{aligned} R(T) &= \sum_{t=1}^{T} E[r_{u,a_t^*,t}] - E[r_{u,a_t,t}] \\ &= \sum_{t=1}^{T} \left(\sum_p w_{u,*}^p r_{u,a_{t,*}^p,*} - r_{u,a_t,*} \right) \\ &= \underbrace{\sum_{t=1}^{T} \sum_p w_{u,*}^p (r_{u,a_{t,*}^p,*} - r_{u,a_t^p,*})}_{(A_1)} \\ &+ \underbrace{\sum_{t=1}^{T} \sum_p w_{u,*}^p (r_{u,a_t^p,*} - r_{u,a_t,*})}_{(A_2)}, \end{aligned}$$

where $a_{t,*}^p$ denotes the optimal arm at time t under meta-path p, and a_t^p denote the selected arm by the base bandit algorithm under meta-path p at time t.

[4]We use a smaller dimension since the dataset is larger.

For (A_1), following the similar procedure of Theorem 4, we can prove that $(A_1) \leq \sum_p w^p_{u,*} R^p(T)$.

Next, we try to bound (A_2). Let p_* denotes the best base bandit algorithm, i.e., $p_* = \arg\max_p \sum_{t=1}^T r_{u,a_t^p,*}$. Then we can first get $(A_2) \leq \sum_{t=1}^T r_{u,a_t^{p^*},*} - r_{u,a_t,*}$. Let $W_t = w^1_{u,t} + \ldots + w^{|\mathcal{P}|}_{u,t}$, then:

$$\frac{W_{t+1}}{W_t} = \sum_p \frac{w^p_{u,t+1}}{W_t} = \sum_p \frac{\hat{w}^p_{u,t} - \frac{\gamma}{|\mathcal{P}|}}{1-\gamma} \exp(\eta l_{p,t}).$$

With the fact $e^x \leq 1 + x + (e-2)x^2$ for $x \leq 1$, we can get:

$$\begin{aligned}\frac{W_{t+1}}{W_t} &\leq \sum_p \frac{\hat{w}^p_{u,t} - \frac{\gamma}{|\mathcal{P}|}}{1-\gamma} \left(1 + \eta l_{p,t} + (e-2)\eta^2 l^2_{p,t}\right) \\ &\leq 1 + \frac{\eta}{1-\gamma} \sum_p \hat{w}^p_{u,t} l_{p,t} + \frac{(e-2)\eta^2}{1-\gamma} \sum_p \hat{w}^p_{u,t} (l_{p,t})^2.\end{aligned}$$

According to the definition of $l_{p,t}$, we can get:

$$\begin{aligned}&(1) \sum_p \hat{w}^p_{u,t} l_{p,t} = r_{u,a_t,t}; \\ &(2) \sum_p \hat{w}^p_{u,t} (l_{p,t})^2 = r_{u,a_t,t} * l_{p,t} \leq l_{p,t} \leq \sum_p l_{p,t}.\end{aligned}$$

Together with $1 + x \leq e^x, \forall x \geq 0$, we arrives:

$$\ln\left(\frac{W_{t+1}}{W_t}\right) \leq \frac{\eta}{1-\gamma} r_{u,a_t,t} + \frac{(e-2)\eta^2}{1-\gamma} \sum_p l_{p,t}.$$

Summarizing over t, we get:

$$\ln(W_{T+1}) \leq \frac{\eta}{1-\gamma} \sum_{t=1}^T r_{u,a_t,t} + \frac{(e-2)\eta^2}{1-\gamma} \sum_{t=1}^T \sum_p l_{p,t}. \tag{8}$$

On the other hand, for any base bandit under meta-path p, we have:

$$\ln(W_{T+1}) \geq \ln(w^p_{u,t}) = \eta \sum_{t=1}^T l_{p,t} - \ln(|\mathcal{P}|). \tag{9}$$

Combining Equations (8) and (9), we can get:

$$\sum_{t=1}^T r_{u,a_t,t} \geq (1-\gamma) \sum_{t=1}^T l_{p,t} - \frac{\ln(|\mathcal{P}|)}{\eta} - (e-2)\eta \sum_{t=1}^T \sum_p l_{p,t}.$$

Let $s_{a,t}$ denote the probability of HUCB to select arm a at time t, then $s_{a,t} = \sum_{p':a_t^{p'}=a} \hat{w}^{p'}_{u,t}$. Note that

$$E[l_{p,t}] = s_{a_t^p,t} \cdot \frac{E[r_{u,a_t^p,t}]}{s_{a_t^p,t}} + (1 - s_{a_t^p,t}) * 0 = r_{u,a_t^p,*}.$$

Taking expectation for bother sides, for the base bandit p_*:

$$\begin{aligned}
&\sum_{t=1}^{T} r_{u,a_t^{p_*},*} - \sum_{t=1}^{T} r_{u,a_t,*} \\
&\leq ((e-2)\eta|\mathcal{P}| + \gamma) \sum_{t=1}^{T} r_{u,a_t^{p_*},*} + \frac{\ln(|\mathcal{P}|)}{\eta} \\
&\leq ((e-2)\eta|\mathcal{P}| + \gamma)T + \frac{\ln(|\mathcal{P}|)}{\eta}.
\end{aligned}$$

Note that $\eta = \frac{k\gamma}{|\mathcal{P}|}, k \geq 0$, and $\gamma = \sqrt{\frac{\ln(|\mathcal{P}|)|\mathcal{P}|}{Tk((e-2)k+1)}}$. This proof is then complete. □

6.2 Proof of Lemma 3

For ease of illustration, we denote

$$\tilde{\mathbf{x}}_{a,t} = \sum_{j\in\mathcal{A}} \mathrm{M}_{p,t}^{a,j} \mathbf{x}_{j,t}, \quad \tilde{\mathbf{v}}_{p,a,t} = \sum_{j\in\mathcal{A}} \mathrm{M}_{p,t}^{a,j} \hat{\mathbf{v}}_{p,a,t}$$

and

$$\tilde{\mathbf{v}}_{p,a,*} = \sum_{j\in\mathcal{A}} \mathrm{M}_{p,t}^{a,j} \mathbf{v}_{p,a,*}.$$

We first derive the confidence interval regarding to $\hat{\theta}_{p,u,t}$. With $r_{u,a,\tau} = (\tilde{\mathbf{x}}_{a,t}, \tilde{\mathbf{v}}_{p,a,*})^T \theta_{p,u,*} + \epsilon_\tau$ and Equation (5), we have:

$$\begin{aligned}
\hat{\theta}_{p,u,t} =& \mathbf{A}_{p,u,t}^{-1} \left(\sum_{\tau=1}^{t-1} (\tilde{\mathbf{x}}_{a_\tau,\tau}, \tilde{\mathbf{v}}_{p,a_\tau,\tau}) r_{u,a_\tau,\tau} \right) \\
=& \mathbf{A}_{p,u,t}^{-1} \left(\sum_{\tau=1}^{t-1} (\tilde{\mathbf{x}}_{a_\tau,\tau}, \tilde{\mathbf{v}}_{p,a_\tau,\tau}) \left((\tilde{\mathbf{x}}_{a_\tau,\tau}, \tilde{\mathbf{v}}_{p,a_\tau,*})^T \theta_{p,u,*} + \epsilon_\tau \right) \right) \\
=& \theta_{p,u,*} - \lambda_1 \mathbf{A}_{p,u,t}^{-1} \theta_{p,u,*} + \mathbf{A}_{p,u,t}^{-1} \left(\sum_{\tau=1}^{t-1} (\tilde{\mathbf{x}}_{a_\tau,\tau}, \tilde{\mathbf{v}}_{p,a_\tau,\tau}) \epsilon_\tau \right) \\
&+ \mathbf{A}_{p,u,t}^{-1} \left(\sum_{\tau=1}^{t-1} (\tilde{\mathbf{x}}_{a_\tau,\tau}, \tilde{\mathbf{v}}_{p,a_\tau,\tau}) (\tilde{\mathbf{v}}_{p,a_\tau,*} - \tilde{\mathbf{v}}_{p,a_\tau,\tau})^T \theta_{p,u,*}^{v} \right).
\end{aligned}$$

Note that

$$\begin{aligned}
\|(\tilde{\mathbf{x}}_{a,t}, \tilde{\mathbf{v}}_{p,a,t})\|_2^2 &= \sum_{j\in\mathcal{A}} \sum_{k\in\mathcal{A}} \mathrm{M}_{p,t}^{a,j} \mathrm{M}_{p,t}^{a,k} (\mathbf{x}_{j,t}, \hat{\mathbf{v}}_{p,j,t})^T (\mathbf{x}_{k,t}, \hat{\mathbf{v}}_{p,k,t}) \\
&\leq L^2.
\end{aligned}$$

Then it follows that

$$\|\hat{\theta}_{p,u,t} - \theta_{p,u,*}\|_{\mathbf{A}_{p,u,t}} \leq \left\| \sum_{\tau=1}^{t-1} (\tilde{\mathbf{x}}_{a_\tau,\tau}, \tilde{\mathbf{v}}_{p,a_\tau,\tau}) \epsilon_\tau \right\|_{\mathbf{A}_{p,u,t}^{-1}} + \lambda_1 \|\theta_{p,u,*}\|_{\mathbf{A}_{p,u,t}^{-1}} + \frac{LS}{\sqrt{\lambda_1}} \sum_{\tau=1}^{t-1} \|\tilde{\mathbf{v}}_{p,a,*} - \tilde{\mathbf{v}}_{p,a,t}\|_2.$$

If the regularization parameter λ_1 is sufficiently large, the Hessian matrix of Equation (4) is positive definite at the optimizer, then the estimation of $\theta_{p,u}$ and $\mathbf{v}_{p,a}$ is $q-$linearly convergent to

the optimizer [26]. In other words, for any arm a, for every $\mu_1 \geq 0$, $\mu_1 \geq 0$, we have

$$\|\mathbf{v}_{p,a,*} - \hat{\mathbf{v}}_{p,a,t+1}\|_2 \leq (q_1 + \mu_1)\|\mathbf{v}_{p,a,*} - \hat{\mathbf{v}}_{p,a,t}\|_2,$$
$$\|\theta_{p,u,*} - \hat{\theta}_{p,u,t+1}\|_2 \leq (q_2 + \mu_2)\|\theta_{p,u,*} - \hat{\theta}_{p,u,t}\|_2,$$

where $0 \leq q_1 \leq 1$ and $0 \leq q_2 \leq 1$. Thus we have

$$\|\tilde{\mathbf{v}}_{p,a,*} - \tilde{\mathbf{v}}_{p,a,t+1}\|_2 \leq (q_1 + \mu_1)\|\tilde{\mathbf{v}}_{p,a,*} - \tilde{\mathbf{v}}_{p,a,t}\|_2.$$

Finally, applying Theorem 1 in [1], we can get

$$\|\hat{\theta}_{p,u,t} - \theta_{p,u,*}\|_{\mathbf{A}_{p,u,t}} \leq \sqrt{\lambda_1}S + \frac{L^2 S}{\sqrt{\lambda_1}} \frac{(q_1 + \mu_1)(1 - (q_1 + \mu_1)^t)}{1 - (q_1 + \mu_1)} + (d + l)\log\left(\left(1 + \frac{\sum_{\tau=1}^{t-1} \| \sum_{j \in \mathcal{A}} \mathrm{M}_{p,\tau}^{a_\tau, j}(\mathbf{x}_{j,\tau}, \hat{\mathbf{v}}_{p,j,\tau})\|_2^2}{\lambda_1(d + l)}\right)/\sigma\right).$$

Following similar procedure, we can derive the confidence interval regarding to $\hat{\mathbf{v}}_{p,a,t}$ and get:

$$\|\hat{\mathbf{v}}_{p,a,t} - \mathbf{v}_{p,a,*}\|_{\mathrm{C}_{p,a,t}} \leq l\log\left(\left(1 + \frac{\sum_{\tau=1}^{t-1}(\mathrm{M}_{p,\tau}^{a_\tau, a})^2 S^2}{\lambda_2 l}\right)/\sigma\right) + \sqrt{\lambda_2}L + \frac{S^2 L}{\sqrt{\lambda_2}}\left(\frac{(q_1 + \mu_1)(1 - (q_1 + \mu_1)^t)}{1 - (q_1 + \mu_1)} + \frac{(q_2 + \mu_2)(1 - (q_2 + \mu_2)^t)}{1 - (q_2 + \mu_2)}\right). \tag{10}$$

This proof is then complete. □

6.3 Proof of Theorem 4

For ease of illustration, we denote

$$\tilde{\mathbf{x}}_{a,t} = \sum_{j \in \mathcal{A}} \mathrm{M}_{p,t}^{a,j} \mathbf{x}_{j,t}, \quad \tilde{\mathbf{v}}_{p,a,t} = \sum_{j \in \mathcal{A}} \mathrm{M}_{p,t}^{a,j} \hat{\mathbf{v}}_{p,a,t}$$

and

$$\tilde{\mathbf{v}}_{p,a,*} = \sum_{j \in \mathcal{A}} \mathrm{M}_{p,t}^{a,j} \mathbf{v}_{p,a,*}.$$

Let a_t^* denote the best arm for u at time t under meta-path p, then the regret at time t is

$$\begin{aligned}
&R_t^p \\
&= (\tilde{\mathbf{x}}_{a_t^*,t}, \tilde{\mathbf{v}}_{p,a_t^*,*})^T \theta_{p,u,*} - (\tilde{\mathbf{x}}_{a_t,t}, \tilde{\mathbf{v}}_{p,a_t,*})^T \theta_{p,u,*} \\
&= (\tilde{\mathbf{x}}_{a_t^*,t}, \tilde{\mathbf{v}}_{p,a_t^*,t})^T \hat{\theta}_{p,u,t} + \alpha_t^\theta \|(\tilde{\mathbf{x}}_{a_t^*,t}, \tilde{\mathbf{v}}_{p,a_t^*,t})\|_{\mathbf{A}_{p,u,t}^{-1}} \\
&\quad + \alpha_t^v \sum_{j \in \mathcal{A}} \mathrm{M}_{p,t}^{a_t^*,j} \|\hat{\theta}_{p,u,t}^v\|_{\mathrm{C}_{p,j,t}^{-1}} + \alpha_t^\theta \|\tilde{\mathbf{v}}_{p,a_t^*,*} - \tilde{\mathbf{v}}_{p,a_t^*,t}\|_{\mathbf{A}_{p,u,t}^{-1}} - (\tilde{\mathbf{x}}_{a_t,t}, \tilde{\mathbf{v}}_{p,a_t,*})^T \theta_{p,u,*} \qquad \text{(a1)} \\
&= (\tilde{\mathbf{x}}_{a_t,t}, \tilde{\mathbf{v}}_{p,a_t,t})^T \hat{\theta}_{p,u,t} + \alpha_t^\theta \|(\tilde{\mathbf{x}}_{a_t,t}, \tilde{\mathbf{v}}_{p,a_t,t})\|_{\mathbf{A}_{p,u,t}^{-1}} \\
&\quad + \alpha_t^v \sum_{j \in \mathcal{A}} \mathrm{M}_{p,t}^{a_t,j} \|\hat{\theta}_{p,u,t}^v\|_{\mathrm{C}_{p,j,t}^{-1}} + \alpha_t^\theta \|\tilde{\mathbf{v}}_{p,a_t^*,*} - \tilde{\mathbf{v}}_{p,a_t^*,t}\|_{\mathbf{A}_{p,u,t}^{-1}} - (\tilde{\mathbf{x}}_{a_t,t}, \tilde{\mathbf{v}}_{p,a_t,*})^T \theta_{p,u,*} \qquad \text{(a2)} \\
&\leq 2\alpha_t^\theta \|(\tilde{\mathbf{x}}_{a_t,t}, \tilde{\mathbf{v}}_{p,a_t,t})\|_{\mathbf{A}_{p,u,t}^{-1}} + 2\alpha_t^v \sum_{j \in \mathcal{A}} \mathrm{M}_{p,t}^{a_t,j} \|\hat{\theta}_{p,u,t}^v\|_{\mathrm{C}_{p,j,t}^{-1}} \\
&\quad + \alpha_t^\theta \|\tilde{\mathbf{v}}_{p,a_t^*,*} - \tilde{\mathbf{v}}_{p,a_t^*,t}\|_{\mathbf{A}_{p,u,t}^{-1}} + \alpha_t^\theta \|\tilde{\mathbf{v}}_{p,a_t,*} - \tilde{\mathbf{v}}_{p,a_t,t}\|_{\mathbf{A}_{p,u,t}^{-1}} \qquad \text{(a3)}
\end{aligned}$$

where the inequality $(a2)$ is according to the arm-selection strategy, while inequalities $(a1)$ and $(a3)$ are out of:

$$
\begin{aligned}
&|(\tilde{\mathbf{x}}_{a,t}, \tilde{\mathbf{v}}_{p,a,*})^T \theta_{p,u,*} - (\tilde{\mathbf{x}}_{a,t}, \tilde{\mathbf{v}}_{p,a,t})^T \hat{\theta}_{p,u,t}| \\
&= |(\tilde{\mathbf{x}}_{a,t}, \tilde{\mathbf{v}}_{p,a,*})^T (\theta_{p,u,*} - \hat{\theta}_{p,u,t}) + (\tilde{\mathbf{v}}_{p,a,*} - \tilde{\mathbf{v}}_{p,a,t})^T \hat{\theta}^v_{p,u,t}| \\
&= |(\tilde{\mathbf{x}}_{a,t}, \tilde{\mathbf{v}}_{p,a,t})^T (\theta_{p,u,*} - \hat{\theta}_{p,u,t}) + (\tilde{\mathbf{v}}_{p,a,*} - \tilde{\mathbf{v}}_{p,a,t})^T \hat{\theta}^v_{p,u,t} + (\tilde{\mathbf{v}}_{p,a,*} - \tilde{\mathbf{v}}_{p,a,t})^T (\theta_{p,u,*} - \hat{\theta}_{p,u,t})| \\
&\leq \alpha^\theta_t \|(\tilde{\mathbf{x}}_{a,t}, \tilde{\mathbf{v}}_{p,a,t})\|_{\mathbf{A}^{-1}_{p,u,t}} + \alpha^v_t \sum_{j \in \mathcal{A}} \mathrm{M}^{a,j}_{p,t} \|\hat{\theta}^v_{p,u,t}\|_{\mathbf{C}^{-1}_{p,j,t}} + \alpha^\theta_t \|\tilde{\mathbf{v}}_{p,a,*} - \tilde{\mathbf{v}}_{p,a,t}\|_{\mathbf{A}^{-1}_{p,u,t}}.
\end{aligned}
$$

Then the cumulative regret up to T is bounded by

$$
\begin{aligned}
R^p(T) &= \sum_{t=1}^{T} R^p_t \\
&\leq 2 \sum_{t=1}^{T} \alpha^\theta_t \|(\tilde{\mathbf{x}}_{a_t,t}, \tilde{\mathbf{v}}_{p,a_t,t})\|_{\mathbf{A}^{-1}_{p,u,t}} + 2 \sum_{t=1}^{T} \alpha^v_t \sum_{j \in \mathcal{A}} \|\mathrm{M}^{a_t,j}_{p,t} \hat{\theta}^v_{p,u,t}\|_{\mathbf{C}^{-1}_{p,j,t}} + 2 \sum_{t=1}^{T} \alpha^\theta_t \|\tilde{\mathbf{v}}_{p,a_t,*} - \tilde{\mathbf{v}}_{p,a_t,t}\|_{\mathbf{A}^{-1}_{p,u,t}} \\
&\leq 2\alpha^\theta_T \sqrt{T \sum_{t=1}^{T} \|(\tilde{\mathbf{x}}_{a_t,t}, \tilde{\mathbf{v}}_{p,a_t,t})\|^2_{\mathbf{A}^{-1}_{p,u,t}}} + 2\alpha^v_T \sum_{j \in \mathcal{A}} \sqrt{T \sum_{t=1}^{T} \|\mathrm{M}^{a_t,j}_{p,t} \hat{\theta}^v_{p,u,t}\|^2_{\mathbf{C}^{-1}_{p,j,t}}} \\
&\quad + 2 \frac{\alpha^\theta_T}{\sqrt{\lambda_1}} \sum_{t=1}^{T} \|\tilde{\mathbf{v}}_{p,a_t,*} - \tilde{\mathbf{v}}_{p,a_t,t}\|_2.
\end{aligned} \tag{11}
$$

Following the similar procedure of Theorem 3 in article [1], we can get:

$$
\begin{aligned}
&\sum_{t=1}^{T} \|(\tilde{\mathbf{x}}_{a_t,t}, \tilde{\mathbf{v}}_{p,a_t,t})\|^2_{\mathbf{A}^{-1}_{p,u,t}} \\
&\leq 2 \log \left(\frac{\det(\mathbf{A}_{p,u,t})}{\det(\lambda_1 \mathbf{I}_1)} \right) \leq 2(d+l) \log \left(1 + \frac{\sum_{t=1}^{T} \| \sum_{j \in \mathcal{A}} \mathrm{M}^{a_t,j}_{p,t} (\mathbf{x}_{j,t}, \hat{\mathbf{v}}_{p,j,t})\|^2_2}{\lambda_1 (d+l)} \right),
\end{aligned}
$$

and

$$
\sum_{t=1}^{T} \|\mathrm{M}^{a_t,j}_{p,t} \hat{\theta}^v_{p,u,t}\|^2_{\mathbf{C}^{-1}_{p,j,t}} \leq 2 \log \left(\frac{\det(\mathbf{C}_{p,u,t})}{\det(\lambda_2 \mathbf{I}_2)} \right) \leq 2l \log \left(1 + \frac{\sum_{t=1}^{T} (\mathrm{M}^{a_t,j}_{p,t})^2 S^2}{\lambda_2 l} \right).
$$

Moreover, with the q-linearly convergence of $\hat{\mathbf{v}}_{p,j,t}$, we can concludes the proof. □

7 RELATED WORK

To the best of our knowledge, no previous work has studied contextual bandit with a heterogeneous information network. However, our work is closely related to the following two lines of work.

Contextual bandit algorithms. Contextual bandit is an important technique to balance the exploitation-exploration tradeoff, in various applications such as recommender systems and information retrieval [8, 27]. LinUCB [15] and Thompson Sampling [4] are two representative algorithms for contextual bandits. A large number of algorithms have leveraged various side information to assist bandit learning. For example, relationships among users were leveraged in [6, 19, 29, 30]. And in this article, we only compare with [29] since it has the best performance among these works. Wang et al. [28] developed the hLinUCB algorithm to learn hidden features in contextual bandit. Zeng et al. [32] designed algorithms for contextual bandits with a time-varying

reward function. Above algorithms either do not leverage relations among users and arms, or leverage only one type of relation. Different from them, in this article, we simultaneously leverage rich relations from heterogeneous information network to assist bandit learning. Two previous works [3, 23] also designed algorithms to combine multiple bandit algorithms. However, they consider a different setting, where each time only the selected base bandit algorithm can be updated. In our work, each base bandit algorithm captures users' preference under the corresponding meta-path, thus we need to update each base bandit algorithm with the received feedback. The difference in problem settings requires us to design different weight updating procedure and arm selection strategy. Moreover, although in this work, all base bandit algorithms are built on hLinUCB [28], it is straightforward to leverage other base bandit algorithms of non-linear reward model [17] or Thompson Sampling [4], and so on. We are also aware of works in the research line of applying deep reinforcement learning algorithms to recommendations [7, 35–37]. Note that deep reinforcement learning algorithms requires a large amount of training data, which makes them not suitable for the cold-start scenario. Our HUCB does not require training data and it is suitable for the cold-start scenario. Furthermore, these algorithms do not have theoretical guarantees on the regret, while our HUCB has such theoretical guarantees.

HIN and its application in recommendation. HIN is a powerful tool to capture multiple heterogeneous relations among users and items [24]. Our work incorporates HIN into contextual bandit through the similarity measure in HIN [25]. Interested readers can refer to [12] for clustering algorithms in HIN, refer to [21] for relevance measure in HIN and refer to monograph [24] for a thorough treatment of HIN. Several algorithms were proposed to tackle the recommendation task based on HIN. Based on existing data, Yu et al. [31] proposed a framework, which first learns users' and items' latent features under multiple meta-paths, then combines these latent features by a weighted mechanism to do recommendation. Shi et al. [22] took users' ratings to items to build a weighted HIN, based on which meta-path based methods are used to do recommendation. Zhao et al. [33] further generalized meta-path to meta-graph, and combined it with factorization machine for a recommendation. Gupta et al. [9] proposed to use personalized weight of meta-paths in HIN to do recommendation. Hu et al. [11] applied meta-paths to do top-n recommendations. Shi et al. [20] utilized the embedding of HIN to do recommendation. Information fusion-based approaches for utilizing HIN for recommendation were proposed in [10, 33, 34]. Jin et al. [13] proposed an efficient neighborhood-based interaction model for recommendation in HIN. However, these algorithms are only applied to offline learning, while our algorithm, based on the bandit technique, is an online learning algorithm. Moreover, our algorithm can be easily extended to leverage weighted HIN and meta-Graph.

8 CONCLUSION

This article proposes a novel contextual bandit framework, which utilizes a given HIN to improve bandit learning. We develop the HUCB algorithm to leverage rich heterogeneous information in HIN by dynamic ensembling a set of base bandit algorithms that learn users' preferences under different meta-paths. We prove that the HUCB algorithm can achieve similar performance as compared with the optimal algorithm where each user is served according to his true preference over meta-paths (assuming the optimal algorithm knows this preference). Moreover, the HUCB algorithm is proved to benefit from leveraging HIN in achieving a smaller regret upper bound, compared to the baseline algorithm without leveraging HIN. Experiments on synthetic datasets, as well as real datasets from LastFM and Yelp demonstrate the superior performance of the HUCB algorithm.

REFERENCES

[1] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. 2011. Improved algorithms for linear stochastic bandits. In *Proceedings of the Advances in Neural Information Processing Systems*. 2312–2320.

[2] Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. 2014. Taming the monster: A fast and simple algorithm for contextual bandits. In *Proceedings of the International Conference on Machine Learning*. 1638–1646.

[3] Alekh Agarwal, Haipeng Luo, Behnam Neyshabur, and Robert E. Schapire. 2017. Corralling a Band of Bandit Algorithms. In *Proceedings of Annual Conference on Learning Theory*. 12–38.

[4] Shipra Agrawal and Navin Goyal. 2013. Thompson sampling for contextual bandits with linear payoffs. In *Proceedings of the International Conference on Machine Learning*. 127–135.

[5] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. 2002. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing* 32, 1 (2002), 48–77.

[6] Nicolo Cesa-Bianchi, Claudio Gentile, and Giovanni Zappella. 2013. A gang of bandits. In *Proceedings of the Advances in Neural Information Processing Systems*. 737–745.

[7] Haokun Chen, Xinyi Dai, Han Cai, Weinan Zhang, Xuejian Wang, Ruiming Tang, Yuzhou Zhang, and Yong Yu. 2019. Large-scale interactive recommendation with tree-structured policy gradient. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 3312–3320.

[8] Dorota Glowacka. 2019. Bandit algorithms in information retrieval. *Foundations and Trends in Information Retrieval* 13, 4 (2019), 299–424.

[9] Mukul Gupta and Pradeep Kumar. 2020. Recommendation generation using personalized weight of meta-paths in heterogeneous information networks. *European Journal of Operational Research* 284, 2 (2020), 660–674.

[10] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Tianchi Yang. 2018. Local and global information fusion for top-n recommendation in heterogeneous information network. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 1683–1686.

[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 *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1531–1540.

[12] Yue Huang and Xuedong Gao. 2014. Clustering on heterogeneous networks. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 4, 3 (2014), 213–233.

[13] Jiarui Jin, Jiarui Qin, Yuchen Fang, Kounianhua Du, Weinan Zhang, Yong Yu, Zheng Zhang, and Alexander J. Smola. 2020. An efficient neighborhood-based interaction model for recommendation on heterogeneous graph. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 75–84.

[14] John Langford and Tong Zhang. 2008. The epoch-greedy algorithm for multi-armed bandits with side information. In *Proceedings of the Advances in Neural Information Processing Systems*. 817–824.

[15] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*. ACM, 661–670.

[16] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. 2011. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*. ACM, 297–306.

[17] Lihong Li, Yu Lu, and Dengyong Zhou. 2017. Provably optimal algorithms for generalized linear contextual bandits. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2071–2080.

[18] Shuai Li, Wei Chen, Shuai Li, and Kwong-Sak Leung. 2019. Improved algorithm on online clustering of bandits. In *International Joint Conference on Artificial Intelligence*. 2923–2929.

[19] Shuai Li, Alexandros Karatzoglou, and Claudio Gentile. 2016. Collaborative filtering bandits. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 539–548.

[20] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and Philip S. Yu. 2018. Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering* 31, 2 (2018), 357–370.

[21] Chuan Shi, Xiangnan Kong, Yue Huang, Philip S. Yu, and Bin Wu. 2014. Hetesim: A general framework for relevance measure in heterogeneous networks. *IEEE Transactions on Knowledge and Data Engineering* 26, 10 (2014), 2479–2492.

[22] Chuan Shi, Zhiqiang Zhang, Ping Luo, Philip S. Yu, Yading Yue, and Bin Wu. 2015. Semantic path based personalized recommendation on weighted heterogeneous information networks. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 453–462.

[23] Adish Singla, Hamed Hassani, and Andreas Krause. 2018. Learning to interact with learning agents. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.

[24] Yizhou Sun and Jiawei Han. 2012. Mining heterogeneous information networks: Principles and methodologies. *Synthesis Lectures on Data Mining and Knowledge Discovery* 3, 2 (2012), 1–159.

[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. *Proceedings of the VLDB Endowment* 4, 11 (2011), 992–1003.

[26] André Uschmajew. 2012. Local convergence of the alternating least squares algorithm for canonical tensor approximation. *SIAM Journal on Matrix Analysis and Applications* 33, 2 (2012), 639–652.

[27] Huazheng Wang, Yiling Jia, and Hongning Wang. 2021. Interactive information retrieval with bandit feedback. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2658–2661.

[28] Huazheng Wang, Qingyun Wu, and Hongning Wang. 2016. Learning hidden features for contextual bandits. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 1633–1642.

[29] Huazheng Wang, Qingyun Wu, and Hongning Wang. 2017. Factorization bandits for interactive recommendation. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*.

[30] Qingyun Wu, Huazheng Wang, Quanquan Gu, and Hongning Wang. 2016. Contextual bandits in a collaborative environment. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 529–538.

[31] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. 2014. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*. ACM, 283–292.

[32] Chunqiu Zeng, Qing Wang, Shekoofeh Mokhtari, and Tao Li. 2016. Online context-aware recommendation with time varying multi-armed bandit. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2025–2034.

[33] Huan Zhao, Quanming Yao, Jianda Li, Yangqiu Song, and Dik Lun Lee. 2017. Meta-graph based recommendation fusion over heterogeneous information networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 635–644.

[34] Huan Zhao, Quanming Yao, Yangqiu Song, James T. Kwok, and Dik Lun Lee. 2021. Side information fusion for recommender systems over heterogeneous information network. *ACM Transactions on Knowledge Discovery from Data* 15, 4 (2021), 1–32.

[35] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. 2018. DRN: A deep reinforcement learning framework for news recommendation. In *Proceedings of the 2018 World Wide Web Conference*. 167–176.

[36] Lixin Zou, Long Xia, Zhuoye Ding, Jiaxing Song, Weidong Liu, and Dawei Yin. 2019. Reinforcement learning to optimize long-term user engagement in recommender systems. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2810–2818.

[37] Lixin Zou, Long Xia, Pan Du, Zhuo Zhang, Ting Bai, Weidong Liu, Jian-Yun Nie, and Dawei Yin. 2020. Pseudo dyna-q: A reinforcement learning framework for interactive recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 816–824.

Received July 2021; revised December 2021; accepted February 2022