2021 IEEE 37th International Conference on Data Engineering (ICDE)

Heterogeneous Information Assisted Bandit Learning: Theory and Application

Xiaoying Zhang
CSE Department
The Chinese University of Hong Kong
xyzhang@cse.cuhk.edu.hk

Hong Xie
College of Computer Science
Chongqing University, China
xiehong2018@cqu.edu.cn

John C.S. Lui
CSE Department
The Chinese University of Hong Kong
cslui@cse.cuhk.edu.hk

2021 IEEE 37th International Conference on Data Engineering (ICDE) | 978-1-7281-9184-3/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICDE51399.2021.00213

***Abstract*—Contextual bandit serves as an invaluable tool to balance the *exploration vs. exploitation* trade-off in various applications like online recommendation. In many applications, heterogeneous information network (HIN) can be derived to provide rich side information for contextual bandits, such as different types of attributes and relationships among users and items. In this paper, 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. We propose the HUCB algorithm to address such a challenge. For each meta-path, the HUCB algorithm employs an independent base bandit algorithm to handle online item recommendation by leveraging the relationship captured in this meta-path. The bandit master is then employed to learn users' preference over meta-paths to dynamically combine base bandit algorithms with a balance of exploration-exploitation trade-off. Experimental results on real datasets from LastFM and Yelp demonstrate the efficacy of the HUCB algorithm.**

I. Introduction

Contextual bandit provides a principled online method to optimize the performance of various systems, e.g., recommender systems, through learning from interactions with the user. For the contextual bandit based online recommendation algorithms [1], [2], each item is mapped as an arm in the contextual bandit, the observed information of an item with regarding 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 to 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 feedbacks in the long run can be optimized, i.e., cumulative reward is maximized. In general, the algorithm needs to make a trade-off between exploitation (i.e., leveraging users' known preference) and exploration (i.e., revealing users' unknown preference).

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, 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 [2], [3], or leverage only one single relationship, e.g., users' friendships [4], [5], [6]. This paper is the first to utilize rich heterogeneous information to assist bandit learning.

This paper 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. Formally, the HIN [7] 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), etc. Each meta-path defines a new composite relation on HIN. 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 manner while balancing the exploration-exploitation trade-off. To address the challenge, we design the HUCB algorithm. In the HUCB algorithm, users' preference over arms under different meta-paths are learned online by a group of independent base bandit algorithms which handle the exploitation-exploration trade-off. Furthermore, we learn the user's preference over meta-paths based on the performance of base bandit algorithms, i.e., if one base bandit algorithm can predict the user's preference over arms more accurately, the user's preference on this meta-path will be enlarged. However, inferring users' preference

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

978-1-7281-9184-3/21/$31.00 ©2021 IEEE
DOI 10.1109/ICDE51399.2021.00213

over meta-paths solely based on historical performance of base bandit algorithms will lead to an suboptimal solution, 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 on may not be selected. Thus, we also develop a bandit master to dynamically ensemble base bandit algorithms while balancing the explore/exploit trade-off in learning user's preference over meta-paths. Experimental results on 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 II).
- We design the HUCB algorithm for HIN-assisted contextual bandit by dynamic ensembling a set of base bandit algorithms that learn users' preference under different meta-paths (Section III).
- We conduct extensive experiments on real datasets from Yelp and LastFM, and demonstrate the efficacy of the HUCB algorithm (Section IV).

II. Problem Formulation

In this section, we first briefly introduce the traditional contextual bandit, then we generalize it to leverage heterogeneous information represented in a *heterogeneous information network* (HIN).

A. Contextual Bandit

In contextual bandit, given a finite set of $N \in \mathbb{N}_+$ arms denoted by $\mathcal{A}$, an agent aims to maximize 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 $\{\mathbf{x}_{a,t}\}_{a \in \mathcal{A}_t}$, as well as the selected arms and received rewards at 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., $\mathbb{E}[r_{u,a_t^*,t}] \geq \mathbb{E}[r_{u,a,t}], \forall a \in \mathcal{A}_t$. The goal of contextual bandit is to minimize the cumulative regret in T rounds:

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

A smaller regret $R(T)$ implies that the cumulative reward is close to the optimal cumulative reward. The agent needs to make a trade-off 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 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 is what the agent wants to learn. Let ϵ_t denote a random variable representing the random noise in the reward. The reward in the LinUCB algorithm [2] is:

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

while hLinUCB algorithm [3] 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.

B. HIN-assisted Contextual Bandit

Previous works estimate $\{\theta_u\}$ (and $\{\mathbf{v}_a\}$ if applicable) either independently for each user (for each arm) [2], [3], or considering a single relationship, for example, users' friendship [6]. 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 parameter learning of other users or arms. 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 [8], [9]. Moreover, different types of relations among users and arms can be obtained in heterogeneous information network and we aim to leverage those relations to assist bandit learning.

Heterogeneous information network (HIN). 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, etc.;*
- *$\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ 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 $v_2 \in \mathcal{V}$;*
- *$\mathcal{K}$ 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 $[v_1, v_2] \in \mathcal{E}$.*

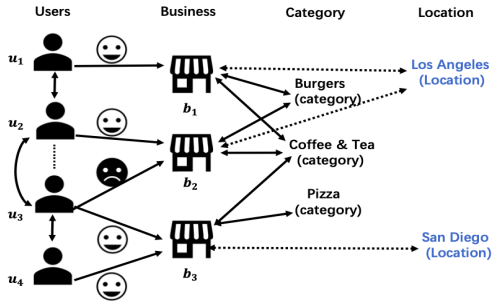


Fig. 1. A single example HIN from Yelp.

Figure 1 shows an example of heterogeneous information network built on Yelp. It contains four types of nodes, i.e., $\mathcal{K} =$ {"user", "business", "location", "category"}, and four types of links, i.e., $\mathcal{R} =$ { "user→business", "business↔location", "business↔category", "user↔user"}. One can observe that $\phi(u_1)$="user", $\phi(b_1)$="business" and $\psi(u_1 \rightarrow b_1) =$ "user→business".

In this paper, 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 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. For example, Figure 1 satisfies above property and $\psi(u_1 \rightarrow b_1) = \psi(u_3 \rightarrow b_2) =$ "user→business". To simplify presentation, we define a relation function $R : \mathcal{K} \times \mathcal{K} \rightarrow \mathcal{R}$ to capture above property, which satisfies that $\psi(v_1 \rightarrow v_2) = R(\phi(v_1), \phi(v_2))$, where $v_1, v_2 \in \mathcal{V}_t$.

To extract rich relations from HIN, one can use the meta-path technique [8], [9], [10]. 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*

$$p \triangleq (K_0 \rightarrow K_1 \rightarrow \ldots \rightarrow K_m),$$

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' preference on the business with similar categories. The semantics of a path $(v_0 \rightarrow v_1 \rightarrow \ldots \rightarrow v_m)$ in a HIN, where $v_0, v_1, \ldots, v_m \in \mathcal{V}_t$, can be summarized by a meta-path $p = (\phi(v_0) \rightarrow \phi(v_1) \rightarrow \ldots \rightarrow \phi(v_m))$. For example, the semantics of the path "$u_1 \rightarrow b_1 \rightarrow$ Coffee&Tea $\rightarrow b_2$" and the path "$u_2 \rightarrow b_2 \rightarrow$ Coffee&Tea $\rightarrow b_3$" 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 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 Eq. (1) is minimized.

III. 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.

A. 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*. Then, two base bandit algorithms are developed to leverage above two types of similarities.

Similarities induced by a meta-path. This paper mainly focuses on two classes of meta-paths characterized by the format "user→ $\ldots$ →user→arm" or "user→arm→ $\ldots$ →arm", formally:

- *User-centric meta-path:* $p = (K_0 \rightarrow \ldots \rightarrow K_{m-1} \rightarrow K_m)$ with $K_0 = K_{m-1} =$"user" and $K_m =$"arm".
- *Arm-centric meta-path:* $p = (K_0 \rightarrow K_1 \rightarrow \ldots \rightarrow K_m)$ with $K_0 =$"user" and $K_1 = K_m =$"arm".

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.

Given a user-centric (or an arm-centric) meta-path, we apply the commonly-used approach, i.e., computing commuting matrices [7], to quantify similarities among users (or among arms). For a user-centric (or an arm-centric) meta-path p, we denote the derived similarity matrix between users, i.e., between K_0 and K_{m-1} (or between arms, i.e., between K_1 and K_m) as $\tilde{\boldsymbol{S}}_{p,t}$ ($\boldsymbol{S}_{p,t}$). Due to page limit, one can refer to technical report [11] for details.

Base bandit algorithm for user-centric meta-path. Wang et. al. [6] 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, with similarities among users as $\widetilde{\mathbf{S}}_{p,t}$.

Base bandit algorithm for arm-centric meta-path. Due to page limit, please refer to technical report [11] for details of base bandit algorithms for arm-centric meta-path.

B. A Dynamic Ensemble of Base Bandit Algorithms

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 III-A to leverage the relation under meta-path p. Next, we consider how to learn users' preference over different meta-paths so 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 an 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 trade-off balanced, so to dynamically ensemble base bandit algorithms.

More specifically, the bandit master uses the vector $\mathbf{w}_{u,t} = [w^1_{u,t}, \ldots, w^{|\mathcal{P}|}_{u,t}] \in \mathbb{R}^{|\mathcal{P}|}$ 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-exploitation trade-off, since $\mathbf{w}_{u,t}$ is updated based on 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. 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 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

Algorithm 1: The HUCB algorithm

Input: $\lambda_1, \lambda_2 \in (0, +\infty), \gamma, \beta \in (0, 1)$.
Init: $\mathbf{w}^i_{u,0} = \frac{1}{|\mathcal{P}|}, \forall i = 1, \ldots |\mathcal{P}|$;
for $p = 1, 2, \ldots, |\mathcal{P}|$ **do**
  initialize base bandit algorithm p;
1 **for** $t = 1, 2, ..., T$ **do**
2 set $\hat{w}^p_{u,t} = (1-\gamma)\frac{w^p_{u,t-1}}{\sum_j w^j_{u,t-1}} + \frac{\gamma}{|\mathcal{P}|}$, for $p \in \mathcal{P}$;
3 sample a base bandit algorithm p_t according to $\hat{\mathbf{w}}_{u,t}$;
4 select the arm a_t using base bandit algorithm p_t;
5 get the user's feedback $r_{u,a_t,t}$;
6 **for** $p = 1, 2, \ldots, |\mathcal{P}|$ **do**
7 with the interaction record $(u, a_t, r_{u,a_t,t})$, update the base bandit algorithm p;
8 take $l_{p,t} = \frac{r_{u,a_t,t}}{\sum_{p':a^{p'}_t = a_t} \hat{w}^{p'}_{u,t}}$ if $a^p_t = a_t$, otherwise $l_{p,t} = 0$;
9 update $w^p_{u,t}$ by $w^p_{u,t} = w^p_{u,t-1}\exp(\eta l_{p,t})$.

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 $6-9$). The updating process includes two parts: (1) updating the base bandit model (line 7); (2) updating the weight vector $\mathbf{w}_{u,t}$ (lines $8-9$): 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 $\eta \cdot \frac{r_{u,a_t,t}}{\sum_{p':a^{p'}_t = a_t} \hat{w}^{p'}_{u,t}}$, 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 [12], we note that one can also use other algorithms that learns from experts [13].

C. Regret Analysis of HUCB

Due to space limit, one can refer to technical report [11] for more details of regret analysis.

IV. Experiments on Real Datasets

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

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

- LinUCB [2]: 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 [3]: it extends LinUCB to consider hidden features, but it does not leverage any other relations.
- factorUCB [6]: it builds from hLinUCB while considering users' friendships. It is the base bandit algorithm under the meta-path "user→user→arm".
- Best base bandit: the base bandit algorithm with the best performance.

- HUCB-EW: a variant of HUCB that randomly selects base bandit algorithms at each round t, i.e., $w^p_{u,t} = \frac{1}{|\mathcal{P}|}, \forall p \in \mathcal{P}$.

A. Experiments on LastFM Dataset.

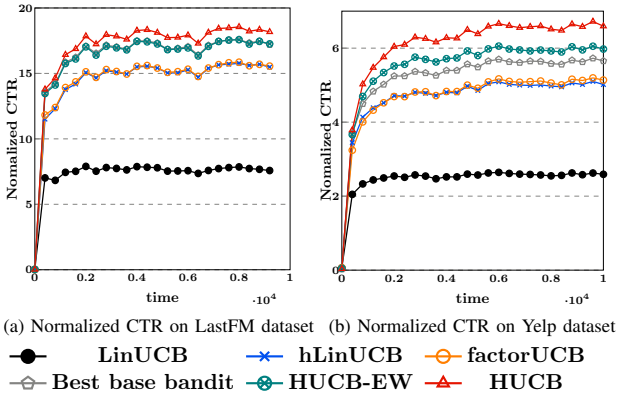
The LastFM dataset is extracted from the online music streaming service Last.fm[2]. It contains three types of nodes: "user", "artist" and "tag", and four types of edges: "user↔user", "user→artist", "artist↔tag", "user→tag". The LastFM dataset contains $1,892$ users and $17,362$ artists. We take each 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 [3], 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"$\}$.

The unbiased offline evaluation protocol proposed in [14] is applied to evaluate algorithms. At each time t, we store the arm presented to the user (a_t), and its received feedback. Then we create the candidate pool $\mathcal{A}_t$ 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 reward 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 [2], we normalize the resulting CTR from different algorithms by the corresponding logged random strategy's CTR.

Evaluation results. Figure 2a 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 that under the meta-path "user→artist→tag→artist", and its performance is similar with the HUCB-EW algorithm. Although it may not be obvious in Figure 2a due to the scale of y-values, the factorUCB algorithm is slightly better than the hLinUCB algorithm, especially in the beginning phrase.

B. 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 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 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 feedbacks.



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

Fig. 2. Experimental results on real datasets.

Evaluation results. Following similar procedure of experiments on LastFM dataset, we compare all algorithms with normalized CTRs. The results are shown in Figure 2b. Similarly, we can observe that HUCB achieves the highest CTRs, followed by HUCB-EW, the best base bandit algorithm, factorUCB, hLinUCB and LinUCB. On Yelp dataset, the best base bandit algorithm is the one under the meta-path "user→item→location→item". 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.

V. Related work

To the best of our knowledge, no previous work has studied contextual bandit with heterogeneous information network.

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

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

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

However, two lines of work are closely related.

Contextual bandit algorithms. Contextual bandit is an important technique to balance the exploitation-exploration trade-off, in various applications such as recommender systems and information retrieval [15]. LinUCB [2] and Thompson Sampling [16] are two representative algorithms for contextual bandits. A large number of algorithms have leveraged various side information to assist bandit learning. For example, relationship among users were leveraged in [4], [5], [6]. And in this paper, we only compare with [6] since it has the best performance among these works. Wang *et. al.* [3] developed the hLinUCB algorithm to learn hidden features in contextual bandit. Zeng *et. al.* [17] 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 paper, we simultaneously leverage rich relations from heterogeneous information network to assist bandit learning. Two previous works [18], [19] 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, it is straightforward to leverage other base bandit algorithms of non-linear reward model [20], Thompson Sampling [16], etc.

Recommendation with HIN. Several algorithms were proposed to tackle the recommendation task based on HIN. Based on existed data, Yu *et. al.* [8] 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.* [9] 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.* [10] further generalized meta-path to meta-graph, and combined it with factorization machine for recommendation. 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.

VI. Conclusion

This paper 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' preference under different meta-paths. Experiments on real datasets from LastFM and Yelp demonstrate the superior performance of the HUCB algorithm.

VII. Acknowledgments

The work of John C.S. Lui is supported in part by the GRF 14201819. The work of Hong Xie was supported in part by National Nature Science Foundation of China (61902042), Chongqing Natural Science Foundation (cstc2020jcyj-msxmX0652), and the Fundamental Research Funds for the Central Universities (2020CDJ-LHZZ-057). Hong Xie is the corresponding author.

References

[1] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári, "Improved algorithms for linear stochastic bandits," in *Advances in Neural Information Processing Systems*, 2011, pp. 2312–2320.

[2] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proc. of WWW*. ACM, 2010, pp. 661–670.

[3] H. Wang, Q. Wu, and H. Wang, "Learning hidden features for contextual bandits," in *Proc. of ACM CIKM*. ACM, 2016, pp. 1633–1642.

[4] Q. Wu, H. Wang, Q. Gu, and H. Wang, "Contextual bandits in a collaborative environment," in *Proc. of ACM SIGIR*. ACM, 2016, pp. 529–538.

[5] S. Li, A. Karatzoglou, and C. Gentile, "Collaborative filtering bandits," in *Proc. of ACM SIGIR*. ACM, 2016, pp. 539–548.

[6] H. Wang, Q. Wu, and H. Wang, "Factorization bandits for interactive recommendation," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[7] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, 2011.

[8] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han, "Personalized entity recommendation: A heterogeneous information network approach," in *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM, 2014, pp. 283–292.

[9] C. Shi, Z. Zhang, P. Luo, P. S. Yu, Y. Yue, and B. Wu, "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, 2015, pp. 453–462.

[10] H. Zhao, Q. Yao, J. Li, Y. Song, and D. L. Lee, "Meta-graph based recommendation fusion over heterogeneous information networks," in *Proc. of ACM SIGKDD*. ACM, 2017, pp. 635–644.

[11] X. Zhang, H. Xie, and J. C. Lui, "[technical report] heterogeneous information assisted bandit learning: Theory and application," https://drive.google.com/file/d/1fcrIFA50L3Tp3JRFvbJbXvtKUNTrTTMV/view?usp=sharing, 2019, online.

[12] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The non-stochastic multiarmed bandit problem," *SIAM journal on computing*, vol. 32, no. 1, pp. 48–77, 2002.

[13] A. Agarwal, D. Hsu, S. Kale, J. Langford, L. Li, and R. Schapire, "Taming the monster: A fast and simple algorithm for contextual bandits," in *International Conference on Machine Learning*, 2014, pp. 1638–1646.

[14] L. Li, W. Chu, J. Langford, and X. Wang, "Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms," in *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 2011, pp. 297–306.

[15] D. Glowacka *et al.*, "Bandit algorithms in information retrieval," *Foundations and Trends® in Information Retrieval*, vol. 13, no. 4, pp. 299–424, 2019.

[16] S. Agrawal and N. Goyal, "Thompson sampling for contextual bandits with linear payoffs," in *International Conference on Machine Learning*, 2013, pp. 127–135.

[17] C. Zeng, Q. Wang, S. Mokhtari, and T. Li, "Online context-aware recommendation with time varying multi-armed bandit," in *KDD*. ACM, 2016, pp. 2025–2034.

[18] A. Agarwal, H. Luo, B. Neyshabur, and R. E. Schapire, "Corralling a band of bandit algorithms," *arXiv preprint arXiv:1612.06246*, 2016.

[19] A. Singla, H. Hassani, and A. Krause, "Learning to interact with learning agents," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[20] L. Li, Y. Lu, and D. Zhou, "Provably optimal algorithms for generalized linear contextual bandits," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 2071–2080.