Interactive Unknowns Recommendation in E-Learning Systems

Shan-Yun Teng*, Jundong Li†, Lo Pang-Yun Ting*, Kun-Ta Chuang*, Huan Liu†
* Dept. of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan
† School of Computing, Informatics and Decision Systems Engineering, Arizona State University, Tempe, USA
syteng@netdb.csie.ncku.edu.tw, jundongl@asu.edu, lpyting@netdb.csie.ncku.edu.tw, ktchuang@mail.ncku.edu.tw, huan.liu@asu.edu

Abstract*—The arise of E-learning systems has led to an anytime-anywhere-learning environment for everyone by providing various online courses and tests. However, due to the lack of teacher-student interaction, such ubiquitous learning is generally not as effective as offline classes. In traditional offline courses, teachers facilitate real-time interaction to teach students in accordance with personal aptitude from students' feedback in classes. Without the interruption of instructors, it is difficult for users to be aware of personal unknowns. In this paper, we address an important issue on the exploration of user unknowns from an interactive question-answering process in E-learning systems. A novel interactive learning system, called *CagMab*, is devised to interactively recommend questions with a round-by-round strategy, which contributes to applications such as a conversational bot for self-evaluation. The flow enables users to discover their weakness and further helps them to progress. In fact, despite its importance, discovering personal unknowns remains a challenging problem in E-learning systems. Even though formulating the problem with the multi-armed bandit framework provides a solution, it often leads to suboptimal results for interactive unknowns recommendation as it simply relies on the contextual features of answered questions. Note that each question is associated with concepts and similar concepts are likely to be linked manually or systematically, which naturally forms the concept graphs. Mining the rich relationships among users, questions and concepts could be potentially helpful in providing better unknowns recommendation. To this end, in this paper, we develop a novel interactive learning framework by borrowing strengths from concept-aware graph embedding for learning user unknowns. Our experimental studies on real data show that the proposed framework can effectively discover user unknowns in an interactive fashion for the recommendation in E-learning systems.

***Keywords*-Unknowns Recommender System; Multi-Armed Bandit; Concept-Aware Graph Embedding; E-learning System;**

I. Introduction

E-learning systems such as Coursera and edX are so convenient and flexible that users can choose abundant knowledge they want to learn. However, such flexibility cannot guarantee the education quality and may lead to the dilemma that users do not know what knowledge they should learn at the present stage, especially for the youth. As illustrated in Figure 1, the user is familiar with geometry, while not good at arithmetic. E-learning systems should provide supplementary resources to enrich the knowledge of users' personal unknowns. Unfortunately, according to our analysis on the real log of user learning process, most users are not aware of his/her personal unknowns. As shown in Figure 2, we draw a case study on 100 users for their known and unknown rate of the answered questions. The studied dataset is provided from a nonprofit-based E-learning service. The known (unknown) rate of a user refers to the fraction of answered questions which are correctly (incorrectly) answered by this user. Obviously, the known rates of most users are higher than their unknown rates. The conclusion can be drawn that most users tend to practice questions which they knew answers because of their unawareness of personal unknowns. In addition, solving questions which users already understood cannot effectively enrich their knowledge because of the lack of unknowns exploration. How to recommend users their unknowns will be the key to the success of learning new knowledge.

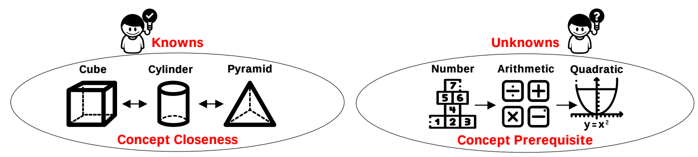


Fig. 1. The illustrative examples of user knowns, user unknowns, concept closeness, and concept prerequisite.

Traditional education policy places considerable significance on interactive learning for students. However, different from the offline classes in schools, it is extremely difficult to perform online interactive recommendation in E-learning systems without the interruption of teachers. The need for online interaction for users in E-learning systems grows because of two reasons. First, without the face-to-face interaction, there is no instructor for users to interact with and learn from on the platform. Second, users' knowledge changes over time, which cannot easily be aware by themselves. How to exploit both users' existing unknowns and user feedback in exploring new unknowns interactively becomes an important issue. However, the traditional offline training paradigm for recommendation becomes incompetent to handle such exploration-exploitation dilemma. The need for both knowledge exploitation and knowledge exploration indicate the necessity to propose an interactive unknowns recommendation.

To address this issue, we try to interactively explore user unknowns from users' question-answer records. Generally, a question has its main concepts, for example, the main concept of the question 'What is the greatest common factor of 12 and

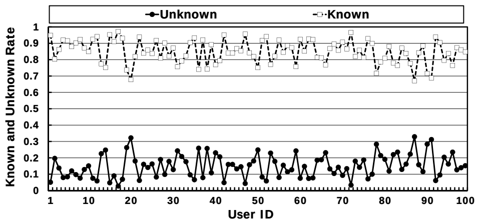


Fig. 2. The known and unknown rate of 100 users.

18?' is 'greatest common factor'. As illustrated in Figure 1, users have personal knowns and unknown concepts. Intuitively, questions correlated to a user's unknown concepts cannot be correctly answered by the user. In this paper, we propose two concept relations, concept prerequisite, and concept closeness. As shown in Figure 1, concept 'Number' is the prerequisite of concept 'Arithmetic', because we basically know 'Number' first before learning 'Arithmetic'. For concept closeness, concept 'Cube' is linked with concept 'Cylinder', because these two concepts describe similar perception. Therefore, if a user does not learn concept c_i which is the prerequisite of concept c_j well, it is more difficult for the user to understand concept c_j. On the other hand, if a user is not good at concept c_i which is similar to concept c_j, it is likely that the user is not familiar with concept c_j either. Thus, exploiting concept relations has potential to improve interactive unknowns recommendation. However, to the best of our knowledge, there is no existing work investigating either concept graphs or interactive unknowns recommendation.

Therefore, in this paper, we study a novel problem of exploiting concept relations for interactive unknowns exploration. The objective of interactive unknowns recommendation in this paper is to interactively provide questions which users do not understand, which can promote users to learn their weaknesses and help them to progress efficiently. The exploration of user unknowns contributes to a series of applications for indispensable educating services such as question recommendation, user aptitude identification, and personalized course navigation. In essence, we address three challenges: (i) how to exploit the proposed concept relations to interactively learn user unknowns; (ii) how to effectively explore new unknowns from user feedback; and (iii) how to retrieve existing user unknowns for users as we have the imbalanced data which is filled with user knowns. We propose the *CagMab* framework to tackle these challenges. The main contributions are:

- We conduct the first study of a new problem, interactive unknowns recommendation, and comprehensively analyze the statistical properties between knowledge concepts.
- We devise a novel *CagMab* framework which incorporates concept-aware graph embedding into a multi-armed bandit model for learning user unknowns.
- We perform experiments on real-world datasets to demonstrate the effectiveness and practicability of the proposed framework.

II. Problem Definition

Before formally introducing the *CagMab* framework, we give the necessary definitions as follows.

Definition 1 (User profile): A user profile $\mathcal{D}_u$ for each user u is a set of question-answer records associated with u. A question-answer record r_i is denoted by a 4-tuple $(u, q, s, \mathcal{B}_q)$ that user u answered question q with the score s. The score s equals to 1 if user u answers the question correctly, otherwise, s is 0. In addition, $\mathcal{B}_q$ denotes the content of question q, and the content includes the question description and the question concepts. For example, a typical record is in the following format: $r_1 = (u_2$,q_4,1,$\{$"What is the greatest common factor of 12 and 18?","greatest common factor"$\})$.

Definition 2 (Question-concept graph): A question-concept graph $\mathcal{G}_{qc} = \{\mathcal{Q} \cup \mathcal{C}, \mathcal{E}_{qc}\}$ is a directed bipartite graph, where $\mathcal{Q}$ is a set of questions, $\mathcal{C}$ is a set of concepts, and $\mathcal{E}_{qc}$ is the set of edges from questions to concepts. In addition, the edge $e_{q_i \rightarrow c_j} = 1$ refers to that question q_i includes the main concept c_j, and the weight of each existing edge is 1. For example, as question-concept graph shown in Figure 3, question q_1 has main concepts c_2 and c_6.

Definition 3 (Concept relational graph): A concept relational graph $\mathcal{G}_{cr} = \{\mathcal{C} \cup \mathcal{C}, \mathcal{E}_{cr}\}$ is an undirected graph which illustrates the concept closeness as we mentioned in Section I. In this graph, $\mathcal{C}$ is a set of concepts, and $\mathcal{E}_{cr}$ refers to the set of edges between different concepts. If $e_{c_i \leftrightarrow c_j} = 1$ holds, concepts c_i and c_j have similar perceptions. The weight of each edge is inversely proportional to the cosine similarity of word embedding vectors [1] of two concepts. For example, as concept relational graph shown in Figure 3, concept c_1 has similar perception with concept c_2, c_3, and c_5.

Definition 4 (Concept progress graph): A concept progress graph $\mathcal{G}_{cp} = \{\mathcal{C} \cup \mathcal{C}, \mathcal{E}_{cp}\}$ is a directed acyclic graph which illustrates the concept prerequisite. For this concept graph, if $e_{c_i \rightarrow c_j} = 1$ holds, concept c_i is the prerequisite of concept c_j. The weight of edges is based on the similarity as that of relational graph. As the progress graph shown in Figure 3, concept c_1 is the prerequisite of concepts c_4 and c_5.

In this paper, bold uppercase characters (e.g., $\mathbf{A}$), bold lowercase characters (e.g., $\vec{\mathbf{a}}$), and normal lowercase characters (e.g., a) are used for matrices, vectors, and scalars, respectively. For an arbitrary matrix $\mathbf{A}$, we represent the i-th row as $\mathbf{A}_{i*}$, the j-th column as $\mathbf{A}_{*j}$, the (i, j)-th entry as $\mathbf{A}_{ij}$, transpose as $\mathbf{A}^T$, and trace as tr($\mathbf{A}$) if $\mathbf{A}$ is a square matrix. Also, the Frobenius norm of a matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$ is defined as $||\mathbf{A}||_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{d} \mathbf{A}_{ij}{}^2}$, $vec(\mathbf{A}) = (\mathbf{A}^T_{*1}, ..., \mathbf{A}^T_{*n})^T$ is the vectorization of matrix $\mathbf{A}$.

Problem Statement: Let $\mathcal{U} = \{u_1, u_2, ..., u_n\}$ be the set of n users, $\mathcal{Q} = \{q_1, q_2, ..., q_m\}$ be the set of m questions, and $\mathcal{C} = \{c_1, c_2, ..., c_o\}$ be the set of o concepts. For a user-question rating matrix $\mathbf{R} \in \mathbb{R}^{n \times m}$, $\mathbf{R}_{ij} = 1$ if user u_i correctly answers question q_j, $\mathbf{R}_{ij} = -1$ denotes u_i gives wrong answer to q_j, and $\mathbf{R}_{ij} =$'?' means u_i has not answered q_j yet. Also, we use $\mathbf{A}^q \in \mathbb{R}^{m \times o}$, $\mathbf{A}^r \in \mathbb{R}^{o \times o}$, and $\mathbf{A}^p \in \mathbb{R}^{o \times o}$ as the corresponding adjacency matrices of graphs $\mathcal{G}_{qc}$, $\mathcal{G}_{cr}$, and $\mathcal{G}_{cp}$, respectively.

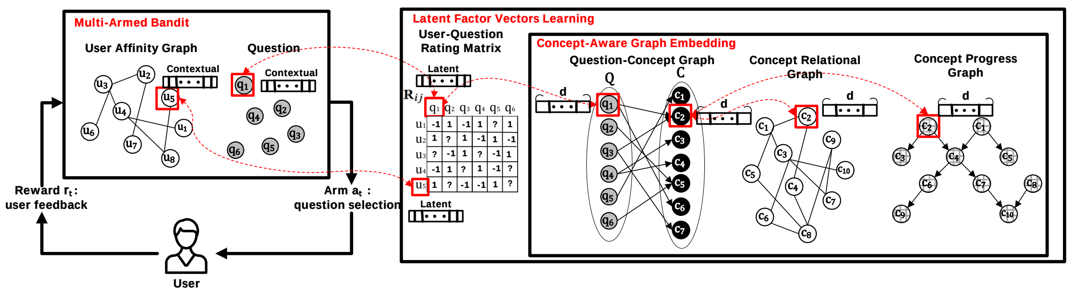


Fig. 3. The overview of *CagMab* framework.

Finally, we give a clear problem formulation as follows.

Given: users $\mathcal{U}$, questions $\mathcal{Q}$, concepts $\mathcal{C}$, user-question rating matrix $\mathbf{R}$, and the proposed graphs $\mathcal{G}_{qc}$, $\mathcal{G}_{cr}$, and $\mathcal{G}_{cp}$;

Select: a set of k questions (user unknowns) for users by interactively exploring rating matrix $\mathbf{R}$ along with proposed graph structures encoded in the adjacency matrices $\mathbf{A}^q$, $\mathbf{A}^r$, and $\mathbf{A}^p$. Note that user unknown indicates that a question score is predicted as '-1' for a user.

III. The CagMab Framework

In this section, we propose the *CagMab* framework incorporating concept-aware graph embedding into multi-armed bandit. As shown in Figure 3, we propose a concept-aware graph embedding to incorporate the graph structure of graphs, $\mathcal{G}_{qc}$, $\mathcal{G}_{cr}$, and $\mathcal{G}_{cp}$ into our model, which helps to capture the interactions between question-concept, concept closeness and concept prerequisite. By capturing rich relations between questions and concepts, we are able to well optimize the latent factors of users and questions. Finally, we introduce both the latent factors and contextual factors of users and questions to the reward with user affinity graph in a multi-armed bandit framework.

A. Multi-Armed Bandit Formulation

To interactively explore user unknowns from users feedback, we propose a multi-armed bandit (MAB) process to reinforce unknowns learning. A multi-armed bandit problem contains a finite set of arms which correspond to the candidate items to be recommended (e.g., unknown questions in an E-learning system). In our scenario, questions $\mathcal{Q}$ correspond to a set of arms. At each trial t, the reward of an observed arm q_t for current user u_i is defined as r_{u_i,q_t}. Thus, the total T-trail reward and the expected T-trail reward are $\sum_{t=1}^{T} r_{u_i,q_t}$ and $\sum_{t=1}^{T} r_{u_i,q_t}$, respectively, where q_t is the arm with maximum reward at trail t. Finally, the accumulated T-trial regret $R_\pi(T)$ can be derived as

$$R_\pi(T) = \mathsf{E}[\sum_{t=1}^{T} r_{u_i,q_t}] - \mathsf{E}[\sum_{t=1}^{T} r_{u_i,q_t}] \tag{1}$$

The goal of MAB algorithm is to interactively update its arm-selection policy π with respect to the user feedback, such that the total regret of the policy is minimized after T trials. Motivated by [2], we define the expected reward r_{u_i,q_t} of an arm q_t at trail t as

$$r_{u_i,q_t} = \mathsf{E}[r_{u_i,q_t}|\vec{\mathbf{x}}_{u_i,q_t}] = \vec{\mathbf{x}}_{u_i,q_t}^{\mathrm{T}}\theta_{u_i} + \varepsilon_t, \tag{2}$$

where $\theta_{u_i} \in \mathsf{R}^{d_\theta}$ (with $||\theta_{u_i}|| \leq l_\theta$) is the unknown parameter vector independently associated with each individual user u_i, $\vec{\mathbf{x}}_{u_i,q_t} \in \mathsf{R}^{d_x}$ (with $||\vec{\mathbf{x}}_{u_i,q_t}|| \leq l_x$) denotes a feature vector representing the contextual information of user u_i and arm q_t, and ε_t is a zero-mean Gaussian noise with variance σ_t^2, that is, $\varepsilon_t \sim \mathcal{N}(0, \sigma_t^2)$. Due to the mutual influence among users [3], it is assumed that the reward r_{u_i,q_t} can be affected by user u_i's neighbors. Let $\mathcal{G}_u = \{\mathcal{U} \cup \mathcal{U}, \mathcal{E}_u\}$ be a weighted graph encoding the affinity relationship among n users. We use $\mathbf{A} \in \mathsf{R}^{n \quad n}$ as the corresponding adjacency matrices of graph $\mathcal{G}_u$, and each element $a_{i,j}$ is the link weight proportional to the influence that user u_j has on user u_i. Thus, $\vec{\mathbf{a}}_{u_i} = \{a_{i,1}, ..., a_{i,n}\}$ is the user influence vector from the neighbors of user u_i, and the expected reward of an arm q_t at trail t can be rewritten as

$$r_{u_i,q_t} = \mathsf{E}[r_{u_i,q_t}|\vec{\mathbf{x}}_{u_i,q_t}, \vec{\mathbf{a}}_{u_i}] = \vec{\mathbf{x}}_{u_i,q_t}^{\mathrm{T}}\mathbf{\Theta}\,\vec{\mathbf{a}}_{u_i} + \varepsilon_t, \tag{3}$$

where $\mathbf{\Theta} = \{\theta_{u_1}, ..., \theta_{u_n}\}$ is the global bandit parameter matrix. However, simply relying on the contextual features of users and question often leads to suboptimal results for interactive unknowns recommendation as it ignores the rich relations between the recommended questions and the associated concepts. To this end, we propose to learn the latent representations of users and questions through the concept-aware graph embedding and then introduce the learned user latent factor representation $\vec{\mathbf{u}}_i \in \mathsf{R}^{d_u}$ and question latent factor representation $\vec{\mathbf{q}}_t \in \mathsf{R}^{d_q}$ into the reward function. Finally, the expected reward can be derived as

$$\begin{aligned} r_{u_i,q_t} &= \mathsf{E}[r_{u_i,q_t}|\vec{\mathbf{x}}_{u_i,q_t}, \vec{\mathbf{a}}_{u_i}, \vec{\mathbf{u}}_i, \vec{\mathbf{q}}_t] \\ &= \underbrace{\vec{\mathbf{x}}_{u_i,q_t}^{\mathrm{T}}\mathbf{\Theta}\,\vec{\mathbf{a}}_{u_i}}_{\text{contextual term}} + \underbrace{\vec{\mathbf{u}}_i\vec{\mathbf{q}}_t}_{\text{latent term}} + \varepsilon_t. \end{aligned} \tag{4}$$

Accordingly, we define a contextual feature matrix, a bandit parameter matrix, a user latent factor matrix, an item latent matrix, and a user influence matrix as $\mathbf{X}_{q_t} = \{\vec{\mathbf{x}}_{u_1,q_t}, ..., \vec{\mathbf{x}}_{u_n,q_t}\}$, $\mathbf{\Theta} = \{\theta_{u_1}, ..., \theta_{u_n}\}$, $\mathbf{U} = \{\vec{\mathbf{u}}_1, ..., \vec{\mathbf{u}}_n\}$, $\mathbf{Q} = \{\vec{\mathbf{q}}_1, ..., \vec{\mathbf{q}}_m\}$, and $\mathbf{A} = \{\vec{\mathbf{a}}_{u_1}, ..., \vec{\mathbf{a}}_{u_n}\}$, respectively. The contextual feature vectors $\vec{\mathbf{x}}_{u_i,q_t}$ and user influence vectors $\vec{\mathbf{a}}_{u_i}$ are observed, and the detailed learning of latent factors $\vec{\mathbf{q}}_t$ and $\vec{\mathbf{u}}_i$ will be given in the next subsection. For the unknown bandit parameters of each user, we apply ridge regression to estimate the global bandit parameter matrix $\hat{\mathbf{\Theta}}$ as follows

$$\arg\max_{\mathbf{\Theta}} \frac{1}{2}\sum_{t=1}^{T}(vec(\widetilde{\mathbf{X}}_{u_t,q_t})^{\mathrm{T}} vec(\mathbf{\Theta}_t\mathbf{A}) + \vec{\mathbf{u}}_t\vec{\mathbf{q}}_t - r_{u_t,q_t})^2 + \frac{\lambda}{2}tr(\mathbf{\Theta}^{\mathrm{T}}\mathbf{\Theta}), \quad (5)$$

where $\lambda \in [0,1]$ is the model specific regularization parameter in ridge regression, and contextual matrix $\widetilde{\mathbf{X}}_{u_i,q_t}$ of user u_i contains only the vector $\vec{\mathbf{x}}_{u_j,q_t}$ of user u_j correlated to user u_i and sets other unrelated ones as zero. With the regularized quadratic function, we can easily derive the closed-form solution of $\mathbf{\Theta}$ as $\hat{\vartheta}_t = vec(\hat{\mathbf{\Theta}}_t) = \mathbf{E}_t^{-1}\mathbf{f}_t$, and

$$\begin{cases} \mathbf{E}_t = \sum_{t'=1}^{t} vec(\widetilde{\mathbf{X}}_{u_{t'},q_{t'}}\mathbf{A}^{\mathrm{T}})vec(\widetilde{\mathbf{X}}_{u_{t'},q_{t'}}\mathbf{A}^{\mathrm{T}})^{\mathrm{T}} + \lambda\mathbf{I}. \\ \mathbf{f}_t = \sum_{t'=1}^{t} vec(\widetilde{\mathbf{X}}_{u_{t'},q_{t'}}\mathbf{A}^{\mathrm{T}})(r_{u_{t'},q_{t'}} - \vec{\mathbf{u}}_{t'}\vec{\mathbf{q}}_{t'}). \end{cases} \quad (6)$$

Finally, the estimated bandit parameters $\hat{\mathbf{\Theta}}$ can be used to predict the expected reward of a particular arm for the current user.

To implement an adaptive bandit algorithm, the exploration strategy should be devised for each user. According to our definition, rewards r_{u_i,q_t} for users are independent of each other. Thus, with probability at least $1-\delta$, the following inequality holds

$$|r_{u_i,q_t} - r_{u_i,q_t}| \le \alpha_\theta \sqrt{vec(\widetilde{\mathbf{X}}_{u_i,q_t}\mathbf{A}^{\mathrm{T}})^{\mathrm{T}}\mathbf{E}_t^{-1}vec(\widetilde{\mathbf{X}}_{u_i,q_t}\mathbf{A}^{\mathrm{T}})} \quad (7)$$

for any $\delta > 0$, where $\alpha_\theta = 1 + \sqrt{\ln(2/\delta)/2}$ is a constant, and $\delta \in [0,1]$. The inequality above gives upper confidence bound for the expected reward of arm q_t, and arm-selection policy can be derived as

$$q_{t,u_i} = \arg\max_{q_t \in Q}\Big(\vec{\mathbf{x}}_{u_i,q_t}^{\mathrm{T}}\hat{\mathbf{\Theta}}_t\,\vec{\mathbf{a}}_{u_i} + \vec{\mathbf{u}}_i\vec{\mathbf{q}}_t + \alpha_\theta\sqrt{vec(\widetilde{\mathbf{X}}_{u_i,q_t}\mathbf{A}^{\mathrm{T}})^{\mathrm{T}}\mathbf{E}_t^{-1}vec(\widetilde{\mathbf{X}}_{u_i,q_t}\mathbf{A}^{\mathrm{T}})}\Big). \quad (8)$$

B. Learning Latent Factors

In this subsection, we investigate how to learn the optimal user embedding vector $\vec{\mathbf{u}}_i$ and question embedding vector $\vec{\mathbf{q}}_j$ by using concept-aware graphs embedding. In this way, it can capture the rich relations between questions and the associated concepts for interactive unknowns recommendation.

1) Modeling User-Question Interaction: To learn the user-question interaction, we recursively optimize the embedding vectors of users and questions with regularization technique. In our scenario, the rating matrix $\mathbf{R} \subseteq \mathcal{U} \times \mathcal{Q}$ shows the relationship between users and questions. The estimated rating $\mathbf{R}_{ij}$ is often expressed as $\mathbf{R}_{ij} := \vec{\mathbf{u}}_i \cdot \vec{\mathbf{q}}_j$, where $\vec{\mathbf{u}}_i \in \mathsf{R}^{d_u}$ is the latent factor vector representing the preferences of user u_i, and $\vec{\mathbf{q}}_j \in \mathsf{R}^{d_q}$ is the latent factor vector representing question q_j. The goal of our approaches is to use the observed ratings to optimize the latent factors for estimating the unobserved ratings. To estimate the unobserved ratings, we assume that the rating $\mathbf{R}_{ij}$ follows a normal distribution $\mathbf{R}_{ij} \sim \mathcal{N}(\mathbf{U}_i \cdot \mathbf{Q}_j, \sigma^2)$. Motivated by [4], the posterior probability of latent matrix $\mathbf{U}$ and $\mathbf{Q}$ with observed rating matrix $\mathbf{R}$ can be derived as

$$\mathcal{P}(\mathbf{U},\mathbf{Q}|\mathbf{R},\sigma_u^2,\sigma_q^2,\sigma^2) \propto \mathcal{P}(\mathbf{R}|\mathbf{U},\mathbf{Q},\sigma^2)\mathcal{P}(\mathbf{U}|\sigma_u^2)\mathcal{P}(\mathbf{Q}|\sigma_q^2) = \prod_i\prod_j \mathcal{N}(\mathbf{U}_i\cdot\mathbf{Q}_j,\sigma^2)^{\mathbf{O}_{ij}}\prod_i\mathcal{N}(0,\sigma_u^2\mathbf{O})\prod_j\mathcal{N}(0,\sigma_q^2\mathbf{O}), \quad (9)$$

where $\mathbf{O} \in \mathsf{R}^{m \times n}$ denotes the indicator matrix, and $\mathbf{O}_{ij} = 1$ indicates that u_i rates q_j; otherwise, $\mathbf{O}_{ij} = 0$.

Finally, we can infer the estimated rating list of questions given from users by minimizing the following objective function

$$O_{uq} = \frac{1}{2}\sum_{i,j}\mathbf{O}_{ij}(\mathbf{R}_{ij} - \vec{\mathbf{u}}_i\cdot\vec{\mathbf{q}}_j)^2 + \frac{\lambda_u}{2}||\mathbf{U}||_F^2 + \frac{\lambda_q}{2}||\mathbf{Q}||_F^2, \quad (10)$$

where $\lambda_u = \frac{\sigma^2}{\sigma_u^2}$ and $\lambda_q = \frac{\sigma^2}{\sigma_q^2}$ are the regularization coefficients to avoid over-fitting.

2) Modeling Concept-Aware Graphs (Cag): By extending the graph embedding approach [5], we model the concept-aware graph structure into our method to learn the optimal user embedding vector $\vec{\mathbf{u}}_i$ and question embedding vector $\vec{\mathbf{q}}_j$. Thus, we devise a graph embedding approach on question-concept graph $\mathcal{G}_{qc}$, concept relational graph $\mathcal{G}_{cr}$, and concept progress graph $\mathcal{G}_{cp}$, and optimize an objective which preserves these graph structures. In our graph embedding technique, we model both the first-order proximity and second-order proximity for concept graphs.

Concept Relational Graph: Generally, the first-order proximity refers to the local pairwise proximity between the vertices in the graph. Given the concept relational graph $\mathcal{G}_{cr} = \{\mathcal{C} \cup \mathcal{C}, \mathcal{E}_{cr}\}$, we model the first-order proximity for each edge (c_i, c_j) as the joint probability between concepts c_i and c_j

$$p(c_i, c_j) = \frac{1}{1 + \exp(-\vec{\mathbf{c}}_i^{\mathrm{T}}\cdot\vec{\mathbf{c}}_j)}, \quad (11)$$

where $\vec{\mathbf{c}}_j \in \mathsf{R}^d$ is the embedding vector representation of vertex c_j. The second-order proximity assumes that vertices sharing many connections to other vertices are similar to each other. For each 2-hop neighbor pair (c_i, c_j), we define the conditional probability of c_j generated by vertex c_i as

$$p(c_j|c_i) = \frac{\exp(\vec{\mathbf{c}}_j^{\mathrm{T}}\cdot\vec{\mathbf{c}}_i)}{\sum_{c_k \in C}\exp(\vec{\mathbf{c}}_k^{\mathrm{T}}\cdot\vec{\mathbf{c}}_i)}. \quad (12)$$

The notations $p(\cdot,\cdot)$ and $p(\cdot|\cdot)$ define a joint distribution and a conditional distribution related to different pairs of concepts, and these two distributions should be close to their empirical distributions $\hat{p}(\cdot,\cdot)$ and $\hat{p}(\cdot|\cdot)$. The empirical distributions of first-order proximity and second-order proximity can be defined as

$$\begin{cases} \hat{p}(c_i,c_j) = \frac{w_{ij}}{W_1} \quad with \quad w_{ij} = w_{ij}, \\ \hat{p}(c_i|c_j) = \frac{w_{ij}}{W_2} \quad with \quad w_{ij} = \frac{\sum_{(e_{io},e_{oj}) \in E_{cr}} w_{io} \times w_{oj}}{\sigma_{ij}}, \end{cases} \quad (13)$$

where $W_1 = \sum_{e_{ij} \in E_{cr}} w_{ij}$, $W_2 = \sum_{c_j \in N_2(c_i)} w_{ij}$, $N_2(c_i)$ is the set of second-order neighbors of vertex c_i, and σ_{ij} is the total number of shortest paths from node c_i to node c_j. With these empirical distributions, we minimize the following objective functions for first-order proximity and second-order proximity

$$\left\{ \begin{array}{l} O_1 = d(p(\cdot,\cdot), \hat{p}(\cdot,\cdot)), \\ O_2 = d(p(\cdot|\cdot), \hat{p}(\cdot|\cdot)), \end{array} \right. \tag{14}$$

where $d(\cdot,\cdot)$ is the distance between two distributions. By replacing $d(\cdot,\cdot)$ with KL-divergence and omitting some constants, the objective functions to preserve first-order proximity and second-order proximity can be derived as

$$\left\{ \begin{array}{l} O_1 = -\sum_{e_{ij} \in E_{cr}} w_{ij} \log p(c_i, c_j). \\ O_2 = -\sum_{c_i \in C, c_k \in N_2(c_i)} w_{ik} \log p(c_k|c_i). \end{array} \right. \tag{15}$$

Finally, we are able to model the structure of concept relational graph based on first-order and second-order proximity by minimizing the objective function

$$\begin{aligned} O_{cr} &= O_1 + O_2 \\ &= -(\sum_{e_{ij} \in E_{cr}} w_{ij} \log p(c_i, c_j) + \sum_{c_i \in C, c_k \in N_2(c_i)} w_{ik} \log p(c_k|c_i)). \end{aligned} \tag{16}$$

Concept Progress Graph: For the concept progress graph $\mathcal{G}_{cp}$, we also model the first-order proximity and second-order proximity for each directed path $p_{c_i \to c_j}$ as the joint probability (Eq.(11)) and conditional probability (Eq.(12)) between concepts c_i and c_j. However, since progress graph is directed acyclic, the empirical distributions of first-order proximity and second-order proximity are different from that of concept relational graph by considering the transition probability. Given a path set $\mathcal{P}$ containing all directed shortest paths in graph $\mathcal{G}_{cp}$, we first define the proximity of two vertices $c_i, c_j \in \mathcal{C}$ with respect to $\mathcal{P}$ as

$$r(c_i, c_j|\mathcal{P}) = \sum_{p_{c_i \to c_j} \in P} r(c_i, c_j|p_{c_i \to c_j}), \tag{17}$$

where $p_{c_i \to c_j}$ is a path linking from vertex c_i to vertex c_j. In this paper, we use a truncated estimation of proximity, which only considers paths up to a length threshold l. Because we only aim at preserving the first-order and second-order proximity. We define the truncated proximity $r_l(c_i, c_j)$ between two vertices c_i and c_j as

$$\begin{aligned} r_l(c_i, c_j) &= \sum_{len(P) \leq l} r(c_i, c_j|\mathcal{P}) \\ &= \sum_{p_{c_i \to c'} \in P} p_{c_i \to c'}^{\psi(c_i, c')} \times r_{l-1}(c', c_j), \end{aligned} \tag{18}$$

where $p_{c_i \to c'}^{\psi(c_i, c')}$ is the transition probability from c_i to c'. If there are n edges on the shortest path from c_i to c', then $\psi(c_i, c') = \frac{1}{n}$. Since we consider both first-order and second-order proximity, the n is either 1 or 2. With the truncated proximity, the empirical distribution of vertices c_i and c_j in first-order and second-order cases can be defined as

$$\left\{ \begin{array}{l} \hat{p}(c_i, c_j) = \frac{r_1(c_i, c_j)}{\sum_{c' \in C} r_1(c_i, c')}, \\ \hat{p}(c_i|c_j) = \frac{r_2(c_i, c_j)}{\sum_{c' \in C} r_2(c_i, c')}, \end{array} \right. \tag{19}$$

With these empirical distributions, we are able to model the structure of concept progress graph based on first-order proximity and second-order proximity by minimizing the objective function

$$\begin{aligned} O_{cp} = -(&\sum_{e_{ij} \in E_{cp}} r_1(c_i, c_j) \log p(c_i, c_j) \\ &+ \sum_{p_{c_i \to c_k} \in P} r_2(c_i, c_k) \log p(c_k|c_i)). \end{aligned} \tag{20}$$

Question-Concept Graph: For the question-concept graph $\mathcal{G}_{qc}$, we only consider the first-order proximity, since graph $\mathcal{G}_{qc}$ is a bipartite graph. It is assumed that nodes with shared neighbors being likely to be similar. Thus, for the first-order proximity of graph $\mathcal{G}_{qc} = \{\mathcal{Q} \cup \mathcal{C}, \mathcal{E}_{qc}\}$, we use a conditional distribution of vertex $c_j \in \mathcal{C}$ related to vertex $q_i \in \mathcal{Q}$ as

$$p(c_j|q_i) = \frac{\exp(\vec{\mathbf{c}}_j^{\mathrm{T}} \cdot \vec{\mathbf{q}}_i)}{\sum_{c_k \in C} \exp(\vec{\mathbf{c}}_k^{\mathrm{T}} \cdot \vec{\mathbf{q}}_i)}, \tag{21}$$

where $\vec{\mathbf{q}}_i$ is the embedding vector of vertex $q_i \in \mathcal{Q}$, and $\vec{\mathbf{c}}_j$ is the embedding vector of $c_j \in \mathcal{C}$. The empirical distribution $\hat{p}(\cdot|q_i)$ can be defined as $\hat{p}(c_j|q_i) = \frac{w_{ij}}{W}$, where $W = \sum_{e_{ij} \in E_{qc}} w_{ij}$. The objective function that we should minimize for question-concept graph can be inferred as

$$O_{qc} = -\sum_{e_{ij} \in E_{qc}} w_{ij} (\log p(c_j|q_i) + \log p(q_i|c_j)). \tag{22}$$

Finally, we are able to preserve the structures of question-concept graph, concept relational graph, and concept progress graph by minimizing the objective functions O_{qc}, O_{cr}, and O_{cp}, respectively. The final objective function of *Cag* is formulated as follows

$$\begin{aligned} &\min_{U,Q,C} \beta(O_{uq}) + \gamma(O_{qc} + O_{cr} + O_{cp}) \\ &= \beta \frac{1}{2} \sum_{i,j} \mathbf{O}_{ij} (\mathbf{R}_{ij} - \vec{\mathbf{u}}_i \cdot \vec{\mathbf{q}}_j)^2 + \frac{\lambda_u}{2} ||\mathbf{U}||_F^2 + \frac{\lambda_q}{2} ||\mathbf{Q}||_F^2 \\ &- \gamma \Bigg(\sum_{e_{ij} \in E_{qc}} w_{ij} (\log p(c_j|q_i) + \log p(q_i|c_j)) \\ &+ \sum_{e_{ij} \in E_{cr}} w_{ij} \log p(c_i, c_j) + \sum_{c_i \in C, c_k \in N_2(c_i)} w_{ik} \log p(c_k|c_i) \\ &+ \sum_{e_{ij} \in E_{cp}} r_1(c_i, c_j) \log p(c_i, c_j) + \sum_{p_{c_i \to c_k} \in P} r_2(c_i, c_k) \log p(c_k|c_i) \Bigg) \end{aligned} \tag{23}$$

where parameter β controls the contribution of first term which models the interaction between users and questions; and parameter γ controls the importance of the rest terms which capture the effect of concept-aware graphs.

3) The Optimization of Cag*:* To optimize objective functions O_{uq}, O_{qc}, O_{cr}, O_{cp}, we adopt the approach of negative sampling [1] which randomly chooses multiple negative samples, and we use the asynchronous stochastic gradient algorithm (ASGD) [6] for optimization. To simplify notation, we use $\mathcal{J}$ to denote the objective function in Eq.(23). In particular, the partial derivative of the objective function $\mathcal{J}$ w.r.t $\vec{\mathbf{u}}_i$, $\vec{\mathbf{q}}_j$, and $\vec{\mathbf{c}}_m$ can be calculated as follows:

$$\frac{\partial \mathcal{J}}{\partial \vec{\mathbf{u}}_i} = \beta(-(\mathbf{R}_{ij} - \vec{\mathbf{u}}_i \cdot \vec{\mathbf{q}}_j)\vec{\mathbf{q}}_j + \lambda_u \vec{\mathbf{u}}_i). \tag{24}$$

$$\begin{aligned} \frac{\partial \mathcal{J}}{\partial \vec{\mathbf{q}}_j} &= \beta(-(\mathbf{R}_{ij} - \vec{\mathbf{u}}_i \cdot \vec{\mathbf{q}}_j)\vec{\mathbf{u}}_i + \lambda_q \mathbf{q}_j) \\ &\quad - \gamma \cdot w_{jm}(\frac{\partial \log p(c_m|q_j)}{\partial \vec{\mathbf{q}}_j} + \frac{\partial \log p(q_j|c_m)}{\partial \vec{\mathbf{q}}_j}). \end{aligned} \tag{25}$$

$$\begin{aligned} \frac{\partial \mathcal{J}}{\partial \vec{\mathbf{c}}_m} &= -\gamma \Bigg(w_{jm}(\frac{\partial \log p(c_m|q_j)}{\partial \vec{\mathbf{c}}_m} + \frac{\partial \log p(q_j|c_m)}{\partial \vec{\mathbf{c}}_m}) \\ &+ (w_{mn}\frac{\partial \log p(c_m, c_n)}{\partial \vec{\mathbf{c}}_m} + w_{mo}\frac{\partial \log p(c_o|c_m)}{\partial \vec{\mathbf{c}}_m}) \\ &+ (r_1(c_m, c_n)\frac{\partial \log p(c_m, c_n)}{\partial \vec{\mathbf{c}}_m} + r_2(c_m, c_o)\frac{\partial \log p(c_o|c_m)}{\partial \vec{\mathbf{c}}_m})\Bigg). \end{aligned} \tag{26}$$

The embedding vectors are updated with the learning rate η for $(t+1)$-th iteration as

$$\begin{cases} \vec{\mathbf{u}}_i^{t+1} = \vec{\mathbf{u}}_i^{t} - \eta \frac{\partial J\ (\mathbf{U}^t, \mathbf{Q}^t, \mathbf{C}^t)}{\partial \vec{\mathbf{u}}_i}, \forall i = 1, ..., a \\ \vec{\mathbf{q}}_j^{t+1} = \vec{\mathbf{q}}_j^{t} - \eta \frac{\partial J\ (\mathbf{U}^{t+1}, \mathbf{Q}^t, \mathbf{C}^t)}{\partial \vec{\mathbf{q}}_j}, \forall j = 1, ..., b \\ \vec{\mathbf{c}}_m^{t+1} = \vec{\mathbf{c}}_m^{t} - \eta \frac{\partial J\ (\mathbf{U}^{t+1}, \mathbf{Q}^{t+1}, \mathbf{C}^t)}{\partial \vec{\mathbf{c}}_m}, \forall m = 1, ..., c \end{cases} \tag{27}$$

Finally, by learning the optimal user latent factor $\vec{\mathbf{u}}_i$ and question latent factor $\vec{\mathbf{q}}_j$, the reward of arms can be computed for arm-selection policy in multi-armed bandit.

C. Regret Analysis

We provide regret analysis of *CagMab* in this section. We first introduce the upper bound of the estimation error of bandit parameters $\hat{\boldsymbol{\Theta}}$ in Lemma III.1.

Lemma III.1. *According to the proof in [3], for any $\delta > 0$, with probability at least $1 - \delta$, the estimation error of bandit parameters of* CagMab *can be derived as bounded by*

$$||\hat{\vartheta}_t - \vartheta\ ||_{\mathbf{E}_t} \leq \sqrt{d_\theta n \ln(1 + \frac{l_x^2 \sum_{t^0=1}^{t}\sum_{j=1}^{n} a_{t^0,j}^2}{\delta \lambda d_\theta n})} + \sqrt{\lambda}||\vartheta\ ||. \tag{28}$$

Then, we prove the following theorem for the upper bound of the regret of *CagMab*.

Theorem III.2. *With probability at least $1 - \delta$, the accumulated regret of* CagMab *is bounded by*

$$R_\pi(T) \leq 2\alpha_{\boldsymbol{\Theta}} \sqrt{T d_\theta n \ln(1 + \frac{l_x^2 \sum_{t=1}^{T}\sum_{j=1}^{n} a_{t,j}^2}{\delta \lambda d_\theta n})}. \tag{29}$$

Proof: Before inferring the accumulated regret of *CagMab*, we first give some lemmas as below.

Lemma III.3. *According to our arm selection policy in Eq.(8), if arm q_t is selected at time t, the following inequality holds.*

$$\begin{aligned} &vec(\widetilde{\mathbf{X}}_{u_t,q_t}\mathbf{A}^{\mathrm{T}})^{\mathrm{T}}\hat{\vartheta}_{t\ \ 1} + \vec{\mathbf{u}}_t\vec{\mathbf{q}}_t + \alpha_\theta||vec(\widetilde{\mathbf{X}}_{u_t,q_t}\mathbf{A}^{\mathrm{T}})||_{\mathbf{E}_{t\ \ 1}^{1}} \\ &\leq vec(\widetilde{\mathbf{X}}_{u_t,q_t}\mathbf{A}^{\mathrm{T}})^{\mathrm{T}}\hat{\vartheta}_{t\ \ 1} + \vec{\mathbf{u}}_t\vec{\mathbf{q}}_t + \alpha_\theta||vec(\widetilde{\mathbf{X}}_{u_t,q_t}\mathbf{A}^{\mathrm{T}})||_{\mathbf{E}_{t\ \ 1}^{1}} \end{aligned} \tag{30}$$

Lemma III.4. *Based on Cauchy-Schwarz inequality and Lemma III.3, the regret of* CagMab *at time t is inferred as*

$$\begin{aligned} R_t &= r_{u_t,q_t} - r_{u_t,q_t} \\ &= (vec(\widetilde{\mathbf{X}}_{u_t,q_t}\mathbf{A}^{\mathrm{T}})^{\mathrm{T}}\vartheta\ \ + \vec{\mathbf{u}}_t\vec{\mathbf{q}}_t) - (vec(\widetilde{\mathbf{X}}_{u_t,q_t}\mathbf{A}^{\mathrm{T}})^{\mathrm{T}}\vartheta\ \ + \vec{\mathbf{u}}_t\vec{\mathbf{q}}_t) \\ &\leq vec(\widetilde{\mathbf{X}}_{u_t,q_t}\mathbf{A}^{\mathrm{T}})^{\mathrm{T}}\hat{\vartheta}_{t\ \ 1} + \alpha_\theta||vec(\widetilde{\mathbf{X}}_{u_t,q_t}\mathbf{A}^{\mathrm{T}})||_{\mathbf{E}_{t\ \ 1}^{1}} \\ &\quad - vec(\widetilde{\mathbf{X}}_{u_t,q_t}\mathbf{A}^{\mathrm{T}})^{\mathrm{T}}\vartheta \\ &\leq ||vec(\widetilde{\mathbf{X}}_{u_t,q_t}\mathbf{A}^{\mathrm{T}})||_{\mathbf{E}_{t\ \ 1}^{1}}||\hat{\vartheta}_{t\ \ 1} - \vartheta\ \ ||_{\mathbf{E}_{t\ \ 1}^{1}} \\ &\quad + \alpha_\theta||vec(\widetilde{\mathbf{X}}_{u_t,q_t}\mathbf{A}^{\mathrm{T}})||_{\mathbf{E}_{t\ \ 1}^{1}} \\ &\leq 2\alpha_\theta||vec(\widetilde{\mathbf{X}}_{u_t,q_t}\mathbf{A}^{\mathrm{T}})||_{\mathbf{E}_{t\ \ 1}^{1}}. \end{aligned} \tag{31}$$

Lemma III.5. *As proved in [7], $\sum_{t=1}^{T}||vec(\widetilde{\boldsymbol{X}}_{u_t,q_t}\boldsymbol{A}^{\mathrm{T}})||_{\boldsymbol{E}_{t\ \ 1}^{1}}^2$ is bounded by $det(\boldsymbol{E}_T)$ as*

$$\ln(\frac{det(\mathbf{E}_T)}{det(\lambda\mathbf{I})}) \leq \sum_{t=1}^{T}||vec(\widetilde{\mathbf{X}}_{u_t,q_t}\mathbf{A}^{\mathrm{T}})||_{\mathbf{E}_{t\ \ 1}^{1}}^2 \leq 2\ln(\frac{det(\mathbf{E}_T)}{det(\lambda\mathbf{I})}) \tag{32}$$

Referring to the following inequality

$$\begin{cases} det(\mathbf{E}_T) \leq (\frac{tr(E_T)}{\delta d_\theta n})^{d_\theta n} \leq (\lambda + \frac{l_x^2 \sum_{t^0=1}^{t}\sum_{j=1}^{n} a_{t,j}^2}{\delta d_\theta n})^{d_\theta n}, \\ det(\lambda\mathbf{I}) \leq \lambda^{d_\theta n}, \end{cases} \tag{33}$$

Lemma III.4, and Lemma III.5, the accumulated regret of *CagMab* can be derived as

$$\begin{aligned} R_\pi(T) &= \mathsf{E}[\sum_{t=1}^{T} r_{u_t,q_t}] - \mathsf{E}[\sum_{t=1}^{T} r_{u_t,q_t}] \leq \sqrt{T\sum_{t=1}^{T} R_t^2} \\ &\leq \sqrt{T\sum_{t=1}^{T} 4\alpha_{\boldsymbol{\Theta}}^2||vec(\widetilde{\mathbf{X}}_{u_t,q_t}\mathbf{A}^{\mathrm{T}})||_{\mathbf{E}_{t\ \ 1}^{1}}^2} \leq \sqrt{T8\alpha_{\boldsymbol{\Theta}}^2 \ln(\frac{det(\mathbf{E}_T)}{det(\lambda\mathbf{I})})} \\ &\leq 2\alpha_{\boldsymbol{\Theta}}\sqrt{T d_\theta n \ln(1 + \frac{l_x^2 \sum_{t^0=1}^{t}\sum_{j=1}^{n} a_{t,j}^2}{\delta \lambda d_\theta n})}, \end{aligned} \tag{34}$$

where $\alpha_{\boldsymbol{\Theta}}$ is the upper bound of $||\hat{\vartheta}_t - \vartheta\ ||_{\mathbf{E}_t}$ (Lemma III.1) over all $t \in T$. Finally, the proof of the regret of *CagMab* is completed. ■

IV. Experimental Results

In this section, we conduct experiments to evaluate the effectiveness of the proposed framework *CagMab*. More specifically, we aim to answer the following research questions:

Can the *CagMab* framework outperform other multi-armed bandit algorithms by capturing rich relations between questions and concepts in the reward for interactive unknowns recommendation?

TABLE I
PERFORMANCE COMPARISON IN TERMS OF PRECISION@5, RECALL@5 AND AUC@5. (THE ROW 'IMPROVE' IS THE ACCURACY DIFFERENCE RANGE BETWEEN OUR METHOD AND PREVIOUS WORK IN PERCENTAGE.)

Metric	Naïve	ϵ-greedy	TS	UCB	LinUCB	COFIBA	factorUCB	GOBLin	CM-CR	CM-CP	**CagMab**	**Improve**
Precision@5	0.1420	0.1500	0.1320	0.1520	0.1300	0.1420	0.1360	0.1440	0.3139	0.3179	**0.3319**	[121%-155%]
Recall@5	0.1404	0.1329	0.1168	0.1470	0.1258	0.1099	0.1059	0.1343	0.2738	0.2709	**0.2886**	[96%-172%]
AUC@5	0.3310	0.3200	0.3160	0.3510	0.3200	0.3260	0.3060	0.3520	0.5570	0.5640	**0.5810**	[60%-89%]

TABLE II
PERFORMANCE COMPARISON IN TERMS OF PRECISION@10, RECALL@10 AND AUC@10. (THE ROW 'IMPROVE' IS THE ACCURACY DIFFERENCE RANGE BETWEEN OUR METHOD AND PREVIOUS WORK IN PERCENTAGE.)

Metric	Naïve	ϵ-greedy	TS	UCB	LinUCB	COFIBA	factorUCB	GOBLin	CM-CR	CM-CP	**CagMab**	**Improve**
Precision@10	0.1480	0.1440	0.1410	0.1460	0.1310	0.1340	0.1340	0.1470	0.2419	0.2510	**0.2580**	[74%-96%]
Recall@10	0.2727	0.2736	0.2800	0.2512	0.2296	0.2114	0.2178	0.2371	0.3792	0.3994	**0.4084**	[46%-93%]
AUC@10	0.4640	0.4570	0.4705	0.4630	0.4455	0.4270	0.4320	0.4535	0.5510	0.5605	**0.5790**	[23%-36%]

TABLE III
STATISTICS OF DATASET AND GRAPHS.

Data	**Dataset**	Graph	**Relational**	**Progress**
#Users	100	#Nodes	3019	283
#Questions	324190	#Links	27440	309
#Concepts	3019	CC	0.9066	0.0888
#QA records	1072910	SP	1.9166	7.7385
Avg. #QA/user	10729.1	Diam.	6	19

Is *CagMab* able to mitigate the cold-start problem for interactive unknowns recommendation?
How the performance of *CagMab* varies with respect to different parameter settings?
How is the diversity of the recommended questions generated by our interactive system for each user?
Can *CagMab* explore unknown unknowns for users as we only have known unknowns in training phases?

A. Dataset and Experimental Setup

Dataset: For the purpose of this study, we apply a real data from a nonprofit-based E-learning service, Junyi online learning system[1], providing courses and tests for users. The Junyi academy platform is a Chinese online learning website. The collected data contains user profiles, question answering records, and question descriptions of 100 users during their surfing time in the platform. For concept relational graph, the link of two concepts is generated if the similarity score is larger than a given threshold. The score is the cosine similarity of two concepts by using average word embedding vectors. With regards to concept progress graph[2], domain experts provide the links among concepts from their domain knowledge. More statistical information of our data is shown in Table III, and the graph measurement, including CC (clustering coefficient), SP (average shortest path length), and Diam (diameter), indicates that the relational graph is much denser than the progress one.

Other Methods: In this paper, three widely used evaluation metrics, Precision@k, Recall@k, and AUC@k are adopted to evaluate the recommendation performance. In our experiment, k is set to 5 and 10. We compare *CagMab*[3] with representative and state-of-the-art algorithms:

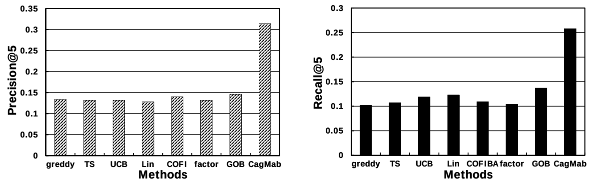
Naïve: In each trail, the naïve algorithm selects an arm which has the highest mean reward observed by sampling different arms in previous iterations.
ϵ-greedy: This method selects the optimal arm with the highest reward for a proportion of $1 - \epsilon$, and randomly chooses an arm for a proportion of ϵ.
TS: Based on Thompson Sampling (TS), this method chooses an arm referring to its probability of being the best arm from the observed trails.
UCB: This method provides upper bound confidence intervals for mean reward and selects the best arm with reward by adding its upper bound confidence.
LinUCB: With the upper bound confidence, LinUCB designs the expected reward being linear in contextual vectors of both users and arms.
COFIBA: COFIBA uses collaborative filtering to group arms and users, and the reward is estimated from the interaction of users and arms in the clustering.
FactorUCB: FactorUCB devises a factorization based bandit, and the arm selection is based on the upper confidence bound over user-arm rating matrix.
GOBLin: GOBLin is proposed by modeling the perception that users would share their contextual information and rewards with neighbors in a network.
CM-CR: To understand the effect of concept progress graph, we remove concept relational graph in this method which is a variant of our developed method.
CM-CP: Conversely, we do not consider concept progress graph in this method as a variant.

Finally, for each individual user, we randomly select 75% of his/her question-answer records for training. The rest of the observed user-question pairs are used as testing. The accuracy performance reported by 4-fold cross-validation is the average of accuracy computed in each training and testing phase.

B. Interactive Recommendation Performance

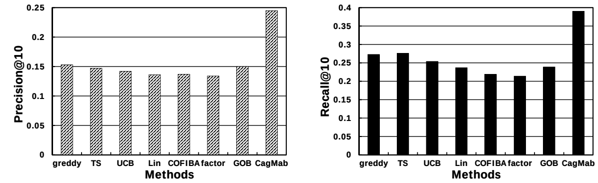
To answer the first question, we compare *CagMab* with several representative MAB algorithms. The comparison results are summarized in Table I and Table II. For parameters

[1]Please refer to https://www.junyiacademy.org/ for the Junyi website.
[2]Please refer to https://www.junyiacademy.org/exercisedashboard for the visualization of our concept graph.
[3]Please refer to https://goo.gl/dvPPdG for the code of CagMab.



(a) Precision@5. (b) Recall@5.

Fig. 4. Performance comparison on dataset with 10% cold-start users in terms of Precision@5 and Recall@5.



(a) Precision@10. (b) Recall@10.

Fig. 5. Performance comparison on dataset with 10% cold-start users in terms of Precision@10 and Recall@10.

in *CagMab* model, we set $\beta = 9$, $\gamma = 0.1$, $\lambda = 0.2$, $\delta = 0.1$, and $d = 120$ through the experiments. More details about parameter selection for *CagMab* will be discussed in the following subsection. From the results in Table I and Table II, we have the following observations:

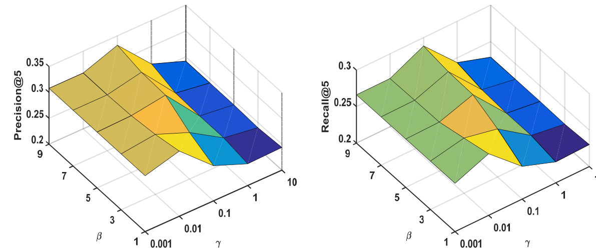
Unknowns recommendation is a difficult task, since users generally provide their knowns in the dataset of E-learning service. However, the *CagMab* framework still outperforms other methods. Either factorUCB considering factorization or LinUCB using contextual features does not show as good performance as *CagMab* which models concept graph structures to optimize latent factors by learning their embedding vectors. The results indicate that incorporating the proposed concept-aware graph embedding with multi-armed bandit can improve the interactive recommendation performance.

The performance of most baseline methods is not satisfactory. Obviously, the accuracy of UCB would be higher than other baselines. Most of the precision and recall of baselines are not over 0.15, so their recommended unknown questions are unconvincing. The reasons can be drawn that previous methods do not consider the factor of concept relations modeling by graph embedding. For our work, previous methods such as LinUCB, factorUCB, and GOBLin which consider contextual features are not more effective than methods without modeling contextual factors.

For the effect of two concept graphs, though CM-CR and CM-CP do not perform as good as *CagMab*, their accuracy performances still outperform other baselines indicating that both graphs are important and have big impact on the recommendation results.

C. Capability of Handling Cold-Start Users

To answer the second question, we investigate the capability of the proposed framework *CagMab* in handling cold-start users. Note that for our recommendation, a cold-start user refers to a user who barely has question answering history. For the cold-start setting, we randomly select 10% users and remove 90% of their records from the training dataset. The performance of different methods with cold-start users is summarized in Figure 4 and Figure 5. As shown in these figures, *CagMab* still outperforms other methods with cold-start users. In addition, the performance of *CagMab* barely degenerates when we introduce cold-start users. These results support that *CagMab* can mitigate the cold-start problem for interactive unknowns recommendation.



(a) Precision@5 with $d = 120$. (b) Recall@5 with $d = 120$.

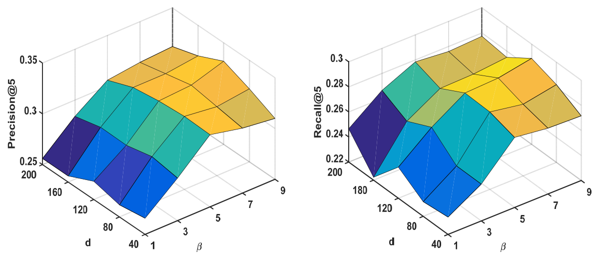
Fig. 6. Parameter sensitivity of *CagMab* w.r.t (β,γ).

D. Parameter Sensitivity

Due to the space limit, we mainly discuss the parameters of latent factor learning phase, since the parameters of multi-armed bandit have been thoroughly discussed in previous work. The Cag phase has a few important parameters, including β which controls the contribution of learning latent factors by regularization technique and γ which controls the learning of graph embedding of $\mathcal{G}_{qc}$, $\mathcal{G}_{cr}$, and $\mathcal{G}_{cp}$. Also, the parameter d is the dimension of learning embedding vectors of nodes in the proposed graphs. By varying the values of β as $\{1, 3, 5, 7, 9\}$, γ as $\{0.001, 0.01, 0.1, 1, 10\}$, and d as $\{40, 80, 120, 160, 200\}$, the results of different settings on (β,γ,$d = 120$), (β,d,$\gamma = 0.1$), and (γ,d,$\beta = 9$) are shown in Figure 6, Figure 7, and Figure 8, respectively. The following observations can be drawn from these figures:

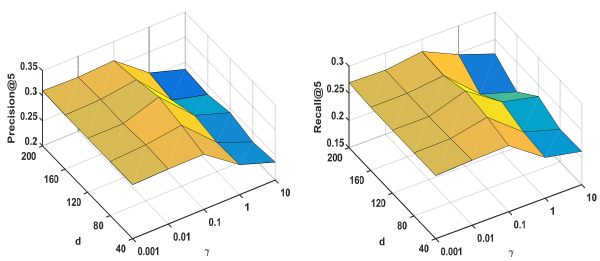
Obviously, for the sensitivity of (β, γ), as γ grows, the accuracy firstly increases, and then decreases with $\gamma > 1$. On the other hand, the performance of our system has smaller difference as β is varied among $\{1, 3, 5, 7, 9\}$. According to our observation, as (β, γ) increase from $(1, 0.001)$ to $(9, 0.1)$, the performance firstly increases and finally reaches the highest accuracy.

For the sensitivity of (β, d), as β increases, the accuracy performance tends to increase, and then becomes stable at certain region, which eases the parameter selection for *CagMab* in practice. For the sensitivity of (γ, d), the best performance reveals with the setting of ($\gamma = 0.1, d = 120$). On the other hand, the dimension d does not affect the performance as much as β and γ. Such observation is desired since larger dimension d leads to more computational cost.



(a) Precision@5 with $\gamma = 0.1$. (b) Recall@5 with $\gamma = 0.1$.

Fig. 7. Parameter sensitivity of *CagMab* w.r.t (β,d).



(a) Precision@5 with $\beta = 9$. (b) Recall@5 with $\beta = 9$.

Fig. 8. Parameter sensitivity of *CagMab* w.r.t (γ,d).

E. Diversity of Recommended Questions

The term 'personalization' refers to a diverse variety of educational programs and learning experiences, which addresses the distinct learning needs of individuals. Therefore, in this subsection, we focus on analyzing the diversity of the recommended questions (user unknowns) generated by *CagMab* for each individual to show that if our system can implement personalized learning. Margalef's richness ($\frac{s\;\,1}{\ln n}$, where s is the number of concepts of the recommended questions, and n is the number of recommended questions which is set as 10) and Shannon's diversity index ($-\sum_{i=1}^{s} p_i \ln p_i$, where p_i is the proportional abundance of the i-th concept) are two important measures which can evaluate the concept richness and concept diversity in our recommended questions (unknowns). The former mainly focus on the number of recommended unknown concepts, while the latter consider both the richness and the evenness of recommended unknown concepts.

As shown in Figure 9, we retrieve the main concepts of recommended questions for each user and draw a case study on 100 users to see the richness and the diversity of these recommended questions. Note that the richness result is associated with the left y-axis, while the diversity result is associated with the right one. The maximum richness and diversity are

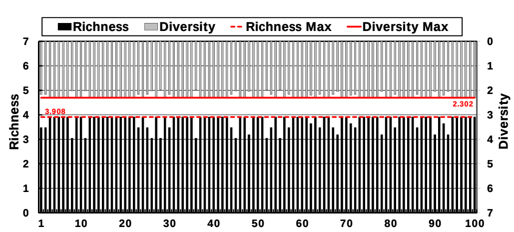


Fig. 9. The richness and the diversity of recommended questions.

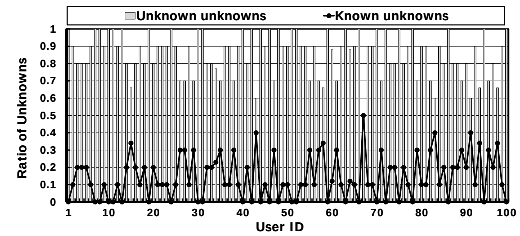


Fig. 10. The ratio of known unknowns and unknown unknowns in recommended questions.

($\frac{s\;\,1}{\ln n} = \frac{10\;\,1}{\ln 10} = 3.908$) and ($\ln n = \ln 10 = 2.302$), respectively. According to our analysis, the recommended questions of most users have high richness. The average richness and diversity of recommended questions are 3.738 and 2.244 which are close to the maximum values of richness (as the dot line shown in Figure 9) and diversity (as the solid line shown in Figure 9). Obviously, *CagMab* can interactively recommend unknowns with high richness and diversity for users.

F. Capability of Exploring Unknown Unknowns

"We know there are known unknowns; that is to say we know there are some things we do not know. But there are also unknown unknowns - the ones we don't know we don't know. And it is the latter category that tends to be the difficult ones." said by United States Secretary of Defense, Donald Rumsfeld. In our system, we also try to examine if we can explore unknown unknowns for user learning progress. Since it is important if users are aware of personal unknowns, and their knowledge can be enriched effectively. However, it is not an easy task to explore unknown unknowns as we only have known unknowns in training phases.

In this experiment, we retrieve user unknown concepts from training data, which is assumed as 'known unknowns'. Since users have incorrectly answered these questions, they should know the concepts of these questions are their unknowns. We also retrieve user unknown concepts from the recommended questions. If the unknown concepts are not included in known unknowns, these unknown concepts are identified as 'unknown unknowns'. By investigating the ratio of 'unknown unknowns' and 'known unknowns' in our recommended questions, we can see if our unknowns learning system can effectively explore unknown unknowns for users. As shown in Figure 10, we draw a case study on the interactive recommended questions of 100 users for the ratio of known unknowns and unknown unknowns. According to our analysis, most recommended unknowns are unknown ones, and the average ratio of recommended unknown unknowns is 0.854 which is much higher than that of recommended known unknowns. Obviously, *CagMab* can explore unknown unknowns for users. These unknown unknowns are mainly discovered (i) from the concept relations among known unknowns and unknown unknowns, which is modeled by concept-aware graph embedding; (ii) from the exploration phase of multi-armed bandit algorithm. Overall, the results imply that *CagMab* can explore unknown unknowns for effective user learning progress.

V. Related Work

Multi-Armed Bandit (MAB) aims to model an agent which can simultaneously acquire new knowledge (exploration) and optimize users' decisions based on the existing knowledge (exploitation). This class of models have been widely used in many real-world applications, including recommender systems [2], display advertising [8] and network embedding [9]. The MAB problem was early addressed in [10]. Aure et al. firstly developed upper confidence bounds (UCB) based approaches for all reward distributions with bounded support [11] [12]. Recently, more and more works focus on the contextual bandit problem. Li et al. [2] first designed a general contextual bandit algorithm (LinUCB). In their work, they modeled a MAB problem by considering the contextual information of both users and articles for personalized news recommendation. Li et al. also investigated various clustering techniques for the contextual bandit problem [13]. They applied a collaborative filtering method (COFIBA) to group items based on the similarity of user clusters and the mined preference patterns. Cesa et al. [14] exploited network structures for the networked bandit problem. A global arm selection strategy (GOBLin) is proposed such that each node (user) is assumed to share signals (contexts and rewards) with its neighbors in a network. Later on, Wang et al. [15] developed a factorization-based bandit (factorUCB) algorithm. In their method, an item selection strategy is developed based on the upper confidence bound over an incrementally constructed user-item rating matrix. Different from the aforementioned approaches, Agrawal et al. [16] designed a general contextual bandit problem with Thompson Sampling. Based on the linear reward functions, the contexts are provided in an adaptive way.

In recent, studies in E-learning systems focus on providing better services with course recommendation and learning paths discovery. These research [17] [18] applied content-based and collaborative filtering recommendation to provide learning courses for users. In addition, Meshram et al. [19] proposed a Thompson sampling-based online reinforcement learning to recommend online courses referring to user preference. Abel et al. [20] evaluated different strategies such as collaborative filtering for recommending online learning forums to users. Another popular recommendation issue on E-learning services is to generate user learning paths [21]. The solution of such problem comprises all possible learning sequences and the objective function is to minimize the devised penalty function to evaluate the sequencing, such as ontology [22] based learning and Bayesian probability theory [23]. However, none of these studies considers user unknowns exploration nor concept-aware graphs illustrating relationships among items. Moreover, previous bandit algorithms on interactive learning systems lead to a very low accuracy performance for unknowns learning. In our work, we fully explore the rich information embedded among the concept graphs of items and integrate it seamlessly with the multi-armed bandit framework for better interactive unknowns recommendation.

VI. Conclusions

In this paper, we investigate if the concept graphs can be leveraged to enhance the performance of interactive unknowns recommendation on E-learning systems. To effectively discover user unknowns, we incorporate concept-aware graph embedding into a multi-armed bandit model, which leads to a novel framework *CagMab*. Moreover, we implement our method for the case studies on real E-learning dataset. The data analysis and experimental results show some particular phenomenon and demonstrate that the proposed system framework is effective and practical.

Acknowledgement: This work was supported in part by Junyi Academy and by Ministry of Science and Technology, R.O.C., under Contract 107-2221-E-006 -165-MY2 and 107-2218-E-006-040.

References

[1] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013.

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

[3] Q. Wu, H. Wang, Q. Gu, and H. Wang, "Contextual bandits in a collaborative environment," in *SIGIR*, 2016.

[4] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *NIPS*, 2008.

[5] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *WWW*, 2015.

[6] B. Recht, C. Re, S. Wright, and F. Niu, "Hogwild: A lock-free approach to parallelizing stochastic gradient descent," in *NIPS*, 2011.

[7] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári, "Improved algorithms for linear stochastic bandits," in *NIPS*, 2011.

[8] W. Li, X. Wang, R. Zhang, Y. Cui, J. Mao, and R. Jin, "Exploitation and exploration in a performance based contextual advertising system," in *KDD*, 2010.

[9] X. Huang, Q. Song, J. Li, and X. Hu, "Exploring expert cognition for attributed network embedding," in *WSDM*, 2018.

[10] J. C. Gittins, "Bandit processes and dynamic allocation indices," *Journal of the Royal Statistical Society. Series B (Methodological)*, 1979.

[11] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, 2002.

[12] P. Auer, "Using confidence bounds for exploitation-exploration trade-offs," *Journal of Machine Learning Research*, 2002.

[13] S. Li, A. Karatzoglou, and C. Gentile, "Collaborative filtering bandits," in *SIGIR*, 2016.

[14] N. Cesa-Bianchi, C. Gentile, and G. Zappella, "A gang of bandits," in *NIPS*, 2013.

[15] H. Wang, Q. Wu, and H. Wang, "Factorization bandits for interactive recommendation." in *AAAI*, 2017.

[16] S. Agrawal and N. Goyal, "Thompson sampling for contextual bandits with linear payoffs," in *ICML*, 2013.

[17] J. Lu, "A personalized e-learning material recommender system," in *ICITA*, 2004.

[18] M. Otair and A. Hamad, "Expert personalized e-learning recommender system," *Retrieved June*, 2005.

[19] R. Meshram, A. Gopalan, and D. Manjunath, "Optimal recommendation to users that react: Online learning for a class of pomdps," in *CDC*, 2016.

[20] F. Abel, I. I. Bittencourt, E. Costa, N. Henze, D. Krause, and J. Vassileva, "Recommendations in online discussion forums for e-learning systems," *IEEE transactions on learning technologies*, 2010.

[21] P. Basu, S. Bhattacharya, and S. Roy, "Online recommendation of learning path for an e-learner under virtual university," in *ICDCIT*, 2013.

[22] J. M. Gascueña, A. Fernandez-Caballero, and P. Gonzalez, "Domain ontology for personalized e-learning in educational systems," in *ICALT*, 2006.

[23] J. M. Marquez, J. A. Ortega, L. Gonzalez-Abril, and F. Velasco, "Creating adaptive learning paths using ant colony optimization and bayesian networks," in *IJCNN*, 2008.