Knowledge-Based Systems 251 (2022) 109158

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A knowledge-enhanced contextual bandit approach for personalized recommendation in dynamic domains

Mingxin Gan [a,*], O-Chol Kwon [a,b]

[a] Department of Management Science and Engineering, School of Economics and Management, University of Science and Technology Beijing, Beijing 100083, China
[b] Department of Information Science, Kim Il Sung University, Pyongyang, Democratic People's Republic of Korea

ARTICLE INFO

Article history:
Received 27 October 2021
Received in revised form 25 May 2022
Accepted 26 May 2022
Available online 3 June 2022

Keywords:
Recommender system
Knowledge graph
Contextual multi-armed bandit
Dynamic domain

ABSTRACT

Recently, contextual multiarmed bandits (CMAB)-based recommendation has shown promise for applications in dynamic domains such as news or short video recommendation, where items are changing over time. There are two key challenges in constructing a CMAB-based system. First, a user's historical rating data are usually sparse, which restricts the precise representation of dynamic contextual information. Second, it is difficult to design a personalized recommendation policy because of the high diversity of users' selection behaviors over time. Therefore, existing CMAB-based recommendation methods mainly focus on the improvement of recommendation accuracy. This leads to difficulty in capturing the dynamic personalized preferences of users. To overcome these limitations, we design a novel knowledge-enhanced contextual-bandit approach that combines the dynamic recommendation strategy of the bandit algorithm and the knowledge representation of the knowledge graph. We argue that a user's selection history for items reflects his/her favorite attributes, which will be repeated in the future. Accordingly we propose the knowledge-enhanced CMAB model, which leverages the contextual information of both users and items obtained by knowledge graph-based embedding. We validate the performance of the proposed model using two public datasets. Experimental results show that our method outperforms state-of-the-art methods in terms of both recommendation accuracy and diversity.

1. Introduction

With the rapid development of internet technology and the information industry, the online environment has become natural for people. It also brings people a crisis of choice every day. A promising solution is to develop an accurate and personalized recommender system to support people in making decisions. Recommender systems are widely applied in many areas of human life, such as movie recommendations [1,2], music recommendations [3,4], and news recommendations [5,6]. The research interest in these fields remains high because consists of many interesting research areas, for example, reinforcement learning-based recommender systems [7,8], and knowledge-based recommender systems [9,10].

Collaborate filtering (CF) is still the main method of recommender systems. CF models user preference based on the similarity of items or users from historical interaction data [11,12]. There are two key examples of CF: Amazon's "people who bought X also bought Y" and Netflix's "people like you liked X" [13]. CF has achieved success and has broad application prospects in many fields, such as e-commerce and social networks [14]. However, CF has some serious drawbacks. First, CF uses the user and item's interaction history to make recommendations, and these interaction data constitute a large sparse matrix. In other words, CF inevitably faces a serious lack of knowledge. To address these problems, several types of auxiliary information, such as item attributes [15–17], contextual information [18–20], user and item reviews [21,22], and user's social relation [23–25], have been used. The challenge in using auxiliary information is that the knowledge sources have not been standardized, which makes it difficult to extract knowledge and to evaluate performance. In addition, CF suffers from the accuracy–diversity dilemma of recommendation. Many studies [26–28] have been conducted to address the accuracy-diversity dilemma in recommender systems. However, some recent studies [13,29] report findings that CF itself negatively affects the diversity of recommendations. Finally, the existing CF learns a static recommendation model using the given training data. Since users select items while interacting with a recommender system in real time, traditional CF fails to accurately capture the dynamic interaction between a user and a recommender system. Consequently, CF cannot capture current

* Corresponding author.
E-mail address: ganmx@ustb.edu.cn (M. Gan).

https://doi.org/10.1016/j.knosys.2022.109158
0950-7051/

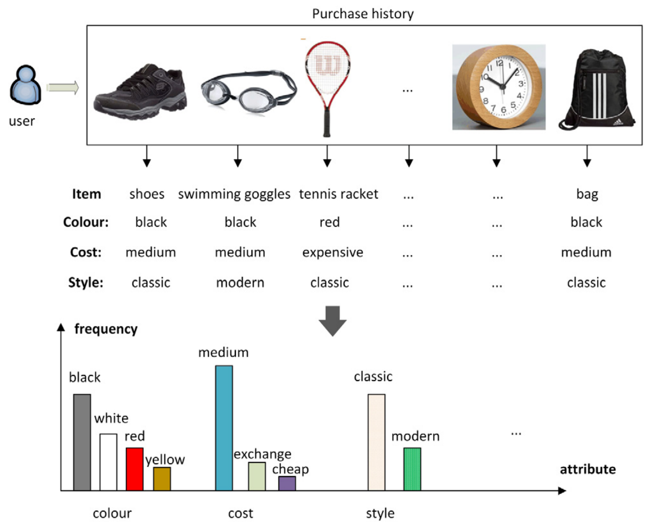

Fig. 1. An example of repeated item attributes in purchase history.

preferences accurately when a user wants to make a selection decision.

In recent years, many studies have adopted knowledge graphs as auxiliary information in recommender systems. A knowledge graph is a multirelational graph composed of entities (nodes) and relations (different types of edges) [30]. Items and their attributes can be mapped into the knowledge graph to understand the mutual relations between items [31]. Knowledge graph-based recommendation is recognized as a promising method that reinforces the knowledge representation of users and items while realizing a diversity of recommendations, through knowledge graph embedding, metapath selection, and the combination of the two methods [30]. However, current methods consume considerable time in the learning process and are therefore regarded as static recommendation methods. This makes it difficult for the recommender system to accurately capture users' dynamic preferences.

On the other hand, exploration–exploitation methods, a.k.a. multiarmed bandits (MAB), have been recognized as an outstanding solution for applications in dynamic domains such as news recommendation or short video recommendation [32]. Bandit-based recommendation methods use an exploration–exploitation mechanism with its inherent dynamic characteristics to balance the short- and long-term benefits of recommendation. This makes it an important solution for the accuracy–diversity dilemma. Bandit-based approaches are classified into general MAB and contextual MAB (CMAB), depending on whether contextual information is used. CMAB shows higher performance by considering contextual information of users and items. However, since CMAB-based methods rely heavily on the contextual representation of items, their performance is limited if the knowledge representation is insufficient.

The mission of a recommender system is to support people's selection decisions. Each person has his or her own preferences, taste or habits, which also affect his or her selection behavior. For example, people have their favorite movie genres or actors, and they have favorite colors. These habits and preferences are relatively stable. Fortunately, although the number of items is enormous, the number of attributes of items is relatively small. Therefore, although it is difficult to predict what a user will select next, the attributes that the user will prefer next are predictable. In other words, a user always selects certain values of these attributes when purchasing a product. Fig. 1 depicts an example of repeated item attributes in a user's purchase history. Each item has some attributes, such as color, price and style. When a user bought shoes, he or she selected a black one, at a medium price and a classic style. As Fig. 1 shows, the user prefers black color, medium price and classic style over other attributes. Thus, we assume that these attributes reflect his/her personal preferences and taste and with a high possibility will be repeated in the future. However, existing studies ignore the effects of the stable attributes reflecting the personalized preference and taste of a user with regards to his or her future selection of items.

To overcome these limitations, we propose a knowledge-enhanced LinUCB (Ke-LinUCB) model that combines the dynamic advantages of the bandit approach and the advantages of the knowledge representation of a knowledge graph. Ke-LinUCB first learns the contextual representation of users and items using a user–item interaction matrix and a knowledge graph. Next, by using the contextual representation, we implement the one choice - multi change strategy for the above assumption. To our knowledge, this is the first study that combines the bandit approach and a knowledge graph.

The main contributions of this study are listed below:

(1) To address the contextual representation problem of bandit-based methods, we propose a new knowledge-enhanced CMAB model called Ke-LinUCB for personalized recommendation in the dynamic domain. Unlike existing methods using given training data, the proposed Ke-LinUCB learns through dynamic interaction between users and a recommender system. This makes it possible to capture the user's preference dynamically. We use the knowledge graph attention network to obtain the knowledge representation of CMAB, which effectively models the high-order connectivities in the knowledge graph.
(2) To model the behavior of a user interacting with a recommender system, we explore an intention-selection mechanism and model an iterative process of attribute selection by using a bandit algorithm. Compared to previous recommendation methods, the proposed method more accurately models the user–system interaction in the real world and effectively copes with the cold-start problem by using the knowledge representation of attributes.
(3) We perform extensive experiments on two real-life datasets. The proposed method improves both the recommendation accuracy and diversity through a more personalized recommendation policy. The results show that our method outperforms state-of-the-art methods for dynamic recommendations.

The rest of the paper is organized as follows. In Section 2, we briefly review the related work on CF, the bandit approach, and knowledge graph-based recommender systems. Section 3 introduces some background concepts and motivation underlying our novel approach. Our proposed KE-LinUCB is described in detail in Section 4. Section 5 presents the experimental results including a comparison with some baselines. Finally, Section 6 concludes the paper.

2. Related work

In this section, we first briefly review CF-based, bandit-based and knowledge graph-based recommendation algorithms and then analyze their limitations.

2.1. CF-based recommender system

CF is the most successful and representative method in the era of recommender systems. CF-based recommendation methods have shown great success and have a wide range of applications in many fields, such as social networks and e-commerce [14]. CF-based approaches are classified into three categories: memory-based [33–35], model-based [36–40] and hybrid methods [41]. The key idea of CF is that if users u and v rate m items similarly, they have similar preferences, and hence, they will rate other items similarly [36]. In other words, CF focuses on similarity in recommendations. Despite the remarkable achievements of CF, it still suffers from the cold-start and data sparsity problems. In addition, some recent studies report that this approach negatively affects the diversity of recommendations. Wang et al. [29] reported that the Matthew effect often leads to unwanted results in user (item)-based CF. Lee and Hosanagar [13] demonstrated across a wide range of product categories that the traditional CF is associated with a decrease in product sales diversity relative to a world without recommendations. This means that the traditional CF reduces users' item selection diversity. In this paper, we adopt a bandit approach to balance the exploration–exploitation of recommendations. Unlike existing CF-based methods, our method focuses on the individualization of recommendations using both the enhanced knowledge representation of users (items) and the exploration strategy of bandit.

2.2. Bandit-based RS

In recent years, bandit-based methods have been widely applied in recommender systems due to their advantages in dealing with dynamic domains and effectiveness for resolving the exploration–exploitation dilemma [32,42–45]. In the context of recommender systems, bandit-based methods are divided into context-free MAB [46,47] and CMAB [44,48], depending on whether the contextual information of users and items is considered. MAB includes ε-greedy, upper confidence bounds (UCB) [49], Thompson sampling (TS) [46,50], EXP3 [47] and so on. UCB is a method that combines exploitation (selecting the arm with the highest reward) and exploration and has become a starting point for many MAB-based methods. UCB's exploration is a way to give more opportunities to arms with less average rewards, and exploration works by chance for such arms. Another representative method is TS, which uses a beta distribution with two parameters; the higher the parameters are, the tighter the concentration of the distribution around its mean [46].

MAB-based recommender systems have the advantage of being simple to implement, but they are far from personalized recommendations because they do not take into account arm and user information. To overcome the limitations of MAB, CMAB-based recommendations have been introduced. Li et al. [44] proposed a famous CMAB algorithm, namely, LinUCB. LinUCB shows superior performance compared to MAB methods by using the user–item features and exploration–exploitation dilemma. Agrawal and Goya [48] proposed the TS, which uses the contextual information of items. Li et al. [32] investigated an adaptive clustering technique for content recommendation based on exploration–exploitation strategies in contextual multiarmed bandit settings. Gutowski et al. [42] recently proposed a new approach of CMAB assessment focused on the evaluation of individual accuracy, called Sliding Window LinUCB (SW-LinUCB) which aims at decreasing the number of very unsatisfied users.

Despite extensive research on recommender systems based on MAB and CMAB, prior studies have a series of problems. First, the exploration function is biased toward randomness. For example, existing algorithms obtain different results depending on the applications or on the datasets they operate on [43]. In contrast, we address the exploration–exploitation dilemma by reducing the randomness of the exploration function on the basis of the assumption that a user's selection of an item is determined by its attribute set. Second, prior work seldom considered the knowledge representation of users and items. Instead, in this paper, we adopt a knowledge graph to acquire standard and global knowledge representations for items and users and use it in the setting of the bandit method.

2.3. Knowledge graph-based recommender system

Recently, knowledge graphs have been increasingly used in recommender systems [30]. Based on the knowledge graphs are utilized, such recommender systems can be divided into three categories: embedding-based methods, path-based methods, and unified methods [12]. The key idea of embedding-based methods [31,51][52] is to embed components of a knowledge graph, including entities and relations, into continuous vector spaces to simplify computations while preserving the inherent structure of the knowledge graph [30]. Knowledge graph embedding (KGE) algorithms are the core part of embedding-based methods, including TransE [53], TransH [54], TransR [55], TransD [56], and DistMult [57]. Path-based methods build a user–item graph and leverage the connectivity patterns of the entity in the graph for recommendation [12]. Path-based methods can be divided into traditional path-based methods and path embedding methods. Traditional path-based methods use metapaths in heterogeneous information networks (HINs) [58]. These methods generally integrate MF with meta-paths [59] and meta-graphs [60]. Hu et al. [61] proposed MCRec, which learns explicit representations for metapath-based context tailored for the recommendation task, and Sun et al. [62] and Wang et al. [63] proposed RKGE and KRRN, which encode the entire path using a recurrent network and LSTM, respectively. The unified method is based on the idea of embedding propagation [12]. RippleNet [64] is the first work to realize this idea. Since RippleNet was proposed, many methods [65–68] have adopted embedding propagation on item or item–user graphs. What these methods have in common is that they all use attention mechanisms.

Although knowledge graph-based recommender systems have achieved notable performance, training requires considerable time, which often leads to considering static(i.e., pretrained) systems. In many scenarios, such as news recommendations and online shopping, recommender systems should immediately capture users' intentions and combine them with prior knowledge to make reasonable recommendations. Accordingly, in this paper, we use a knowledge graph to acquire knowledge representations of items and users, which is combined with the proposed bandit algorithm to make dynamic recommendations.

3. Background and problem formulation

In this section, we first explore the behavior of users who select items and propose an intention-selection mechanism to explain the process. Then, we briefly introduce a contextual bandit problem for recommender systems and analyze the research space of this area.

3.1. Intention-selection mechanism

Existing recommendation methods, especially the CF, aim to recommend items in the item set that are suitable for users based on the similarities between users (items). In general, the number of items is very large, so the recommendation that focuses on item selection is limited by the sparsity of data. In reality, however, in most cases, the user expresses his or her intention before selecting an item.

As we know, in the case of product purchase, the user first inputs key words of the product he or she wants to purchase and then selects favorite items from candidates displayed by a recommendation system. We call this user's selection process an intention-selection mechanism. In the intention-selection mechanism, the goal of a recommender system is to minimize the selection cost by recommending items with attributes that the user prefers. For example, if a user enters "bag" as a key word in an online store, bags should be recommended with various attributes (for example, price, color, shape, etc.) that the user may prefer. In this mechanism, the user's item selection process is the attribute selection of one of the items. Based on this analysis, we assume that a user's selection process for items is the selection process of his favorite attributes and that these attributes are repeated. Since the number of attributes is much smaller than the number of items, this mechanism may be an effective solution to the data sparsity and cold-start problem. We model this mechanism by using the CMAB algorithm and a knowledge graph. In the next section, we introduce the CMAB problem and its representative algorithm, LinUCB.

3.2. The contextual bandit problem

Guided by Langford and Zhang [69] and Li et al. [44], we define the CMAB problem for recommender systems as follows: Let $A = \{a_1, a_2, \ldots, a_k\}$ be a set of k independent arms in the context of a recommender system, which corresponds to items for recommendation. Let $U = \{u_1, u_2, \ldots, u_n\}$ be a set of n players, for example, users in a recommender system. Let $X \subseteq \mathbf{R}^d$ denote the set of information that is obtained from A and U. At each time $t \in [1, T]$, the contextual bandit Algorithm B chooses optimal arm $a_t \in A$ and receives payoff $r_{t,a_t} \in \{0, 1\}$ whose expectation depends on the context $x_{t,a_t} \in X$. $T \in N$ is the time horizon. Algorithm B then improves its selection strategy with the existing information. One simplest and effective approach is to assume that the expected payoff of arm a is linear in its context vector $x_{t,a}$ with an unknown coefficient vector θ_a such that $\mathbf{E}[r_{t,a}|x_t] = x_{t,a}^T\theta_a$. The T-time payoff of Algorithm B is defined as $\mathbf{E}\left[\sum_{t=1}^{T} r_{t,a_t}\right]$.

Research on the CMAB problem can be largely divided into two primary areas. The first is to develop recommendation policies. Let $\Pi : X \to A$ denote a set of recommendation policies where the optimal policy is $\pi^* = \arg\max_{\pi\in\Pi} \mathbf{E}_{r,x}[r_{t,\pi(x)}]$ [43]. Let $\pi_t \in \Pi$ be a policy found by a CMAB Algorithm B at time t. Then the T-time regret of B is $\rho_T(B) = \mathbf{E}_{r,x}[r_{t,\pi^*(x_t)} - r_{t,\pi(x_t)}]$. The goal of B is to minimize $\rho_T(B)$. In this scenario, the context is constant for all algorithms. The representative algorithm is LinUCB. At each trial t, LinUCB chooses the arm a_t so that

$$a_t = \arg\max_{a\in A_t}\left(x_{t,a}^T\hat{\theta}_a + \alpha\sqrt{x_{t,a}^T A_a^{-1} x_{t,a}}\right) \tag{1}$$

where $\hat{\theta}_a = A_a^{-1}b_a$, $A_a = D_a^T D_a + I_d$, and D_a is the design matrix of dimension $n \times d$ (n training inputs of d features). In this paper, we refer to (b_a, A_a) as the internal state of arm a. Like all UCB methods, LinUCB is composed of two parts: expected payoff (exploitation) $x_{t,a}^T\hat{\theta}_a$ and predictive variance (exploration) $\alpha\sqrt{x_{t,a}^T A_a^{-1} x_{t,a}}$, where α plays a role in regulating the trade-off between exploration and exploitation. The first term of the LinUCB formula works like CF, and the second term is a random selection. Thus, LinUCB also cannot avoid data sparsity and cold-start problems. In Section 4.2, we solve this problem with the modified LinUCB algorithm. The second is the contextual representation problem of CMAB. However, few works have been conducted to enhance the contextual knowledge representation of CMAB. In Section 4.1, we use a knowledge graph to learn the contextual representation of users and items.

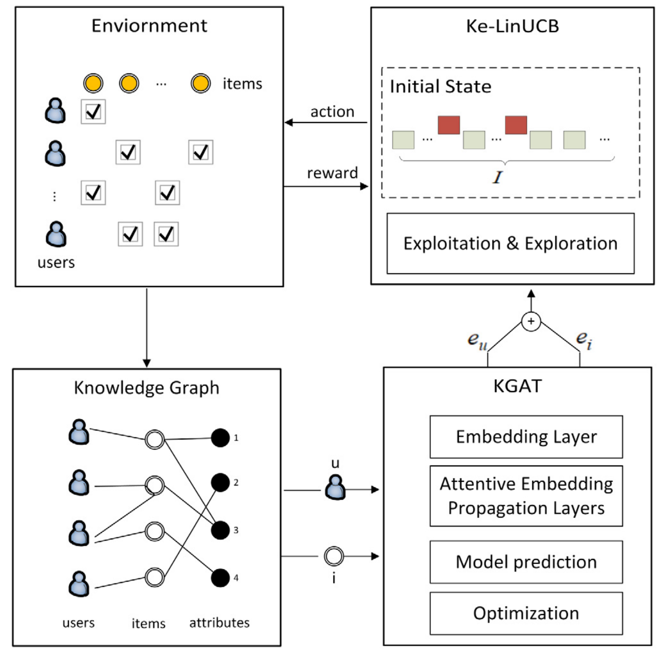

Fig. 2. Overview of the proposed approach.

4. Methods

In this section, we introduce our new approach, which is built upon the combination of the LinUCB [44] algorithm and the method of knowledge representation [68]. Fig. 2 shows an overview of the proposed approach which consists of four parts:

Environment: This includes users and items and interactions between them; for example, a rating matrix. This part receives an action from the CMAB and returns a corresponding reward. The action is to recommend an item to a user, and the reward results from whether the user actually selects the item. Knowledge graph: A knowledge graph is an external database in which users, items, and their properties are linked in a graph structure. It is a knowledge source for the contextual representation of items and users. Knowledge Graph Embedding Module: This module performs representation learning by embedding all nodes and connections in the knowledge graph. We adopt the knowledge graph attention network (KGAT) [68] for knowledge graph embedding. This module provides the learned representation of the user and item to the CMAB module. Knowledge-enhanced LinUCB: This part uses an exploration–exploitation strategy to recommend items to a user. At this time, the knowledge representation of users and items obtained from the Knowledge Graph Embedding Module is used. This module passes recommended items to the environment, receives its rewards, and continues to learn from the environment.

We shall introduce the proposed method in two steps. The first step is a preliminary one and it aims to obtain the knowledge representation of users and items from a user–item interaction matrix and a knowledge graph. The second step is the dynamic recommendation process using knowledge-enhanced LinUCB.

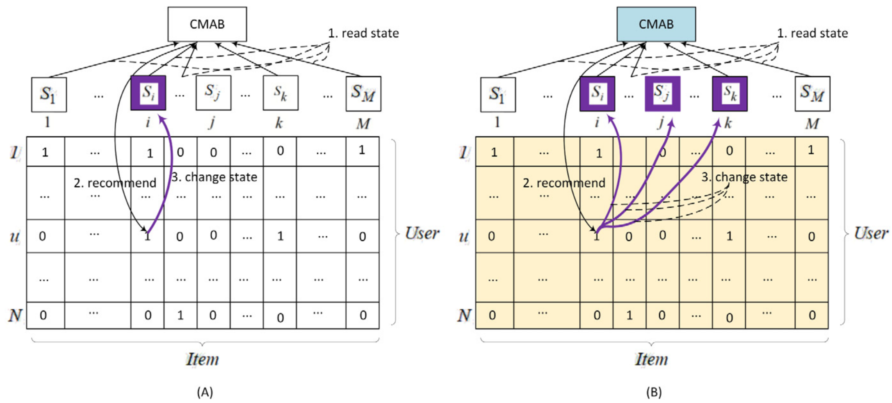

Fig. 3. LinUCB (A) and proposed Model (B).

4.1. Context representation using a knowledge graph

We use KGAT [68] for the context representation of our CMAB method. KGAT has three main components: an embedding layer, an attentive embedding propagation layer and a prediction layer. Given user set U, item set I, user–item interaction matrix R and knowledge graph G, we use the KGAT to extract the user representation $EU = \{e_{u_1}, e_{u_2}, \dots, e_{u_n}\}$ and item representation $EI = \{e_{i_1}, e_{i_2}, \dots, e_{i_m}\}$ as follows.

4.1.1. Embedding layer

This layer is the module used to obtain the vector representation for entities and relations of the knowledge graph. KGAT uses TransR to embed the entities and relations of the knowledge graph. Given a knowledge graph G, for a triplet $(h, r, t) \in G$, the energy score is formulated as follows:

$$En(h, r, t) = \|\mathbf{W}_r e_h + e_r + \mathbf{W}_r e_t\|_2^2, \tag{2}$$

where $\mathbf{W}_r \in \mathbf{R}^{k\times d}$ is the transformation matrix of relation r. For a triple (h, r, t), h and t are user or item attributes, and r is their relation in the knowledge graph. The loss function for training is as follows:

$$L_{KG} = \sum_{(h,r,t,t')\in T} -\ln\sigma(En(h, r, t') - En(h, r, t)), \tag{3}$$

where $T = \{(h, r, t, t')|(h, r, t) \in G, (h, r, t') \notin G\}$, and (h, r, t') is a broken triplet constructed by randomly replacing one entity in a valid triplet, and $\sigma(.)$ is the sigmoid function.

4.1.2. Attentive embedding propagation layers

This module consists of multiple layers of the graph convolution network, and every layer has three components: Information Propagation: This part models the propagation of information from one entity to another. All entities are influenced by surrounding entities in the graph network structure. To represent the first-order connection of an entity, KGAT uses the ego-network [70]. Knowledge-aware Attention: This part uses a relational attention mechanism to control the propagation between each pair of entities. Unlike GCN [71] and GraphSage [72], KGAT uses the discount factor between two entities. Information Aggregation: This part aggregates the entity representation and its ego-network representation as the new representation. In this stage, three types of aggregation, i.e., GCN Aggregator, GraphSage Aggregator and Bi-Interaction Aggregator, are used. It then propagates to multiple layers to explore higher-order connectivity information.

4.1.3. Model prediction

After performing L-order propagation, for the user node and item node, the overall representation is obtained as follows:

$$e_u^* = e_u^0\| \cdots \|e_u^L, e_i^* = e_i^0\| \cdots \|e_i^L, \tag{4}$$

where $||$ is the concatenation operation. Then, KGAT uses the inner product to predict its matching score:

$$\hat{y}(u, i) = {e_u^*}^T e_i^* \tag{5}$$

4.1.4. Optimization

To optimize the recommendation model, KGAT uses the BPR loss [73] as follows:

$$L_{CF} = \sum_{(u,i,j)\in O} -\ln\sigma(\hat{y}(u, i) - \hat{y}(u, j)), \tag{6}$$

where $O = \{(u, i, j)|(u, i) \in R^+, (u, j) \notin R^-\}$ denotes the training set, R^+ indicates the observed (positive) interactions between user u and item j, R^- is the sampled unobserved (negative) interaction set, and $\sigma(.)$ is the sigmoid function. Finally, the objective function to learn Eqs. (3) and (6) jointly, is as follows:

$$L_{KGAT} = L_{KG} + L_{CF} + \lambda\|\Theta\|_2^2, \tag{7}$$

where $\Theta = \{\mathbf{E}, \mathbf{W}_r, \forall l \in R, \mathbf{W}_1^{(l)}, \mathbf{W}_2^{(l)}, \forall l \in \{1, \dots, L\}\}$ is the model parameter set, and $\mathbf{E}$ is the embedding table for all entities and relations, and L_2 regularization parameterized by λ on Θ is conducted to prevent overfitting.

4.2. Knowledge-enhanced linucb for recommender systems

In the original LinUCB, the CMAB algorithm selects an item and only changes its internal state. We refer to this strategy as a one choice-one change strategy. That is, LinUCB is a contextual bandit algorithm based on a one choice-one change strategy. Instead, our approach is to simultaneously change the internal state of several items similar to that item by reward of the optimal item. Fig. 3 shows the contrast between the LinUCB and our model framework. Fig. 3 (A) shows the item selection and internal state changes in LinUCB. The CMAB algorithm reads the internal state of each item and recommends selected item i to user u and then changes the item's internal state S_i according to the reward. The updated S_i will affect other users' selection of item i. However, as described in Section 3.2, LinUCB cannot know items with properties similar to those already recommended in the exploration process. That is, it is based on the blind knowledge discovery method, and the efficiency of personalization of recommendations is low.

In contrast, our key idea is to increase the selectivity of items with attributes similar to those already recommended in the exploration process. Fig. 3 (B) shows the selection and change process of the proposed method. In every trial, the proposed method first selects the optimal item a_t using Eq. (1) and then looks for the items that are most similar to this item. We use the cosine similarity to evaluate the similarity of each item. At this time, the knowledge representation (Section 4.1) obtained by using the knowledge graph is used. Since the knowledge representation of an item by the knowledge graph is determined by the attributes of the items, items with similar attributes have high similarity. After selecting the items that are most similar to the optimal item, the internal states of the selected items are changed together with the internal state of the optimal item. We term this strategy one choice – multi change strategy.

We describe the above process as follows:

(1) Fix M independent items with their knowledge representation and set constants k, α.
(2) Select a user $u \in U$ with its knowledge representation $e_u \in EU$.
(3) For each arm $a_t \in A$, obtain the context information $x_{t,a}$ using e_u and $e_{a_t} \in EI$. Then calculate $p_{t,a} = x_{t,a}^T\hat{\theta}_a + \alpha\sqrt{x_{t,a}^T A_a^{-1} x_{t,a}}$.
(4) Choose the item $a_t^* = \arg\max_{a \in A_t} p_{t,a}$ and observe the pay-off r_t (0 or 1).
(5) Update the internal state of a_t^*.
(6) Select the k items $S_k = \{s_1, s_2, \ldots, s_k\}$ with their context that is most similar to a_t^*.
(7) Update the internal states of every item of S_k.
(8) Repeat from (2) to (7) until some condition is satisfied.

We term this method Ke-LinUCB and term the combination with LinUCB-Hybrid Ke-LinUCB-Hybrid. Algorithms 1 and 2 describe the Ke-LinUCB method and Ke-LinUCB-Hybrid, respectively.

Algorithm 1. Ke-LinUCB
Inputs: EU, EI, $\alpha \in \mathbf{R}^+$, $k \in N$, $m \in N$
1: **for** $t = 1$**to** T **do**
2: Select a user $u \in U$ with his representation $u_t \in EU$
3: **for** all $a \in A$ **do**
4: **if** a is a new arm **then**
5: $Y_a \leftarrow I_d$; $b_a = O_{d\times 1}$
6: **end if**
7: Select the representation v_a of a, $v_a \in EI$
8: $x_{t,a} \leftarrow u_t || v_a$(concatenation)
9: $\hat{\theta}_a \leftarrow Y_a^{-1} b_a$; $p_{t,a} \leftarrow \hat{\theta}_a^T x_{t,a} + \alpha\sqrt{x_{t,a}^T Y_a^{-1} x_{t,a}}$
10: **end for**
11: Choose the arm $a_t = \arg\max_{a \in A_t} p_{t,a}$ and observe the pay-off r_t
12: $Y_{a,t} \leftarrow Y_{a,t} + x_{t,a_t} x_{t,a_t}^T$; $b_{a,t} \leftarrow b_{a,t} + r_t x_{t,a_t}$
13: **if** $r_t = 1$ **and** $t < m$ **then**
14: Select the k arms $S_k = \{s_1, s_2, \ldots s_k\}$ with their context that is most similar to a_t.
15: Get their context $x_{s1}, x_{s2}, \ldots, x_{sk} \in EI$
16: **for** $j = 1$ **to** k **do**
17: $Y_{s_j,t} \leftarrow A_{s_j,t} + x_{s_j} x_{s_j}^T$; $b_{s_j,t} \leftarrow b_{s_j,t} + r_t x_{s_j}$
18: **end for**
19: **end if**
20: **end for**

Unlike LinUCB, we use contextual representations obtained from knowledge graph embedding in steps 2 and 7. Steps 13 to 19 show the main implementation of the proposed method. In practical applications, how to apply this one-choice multichange strategy is also important. In the initial stage of recommendation, it is necessary to recommend items with properties similar to those already recommended because of data sparseness. However, as the data grow over a period, this is unnecessary; otherwise, it may cause overfitting. In other words, it becomes an important issue as to how many items similar to the items to be noticed in the one choice - multi change strategy are considered and for how long to apply this strategy. We confirm these values through experiments in 5.3.2 and 5.3.3. In Algorithm 1, m specifies the length of time for this strategy. For convenience, we set the default value of m as T. When m is less than T, we use the notation KE-LinUCB-m.

Algorithm 2: Ke-LinUCB-Hybrid
Inputs: EU, EI, $\alpha \in \mathbf{R}^+$, $k \in N$,
$Y_0 \leftarrow I_k$; $b_0 \leftarrow O_k$
1: **for** $t = 1$ **to** T**do**
2: Select a user $u \in U$ with his context $x_t \in EU$

$$\beta \leftarrow A_0^{-1} b_0$$

3: **for** all $a \in A$ **do**
4: **if** a is a new arm **then**
5: $Y_a \leftarrow I_d$; $B_a \leftarrow O_{d\times k}$; $b_a = O_{d\times 1}$
6: **end if**
7: Select a'representation $v_{t,a}$, $v_a \in EI$
8: $x_{t,a} \leftarrow u_t || v_a$
9: $\theta_a \leftarrow A_a^{-1}(b_a - B_a\beta)$;
10: $s_{t,a} \leftarrow x_{t,a}^T A_0^{-1} x_{t,a} - 2x_{t,a}^T A_0^{-1} B_a^T A_a^{-1} v_{t,a} + v_{t,a}^T A_a^{-1} v_{t,a} + v_{t,a}^T A_a^{-1} B_a^T A_a^{-1} v_{t,a}$
11: $p_{t,a} \leftarrow x_{t,a}^T\beta + v_{t,a}^T\theta_a + \alpha\sqrt{s_{t,a}}$
12: **end for**
13: Choose the arm $a_t = \arg\max_{a \in A_t} p_{t,a}$ and observe the pay-off r_t
14: **if** $r_t = 1$ **then**
15: Select the k arms $S_k = \{s_1, s_2, \ldots s_k\}$ with their context that are most similar to a_t.
16: $S_{k+1} = \{s_1, s_2, \ldots, s_k, s_{k+1}\} = S_k \cup a_t$
17: **for** $j = 1$ **to** $k + 1$ **do**
18: $A_0 \leftarrow A_0 + B_{a_t}^T A_{a_t}^{-1} B_{a_t}$; $b_0 \leftarrow b_0 + B_{a_t}^T A_{a_t}^{-1} b_{a_t}$
19: $A_{a_t} \leftarrow A_{a_t} + v_{t,a_t} v_{t,a_t}^T$; $B_{a_t} \leftarrow B_{a_t} + v_{t,a_t} x_{t,a_t}^T$; $b_{a_t} \leftarrow b_{a_t} + r_t v_{t,a_t}$
20: $A_0 \leftarrow A_0 + x_{t,a_t} x_{t,a_t}^T - B_{a_t}^T A_{a_t}^{-1} B_{a_t}$; $b_0 \leftarrow b_0 + r_t x_{t,a_t} - B_{a_t}^T A_{a_t}^{-1} b_{a_t}$
21: **end for**
22: **end if**
23: **end for**

Similar to Algorithm 1, we use contextual representations obtained from knowledge graph embedding in steps 2 and 7. Steps 13 to 23 show the main implementation of the proposed method. We also propose Ke-SW-LinUCB combining our method with SW-LinUcb. This is realized by changing step 9 of Algorithm 1 as follows:

$$p_{t,a} \leftarrow (1 - \frac{Occ_w(a,t)}{w})\hat{\theta}_a^T x_{t,a} + \alpha\sqrt{x_{t,a}^T Y_a^{-1} x_{t,a}} \tag{8}$$

where w is the size of the sliding window and $Occ_w(a,t)$ is the number of times an item a has been pulled since the last w trial. As in Algorithm 1, we use the notation Ke-SW-LinUCB-m.

5. Experiments

In this section, we evaluate our method on two real-world datasets. We aim to answer the following questions:

RQ1: How does the improvement of knowledge representation affect the accuracy of recommendation in CMAB?

Table 1
Original datasets.

		Amazon-book	Yelp2018
User–Item interaction	#Uers	70,679	45,919
	#Items	24,915	45,538
	#Interactions	847,733	1,185,068
Knowledge Graph	#Entities	88,572	90,961
	#Relations	39	42
	#Triplets	2,557,746	1,853,704

RQ2: How does the one choice-multi change strategy improve the performance of CMAB algorithms?

RQ3: How are the hyperparameters of KE-LinUCB tuned to optimize model effectiveness?

5.1. Dataset description

To verify the effectiveness of our model, we use two datasets used by KGAT.[1] The benchmark datasets are Amazon-book and Yelp2018. Table 1 shows the original datasets used to obtain the knowledge representation of users and items.

Amazon-book[2]: Amazon product data are widely used for the evaluation of recommender systems. To evaluate the KE-LinUCB algorithm, we randomly select 500 items and 50 users from this database.

Yelp2018[3]: This is the dataset from the 2018 edition of the Yelp challenge. Again, we randomly select 500 items and 50 users for the evaluation of different baselines and our KE-LinUCB algorithm.

We randomly split the user–item interaction data into two halves, build user and item embeddings using the one half and knowledge graph, and then test the performance of the proposed model on the other half of the interaction data.

5.2. Experimental settings

5.2.1. Evaluation metrics

We use two metrics for the experiment: global accuracy and diversity.

Global Accuracy:

Global accuracy is a widely used accuracy evaluation metric for the evaluation of the bandit algorithm. It is defined as follows:

$$acc(T) = \frac{g(t)}{T},$$

where T is the time horizon, $g(t) = \sum_T r_t$, r_t is the feedback of the algorithm and $r_t \in \{0, 1\}$.

Diversity:

Diversity is an important metric for evaluating the performance of an RS. We adopt the diversity metric [42] for the experiment. For k fixed arms, diversity can be calculated as follows:

$$div(N) = 1 - \frac{c_v(N)}{\sqrt{k}},$$

where $N = \{n_{a_1}, n_{a_2}, \ldots, n_{a_k}\}$, n_{a_j} is the number of times arm a_j has been pulled thus far, $c_v(N) = \frac{\sigma(N)}{\bar{N}}$, $\bar{N}$ is the average number of times any arm has been selected by the algorithm, and $\sigma(N)$ is its standard deviation.

[1] https://github.com/xiangwang1223/knowledge_graph_attention_network
[2] http://jmcauley.ucsd.edu/data/amazon
[3] https://www.yelp.com/dataset/challenge

5.2.2. Baselines

Random: The random strategy randomly chooses an arm at a time. This strategy is the simplest and does not require parameters. Evaluating the accuracy of this strategy on a dataset enables us to evaluate the diversity of the dataset.

$\varepsilon-$greedy: This strategy randomly selects an arm with predefined probability ε and selects the arm with the highest cumulative pay-off estimate with probability $1 - \varepsilon$. When ε becomes large, it works like a random search, and when ε becomes small, the Mathew effect appears in item selection, which reduces diversity.

Thompson Sampling (TS) [48]: TS is one of the earliest heuristic methods for MAB problems. This is a bandit algorithm using the beta distribution, which has two parameters. Beta distribution is especially useful for Bernoulli rewards. In each step, TS selects an arm according to its posterior probability of having the best parameter.

LinUCB [44]: This is a well-known contextual bandit algorithm that sequentially chooses items to serve users based on the contextual information of users and items. UCB consists of two parts. The first part is the exploitation part, which works like CF, and the second part is the exploration part, which is a random selection.

SW-LinUCB [42]: This is a CMA algorithm that improves the original LinUCB by combining a diversification mechanism. This algorithm runs like the original LinUCB but penalizes arms, which are selected too frequently so as to select arms fairly among optimum. SW-LinUCB offers a trade-off between global and individual accuracy.

5.2.3. Parameter settings

For the KGAT, we use the source code. We set the entity and relation embedding size to 64. For the knowledge graph and CF, we set the coefficient of L_2 parameters to 10^{-5} and set the learning rate to 10^{-4}. For our Ke-LinUCB, we searched for α in (0.05, 0.1, 0.2, $\cdots$, 0.9, 1). In addition, since a recommender system based on bandit is a dynamic system, the accuracy of the initial stages is important in the evaluation of system performance. If the accuracy of the initial stage is low, users are more likely to have an unpleasant experience. Therefore, we set the time parameter t to (10, 20, 30) and evaluate the accuracy and diversity. To choose the best one choice-multi change strategy, we tune the parameter k in (1, 2, 3, 4, 5).

5.3. Results and analysis

5.3.1. Evaluation of the effectiveness of knowledge representation (RQ1)

To evaluate the effect of knowledge representation on the CMAB algorithm, we classify the baseline and proposed models into three classes. The first class (FC) contains the models that use the rating matrix as contextual information of CMAB. The second class (SC) has the models that use only the knowledge representation of items as contextual information of CMAB. The third class (TC) consists of models that use both user and item knowledge representations. Tables 2 and 3 show the accuracy and diversity according to knowledge representation. For the proposed models, t is 30. In the Amazon dataset, the accuracy of FC and SC of all models hardly increases. On the other hand, in all TC models, the accuracy is stably increased. In the Yelp2018 dataset, the accuracy of FC and SC of the models is relatively increased. However, the accuracy of the TC model of the two databases is much higher than the accuracy of the FC models (Fig. 4).

We can observe the following two facts. First, the performance of CMAB highly depends on its contextual representation. For example, in the Amazon dataset, the accuracy of Ke-LinUCB (TC)

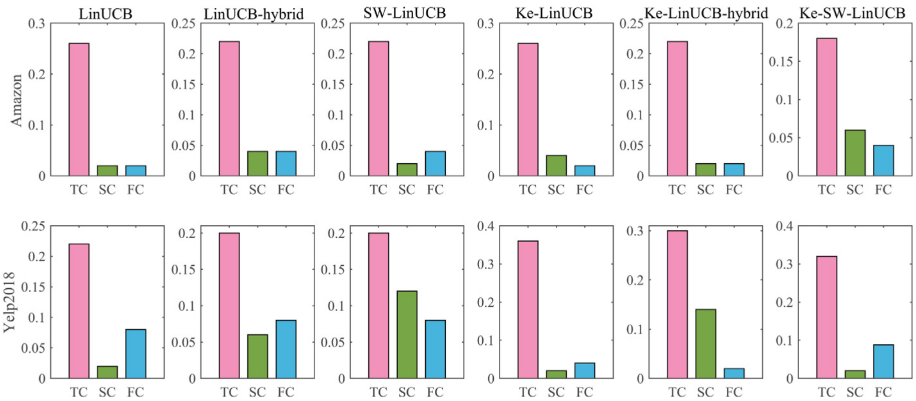

Fig. 4. Accuracy comparison according to knowledge representation in both datasets.

Table 2
Global accuracy and diversity according to the representation (Amazon).

Model	Time step					
	10		20		30	
	acc	div	acc	div	acc	div
LinUCB (TC)	**0.1800**	**0.4595**	**0.2400**	**0.5581**	**0.2600**	**0.5990**
LinUCB (SC)	0.0200	0.5725	0.0200	0.5846	0.0200	0.6707
LinUCB (FC)	0.0400	0.7160	0.0200	0.7587	0.0200	0.7866
LinUCB-hybrid (TC)	**0.1000**	**0.5673**	**0.1800**	**0.6417**	**0.2200**	**0.6624**
LinUCB-hybrid (SC)	0.0200	0.5539	0.0200	0.5725	0.0400	0.7244
LinUCB-hybrid (FC)	0.0200	0.6059	0.0200	0.6954	0.0400	0.7244
SW-LinUCB (TC)	**0.1400**	**0.6165**	**0.2000**	**0.6594**	**0.2200**	**0.6699**
SW-LinUCB (SC)	0.0200	0.5296	0.0400	0.5929	0.0200	0.5514
SW-LinUCB (FC)	0.0600	0.6271	0.0200	0.6059	0.0400	0.6766
Ke-LinUCB (TC)	**0.1400**	**0.5060**	**0.2200**	**0.6058**	**0.2600**	**0.6640**
Ke-LinUCB (SC)	0.0200	0.4717	0.0400	0.5465	0.0400	0.5440
Ke-LinUCB (FC)	0.0400	0.6104	0.0200	0.6848	0.0200	0.7335
Ke-LinUCB–hybrid (TC)	**0.0800**	**0.5010**	**0.1600**	**0.6348**	**0.2200**	**0.6693**
Ke-LinUCB–hybrid (SC)	0.0200	0.6328	0.0200	0.6821	0.0200	0.6821
Ke-LinUCB–hybrid (FC)	0.0200	0.6550	0.0200	0.7387	0.0200	0.6921
Ke-SW-LinUCB (TC)	**0.1200**	**0.6172**	**0.1800**	**0.6459**	**0.1800**	**0.6467**
Ke-SW-LinUCB (SC)	0.0200	0.6737	0.0200	0.7282	0.0600	0.7354
Ke-SW-LinUCB (FC)	0.0200	0.6681	0.0200	0.7608	0.0400	0.7442

Table 3
Global accuracy and diversity according to the representation (Yelp2018).

Model	Time step					
	10		20		30	
	acc	div	acc	div	acc	div
LinUCB (TC)	**0.1600**	**0.5719**	**0.2200**	**0.6061**	**0.2200**	**0.6176**
LinUCB (SC)	0.0200	0.6233	0.0000	0.6853	0.0200	0.7189
LinUCB (FC)	0.0200	0.6681	0.0800	0.4102	0.0800	0.2210
LinUCB-hybrid (TC)	**0.1000**	**0.2430**	**0.1000**	**0.2734**	**0.2000**	**0.4134**
LinUCB-hybrid (SC)	0.0200	0.6914	0.0400	0.5823	0.0600	0.6752
LinUCB-hybrid (FC)	0.1200	0.4965	0.0200	0.5272	0.0800	0.6752
SW-LinUCB (TC)	**0.1400**	**0.5093**	**0.2000**	**0.5700**	**0.2000**	**0.5998**
SW-LinUCB (SC)	0.0200	0.6914	0.0400	0.6452	0.1200	0.5507
SW-LinUCB (FC)	0.0600	0.7495	0.1000	0.4877	0.0800	0.4118
Ke-LinUCB (TC)	**0.1600**	**0.6085**	**0.2400**	**0.6784**	**0.3600**	**0.7262**
Ke-LinUCB (SC)	0.0000	0.2936	0.0000	0.4717	0.0200	0.3718
Ke-LinUCB (FC)	0.0400	0.6681	0.0000	0.7283	0.0400	0.7294
Ke-LinUCB–hybrid (TC)	**0.1200**	**0.5623**	**0.2000**	**0.6356**	**0.3000**	**0.6820**
Ke-LinUCB–hybrid (SC)	0.0800	0.3526	0.1400	0.0865	0.1400	0.0457
Ke-LinUCB–hybrid (FC)	0.0400	0.6271	0.0400	0.6339	0.0200	0.7160
Ke-SW-LinUCB (TC)	**0.1800**	**0.5463**	**0.2600**	**0.5990**	**0.3200**	**0.6293**
Ke-SW-LinUCB (SC)	0.0000	0.5929	0.0000	0.5961	0.0200	0.6328
Ke-SW-LinUCB (FC)	0.0200	0.4918	0.0200	0.4137	0.0800	0.5465

compared to Ke-LinUCB (FC) is increased by 13 times, and in the Yelp2018 dataset, the accuracy of Ke-LinUCB (TC) compared to Ke- LinUCB (FC) is increased by 15 times. The FC model uses a rating matrix to represent the context, and since the rating matrix is sparse, it leads to a lack of knowledge of context representation. The SC models use only the knowledge representation of items as contextual information, and there is no significant difference in performance between the FC model and the SC model. This means that the knowledge representation of users obtained from the knowledge graph greatly affects the performance of CMAB. The result shows that knowledge acquisition by a knowledge graph is an effective method for the contextual expression of CMAB. Second, the accuracy of the models shows different characteristics depending on the dataset. The ratio of accuracy between the TC and FC models in the Amazon dataset is relatively larger than that of Yelp2018. This means that the CMAB's accuracy improvement by knowledge enhancement is affected by the dataset.

5.3.2. Evaluation of the one choice–multi change strategy (RQ2)

Tables 4 and 5 show the performance of all the methods over time. We highlight the proposed methods in bold. In the two datasets, LinUCB and SW-LinUCB have similar accuracies. SW-LinUCB and LinUCB have much higher accuracy than random, ε-greedy, and TS for both datasets. Random has the highest diversity for both datasets, but the accuracy is very low. Since these two datasets are actual purchase datasets, the user's choice itself has a certain diversity.

Table 4
Global accuracy and diversity on the Amazon dataset, **: best, *: good.

Model	Time step					
	10		20		30	
	acc	div	acc	div	acc	div
random	0.0000	0.6478	0.0200	0.6853	0.0200	0.7520
epsilon	0.0000	0.0005	0.0200	0.5114	0.0400	0.5130
tompson	0.0000	0.5010	0.0200	0.7244	0.0200	0.7595
LinUCB	**0.1800****	**0.4595**	**0.2400****	**0.5581**	0.2600	0.5990
LinUCB-hybrid	0.1000	0.5673	0.1800	0.6417	0.2200	0.6624
SW-LinUCB	**0.1400***	**0.6165**	0.2000	0.6594	0.2200	0.6699
Ke-LinUCB-10	**0.1400***	**0.5060**	0.2000	0.5902	**0.2800***	**0.6374**
Ke-LinUCB-20	**0.1400***	**0.5060**	**0.2200***	**0.6058**	**0.3000****	**0.6709**
Ke-LinUCB-30	**0.1400***	**0.5060**	**0.2200***	**0.6058**	0.2600	0.6640
Ke-SW-LinUCB-10	0.0800	0.3586	0.2000	0.5239	0.2400	0.6104
Ke-SW-LinUCB-20	0.0800	0.5267	0.1000	0.5586	0.1800	0.6145
Ke-SW-LinUCB-30	0.1200	0.6172	0.1800	0.6459	0.1800	0.6467

In the Amazon dataset (Table 4), the proposed methods have lower performance than LinUCB up to 20 epochs but show high accuracy after 20 epochs. SW-LinUCB has relatively increased diversity compared to LinUCB; however, its accuracy is lower than that of the proposed methods. The proposed methods have a high diversity in the total epochs compared to the baseline methods. In the Yelp2018 dataset (Table 5), the proposed models have higher accuracy and diversity than the baseline models in the total epochs. This shows the effectiveness of the one choice -

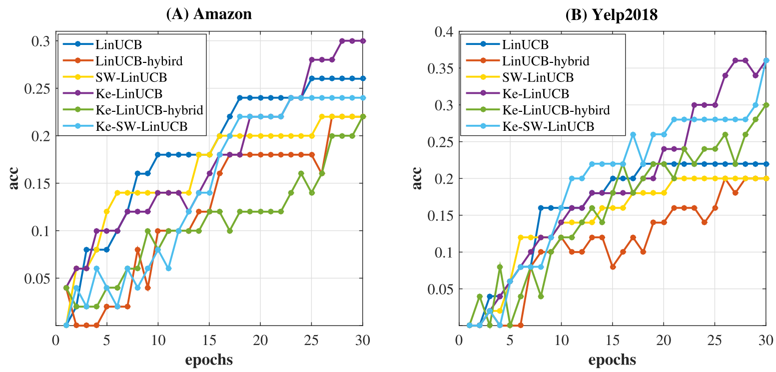

Fig. 5. Comparisons of our proposed models and baseline models.

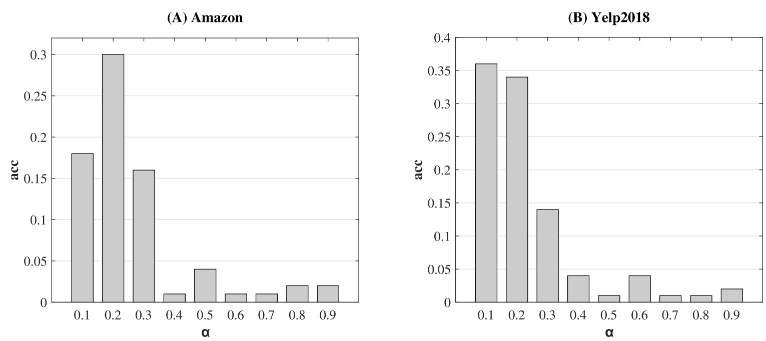

Fig. 6. Global accuracy according to α in two datasets.

Table 5
Global accuracy and diversity on the Yelp2018 dataset, **: best, *: good.

Model	Time step					
	10		20		30	
	acc	div	acc	div	acc	div
random	0.0400	0.6233	0.0000	0.6853	0.0000	0.7520
epsilon	0.0400	0.0005	0.0400	0.0005	0.0400	0.0188
tompson	0.0200	0.5539	0.0400	0.6737	0.0200	0.7367
LinUCB	0.1600	0.5719	0.2200	0.6061	0.2200	0.6176
LinUCB-hybrid	0.1000	0.2430	0.1000	0.2734	0.2000	0.4134
SW-LinUCB	0.1400	0.5093	0.2000	0.5700	0.2000	0.5998
Ke-LinUCB-10	0.1600	0.5611	0.2400	0.6274	0.2600	0.6501
Ke-LinUCB-20	0.1600	0.5611	0.2200	0.6129	**0.3400***	**0.6707**
Ke-LinUCB-30	0.1600	0.6085	0.2400	0.6784	**0.3600****	**0.7262**
Ke-SW-LinUCB-10	**0.2000****	**0.6288**	**0.2600***	**0.6638**	0.3000	0.6743
Ke-SW-LinUCB-20	**0.2000****	**0.6288**	**0.2800****	**0.6915**	**0.3600****	**0.7015**
Ke-SW-LinUCB-30	**0.1800***	**0.5463**	**0.2600***	**0.5990**	0.3200	0.6293

Table 6
Global accuracy according to k of Ke-LinUCB-20 on the Amazon dataset, **: best, *: good.

	Time step					
	10		20		30	
k	acc	div	acc	div	acc	div
1	**0.1200***	**0.4424**	**0.2000***	**0.5664**	**0.2200***	**0.6175**
2	**0.1400****	**0.5060**	**0.2200****	**0.6058**	**0.3000****	**0.6709**
3	0.1000	0.4772	0.1800	0.5472	**0.2200***	**0.6158**
4	**0.1200***	**0.5480**	0.1800	0.6254	**0.2200***	**0.6677**
5	**0.1200***	**0.5377**	0.1600	0.6265	0.2000	0.6571

multi change strategy. The time length of the one choice - multi change strategy affects the performance of this algorithm. In two datasets, this strategy shows the best performance when the time length is 20. This result shows that the one choice - multi change strategy is effective in addressing the cold start problem. The balance of accuracy–diversity in off-line testing mainly depends on the dataset. Overall, the knowledge enhancement models show stable higher accuracy and diversity than the baseline models in both datasets.

Fig. 5 shows the dynamic behavior of our proposed models and baseline models. In the Amazon dataset, the accuracy of Ke-LinUCB is relatively higher than that of the other models from the 20th epoch. In the Yelp 2018 database, the accuracies of Ke-LinUCB, Ke-LinUCB-hybrid and Ke-SW-LinUCB after the 10th epoch are much higher than those of the baseline models. This shows that one choice-multi change strategy is effective in dynamic recommendation.

5.3.3. Effect of hyperparameters in KE-LinUCB (RQ3)

We conducted experiments to evaluate the effectiveness of the hyperparameters in our proposed method. Fig. 6 shows the global accuracy of Ke-LinUCB-20 according to α. The value of α for the maximum accuracy exists within a certain range. For example, for the Amazon dataset and the Yelp2018 dataset, Ke-LinUCB-20 has the highest accuracy in [0.1, 0.2]. In addition, in the Yelp2018 dataset, the performance change because α is not large compared to the Amazon dataset.

Tables 6 and 7 show the global accuracy of the proposed model according to the different numbers of k. We recall that k is the number of items that change simultaneously in the one choice - multi change strategy. This result explains the following facts: (1) By adjusting the value of k in the initial stage, the recommendation accuracy can be increased. For instance, our proposed model has the highest accuracy when k = 2 in the Amazon dataset, whereas it has the highest accuracy when k = 4 in the Yelp2018 dataset. (2) The accuracy–diversity trade-off according to the value of k has different characteristics depending on the datasets. This difference in parameters can be seen to be related to the characteristics of the datasets.

Table 7
Global accuracy according to k of Ke-LinUCB-20 on the Yelp2018 dataset, **: best, *: good.

	Time step					
	10		20		30	
k	acc	div	acc	div	acc	div
1	**0.1400***	**0.4911**	**0.2000***	**0.5814**	0.2200	0.6145
2	0.0600	0.4181	0.0800	0.4689	0.1400	0.5585
3	0.1200	0.4077	0.1400	0.5223	0.2600	0.6158
4	**0.1600****	**0.5611**	**0.2200****	**0.6129**	**0.3400****	**0.6707**
5	0.1200	0.5550	0.1400	0.5901	**0.3200***	**0.6631**

6. Conclusion and future work

In this paper, we have proposed a novel knowledge-enhanced contextual bandit approach for personalized recommendation. Based on the assumption that the attributes that a user likes are repeated, we develop the one choice - multi change strategy for the contextual bandit algorithm. Extensive experiments on two datasets show the effectiveness of our approach.

Through this study, we can draw the following conclusions:

(1) In the CMAB algorithm, the recommendation performance highly depends on the representation of contextual information. When the accuracy of the contextual representation of the item and user increases, the recommendation accuracy is improved. This shows that the study of the precise representation of contextual information is one of the important research directions of the CMAB-based recommender system. (2) The one choice - multi change strategy increases the recommendation accuracy and diversity. We argue that while it is difficult to predict what a user will choose next, the attributes that the user will choose next are predictable. In KG, each item is represented by attributes and thus the expressions of items with similar attributes are similar. Especially in the early stage of recommendation, this strategy is effective. (3) The values of parameters for the best performance of the CMAB algorithm reflect the characteristics of the datasets. In particular, the values of α and k reflect the characteristics of the system that created the dataset. For example, the accuracy of the random method is an indicator of the diversity of the dataset, and the α and k of CMAB are also important indicators of the diversity. This work uses offline datasets to simulate an online recommendation process, and the experimental results imply some unique properties of every dataset. Our results show that CMAB can be a measure of the system that created the database.

If we can understand the behavior of the system that created it through the dataset, it will help build a more accurate and diversified recommender system. As future work, we will continue to investigate the behavior of the dynamic recommendation process using external knowledge such as KG. The study will not only provide a more accurate understanding of the recommender system that has created the current benchmark dataset but will also serve as an important milestone in making more accurate and diverse recommendations.

CRediT authorship contribution statement

Mingxin Gan: Resources, Data curation, Writing – review & editing, Supervision, Funding acquisition. **O-Chol Kwon:** Conceptualization, Methodology, Software, Validation, Investigation, Writing – original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study is supported by the National Natural Science Foundation of China (71871019, 71471016).

References

[1] M.-H. Chen, C.-H. Teng, P.-C. Chang, Applying artificial immune systems to collaborative filtering for movie recommendation, Adv. Eng. Inf. 29 (4) (2015) 830–839.
[2] Q. Diao, M. Qiu, C.-Y. Wu, A.J. Smola, J. Jiang, C. Wang, Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS), in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2014, pp. 193–202.
[3] E. Liebman, M. Saar-Tsechansky, P. Stone, The right music at the right time: Adaptive personalized playlists based on sequence modeling, Mis Q. 43 (3) (2019) 765–786.
[4] M. Schedl, D. Hauger, Tailoring music recommendations to users by considering diversity, mainstreaminess, and novelty, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2015, pp. 947–950.
[5] J. Turcotte, C. York, J. Irving, R.M. Scholl, R.J. Pingree, News recommendations from social media opinion leaders: effects on media trust and information seeking, J. Comput. Mediat. Commun. 20 (5) (2015) 520–535.
[6] F. Hopfgartner, B. Kille, A. Lommatzsch, T. Plumbaum, T. Brodt, T. Heintz, Benchmarking news recommendations in a living lab, in: International Conference of the Cross-Language Evaluation Forum for European Languages. Vol. 213, 2014, pp. 250–267.
[7] L.W. Huang, M.S. Fu, F. Li, H. Qu, Y.J. Liu, W.Y. Chen, A deep reinforcement learning based long-term recommender system, Knowl. Based Syst. 213 (2021) 106706.
[8] F. Liu, R.M. Tang, X.T. LI, W.N. Zhang, Y.M. Ye, H.K. Chen, H.F. Chen, H.F. Guo, Y.Z. Zahng, X.Q. He, State representation modeling for deep reinforcement learning based recommendation, Knowl. Based Syst. 205 (2020) 106170.
[9] Y. Lin, B. Xu, J. Feng, H. Lin, K. Xu, Knowledge-enhanced recommendation using item embedding and path attention, Knowl. Based Syst. 233 (2021) 107484.
[10] K.H. Song, X.Y. Zeng, Y. Zheng, J. De Jonckheere, X.J. Yuan, L. Koehl, An interpretable knowledge-based decision support system and its applications in pregnancy diagnosis, Knowl. Based Syst. 221 (2021) 106835.
[11] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, IEEE Trans. Knowl. Data Eng. 17 (6) (2005) 734–749.
[12] Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, Q. He, A survey on knowledge graph-based recommender systems, IEEE Trans. Knowl. Data Eng. (2020) 1.
[13] D. Lee, K. Hosanagar, How do recommender systems affect sales diversity? A cross-category investigation via randomized field experiment, Inf. Syst. Res. 30 (1) (2019) 239–259.
[14] R. Chen, Q.Y. Hua, Y.S. Chang, B. Wang, L. Zhang, X.J. Kong, A survey of collaborative filtering-based recommender systems: From traditional methods to hybrid methods based on social networks, IEEE Access 6 (2018) 64301–64320.
[15] S. Rendle, Factorization machines with libFM, ACM Trans. Intell. Syst. Technol. 3 (3) (2012) 1–22.
[16] J.-D. Zhang, C.-Y. Chow, J. Xu, Enabling kernel-based attribute-aware matrix factorization for rating prediction, IEEE Trans. Knowl. Data Eng. 29 (4) (2017) 798–812.
[17] Q. Zhao, J.L. Chen, M.M. Chen, S. Jain, A. Beutel, F. Belletti, E.H. Chi, Categorical-attributes-based item classification for recommender systems, in: 12th ACM Conference on Recommender Systems, RecSys, 2018, pp. 320–328.
[18] X. Liu, W. Wu, Learning context-aware latent representations for context-aware collaborative filtering, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2015, pp. 887–890.
[19] L.B. Wu, C. Quan, C.L. Li, Q. Wang, B.L. Zheng, X.Y. Luo, A context-aware user-item representation learning for item recommendation, ACM Trans. Inf. Syst. 37 (2) (2019) 1–29.
[20] S. Wu, Q. Liu, L. Wang, T.N. Tan, Contextual operation for recommender systems, IEEE Trans. Knowl. Data Eng. 28 (8) (2016) 2000–2012.
[21] Y.R. Jin, S.B. Dong, Y. Cai, J.L. Hu, RACRec: Review aware cross-domain recommendation for fully-cold-start user, IEEE Access (8) (2020) 55032–55041.
[22] S.M. Mudambi, D. Schuff, What makes a helpful online review? A study of customer reviews on amazon.com, Mis Q. 34 (1) (2010) 185–200.
[23] F. Ebrahimi, S.A. H. Golpayegani, Personalized recommender system based on social relations, in: 2016 24th Iranian Conference on Electrical Engineering, ICEE, 2016, pp. 218–223.

[24] D. Margaris, D. Spiliotopoulos, C. Vassilakis, Social relations versus near neighbours: Reliable recommenders in limited information social network collaborative filtering for online advertising, in: Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 2019, pp. 1160–1167.

[25] W. Reafee, N. Salim, A. Khan, The power of implicit social relation in rating prediction of social recommender systems, PloS One 11 (5) (2016) e0154848.

[26] M.X. Gan, R. Jiang, Improving accuracy and diversity of personalized recommendation through power law adjustments of user similarities, Decis. Support Syst. 55 (3) (2013) 811–821.

[27] A. Javari, M. Jalili, A probabilistic model to resolve diversity-accuracy challenge of recommendation systems, Knowl. Inf. Syst. 44 (1) (2015) 609–627.

[28] T. Zhou, Z. Kuscsik, J.G. Liu, M. Medo, J.R. Wakeling, Y.C. Zhang, Solving the apparent diversity-accuracy dilemma of recommender systems, Proc. Natl. Acad. Sci. United States Am. 107 (10) (2010) 4511–4515.

[29] H. Wang, Z.H. Wang, W.S. Zhang, Quantitative analysis of matthew effect and sparsity problem of recommender systems, in: 2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis, Icccbda, 2018, pp. 78–82.

[30] Q. Wang, Z.D. Mao, B. Wang, L. Guo, Knowledge graph embedding: A survey of approaches and applications, IEEE Trans. Knowl. Data Eng. 29 (12) (2017) 2724–2743.

[31] F.Z. Zhang, N.J. Yuan, D.F. Lian, X. Xie, W.Y. Ma, Collaborative knowledge base embedding for recommender systems, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 353–362.

[32] S. Li, A. Karatzoglou, C. Gentile, Collaborative filtering bandits, in: Proceedings of the 39th International ACM SIGKDD Conference on Research and Development in Information Retrieval, 2016, pp. 539–548.

[33] J. Bobadilla, F. Ortega, A. Hernando, Á. Arroyo, A balanced memory-based collaborative filtering similarity measure, Int. J. Intell. Syst. 27 (10) (2012) 939–946.

[34] B. Jeong, J. Lee, H. Cho, Improving memory-based collaborative filtering via similarity updating and prediction modulation, Inf. Sci. 180 (5) (2010) 602–612.

[35] K. Yu, A. Schwaighofer, V. Tresp, X.W. Xu, H.P. Kriegel, Probabilistic memory-based collaborative filtering, IEEE Trans. Knowl. Data Eng. 16 (1) (2004) 56–69.

[36] K. Goldberg, T. Roeder, D. Gupta, C. Perkins, Eigentaste: A constant time collaborative filtering algorithm, Inf. Retr. 4 (2) (2001) 133–151.

[37] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (8) (2009) 30–37.

[38] B. Marlin, Modeling user rating profiles for collaborative filtering, Adv. Neural Inf. Process. Syst. (2004) 627–634.

[39] A. Nachev, L. Ganchev, J. Rowland, A model-based approach to collaborative filtering by neural networks, in: Proceedings of the 2005 International Conference on Artificial Intelligence, Vol. 2, 2005, pp. 846–852.

[40] X. Zhou, J. He, G.Y. Huang, Y.C. Zhang, Svd-based incremental approaches for recommender systems, J. Comput. Syst. Sci. 81 (4) (2015) 717–733.

[41] S.J. Gong, H.W. Ye, H.S. Tan, Combining memory-based and model-based collaborative filtering in recommender system, in: 2009 Pacific-Asia Conference on Circuits, Communications and Systems, 2009, pp. 690–693.

[42] N. Gutowski, T. Amghar, O. Camp, F. Chhel, Global versus individual accuracy in contextual multi-armed bandit, in: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, 2019, pp. 1647–1654.

[43] N. Gutowski, T. Amghar, O. Camp, F. Chhel, Gorthaur-Exp3: Bandit-based selection from a portfolio of recommendation algorithms balancing the accuracy-diversity dilemma, Inf. Sci. 546 (2021) 378–396.

[44] L. Li, W. Chu, J. Langford, R.E. Schapire, A contextual-bandit approach to personalized news article recommendation, in: Proceedings of the 19th International Conference on World Wide Web ACM, 2010, pp. 661–670.

[45] Y.P. Wen, F.R. Wang, R. Wu, J.X. Liu, B.Q. Cao, Improving the novelty of retail commodity recommendations using multiarmed bandit and gradient boosting decision tree, Concurr. Comput. Pract. Exp. 32 (14) (2020) e5703.

[46] S. Agrawal, N. Goyal, Analysis of thompson sampling for the multi-armed bandit problem, J. Mach. Learn. Res. (2012).

[47] P. Auer, N. Cesa-Bianchi, Y. Freund, R.E. Schapire, The nonstochastic multiarmed bandit problem, Siam J. Comput. 32 (1) (2003) 48–77.

[48] S. Agrawal, N. Goya, Thompson sampling for contextual bandits with linear payoffs, in: 30th International Conference on Machine Learning, Vol. 28, 2013, pp. 127–135.

[49] P. Auer, N. Cesa-Bianchi, P. Fischer, Finite-time analysis of the multiarmed bandit problem, Mach. Learn. 47 (2–3) (2002) 235–256.

[50] O. Chapelle, L. Li, An empirical evaluation of thompson sampling, in: Proceedings of the 24th International Conference on Neural Information Processing Systems, 2011, pp. 2249–2257.

[51] H. Wang, F.Z. Zhang, X. Xie, M.Y. Guo, DKN: Deep knowledge-aware network for news recommendation, in: Proceedings of the 2018 World Wide Web Conference, 2018, pp. 1835–1844.

[52] Y.T. Ye, X.W. Wang, J.C. Yao, K.Y. Jia, J.R. Zhou, Y.H. Xiao, H.X. Yang, Bayes embedding (Bem): Refining representation by integrating knowledge graphs and behavior-specific networks, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 679–688.

[53] A. Bordes, N. Usunier, J. Garcia-Durán, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: Proceedings of the 26th International Conference on Neural Information Processing Systems, Vol. 2, 2013, pp. 2785–2795.

[54] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph embedding by translating on hyperplanes, in: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014, pp. 1112–1119.

[55] Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion, in: Proceedings of the National Conference on Artificial Intelligence, 2015, pp. 2181–2187.

[56] G. Ji, S. He, L. Xu, K. Liu, J. Zhao, Knowledge graph embedding via dynamic mapping matrix, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Vol. 1, 2015, pp. 687–696.

[57] B. Yang, W.-T. Yih, X. He, J. Gao, L. Deng, Embedding entities and relations for learning and inference in knowledge bases, in: 3rd International Conference on Learning Representations, 2015.

[58] Y. Sun, J. Han, X. Yan, P.S. Yu, T. Wu, Pathsim: Metapath-based top-K similarity search in heterogeneous information networks, Proc. VLDB Endow. 4 (11) (2011) 992–1003.

[59] X. Yu, X. Ren, Q. Gu, Y. Sun, J. Han, Collaborative filtering with entity similarity regularization in heterogeneous information networks, in: Proc. of IJCAI-13 HINA Workshop, 2013.

[60] H. Zhao, Q.M. Yao, J.D. Li, Y.Q. Song, D.L. Lee, Meta-graph based recommendation fusion over heterogeneous information networks, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 635–644.

[61] B.B. Hu, C. Shi, W.X. Zhao, P.S. Yu, Leveraging meta-path based context for top-n recommendation with a neural co-attention model, in: Proceedings of the 24rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018, pp. 1531–1540.

[62] Z. Sun, J. Yang, J. Zhang, A. Bozzon, L.K. Huanga, C. Xu, Recurrent knowledge graph embedding for effective recommendation, in: Proceedings of the 12th ACM Conference on Recommender Systems, 2018, pp. 297–305.

[63] X. Wang, D.X. Wang, C.R. Xu, X.N. He, Y.X. Cao, T.S. Chua, Explainable reasoning over knowledge graphs for recommendation, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, (1) 2019, pp. 5329–5336.

[64] H.W. Wang, F.Z. Zhang, J.L. Wang, M. Zhao, W.J. Li, X. Xie, M.Y. Guo, Ripplenet: Propagating user preferences on the knowledge graph for recommender systems, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018, pp. 417–426.

[65] Y.R. Qu, T. Bai, W.N. Zhang, J.Y. Nie, J. Tang, An end-to-end neighborhood-based interaction model for knowledge-enhanced recommendation, in: Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data, (8) 2019, pp. 1–9.

[66] X.L. Tang, T.Y. Wang, H.Z. Yang, H.J. Song, Akupm: Attention-enhanced knowledge-aware user preference model for recommendation, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019, pp. 1891–1899.

[67] H.W. Wang, M. Zhao, X. Xie, W.J. Li, M.Y. Guo, Knowledge graph convolutional networks for recommender systems, in: Proceedings of the World Wide Web Conference, 2019, pp. 3307–3313.

[68] X. Wang, X.N. He, Y.X. Cao, M. Liu, T.S. Chua, KGAT: Knowledge graph attention network for recommendation, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019, pp. 950–958.

[69] J. Langford, T. Zhang, The epoch-greedy algorithm for multi-armed bandits with side information, in: Advances in Neural Information Processing Systems, Vol. 20, Curran Associates, Inc. 2008, pp. 817–824.

[70] J.Z. Qiu, J. Tang, H. Ma, Y.X. Dong, K.S. Wang, J. Tang, Deepinf: Social influence prediction with deep learning, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018, pp. 2110–2119.

[71] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: 2017 5th International Conference on Learning Representations, 2017.

[72] W.L. Hamilton, R. Ying, J. Leskovec, Inductive representation learning on large graphs, in: Advances in Neural Information Processing Systems, 2017, pp. 1025–1035.

[73] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, BPR: Bayesian personalized ranking from implicit feedback, in: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, 2009, pp. 452–461.