Information Sciences 479 (2019) 231–249

Contents lists available at ScienceDirect

Information Sciences

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

Learning peer recommendation using attention-driven CNN with interaction tripartite graph

Qintai Hu[a,b,*], Zhongmei Han[a,b], Xiaofan Lin[a,b], Qionghao Huang[a], Xiaomei Zhang[a]

[a] School of Information Technology in Education, South China Normal University, Guangzhou, China
[b] Guangdong Engineering Research Center for Smart Learning, South China Normal University, Guangzhou, China

ARTICLE INFO

Article history:
Received 13 September 2018
Revised 22 November 2018
Accepted 4 December 2018
Available online 5 December 2018

Keywords:
Interaction behaviours
Dynamic interaction tripartite graph
Attention-driven CNN
Multi-objective learning peer recommendation

ABSTRACT

Learning peer recommendation (LPR) is one of the effective solutions to overcome the information load of learners. This paper presents a multi-objective LPR framework for online learning. Using a dynamic interaction tripartite graph (DITG), we characterize and model the complex relationships among learners, learning content, and interaction behaviours, followed by capturing the dynamic interactions among learners with an attention-driven convolution neural network (CNN). The proposed attention-driven CNN is leveraged to tune the weights of interaction behaviours according to the features of the learning content. A multi-objective function composed of three conflicting metrics, interaction intensity, diversity and novelty, is optimized to achieve simultaneous multiple recommendations for a group of learners. Compared to the state-of-the-art approaches, the proposed LPR framework and algorithms perform favourably.

1. Introduction

With the rapid development of online learning and social networks, online learners not only interact with each other at any time and from anywhere, but also make full use of learning from their peers' wisdom [17,29]. Confronted by a large body of information, sometimes learners can feel lonely and overwhelmed [6]. However, a learner's needs can be identified by analyzing a large amount of available information, such as grades, learning content and interactions among learners [35]. Therefore, personalized recommendation is becoming increasingly important to alleviate information overload [46]. Unlike traditional recommendation systems for commercial products such as movies, different kinds of interactions among learners will significantly affect the performance of a recommendation system [40].

Traditional approaches for personalized learning recommendation mainly consider the static factors of learners. Static factors refer to learner profile attributes that remain unchanged over a given period, such as historical tags, scores and other information on learning content or learners [23]. As an example, we can use a tripartite graph to model such factors. As we know, interactions among learning peers take place frequently and usually keep changing [43]. From educational perspectives, it is insufficient to use a tripartite graph to model static factors (tags or scores) [12,32,44]. Therefore, in this paper, we propose to use a dynamic interaction tripartite graph (DITG) to model peer interactions.

* Corresponding author at: School of Information Technology in Education, South China Normal University, Guangzhou, China.
E-mail address: huqt8@scnu.edu.cn (Q. Hu).

https://doi.org/10.1016/j.ins.2018.12.003
0020-0255/

To extract deep interaction features from the collected data, conventional neural networks such as multilayer perceptions (MLPs) with the backpropagation algorithm [10,42] perform poorly for learning peer recommendation. In the context of big data, advanced learning models, such as CNN [11] and stochastic configuration networks [37], are better alternatives for capturing complex and dynamic relations in learning analysis [11,13]. It has been observed that highly accurate recommendations can be easily achieved for popular learners [38], but the recommendation diversity is undoubtedly lost. Thus, it is useful and challenging to design a learning peer recommendation system which is well balanced between multiple objectives, such as accuracy, diversity and novelty. In this paper, we propose a multi-objective framework for learning peer recommendation based on DITG and an attention-driven CNN (LPRACNN). Specifically, we first construct a DITG with manually assigned weights that reflect the complex relationships between learners and content from learning objective perspectives. Then, we devise two novel layers of a *scaler* layer and an attention-driven CNN to tune the initial weights of the DITG. After obtaining different interaction intensities among learners, we optimize the proposed system in terms of diversity, novelty and interaction intensity. From the experiment results, it can be seen that the proposed LPRACNN achieves significant improvements in recommendation performance. The following summarizes our contributions:

(1) The development of an attention-driven CNN-based framework for recommending learning peers: multi-objective learning peer recommendation to balance the interaction intensity, novelty and diversity of recommendation lists.

(2) The use of a dynamic interaction tripartite graph (DITG) for peer interactions: considering the dynamic nature of the online learning process, we use DITG to integrate dynamic interaction behaviours into a tripartite graph with dynamic weights in a real-time manner.

(3) Learning the parameters in our model by both manual settings in DITG and automatic adjustment in an attention-driven network: first, the initial weights in DITG are set manually according to the education design principle. Second, we design an attention-driven network to adjust these weights through a normal training process. This differs from the common initial random weights in that it saves training time and takes pedagogies into account.

The remainder of this paper is organized as follows. Section 2 briefly reviews the related work. Section 3 formulates the problem of learning peer recommendation and briefly introduces our framework. Section 4 describes the proposed attention-driven CNN model for multi-objective learning peer recommendation. Section 5 reports our experiment results with comparisons, and Section 6 concludes this paper with suggestions for future studies.

2. Related work

This section briefly reviews some related work, including learning recommendation systems, graph-based recommendation, and deep learning in recommendation.

2.1. Learning recommendation systems

Learning recommendation systems aim to help learners find relevant and suitable information for their needs [36]. There has been a notable upsurge in the use of recommendation techniques in online learning [15]. Roughly, recommendation techniques are classified into the three categories: content-based filtering (CBF), collaborative filtering (CF) and hybrid filtering [33].

In early recommendation systems, CBF was widely used to make recommendations to a target learner based on what another learner with similar preferences liked in the past [25], including teaching guidance, adaptive learning and management decisions. All of these studies need to learn learning preferences for some types of learning content and apply these to other types of learning content [31]. Despite its efficiency, CBF not only suffers from the cold start problem, it is inefficient and even infeasible for large-scale online learning systems. CF is the most popular approach for recommending the favorite things of other learners with similar tastes and preferences to a target user [20]. It relies on the relationship between learners and learning content. Neighborhood models and matrix factorization have achieved some of best results [22]. However, with the explosive growth in the amount of learning data, the efficiency of the algorithms significantly declines. Moreover, these algorithms will be ineffective because of the sparsity of data. Hybrid filtering combines more than one approach, such as CF, CBF, semantics, and ontology [2]. Hybrid filtering, however, cannot change the structure easily and achieves improved results [21].

Though these classic recommendation methods have been improved and widely used in different domain applications, they are still limited when applied to large-scale e-learning systems.

2.2. Graph-based recommendation

Many scholars have proposed different methods based on network structures to enhance the effect of recommendation. Nowell et al. [14] visualized a network structure as a bipartite graph, where two sets of vertexes correspond to users and items, respectively, and an edge represents a selection relation between a user and item. With this definition of a bipartite graph, the recommendation algorithm focuses on resource allocation. In particular, it is assumed that nodes selected by users are used to recommend other information to their adjacent nodes. In this way, the resource is reallocated [3]. Furthermore, a CF algorithm is introduced to the resource allocation of a bipartite graph, which alleviates the low accuracy problem of data sparsity. Other work tries to assign different weights to different items, which forms a weighted bipartite graph [26].

Despite the huge success of bipartite graph approaches, sparsity is still a major challenge, as it adversely affects the accuracy of recommendations.

The aforementioned related works made use of bipartite graphs. The expected accuracy and diversity, however, cannot be achieved merely through information of network structures between users and items [34]. In recent years, other information of users or items is drawn into a bipartite graph to form a tripartite graph to improve the accuracy and diversity of recommendations. Tags of items or users can reflect the personalized characteristics of users or the performance of items, so the tag was taken as a factor to transfer the ternary relations of user-tag-item [44]. Additionally, a score is also considered as a rater attitude [32]. So tripartite graph methods are applied to improve the effect of user recommendation by fusing the score.

However, the majority of the existing works based on tripartite graphs depend on historical scores or tags of users, without focusing on interactions. For online learning, we should integrate frequent interaction behaviours into a tripartite graph for recommendations.

2.3. Deep learning in recommendation

A large body of literature focuses on integrating deep learning into recommender systems. As a simple model, MLP can model the non-linear interactions between users and items to enhance recommendation quality [5]. Alashkar et al. [1] propose two identical MLPs to model labeled examples and expert rules. The parameters of these two networks are updated simultaneously by mining the differences between their outputs. Although it is precise, expertise acquisition needs much human involvement. As an advanced model for feature extraction, Shen et al. [30] build a CNN model for e-learning resource recommendation, which extracts item features from the text information of learning resources, such as the introduction and the content of the material. Attention is a mechanism to flexibly select the reference part of context information, which can facilitate global learning [9]. Chen et al. [8] employ an attention model to tackle the implicit feedback in multimedia recommendation, which builds an item- and component-level attention model to infer the underlying user preferences.

Deep learning provides an in-depth understanding of learner demands and the characteristics of learning content, as well as their interactions. Some researchers exploit the attention models to improve recommendation approaches based on deep learning. Gong et al. [13] adopt CNNs to perform recommendation tasks, which employ an attention layer to produce a weight for a word with its surrounding context. It can effectively select the attention information. Wang et al. [39] use CNNs to effectively learn the representation of interactions and incorporate two kinds of factors into their attention model. It can model varying user preferences and capture the selection interactions to take advantage of from recent knowledge to the current model. Although these deep learning approaches show the effectiveness in recommendation, they are limited when handling real-time recommendation based on dynamic interactions, in which each user/item is assumed to remain unchanged over a period of time. This is inconsistent with real scenarios where user interactions and their learning process are dynamic. This motivates us to design LPRACNN by exploiting the dynamic interaction scene for recognizing an optimal learning peer.

Different to existing work, our framework relies on deep learning to capture non-linear and non-trivial relationships between learners and learning content. In addition, we introduce two additional layers into CNNs from an educational perspective. To the best of our knowledge, this is the first work on seamlessly integrating an attention model and CNN by which to achieve the optimal learning peer recommendation with a trade-off between interaction intensity, diversity and novelty.

3. DITG and attention-driven CNN-based multi-objective learning peer recommendation model

3.1. Problem formulation

To match the optimal learning peers with target learners $\mathbb{L}^t$, we present an LPR approach that utilizes an attention-driven CNN model based on constructing DITG. Before describing our learner recommendation model, we introduce the notations used in this paper.

Assumption 1 (Learning objective). In terms of target learners' learning objectives $LO_{\mathbb{L}^t}$ running through the whole learning process, $\mathbb{L}^t$ should be tightly objective-centric. Therefore, there is a hypothesis that the learning objectives of $\mathbb{L}^t$ are known.

We take into account interaction behaviours to constitute the ternary relations among learners, interaction behaviours, and learning content. Accordingly, an interaction tripartite graph is constructed, with its definition as follows:

Definition 1 (Dynamic interaction tripartite graph, DITG). Given sets of learners $\mathbb{L} = \{l_1, l_2, \ldots, l_m\}$, a set of interaction behaviours $\mathbb{I} = \{i_1, i_2, \ldots, i_k\}$, and a set of learning content $\mathbb{C} = \{c_1, c_2, \ldots, c_n\}$, we denote an undirected DITG as $\mathbb{G} = (\mathbb{L} \cup \mathbb{I} \cup \mathbb{C}, \mathbb{E})$, where $\mathbb{E}$ is the set of edges, together with three disjoint sets of vertices $\mathbb{L}$, $\mathbb{I}$ and $\mathbb{C}$ as shown in Fig. 1. Each triple $Tr = \{l_m, i_k, c_n\}$ derived from $\mathbb{G}$ denotes that learner l_m interacts with learning content c_n by the interaction behaviour i_k.

Note that interaction behaviours can be partitioned into five types with different weights as depicted in Fig. 1. Therefore an interaction behaviour $\mathbb{I}$ is defined in detail as follows:

Definition 2 (Weighted interaction behaviours). We divide interaction behaviours $\mathbb{I}$ into 5 subcategories: $i_1 = \{browse\}$, $i_2 = \{concern\}$, $i_3 = \{comment\}$, $i_4 = \{forward\}$, $i_5 = \{post\}$. Specifically, to clarify each interaction behaviour between learn-

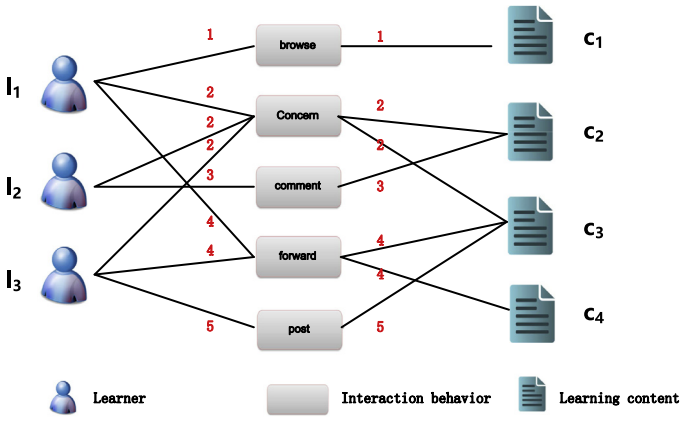


Fig. 1. Construction of a dynamic interaction tripartite graph *DITG*.

ers $\mathbb{L}$ and learning content $\mathbb{C}$, we have

$$\mathbb{I}_k = \left(l_m, l_i, c_n, t_{begin}, t_{end}\right), \tag{1}$$

$$weight(i_k) = k. \tag{2}$$

where $k = 1, 2, 3, 4, 5$. Specifically in Eq. (1), when $k = 1$, $\mathbb{I}_1$ denotes that the learners residence time on the screen exceeds a certain value, expressed as an invalid browse. According to the principles from social network and educational psychology [16,28], we set $t_{end} - t_{begin}$ from 5s to 20s. In other words, if a browse is within [5s, 20s], it is an effective browse or an invalid browse, otherwise. This setting of $t_{end} - t_{begin}$ is also validated by the time distribution from our experimental dataset. When $k = 3$, $\mathbb{I}_3$ is also referred to as an effective comment and an invalid comment in terms of the value of $t_{end} - t_{begin}$. If $0\ day < t_{end} - t_{begin} < \alpha\ days$, i_3 is an effective comment or an invalid comment, otherwise.

According to Eqs. (1) and (2), m learners are studying n learning content with five different interaction behaviours from t_{begin} to t_{end}. In addition, i_1 and i_3 are constrained by time. Further, different weights of interaction behaviours are initially given as a weight vector $\mathbb{W}^d = (1, 2, 3, 4, 5)$, whereas subsequent $\mathbb{W}^d$ are tuned during the learning process. Hence, the dynamic weight $\mathbb{W}^d$ of interaction behaviours is adjusted in accordance with the following rule:

Definition 3 (Dynamic weight adjustment rule). With respect to the weights of interaction behaviours, a combination of manual settings and an attention-driven network is employed to adjust them dynamically. Moreover, the weights vary according to the features of the learning content. On the other hand, the learning content is categorized into two types, including hot and basic content. Hot learning content is that cause learners greater concern in a certain period time, which can enable learners to hold the trend of field and guide learners to learn according to their own needs. The basic content defined in this paper refers that the interaction intensity tends to be stable value for most beginners during a period time. If learning content c_n is hot, we have $\mathbb{W}^d = (1, 2, 3, 5, 4)$, or $\mathbb{W}^d = (1, 2, 3, 4, 5)$ otherwise if c_n is basic.

The reason for defining this rule is that i_1 and i_2 behaviours remain unchanged by setting i_1 and i_2 as 1 and 2, respectively, while other behaviours are different. It is necessary to spread hot learning content among learners by maximizing the weight of i_4 as 5. As i_5 is the weight assigned to content which learners regard as either hot, new or original, i_5 is thereby set to 4. The positive comments made by one learner can persuade other learners to browse or participate so that more peers will search for this content. As such, we assign the weight of i_3 as 3.

On the other hand, our attention-driven CNN model integrated with the attention-driven mechanism adjusts the weight errors caused by the aforementioned manual settings during the training. A given DITG is presented as a no-negative matrix $\mathbf{M} \in \gamma^{\mathbb{L}\times\mathbb{I}\times\mathbb{C}}$. Each entry of $\mathbf{M}$ represents the weight of an interaction behaviour.

In order to formalize the problem of a learning peer recommendation , we introduce the following notations. An interaction intensity matrix **IM** is the interaction intensities between target learners (denoted as $\mathbb{L}^t$) and candidate learners (denoted as $\mathbb{L}^c$), evaluated by a rating function $f(\mathbb{L}^t, \mathbb{L}^c)$. Unlike learning content recommendation, peer recommendation has more constraints, such as the maximum recommendation of learners, the diversity of recommendation lists and novelty. Not only is a rating function of a high accuracy, an efficient multiple metric balancer b is also very significant to the overall performance of the recommendation system.

With the matrix completion of **IM** on DITG, the LPR problem within a certain time window Δt can be formalized as follows: Given $LR_{\Delta t} = \left\{\mathbb{L}^t, \mathbb{L}^c, \mathbb{I}^c, \mathbb{C}^c, \mathbb{Q}, f, b\right\}$, where $\mathbb{L}^c \subset \mathbb{L}$, $\mathbb{C}^c \subset \mathbb{C}$, a set $\mathbb{Q}$ is a collection of constraints or metrics for a balancer b to refer, and the goal is, $\forall i \in \mathbb{L}^t$, to recommend a list of learners $L_i \in \mathbb{L}^c$ through f and b with certain limitations or constraints $\mathbb{Q}$.

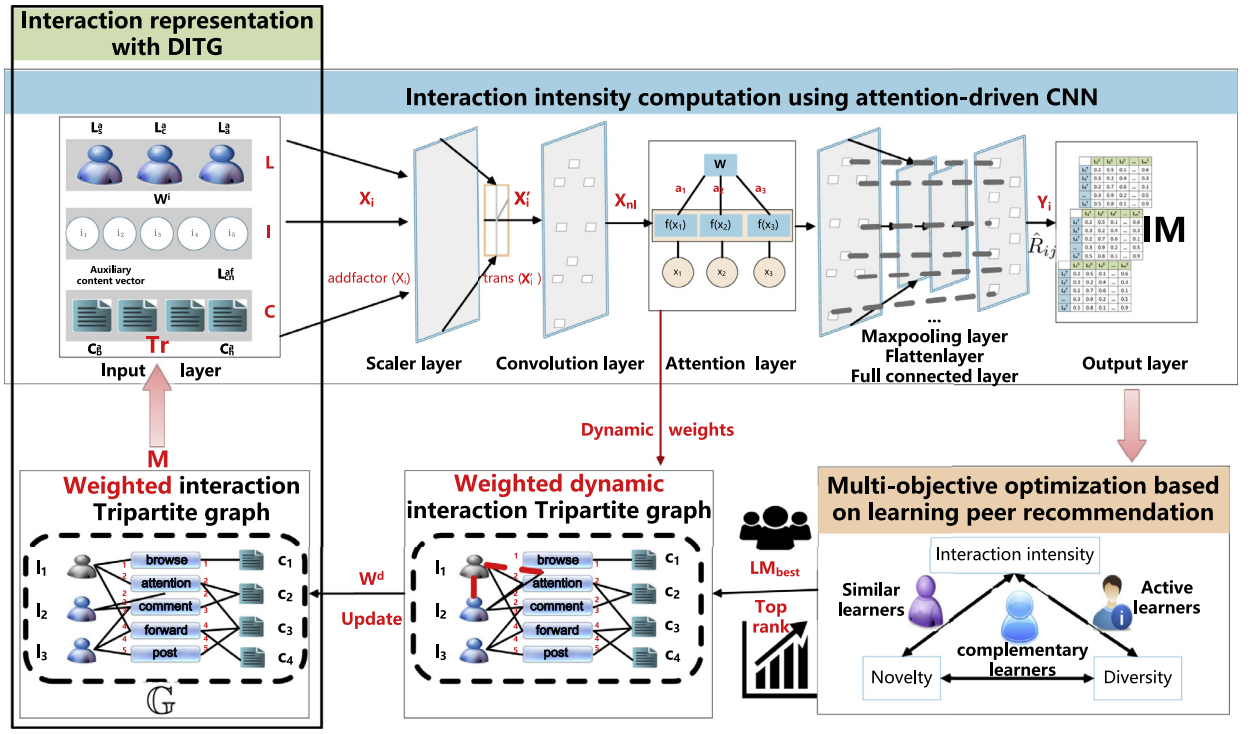


Fig. 2. Our framework for multi-objective LPR using attention-driven CNN and interaction tripartite graph *(LPRACNN)*.

3.2. A framework of multi-objective optimization LPRACNN based on DITG

The working flow of our framework is illustrated in Fig. 2. The framework consists of three main components: interaction representation with a dynamic interaction tripartite graph, interaction intensity computation using an attention-driven CNN, and multi-objective optimization based on learning peer recommendation. We briefly describe these components as follows.

Interaction representation with a dynamic interaction tripartite graph. To capture interaction information, we further refine this component $\mathbb{L}$, $\mathbb{I}$ and $\mathbb{C}$ of DITG, namely, learner attributes and learning content attributes. In terms of interaction intensity, learner attributes can be divided into active learners ($\mathbb{L}_a^a$), similar learners ($\mathbb{L}_s^a$) and complementary learners ($\mathbb{L}_c^a$). Learning content attributes fall into two categories according to the interaction intensity: basic content is denoted as $\mathbb{C}_b^a$ and hot content is denoted as $\mathbb{C}_h^a$. Additionally, considering the preferences of learners, features $\mathbb{C}_{c_n}^{af}$ of learning content attributes $\mathbb{C}_b^a$ and $\mathbb{C}_h^a$ are categorized as text, image, audio, video and animation.

$$\mathbb{C}_{c_n}^{af} = \{text, image, audio, video, animation\}. \tag{3}$$

Also, for a given DITG, the ternary relations that the attributes $\mathbb{L}^a$ of learner l_m interacts with feature $\mathbb{C}_{c_n}^{af}$ of learning content c_n by interaction behaviour i_k can be denoted as

$$Tr = (l_m\mathbb{L}^a, i_k, w_k, c_n\mathbb{C}_{cn}^{af}). \tag{4}$$

Interaction intensity computation using attention-driven CNN. To fill the unobserved entries in interaction intensity matrix **IM** between $\mathbb{L}^t$ and $\mathbb{L}^c$, we employ an attention-driven CNN model as a fitting function f to generate them. First,a DITG $\mathbb{G}$ between $\mathbb{L}^t$ and $\mathbb{L}^c$ is constructed and transferred into the ternary relations Tr as input data at the model training stage. After representing interaction information associated with $\mathbb{L}^t$ and $\mathbb{L}^c$, let $X_i = x_i^L \cup x_i^C \cup x_i^I$ be the learner, learning content and interaction behaviour input vector respectively, and $X_i \in Tr$. Considering the initial error of X_i from DITG, we design an extra scaler layer before the traditional CNN neural network through an operation of $addfactor(X_i)$, then X_i is set to X_i'. Moreover, another operation $trans(X_i')$ transforms the input vector into a matrix. Furthermore, central to learning content $x_i^{C'}$, an attention layer is integrated into the CNN model to adjust the weights $\mathbb{W}^d$ of interaction behaviours $\mathbb{I}$. Through the above attention-driven layer, the entries of **IM** are obtained more precisely. The output Y_i denotes five levels of interaction intensity between $\mathbb{L}^t$ and $\mathbb{L}^c$.

Multi-objective optimization based on learning peer recommendation. With a given number of candidate learners $\mathbb{L}^c$ and matrix $\mathbf{LM}_{best}$, this component produces a final and best learner recommendation for target learners by simultaneously optimizing three objectives of interaction intensity, diversity, and novelty with balancer b.

4. Attention-driven CNN based learning peer recommendation model

In this section, we present a deep learning based learning peer recommendation model with DITG. As shown in Fig. 1, the proposed DITG can represent the interaction behaviours of learners. Therefore, we employ deep learning to develop a learner recommendation model using the DITG records as training data. We detail the model in the following section.

4.1. System input and output of LPRACNN

As we know, a fitting rating function $f:\mathbb{R}^d \to \mathbb{R}^m$ has a fixed dimension of the input and output. As a result, the input and output for the LPRACNN model can be denoted as a vector pair $(\boldsymbol{X}_i, \boldsymbol{Y}_i)$, where $\boldsymbol{X}_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,d}\}$, $\boldsymbol{Y}_i = \{y_{i,1}, y_{i,2}, \ldots, y_{i,m}\}$. Also, the order of $\boldsymbol{X}_i$ is important for model training or model prediction. Nevertheless, the DITG involves thousands of learning content and complex interactions between learners and learning content. To reduce the complexity of the model, the proposed model accepts only two learners of target learners $\mathbb{L}^t$ and candidate learners $\mathbb{L}^c$ with their accompanying DITG as input.

As discussed, the learning content is categorized into a fixed number of groups in terms of types and features (e.g., basic, hot, text). Moreover, according to the categories of learning content, interaction behaviour weights are adjusted dynamically.

For the output of our model, we begin with a top-down design for the final computation of interaction intensity **IM**. An element of **IM** is a non-negative integer that indicates how likely it is that two learners become a pair of positive learning peers. Therefore, the input of the LPRACNN model can be denoted as

$$\boldsymbol{X}_i = \big(cl_1, I_1, \ldots, I_i, \ldots, I_n, cl_2, J_1, \ldots, J_j, \ldots, J_n, cl_3, k_1, \ldots, k_k, \ldots, k_n\big), \tag{5}$$

where cl_1, cl_2, cl_3 are the properties of learners, I_i, J_j, k_k are the attributes of learners, the attributes and features of the learning content, and the categorized and aggregated weights of different interactions, respectively. The output of the LPRACNN model is

$$\boldsymbol{Y}_i = f(\boldsymbol{X}_i) = (y_{i,1}, \ldots, y_{i,j}, \ldots, y_{i,d}), \tag{6}$$

where $y_{i,j}$ equals 0 or 1, and $\sum_d^{j=1} y_{i,j} = 1$. The vector form can be transformed into a scalar form by applying Eq. (7).

$$j = \{j | y_{i,j} = 1, 1 \le j \le d\}, \tag{7}$$

where the larger the value of j, the more positive the learning closer to the learning peer pair.

4.2. Scaler layer for feature extraction

We employ a deep learning model to fit the potential rating function f. As discussed in Section 4.1, the fitting function f accepts a vector as its input [18].

Thus, we design an extra layer named a *scaler* layer before the traditional CNN neural network, which consists of two operations. One is to add a factor to the elements of $\boldsymbol{X}_i$ (given in Eq. (8)), which compensates for the possible loss cost by encoding, and it will be set during the training of the model.

$$\boldsymbol{X}_i' = addfactor(\boldsymbol{X}_i) = \{\alpha_1 x_{i,1}, \ldots, \alpha_d x_{i,d}\}. \tag{8}$$

In the *scaler* layer, another operation is to transform the input vector into a *m*atrix, which can formalized as Eq. (9). The intuition behind this transformation is to explore different combinations of features, through which we can possibly eliminate the limitation as a result of the fixed order of elements in $\boldsymbol{X}_i$ and so it can be compatible with a traditional CNN network.

$$\boldsymbol{Y}_i' = trans(\boldsymbol{X}_i') = (e_{i',j'})_{d_2 \times d_1} = \begin{bmatrix} e_{1,1} & e_{1,2} & \ldots & e_{1,d_1} \\ \vdots & \vdots & \ddots & \vdots \\ e_{d_2,1} & e_{d_2,2} & \ldots & e_{d_2,d_1} \end{bmatrix}, \tag{9}$$

where $e_{i',j'}$ are obtained by applying different operations between $\alpha_i' x_{i,i'}$ and $\alpha_j x_{i,j'}$, or simply put $\alpha_{i'} x_{i,i'}$ and $\alpha_{j'} x_{i,j'}$ together to form an new element of the vector.

As we can see, both these two kinds of operations are still a linear transformation between layers, which enables us to employ neural nodes to perform the transformation. For the operation *addfactor* to $\big(x_{i,i'}\big)_{1\times d}$, we can use a layer of d nodes to perform the operation, which is formalized as:

$$\alpha_{i'} x_{i,i'} = \sum_{i'=1}^{d} z(w_{i'} x_{i,i'} + b_i), i = 1, 2, \ldots, d, \tag{10}$$

where $w_{i'}$ equals 0 if $i' \neq i$, $b_i = 0$, and z is an identical function.

As for the operation *trans*, we can use a layer of $d \times d$ nodes to perform the operation, which is described:

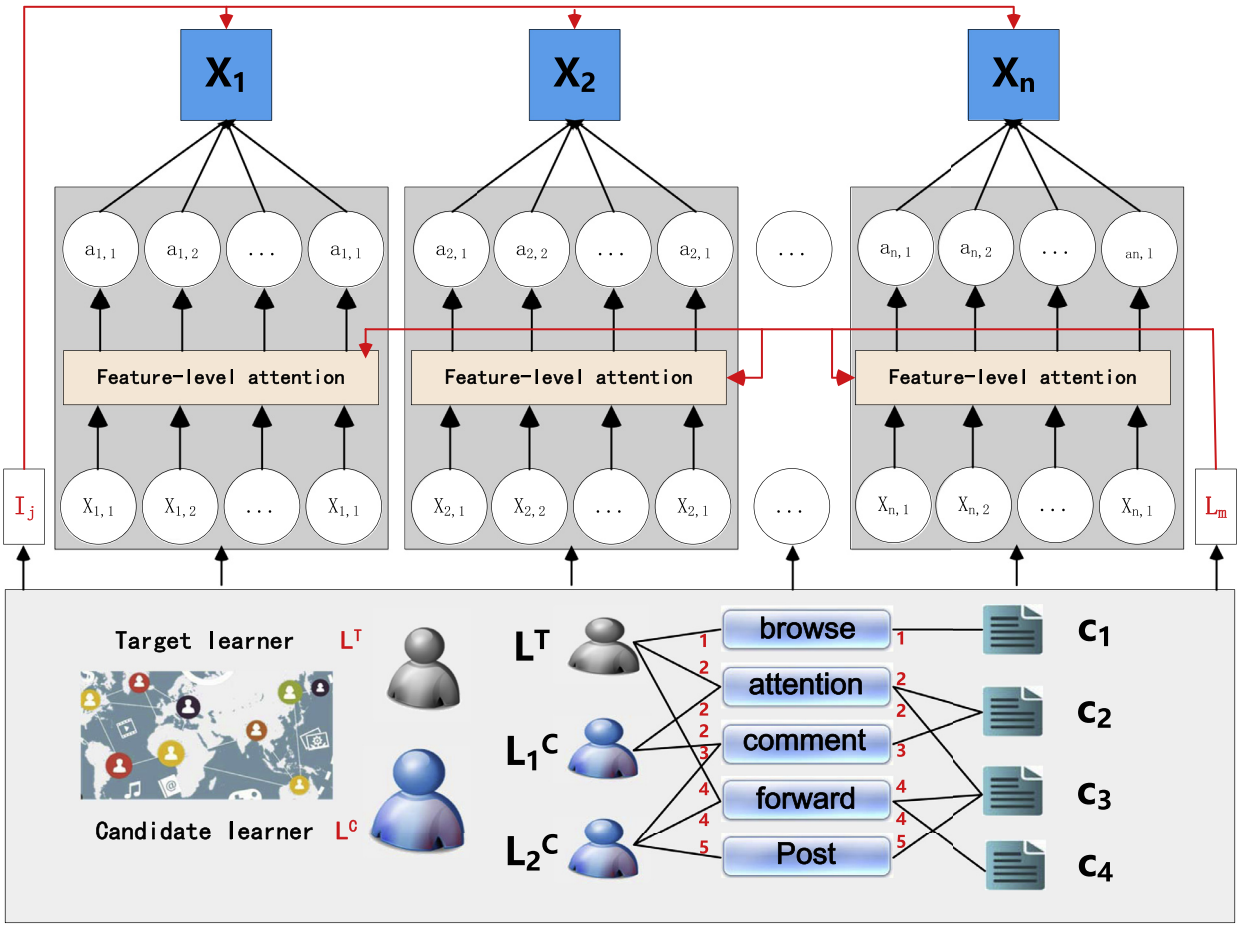


Fig. 3. Attention-driven dynamic weight adjustment of interaction behaviours.

$$e_{i,j} = \sum_{i'=1}^{d} z'(w_{i,j,i'} x_{i,i'} + b_{i,j}), i, j = 1, 2, \ldots, d, \tag{11}$$

where z' is an activation function, $w_{i,j,i'}$ equals 0 if $i' \neq i$ or $i' \neq j$.

Therefore, we integrate the *scaler* layer with traditional CNN layers as a whole network for training.

4.3. Attention-driven dynamic adjustment of interaction weights

Here attention-driven dynamic weights of interaction behaviours are adjusted dynamically. The basic idea is to use the learning content-centred principle. Central to a learning objective, learners select corresponding learning content to construct their own knowledge. For example, a higher portion of hot learning content is associated with a learning objective during a certain period of the learning process. As such, the spread of hot learning content is accelerated. Interaction behaviours such as forwarding and posting should be given higher weights than those for browsing and commenting. With this in mind, we propose a novel attention-driven dynamic weight adjustment model of interaction behaviours, as shown in Fig. 3.

To incorporate these two aspects, an attention-driven solution is taken into account. Specifically, learning content is composed of complex information. Accordingly, for the same learning content, different learners may prefer different parts. To represent the various features of each learning content c_n, we denote x_{n^*} as a variable-sized set of features. $|\{x_{n^*}\}|$ is the size of the set and x_{nl} the l-th features in the set. Different to conventional models that generally extract a unified feature representation by using the average pooling [7], learning content-centred attention assigns features with attention weights that are consistent with the spread characteristics of interaction behaviours, and then uses the weighted sum as the new weights.

A learning content-centred attention score for the l-th features x_{nl} of learning content c_n from learner l_m is calculated by a two-layer network:

$$b(m,n,l) = w_2^T \phi(\mathbf{W}_{2l} l_m + W_{2x} x_{nl} + b_2) + c_2, \tag{12}$$

where the matrixes $\mathbf{W}_{2*}$ and bias b_2 are the first-layer parameters, vector W_2 and bias c_2 are the second-layer parameters, and $\phi(x) = \max(0, x)$ is the *ReLU* function. Then, the final sum of the learning content-centred attention score is normalized as:

$$\beta(m,n,l) = \frac{exp(b(m,n,l))}{\sum_{q=1}^{|\{x_{n*}\}|} exp(b(m,n,l))}, \tag{13}$$

where q is the size of $|\{x_{n*}\}|$.

Once the learning content-centred attention $\beta(m, n, l)$ is obtained, the interaction behaviour weights $\mathbb{W}^d$ of learning content c_n are calculated by the following weighted sum:

$$\bar{x}_n = \sum_{l=1}^{|\{x_{n*}\}|} \beta(m,n,l) X_{nl}. \tag{14}$$

4.4. Interaction intensity score computation using attention-driven CNN

For the aforementioned attention-driven dynamical weights of interaction behaviours, different attention values can be learnt to distribute the significance of five interaction behaviours with respect to the learning content. Simultaneously, the features generated from the above encoded process,i.e., L (denoted as the learner latent vectors $L = [l_1, l_2, l_3, \ldots, l_M] \in \mathbb{R}^{D\times M}$), C (denoted as the learning content latent vectors $C = [c_1, c_2, c_3, \ldots, c_N] \in \mathbb{R}^{D\times N}, D << \min(M, N)$), $\mathbb{W}^d$, and the parameters of the attention networks are obtained. Accordingly, we use a set of convolutional layers and pooling layers in an alternate way with multiple features to produce the outputs.

The layers after the *scaler* layers are two groups of operations (*Convolution2D, ReLU, MaxPooling2D*). After the two groups of layers, we add a *Flatten* layer before the *fully-connected* layer. The *fully-connected* layer is to create fully-connected layers of mono-dimensional features. The last layer is the *SoftMax* layer, which is designed to map the activation from the aggregated previous layers to a class probability distribution.

The scores $\hat{R}_{ij}$ of the interaction intensity for target learners and candidate learners are obtained by the fully-connected layer:

$$\hat{R}_{ij} = \left(l_i + \sum_{n\in\mathcal{R}(i)} \beta(m,n,l) p_n \right)^t v_j, \tag{15}$$

where $\hat{R}_{ij}$ is the final score of interaction intensity between target learner i and candidate learner j. $\mathcal{R}(i)$ is the set of all learning content with which learner i and candidate learner j interact. Note that each learning content in $\mathcal{R}(i)$ is associated with two factor vectors. One is denoted by v_n, which is the basic vector of learning content c_n. The other is the auxiliary feature vector (e.g., $\mathbb{C}_{c_n}^{af}$), denoted by p_n, which is used to characterize learners based on the set of learning content with which they interact. Moreover, training data $\mathcal{R}(B)$ is generated by:

$$\mathcal{R}(B) = \{(i,j,k) | j \in \mathcal{R}(i) \wedge k \in C \backslash \mathcal{R}(i)\}, \tag{16}$$

where C is the set of all learning content and the semantics of $i, j, k \in \mathcal{R}(B)$ is what learner i is assumed to prefer when interacting with j over k. $\hat{R}_{ij}$ can be further transferred into matrix **IM** according to the output scores from the fully-connected layer. So learner recommendation is then reduced to a ranking problem among candidate learners. We can optimize attention-driven recommendation between the positive and non-observable content:

$$\begin{aligned} arg\min_{L,C,P,\Theta} \sum_{(i,j,k)\in\mathcal{R}(B)} & -In\ \sigma\{(l_m + \sum_{l\in\mathcal{R}(i)} \beta(m,n,l) p_l)^T v_j \\ & - (l_i + \sum_{l\in\mathcal{R}(i)} \beta(m,n,l) p_l)^T v_k\} + \lambda(||L||^2 + ||C||^2 + ||P||^2), \end{aligned} \tag{17}$$

where σ is the logistic sigmoid function and λ is the regularization parameter.

An attention-driven algorithm for interaction intensity based on the bootstrap sampling of training triples is proposed for the network. The main procedures of the proposed algorithm for training the model are summarized in Algorithm 1.

For Algorithm 1, we use Θ to denote the set of parameters in two attention layers, and $\hat{R}_{ijk}$ to denote $\hat{R}_{ij} \leftarrow \hat{R}_{ik}$. Note that Line 14 is the gradients of the model parameters updated using chain rules. To optimize the objective function (Eq. (17)), we employ stochastic gradient descent (SGD) — a universal solver for optimizing neural network models. Meanwhile a training instance is randomly selected. Moreover, during the training, each model parameter is updated towards the direction of its negative gradient.

Algorithm 1 Interaction intensity based on the attention-driven deep convolutional neural network model.

Input: Target learner $\mathbb{L}^t$, dynamic interaction tripartite graph $\mathbb{G}$ associated with $\mathbb{L}^t$, and the difference tolerance ϵ_0;
Output: Latent feature matrixes L, C, P, parameters in attention model Θ and matrix
IM between target learner $\mathbb{L}^t$ and candidate $\mathbb{L}^c$;

Construct learner-interaction behaviour-learning content tripartite graph $\mathbb{G}$ with weights $\mathbb{W}$;
Convert $\mathbb{G}$ into matrix **M**;
Initialize L, C, P and Θ;
repeat
  Draw i, j, k from $\mathcal{R}(B)$;
  for each feature l of learning content n in $\mathcal{R}(i)$ **do**
    Calculate $\beta(m, n, l)$ to (Eq. 12) and (13);
    Calculate $\bar{x}_n$ via Eq. (14);
  end for
  $L_i^{'} \leftarrow L_i + \sum_{l \in \mathcal{R}(i)} \beta(m, n, l) p_n$;
  $\hat{R}_{ij} \leftarrow u_i^{'} v_j \leftarrow u_i^{'} v_k$;
  $\boldsymbol{IM}_{[i][j]} = \hat{R}_{ij}$ //The matrix represents target learner i interacting with candidate learner j;
  for each parameter θ in L, C, P, Θ **do**
    Update $\{\theta, \epsilon\} \leftarrow \theta + \eta \left(\frac{exp^{-\hat{R}_{ijk}}}{1+exp^{-\hat{R}_{ijk}}} \frac{\delta \hat{R}_{ijk}}{\delta \eta} + \lambda \theta \right)$;
  end for
until $\epsilon \leq \epsilon_0$
Return L, C, P and Θ;
Select available learners for target learner $\mathbb{L}^t$ from **IM**.

To conclude, given a testing set of relation selection among learners, interaction behaviours and learning content, DITG descriptions are extracted from the target learners and candidate learners, and particularly mapped into the attention-driven deep CNN. Then, they are mapped into different scores by the learnt mapping functions. Finally, the interaction intensity matrix **IM** between the target learner i and candidate learner j are obtained with all scores $\hat{R}_{ij}$.

4.5. Learning peer recommendation by multi-objective optimization

Personalized recommendation should consider multiple preferences of the targeted users, especially in learning peer recommendation [4]. So, it is not enough to recommend learners for a learner in online learning only with the output of the rating function f.

Our work aims to find a trade-off between the proposed three objective functions using a novel multi-objective algorithm, and to return a set of recommendation solutions for the target learners that is optimal in light of three important metrics, namely, interaction intensity, diversity and novelty. Note that the three metrics are weighted according to different features of the learning content for a given target learner; that is, $\mathbb{W}^O = (w_1, w_2, w_3)$, where $w_1 + w_2 + w_3 = 1, 0 \leq w_i \leq 1, i = 1, 2, 3$.

Matrix of selection. First, a decision possibility matrix of learner recommendation for a target learner $\mathbb{L}^t$ is obtained, denoted as **LM**. It is defined as follows:

$$\mathbf{LM} = [l_1, l_2, \ldots, l_i, \ldots, l_n]^T = \begin{bmatrix} l_{1,1} & \ldots & l_{1,j} & \ldots & l_{1,m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n,1} & \ldots & l_{n,j} & \ldots & l_{n,m} \end{bmatrix}, \tag{18}$$

where n is the cardinality of set $\mathbb{L}^t$, and m the cardinality of set $\mathbb{L}^c$, and $l_{i,j}$ equals 1 (recommended) or 0 (not recommended). Each column should meet with $\sum_{i=1}^{n} l_{i,j} \leq n'$ (the maximum count of a learner to be recommended to the target learners), and each row should meet with $\sum_{j=1}^{m} l_{i,j} \leq m'$ (the length of the learner recommendation list).

Interaction intensity. The high-quality interaction behaviours is essential to achieve the learning objective. Through the aforementioned model of the attention-driven deep convolutional neural network, $\hat{R}_{ij}$ between $\mathbb{L}^t$ and $\mathbb{L}^c$ is obtained by forming matrix **IM**.

$$\mathbf{IM} = (\hat{R}_{ij})_{n \times m}, \tag{19}$$

Diversity. Diversity measures the capacity of recommending return learner to different learners for a target learner. Given two learners i and j, Div is an important factor of the recommendation algorithm defined by Hamming distance H_{ij}

$$H_{ij} = 1 - \frac{Q_{ij}}{L}, \tag{20}$$

where L is the length of the prediction list, Q_{ij} is the same nodes between the learner i and the learner j in the prediction list. Therefore, the diversity matrix **Div** is defined as follows:

$$\mathbf{Div} = (H_{ij})_{n\times m}. \tag{21}$$

Novelty. In learner recommendation, learners with few interaction should also be recommended. Novelty quantifies the ability to discover novel learners, denoted as **Nov**. Novelty can be measured in terms of the mean and the variance of interaction behaviours. The novelty value nov_j of learner j proposed by Wang et al. [38] is defined as:

$$nov_j = \frac{1}{\mu_j(\sigma_j+1)^2}, \tag{22}$$

where the value of a candidate learner j is the reciprocal of mean (μ_j) and the variance (σ_j), rated by all learning content with which learner j interacts. To avoid division by zero, a small value is added to the variance. The more active a candidate learner, the lower the value. Thus, the **Nov** of a novelty is formed as a vector

$$\mathbf{Nov} = (div_1, div_2, \ldots, div_m). \tag{23}$$

Objective functions F_1, F_2 and F_3. The different values of $\mathbf{LM}_{n\times m}$ can be regarded as different possibilities of learner recommendation. The values of **IM**, **Div** and **Nov** for each **LM** are presented by

$$\begin{cases} F_1(\mathbf{LM}) = e(\mathbf{LM} \bigodot \mathbf{IM})e^T \\ F_2(\mathbf{LM}) = e(\mathbf{LM} \bigodot \mathbf{Div})e^T \\ F_3(\mathbf{LM}) = e(\mathbf{LM} \bigodot \mathbf{Nov})e^T \end{cases}, \tag{24}$$

where e is a vector of all ones, **LM** is equivalent to $\mathbf{LM}_{n\times m}$, $F_1(\mathbf{LM})$ is the value of **IM** for $\mathbf{LM}_{n\times m}$, analogously with $F_2(\mathbf{LM})$ and $F_3(\mathbf{LM})$. The operator $\odot$ is the Hadamard product. It is desirable to select optimal learners when each objective reaches the maximum with a certain value of **LM**. But this usually does not happen. Further, different recommendations can set different preferences for **IM**, **Div** or **Nov** by specifying w_i. Thus, the overall objective function for learner recommendation is obtained as:

$$F(\mathbf{LM}) = F_1(w_1 \odot \mathbf{LM}) + F_2(w_2 \odot \mathbf{LM}) + F_3(w_3 \odot \mathbf{LM}), \tag{25}$$

where we have $\mathbb{W}^O = (\mathbb{W}^O_{i,j})_{n\times 3}, w_j = (\mathbb{W}^O_{i,j})n \times 1, (j = 1, 2, 3)$.

The max of $F(\mathbf{LM})$ is obtained by finding the best selection matrix $\mathbf{LM}_{best}$ for the objective function. Thus, the solution to learner recommendation is cast into finding the approximate optimal solution to $\max F(\mathbf{LM})$.

We implement a parallel genetic algorithm as a function *solve*(.) in the system to solve the problem. The whole algorithm is given in Algorithm 2. The sequence diagram of learner recommendation is shown in Fig. 4.

Algorithm 2 Multi-objective Optimization for Learner recommendation.

Input: w_i;
Output: $\mathbf{LM}_{best}$;
1: Initialize **LM**;
2: **IM** $\leftarrow (\hat{R}_{ij})$ //The configuration of **IM**;
3: **Div** $\leftarrow (H_{ij})$ //The configuration of **Div**;
4: **Nov** $\leftarrow (nov_j)$ //The configuration of **Nov**;
5: $F(\mathbf{LM}) \leftarrow (F_1, F_2, F_3 \odot \mathbb{W}^O(\mathbf{LM}))$;
6: $\mathbf{LM}_{best} \leftarrow solve(maxF(\mathbf{LM}))$;
7: Return $\mathbf{LM}_{best}$.

Additionally, the sequence diagram of learning peer recommendation based on multi-objective optimization is illustrated in Fig. 4. Different to traditional genetic algorithms, some key steps of the algorithm are summarized.

*S*1: To quickly produce an acceptable solution from a large problem space, we produce the population p_i consisting of p_{1i}, p_{2i}, p_{3i} respectively from the sound results of $F_1(\mathbf{LM})$, $F_2(\mathbf{LM})$ and $F_3(\mathbf{LM})$ after computing the DITG information.

*S*2: If the process does not produce a satisfying solution for target learners after the current loop, and there are still acceptable solutions for $F_1(\mathbf{LM})$, $F_2(\mathbf{LM})$, $F_3(\mathbf{LM})$ respectively, which are not included in the previous loops, then go back to Step *S*1.

*S*3: The numbers of target learners selected from p_{1i}, p_{2i}, p_{3i} are equal.

*S*4: The crossover must be conducted between different $p_{ji}(j = 1, 2, 3)$.

Specifically, a hash table is employed to minimize the time of sorting or conditional statements, which costs more space, but uses less time.

5. Performance evaluation

This section reports the experiment results to demonstrate the performance of our proposed LPRACNN framework. Our experiments consist of three parts. The first is to validate the proposed multi-objective optimization. The second is for

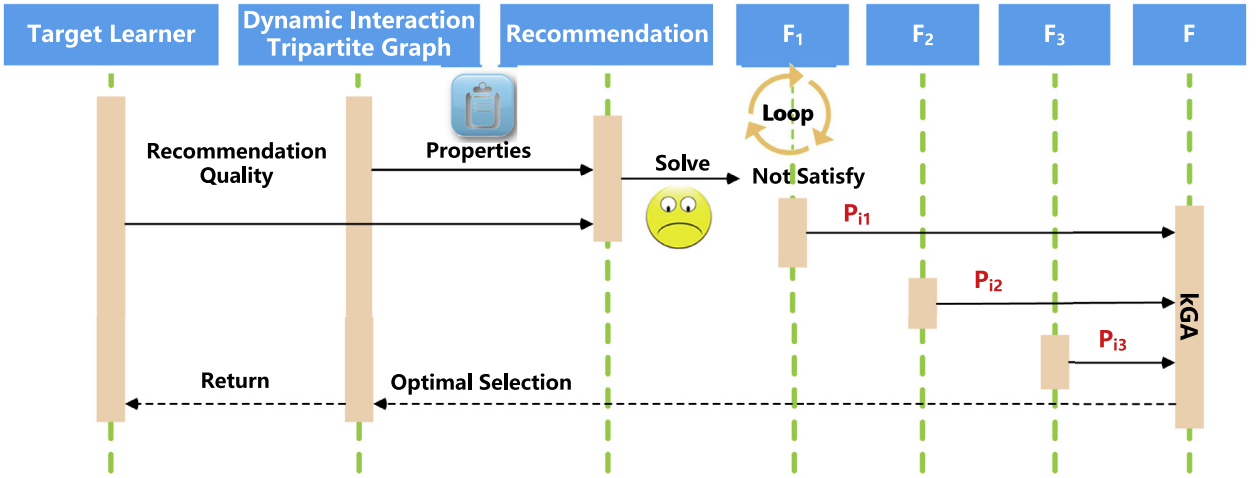


Fig. 4. The sequence diagram of learner recommendation based on multi-objective optimization.

Table 1
World University City Dataset.

Learners	Learning Content	Interaction Behaviour	Learner-Interaction	Content-Interaction
10570	4035	45083	32580	12503

assessing attention-driven dynamic performance, while the last one is to comprehensively compare our framework with other relevant ones.

5.1. Baseline methods for comparisons

For comparisons, we select six algorithms that are similar to our algorithm as baselines: (i) BGR [45], a traditional bipartite graph method that only considers the selection relationships between learners and learning content; (ii) LTC [44], using a tripartite graph to exploit the information of tags; (iii) LSC [32], employing a tripartite graph to represent the relationships among the learners-learning content-scores; (iv) CNNR [30], deep learning recommendation based on CNN; (v) LIC, without the dynamic weight adjustment of interaction behaviours; (vi) MOEALTR [38], based on a multi-objective recommendation in light of interaction intensity, diversity and novelty.

For different problems, most of the existing recommendation methods consider the three metrics of accuracy, diversity and novelty either separately or simultaneously. This is different from our work. As a result, it is difficult to compare the performance of our multi-objective optimization with these methods directly. Accordingly, a hybrid recommendation algorithm (MOEALTR) [38] is selected. For a fair comparison, we set the same value of W^o used in LPRACNN and MOEALTR in our experiments. As for the comparison of dynamic performance, we chose LIC (without weighting interaction behaviours) and DLIC (weighting interaction behaviours using manual settings). Furthermore, we compared LPRACNN with BGR, LTC, LSC, LIC, CNNR and MOEALTR, in several experiments with respect to the evaluation metrics given in Section 5.3.

5.2. Datasets

In order to verify the effectiveness of the proposed algorithm, we choose the dataset from the World University City which is a public dataset adopted by many researchers [19,24,46]. The interaction frequencies among the learners, interaction behaviours and learning content in the dataset follow a power-law distribution. In particular, some learning content was accessed frequently, while other content was accessed infrequently. A considerable number of learners have a low weight for interaction behaviours, which results in high sparsity. As such, it is difficult to make recommendations for learners from such a raw dataset. Therefore, we preprocessed the dataset by reducing its sparsity. Specifically, we removed the learners whose interactions are less than 20 and selected the learning content whose interaction frequency is more than 30. After preprocessing, the dataset consists of 10,570 learners, 4035 examples of learning content, 45,083 examples of interaction information, 32,580 edges between learners and interaction behaviour, and 12,503 edges between interaction behaviour and learning content, as listed in Table 1.

However, the currently available data for education cannot completely meet the requirements of our experiments. So, we need additional simulation data. To ensure that the simulation data is close to the real-world dataset, we first extracted some rules from the real dataset. The simulation dataset was then generated by applying these rules. In addition to these

Table 2
Simulation dataset.

Learners	Learning Content	Interaction Behaviour	Learner-Interaction	Content-Interaction
100300	45050	418006	300952	123500

Table 3
Four possible states of learners to be predicted.

Learner adoption	Recommendation of the algorithm	Not recommendation of the algorithm
yes	k_{tp}	k_{fn}
no	k_{fp}	k_{tn}

rules, we also selected T statistical periods as the time length of the simulation dataset given the periodicity of interaction and learning.

Table 2 summarizes the statistics of the simulation dataset in accordance with Table 1. To verify the algorithm's performance, 60% of the dataset is the training set, while the rest is the testing set for each dataset. All the experiments were conducted on an Amax GPU server with 64G memory and Titan X Pascal GPU X 4.

5.3. Evaluation metrics

In our experiments, we use seven metrics to evaluate the performance of LPRACNN. Each is described in detail in the following.

Precision and Recall: Precision is the percentage of relevant prediction instances among the retrieved instances, while recall is the percentage of relevant prediction instances that have been retrieved over the total amount of relevant instances. There are often four possible cases for adopting the recommended or unrecommended learning peers , as listed in Table 3, where k is the number of learning peers in the testing data.

According to Table 3, two metrics [41], are defined as

$$\mathrm{Precision}_{\mathrm{U_{Test}}}(\mathrm{L}) = \frac{1}{\mathrm{M}} \sum_{\mathrm{U_{Test}}} \mathrm{k_{tp}}, \tag{26}$$

$$\mathrm{Recall}_{\mathrm{U_{Test}}}(\mathrm{L}) = \frac{1}{\mathrm{M}} \sum_{\mathrm{U_{Test}}} \frac{\mathrm{k_{tp}}}{\mathrm{k_{tp}} + \mathrm{k_{fn}}}, \tag{27}$$

where U_{Test} is the set of test learners, and M is the number of test learners. In addition, as given in Eq. (28), F_1 is employed to evaluate the performance of LPRACNN comprehensively.

$$F_1(U) = \frac{2\mathrm{Precision}_{\mathrm{U_{Test}}}(\mathrm{L})\mathrm{Recall}_{\mathrm{U_{Test}}}(\mathrm{L})}{\mathrm{Precision}_{\mathrm{U_{Test}}}(\mathrm{L}) + \mathrm{Recall}_{\mathrm{U_{Test}}}(\mathrm{L})}. \tag{28}$$

MAP (Mean Average Precision): In real-world learning environments, most learners rarely browse a whole recommendation list. This fact indicates that even if the recommendation accuracy is high, the real effectiveness is widely divergent due to the location rearward of appropriate peers in the recommendation list. Hence, a widely accepted metric, *MAP* can select the optimal learning peers, simultaneously guaranteeing the top rankings of the selected learners in a recommendation list [27]. MAP is defined by Eq. (29).

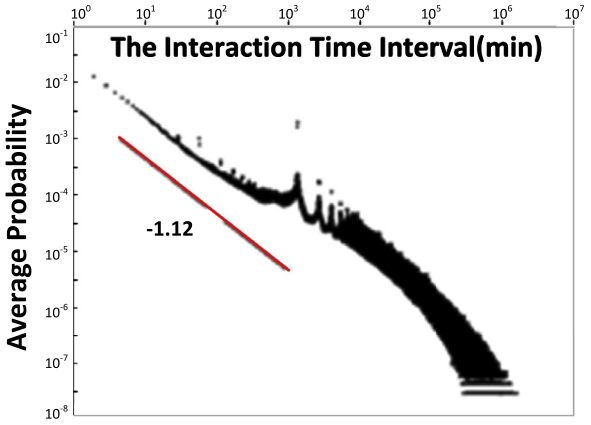
$$\mathrm{MAP} = \frac{1}{\|\mathrm{U}\|} \sum_{\mathrm{u=1}}^{\|\mathrm{U}\|} \left(\frac{1}{\mathrm{T_u}} \sum_{\mathrm{k=1}}^{\mathrm{T_u}} \mathrm{rel_k Precision_U(k)} \right), \tag{29}$$

where T_u is the set of learners with learner u with interactions from the testing data, $\mathrm{Precision_U(k)}$ the precision at top k, and rel_k is a binary variable that is 1 if a learner with rank K in the recommended list appears in T_u or 0 otherwise.

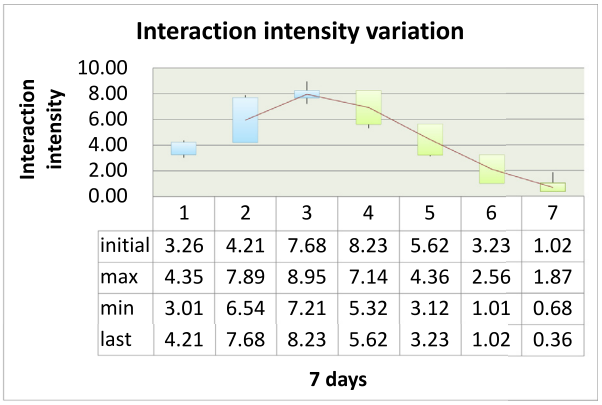
Diversity: Generally speaking, a more personalized prediction list has a larger hamming distance than other prediction lists. Consequently, to avoid repetition with Eq. (20), the mean of the hamming distance measures the diversity of prediction and recommendation.

$$S = \frac{1}{m(m-1)} \sum_{i \neq j} H_{ij}, \tag{30}$$

where $0 \leq S \leq 1$. In a case of $S = 1$, this implies that all the prediction and recommendation lists of the learners are completely different, and $S = 0$ for the exact same lists.



(a) The average interaction distribution of learners



	1	2	3	4	5	6	7
initial	3.26	4.21	7.68	8.23	5.62	3.23	1.02
max	4.35	7.89	8.95	7.14	4.36	2.56	1.87
min	3.01	6.54	7.21	5.32	3.12	1.01	0.68
last	4.21	7.68	8.23	5.62	3.23	1.02	0.36

(b) Interaction variation at different times

Fig. 5. Parameter α of comment time and interaction time.

Novelty: Novelty measures how well an approach recommends unknown learners to target learners. To measure the unexpectedness of candidate learners, novelty is defined as

$$\text{Novelty} = \frac{1}{\text{mk}} \sum_{\text{u=1}}^{\text{m}} \sum_{\text{i} \in \text{L}_\text{u}} \text{d}_\text{i}, \tag{31}$$

where L_u is the top-k list of a learner u, m is the number of learners and d_i is the degree of learning content j.

Hit ratio: By considering a dynamic prediction associated with time, it is necessary to make recommendations for learners at a certain time. Hence, the hit ratio HitRatio^t is employed to measure the dynamic recommendations of LPRACNN. First, the data set is divided into training set U_{train} and testing set U_{test}. By ranking all data instances in order of time, the instances of learners who interacted with other learners in a recent time period are extracted into U_{test} for testing, while the others are assigned to U_{train}. For the prediction, each learner l is regarded as the link prediction node at a given time t, and all its link nodes are denoted as $R_{(l,\ t)}$. If there is just one node in the testing set U_{test}, then one shot hit an appropriate peer. The hit ratio is calculated as:

$$\text{HitRatio}^t = \frac{\sum_l T_l \in R_{(l,t)}}{U_{test}}, \tag{32}$$

where $T_l \in R_{(l,\ t)}$ is an indicator function, and T_l is the learning content with which the learners interacted in the testing set U_{test}

5.4. Parameter settings

In this section, we experimentally set different values of parameter α as effective comment behaviours, which is the time length of an interaction behaviour. In the education field, memory and attentiveness of knowledge cognition are affected by time. So, it is necessary to evaluate the impact of interaction time on cognitive load. The interaction behaviours of learners can occur over time. Therefore, the interaction time length should be adjusted to match the optimal interaction intensity to reach a satisfactory recommendation.

In Experiment 1, the interaction data of learners over a two-week period is extracted. For the five interaction behaviours, the interaction interval is measured in minutes with its distribution shown in Fig. 5.

As shown in Fig. 5(a), the data follows a power law attenuated at the time interval $t < 10^4$ min (about 7 days) with the attenuation exponent of $a = -1.12$. The attenuation speed accelerates at $t > 10^4$ min, which implies that there are no interaction behaviours among the majority of learners. The results of these experiments show that the number of interaction behaviours and interaction time follow a power law. As a result, the optimal interaction time length is selected by computing the interaction intensity of learners at different times. As shown in Fig. 5(b), the probability of interactions is high, with the interaction intensity reaching its peak on the third day. Based on these results, we set 3 days for the time lengths of comments and the interaction count, as well as for the decay time in our experiments.

5.5. Results and discussion

Multi-objective optimization effectiveness of LPRACNN: To examine the final solutions of our proposed algorithm by comparing with MOEALTR, we conducted experiments to balance three objectives simultaneously. Before reporting the comparative results with MOEALTR, the performance of the multi-objective algorithm is affected by some parameters in a way

Table 4
Parameter settings of the multi-objective algorithm.

Parameter	Meaning	Value
L	The length of the recommendation list	20
NP	The size of the population	150
gmax	The number of generations	500

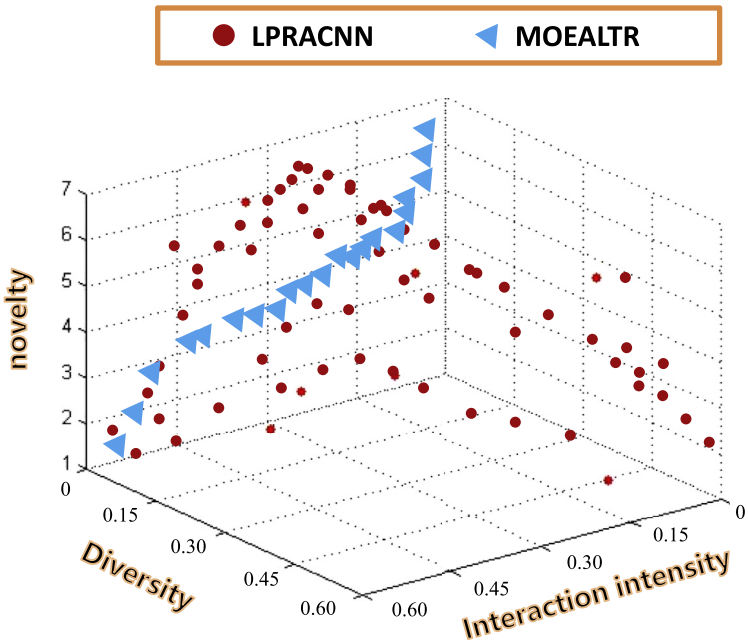


Fig. 6. A final non-dominated solution of LPRACNN and MOEALTR in the interaction intensity-novelty-diversity space.

that causes both a computationally expensive and time-consuming burden. Therefore, the values of the parameters are set as shown in Table 4. In addition, W^o are set as 0.2, 0.3, 0.5, respectively.

The results of LPRACNN and MOEALTR are plotted in Fig. 6. With the interaction intensity ranging from 0.55 to 0.29, LPRACNN obtains a decreasing novelty from 0.99 to 6.53. This proves that two conflicting objectives exist between interaction intensity and novelty. From the reported results, it can be observed that the solution of LPRACNN in three-dimensional space forms a curve, while MOEALTR forms a two-dimensional surface. Clearly, the results indicate the effectiveness of LPRACNN and the optimal trade-off among three objectives, showing that it is able to generate multiple recommendations with a higher diversity, interaction intensity and similar novelty.

As a result, we further give a detailed explanation of this phenomenon by selecting a target learner randomly in the dataset as an example. Fig. 7 details the recommendation results given to the learner. LPRACNN proposed in this paper and MOEALTR both suggest 30 recommendations to the learner. The interaction intensity, diversity and novelty of some representative recommendation lists are given. Compared with MOEALTR, LPRACNN is not as efficient in generating novel learners with a higher interaction intensity, obtaining an increasing interaction intensity from 0.25 to 0.88 with a decreasing novelty from 0.76 to 0.19. In addition to its tendency to suggest less popular learners, MOEALTR only includes learner (9) and learner (26) whose diversity is 0.21 and 0.19 respectively. It can be concluded that MOEALTR cannot balance the three objectives well in comparison with LPRACNN. In addition, one possible explanation is that our approach integrates the attention model based on learning content, which facilitates the achievement of learning objectives whereas it pays relatively less attention to learner preferences irrelevant to learning. Compared with MOEALTR, our algorithm reconciles the demands of the three objectives in the inevitable case of computational cost. This unavoidable drawback coincides with our assumption of the proposed model complexity. Therefore, the inputs of the proposed model are set to two learners with their DITG.

Dynamic recommendation performance: In order to compare the performance of LIC, DLIC and LPRACNN, Fig. 8 illustrates some of the results. At the beginning of learning, LIC, DLIC and LPRACNN achieve the same hit ratio of 0.37. At the end of the learning process, however, LIC without using weighted interaction behaviours remains almost flat, only obtaining an increase of 0.02. For a dynamic recommendation, where dynamic weights are used, the hit ratios of DLIC and LPRACNN improve by 0.14 and 0.25. This verifies the benefits of weighted interaction behaviours.

More importantly, a weight adjustment exerts a significant influence on recommendation performance. As shown in Fig. 8, LPRACNN achieves a remarkable and sharp improvement of 0.25 in the hit ratio, compared with 0.14 for DLIC. This

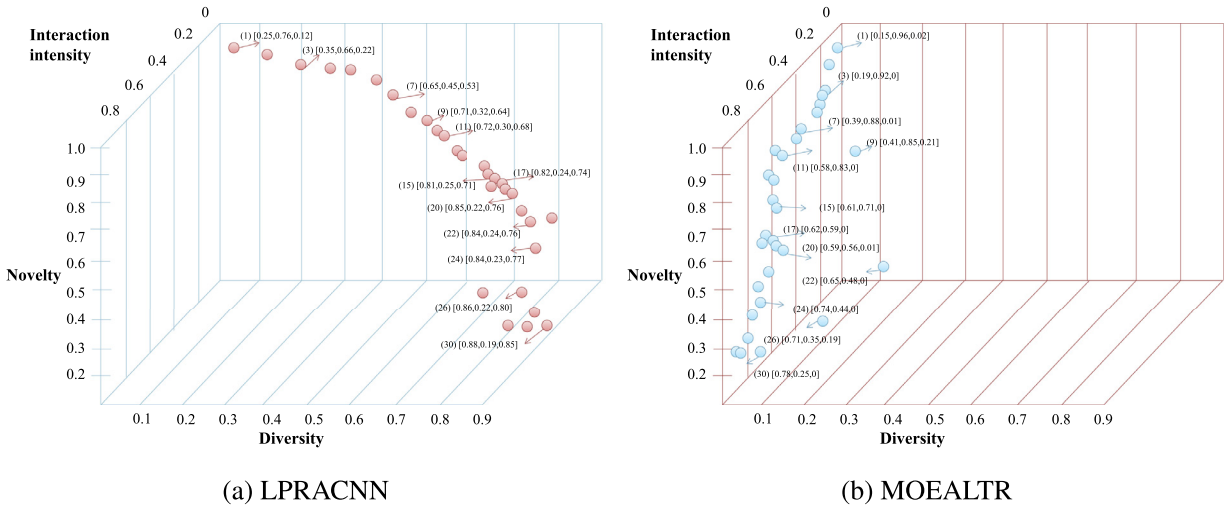


Fig. 7. Interaction Intensity, diversity and novelty of a recommendation to a given learner.

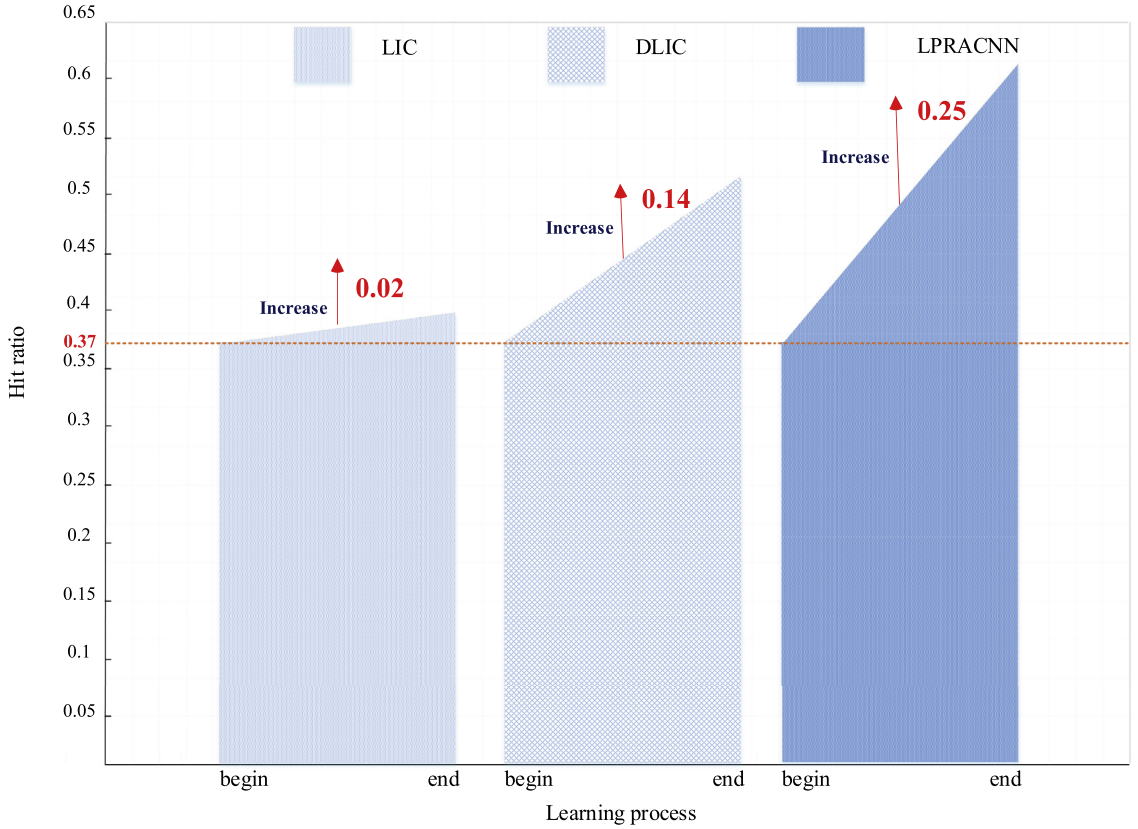


Fig. 8. Comparison of the results of the hit ratios of the two algorithms.

demonstrates that an attention-driven adjustment outperforms the manual settings significantly. The reason for this is that our proposed method is capable of adapting to the features of learning content by flexibly adjusting weights according to a current learning state. In contrast, the parameter values using the manual settings remain unchanged with respect to the different features. To summarize, the proposed LPRACNN is well suited to a dynamic learning process and achieves good performance.

Synthetic comparison results: Although these experiment results demonstrate the effectiveness of our algorithm, it is still necessary to consider the synthetic recommendation performance. In this regard, further experiments are conducted.

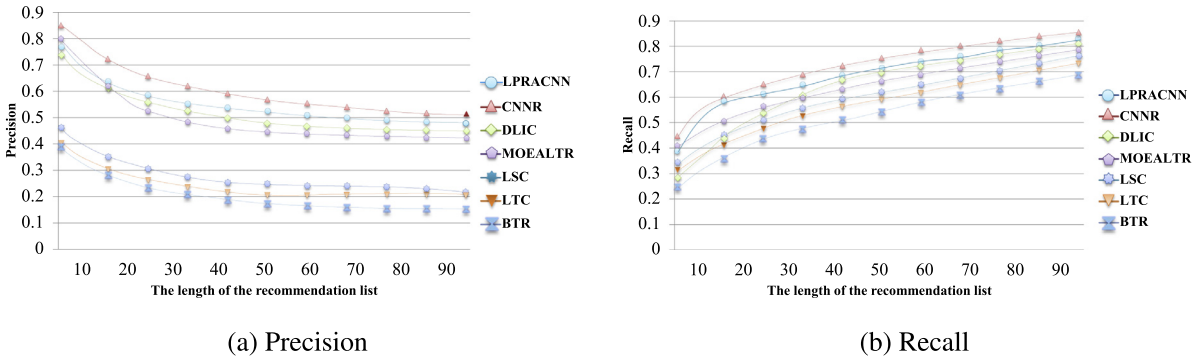


Fig. 9. Comparison of the results for accuracy for the seven algorithms.

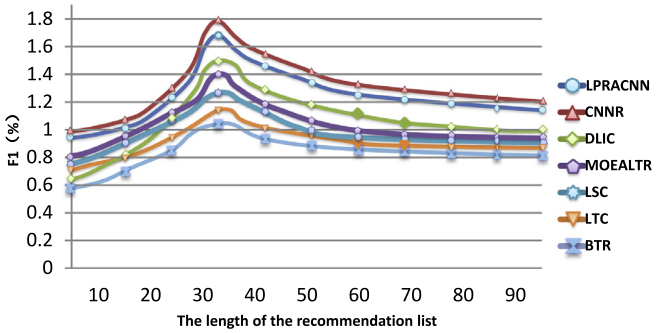


Fig. 10. Comparison of the results for F_1 for the seven algorithms.

We conducted comparative experiments against other baseline methods with respect to accuracy, diversity and novelty. Four kinds of experiments are conducted. First, the recommendation accuracy of the seven aforementioned algorithms are compared in terms of precision, recall and F_1. Then, the MAP of the seven algorithms is compared to further evaluate the effect of LPRACNN. Subsequently, the diversity of recommending learners is compared. Finally, the experiment results for novelty are compared to evaluate the performance of the algorithm.

The proposed LPRACNN in this paper is compared with BGR, LTC LTS, LIC, CNNR and MOEALTR for overall accuracy in terms of precision, recall and F_1. The experiment results of precision and recall are shown in Fig. 9.

Fig. 9 shows that with an increase in the length of the recommendation lists, the precision of these algorithms decreases, whereas recall increases reversely. Both gradually converge at a certain point. Fig. 9(a) and (b) show that the scores of BGR, LTC and LSC are lower than those of DLIC and LPRACNN. In terms of precision and recall, MOEALTR is superior to DLIC and LPRACNN, obtaining a precision and recall result of about 0.79 and 0.4, whereas DLIC and LPRACNN perform better than MOEALTR gradually. The reason for this is that LPRACNN is likely to add interaction behaviours with dynamic weights rather than static factors (e.g., tags, scores). Although it is evident that these results indicate that LPRACNN has the advantage of recommending learners accurately for a target learner, LPRACNN does not embody notable advantages when the length of the recommendation list is over 80. F_1 further indicates the performance of LPRACNN.

Fig. 10 depicts the F_1 score curves. It is clear to see that CNNR and LPRACNN perform better than the other algorithms when the length of a recommendation list ranges from 20 to 60. In particular, they perform best when the length of the list is 30. Note that the performance advantage of CNNR and LPRACNN is gradually lost with the increasing length of the list. These results for accuracy demonstrate that our proposed LPRACNN and CNNR outperform the other algorithms. Specifically, the experiment results suggest that the length of a recommendation list produced by our algorithms should be set to 30 for a target learner.

In addition to ensuring recommendation accuracy, there is a requirement to rank the learning peers which are recommended. The results are shown in Fig. 11. We can observe that LPRACNN has the best performance with a MAP of 95.6%, and ranks the optimal learners in the top ranks of the recommendation list with a better recommendation quality and reduces the selection time of learners. Thus, it is concluded that MAP can compensate for LPRACNN's deficiency in having slightly lower accuracy than CNNR.

To further evaluate the performance of recommendation diversity, we conduct experiments to compare the S values. As shown in Fig. 12, it is clear that the S values of the seven algorithms decrease with an increase in the length of the list. In the end, however, LPRACNN achieves better diversity of 0.48. Initially, it can be clearly observed that the S values of LTC and LSC exceed those of DLIC and LPRACNN. As the learning progresses, LPRACNN is superior to the other algorithms in

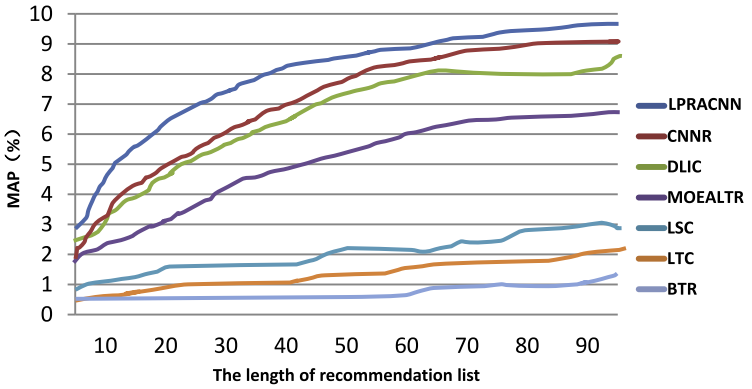


Fig. 11. Comparison of the results for MAP for the seven algorithms.

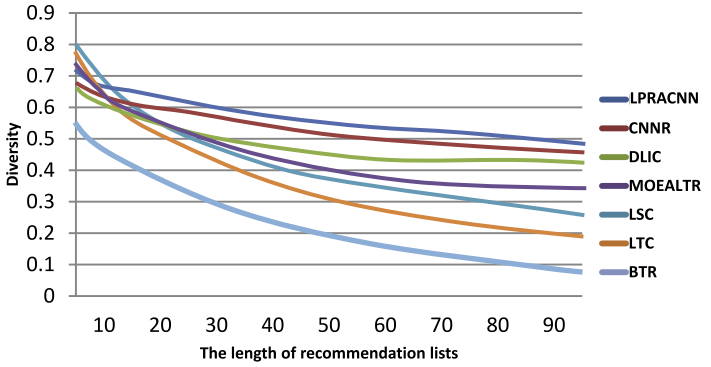


Fig. 12. Comparison of the results for diversity for the seven algorithms.

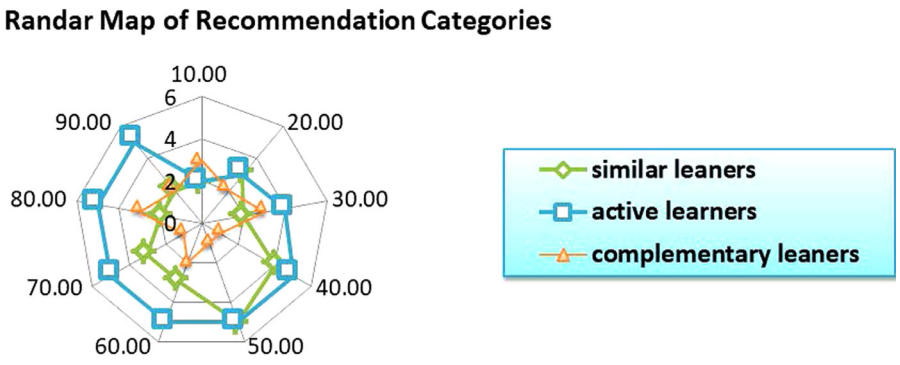


Fig. 13. The results of recommendation categories.

terms of diversity. This may be because online learning systems have a cold start problem, i.e., lacking the full details of learners at the beginning of learning. The learners recommended tend to be more diverse and have more various interaction behaviours with dynamic weights adjusted by learning content-centred attention-driven networks. Hence, we conduct additional experiments to clarify the categories of learners.

In this work, learning peers are recommended according to the attributes of the learning content, including basic and hot content. Learning content comprise five features: text, image, audio, video and animation. Therefore, learners can be categorized into similar learners, active learners, and complementary learners, according to the features of the learning content. For instance, if a target learner is learning from basic learning content, then learning peers who also learn from basic learning content could be recommended to the target learner by further considering the features of the learning content.

More specifically, a target learner is selected randomly to investigate the categories of the learners recommended. From Fig. 13, it can be noted that with an increase in the length of the recommendation list and the number of active learners, more categories should be taken into consideration. Therefore, for hot content, the categories of learning content increase

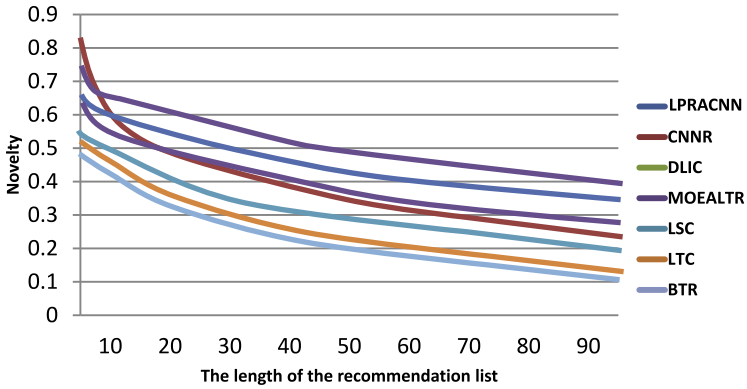


Fig. 14. Comparison of the results for novelty for seven algorithms.

with eventually closing to or equaling five as shown in Fig. 13. If all of learners are similar types, a recommendation will not focus on the categories of learning content too much, rather, the mastery of basic content is emphasized instead. So the categories of learning content are not given priority. This will be determined by the preferences of learners. As shown in Figs. 12 and 13, this is consistent with the recommendation mechanism of diverse learners proposed in this paper, demonstrating the effectiveness of LPRACNN once again.

Analogously, novelty is an important factor to experimentally validate the performance of LPRACNN. Fig. 14 plots the detailed results of the seven algorithms. We can observe that at first CNNR is superior to the other algorithms, and soon DLIC and LPRACNN perform better compared to CNNR, which provides additional evidence to support the integration of interaction behaviours into learner recommendation with dynamic weights.

From the experiment results on the performance of multi-objective optimization and dynamic recommendation and a comparison of the overall results, we conclude that our proposed LPRACNN can provide a trade-off solution with more diverse yet accurate and novel recommendations by integrating interaction behaviours dynamically. As a result, it helps to provide the learners with more accurate and personalized learning guidance so as to better adapt to online learning environments.

6. Conclusion

As online learning environments are popular nowadays, learning peer recommendation is becoming important. In this paper, we presented a novel framework for recommending learning peers using a tripartite graph and CNN. Our framework takes into account dynamic interaction behaviours and an attention mechanism. From the perspective of educational design, we integrated five interaction behaviours with corresponding weights into ternary relations in a dynamic interaction tripartite graph. A deep CNN model was proposed to infer the underlying interaction intensity among learners so as to recommend optimal learners. For the dynamic learning process, an attention-driven mechanism was employed to learn the weights of the interactions, according to the features of the learning content. Finally, we presented a multi-objective function as the objective function of the recommendation model in order to optimize interaction intensity, diversity, and novelty simultaneously. The experiment results show that our approach achieved superior performance compared with several state-of-the-art approaches.

This work brings new perspectives for future explorations. The performance of multi-dimensional recommendation based on DITG will be further enhanced by blending interaction behaviours with multi-modal educational information, such as voice interaction, facial expression and posture movement.

Acknowledgments

This work was supported by the Major Project of the National Social Science Fund of China (No. 18ZDA334), the National Natural Science Foundation of China (No. 61877020 and 61370178), and the S&T Project of Guangdong Province (No. 2015A030401087).

References

[1] T. Alashkar, S. Jiang, S. Wang, Y. Fu, Examples-rules guided deep neural network for makeup recommendation, in: Proceedings of the 31th AAAI Conference on Artificial Intelligence, 2017, pp. 941–947.
[2] A. Bellogin, I. Cantador, P. Castells, A comparative study of heterogeneous item recommendations in social systems, Inf. Sci. 221 (2013) 142–169.
[3] S. Bhagwani, S. Satapathy, H. Karnick, Semantic textual similarity using maximal weighted bipartite graph matching, in: Proceedings of the First Joint Conference on Lexical and Computational Semantics, Association for Computational Linguistics, 2012, pp. 579–585.
[4] J. Bobadilla, A. Gutirrez, F. Ortega, B. Zhu, Reliability quality measures for recommender systems, Inf. Sci. 442 (2018) 145–157.

[5] C. Castro, A. Braga, Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data, IEEE Trans. Neural Netw. Learn. Syst. 24 (6) (2013) 888–899.
[6] S. Cavallari, V. Zheng, H. Cai, K. Chang, E. Cambria, Learning community embedding with community detection and node embedding on graphs, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, ACM, 2017, pp. 377–386.
[7] J. Chen, X. Song, L. Nie, X. Wang, H. Zhang, T. Chua, Micro tells macro: predicting the popularity of micro-videos via a transductive model, in: Proceedings of the 2016 ACM on Multimedia Conference, ACM, 2016, pp. 898–907.
[8] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, T. Chua, Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2017, pp. 335–344.
[9] A. Das, H. Agrawal, L. Zitnick, D. Parikh, D. Batra, Human attention in visual question answering: do humans and deep networks look at the same regions? Comput. Vis. Image Understanding 163 (2017) 90–100.
[10] M.K.K. Devi, P. Venkatesh, Smoothing approach to alleviate the meager rating problem in collaborative recommender systems, Future Gener. Comput. Syst. 29 (1) (2013) 262–270.
[11] X. Gao, T.T. Mu, J.Y. Goulermas, M. Wang, Attention driven multi-modal similarity learning, Inf. Sci. 432 (2017) 530–542.
[12] A. Gogna, A. Majumdar, Diablo: optimization based design for improving diversity in recommender system, Inf. Sci. 378 (2017) 59–74.
[13] Y. Gong, Q. Zhang, Hashtag recommendation using attention-based convolutional neural network., in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI), 2016, pp. 2782–2788.
[14] Z. Huang, H. Chen, D. Zeng, Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering, ACM Trans. Inf. Syst. 22 (1) (2004) 116–142.
[15] M. Jahrer, A. Töscher, R. Legenstein, Combining predictions for accurate recommender systems, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2010, pp. 693–702.
[16] T. Joerding, A temporary user modeling approach for adaptive shopping on the web, in: Proceedings of Second Workshop on Adaptive Systems and User Modeling on the World Wide Web, Toronto and Banff, Canada. Computer Science Report, 1999. 99–07.
[17] C. Kent, E. Laslo, S. Rafaeli, Interactivity in online discussions and learning outcomes, Comput. Educ. 97 (2016) 116–128.
[18] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–445.
[19] D. Li, B. Chen, Technology-enhanced business english writing teaching, Int. J. Inf. Educ. Technol. 7 (9) (2017) 700–706.
[20] Q. Liu, E. Chen, H. Xiong, C. Ding, J. Chen, Enhancing collaborative filtering by user interest expansion via personalized ranking, IEEE Trans. Syst. Man Cybern. Part B (Cybernetics) 42 (1) (2012) 218–233.
[21] J.P. Lucas, N. Luz, M. Moreno, R. Anacleto, A.A. Figueiredo, C. Martins, A hybrid recommendation approach for a tourism system, Expert Syst. Appl. 40 (9) (2013) 3532–3550.
[22] N. Manouselis, R. Vuorikari, F. Van Assche, Collaborative recommendation of e-learning resources: an experimental investigation, J. Comput. Assist. Learn. 26 (4) (2010) 227–242.
[23] S. Min, I. Han, Detection of the customer time-variant pattern for improving recommender systems, Expert Syst. Appl. 28 (2) (2005) 189–199.
[24] J. Qi, An optimization approach for flipped classroom and multimedia teaching innovation by contrast with traditional teaching, Boletín Técnico 55 (8) (2017) 480–488.
[25] X. Qian, H. Feng, G. Zhao, T. Mei, Personalized recommendation combining user interest and social circle, IEEE Trans. Knowl. Data Eng. 26 (7) (2014) 1763–1777.
[26] M. Ramezani, Improving graph-based approaches for personalized tag recommendation, J. Emerg. Technol. Web Intell. 3 (2) (2011) 168–176.
[27] M. Rawashdeh, H. Kim, J. Aljaam, A. El Saddik, Folksonomy link prediction based on a tripartite graph for tag recommendation, J. Intell. Inf. Syst. 40 (2) (2013) 307–325.
[28] B. Schmitz, B.S. Wiese, New perspectives for the evaluation of training sessions in self-regulated learning: time-series analyses of diary data, Contemp. Educ. Psychol. 31 (1) (2006) 64–96.
[29] T. Shamir-Inbal, I. Blau, Developing digital wisdom by students and teachers: the impact of integrating tablet computers on learning and pedagogy in an elementary school, J. Educ. Comput. Res. 54 (7) (2016) 967–996.
[30] X. Shen, B. Yi, Z. Zhang, J. Shu, H. Liu, Automatic recommendation technology for learning resources with convolutional neural network, in: Proceedings of 2016 International Symposium on Educational Technology (ISET), IEEE, 2016, pp. 30–34.
[31] J. Shu, X. Shen, H. Liu, B. Yi, Z. Zhang, A content-based recommendation algorithm for learning resources, Multimedia Syst. 24 (2) (2018) 163–173.
[32] J. Song, L. Gao, F. Nie, H. Shen, Y. Yan, N. Sebe, Optimized graph learning using partial tags and multiple features for image and video annotation, IEEE Trans. Image Process. 25 (11) (2016) 4999–5011.
[33] L. Tern, A.O. Mensah, A. Estorelli, A literature review for recommender systems techniques used in microblogs, Expert Syst. Appl. 103 (2018) 63–73.
[34] Y. Ustunbas, S.G. Oguducu, A recommendation model for social resource sharing systems based on tripartite graph clustering, in: Intelligence and Security Informatics Conference, 2011, pp. 378–381.
[35] A. Van Leeuwen, J. Janssen, G. Erkens, M. Brekelmans, Supporting teachers in guiding collaborating students: effects of learning analytics in CSCL, Comput. Educ. 79 (2014) 28–39.
[36] S. Wan, Z. Niu, An e-learning recommendation approach based on the self-organization of learning resource, Knowl. Based Syst. 160 (2018) 71–87.
[37] D. Wang, M. Li, Stochastic configuration networks: fundamentals and algorithms, IEEE Trans. Cybern. 47 (10) (2017) 3466–3479.
[38] S. Wang, M. Gong, H. Li, J. Yang, Multi-objective optimization for long tail recommendation, Knowl. Based Syst. 104 (2016) 145–155.
[39] X. Wang, L. Yu, K. Ren, G. Tao, W. Zhang, Y. Yu, J. Wang, Dynamic attention deep model for article recommendation by learning human editors' demonstration, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2017, pp. 2051–2059.
[40] J. Wei, J. He, K. Chen, Y. Zhou, Z. Tang, Collaborative filtering and deep learning based recommendation system for cold start items, Expert Syst. Appl. 69 (2017) 29–39.
[41] J. Wu, L. Chen, Q. Yu, P. Han, Z. Wu, Trust-aware media recommendation in heterogeneous social networks, World Wide Web 18 (1) (2015) 139–157.
[42] F. Yang, F. Li, Study on student performance estimation, student progress analysis, and student potential prediction based on data mining, Comput. Educ. 123 (2018) 97–108.
[43] S. Ye, F. Wu, Measuring message propagation and social influence on twitter. com, Int. J. Commun. Netw. Distrib. Syst. 11 (1) (2013) 59–76.
[44] Y.W. Zhang, R. Wang, T.W. Wang, G.D. He, J. Hu, An asymmetric graph recommendation algorithm based on user scores and interests, IOP Conf. Ser. 231 (1) (2017) 012045.
[45] T. Zhou, J. Ren, M. Medo, Y. Zhang, Bipartite network projection and personal recommendation, Phys. Rev. E 76 (4) (2007) 1–7.
[46] Y. Zhou, C. Huang, Q. Hu, J. Zhu, Y. Tang, Personalized learning full-path recommendation model based on LSTM neural networks, Inf. Sci. 444 (2018) 135–152.