Knowledge-Based Systems 254 (2022) 109551

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

A dual learning-based recommendation approach

Shanshan Wan [a,b], Ying Liu [a], Dongwei Qiu [a], James Chambua [c], Zhendong Niu [d,e,*]

[a] Beijing University of Civil Engineering and Architecture, Beijing 100044, China
[b] Beijing Key Laboratory of Intelligent Processing for Building Big Data, Beijing 100044, China
[c] Computer Science and Engineering Department, University of Dar-es-Salaam, Dar es Salaam 35091, Tanzania
[d] Beijing Institute of Technology, Beijing 10081, China
[e] School of Computing and Information, University of Pittsburgh, Pittsburgh 15260, USA

ARTICLE INFO

Article history:
Received 17 March 2021
Received in revised form 20 July 2022
Accepted 22 July 2022
Available online 30 July 2022

Keywords:
Recommender system
Dual learning
Data sparsity
Duality
Hybrid filtering recommendation

ABSTRACT

Data sparsity and cold start are two critical issues which need to be addressed in recommender systems (RSs). Currently, most methods address these issues by applying user history files or some side information to improve the user model and complete the rating matrix. However, such methods cannot perform well when labeled data is scarce or unavailable. In this paper, we propose a dual learning-based recommendation approach (DLRA). DLRA can trigger initial recommendation and improve the quality of recommendations by using the duality characteristics of RSs, even when the available labeled information is scarce. Specifically, DLRA regards the recommendation task as two independent subtasks — primal task and dual task, and these two tasks show strong duality in DLRA. The primal task is item-centered which aims to find users who can rate high for items, while the dual task is user-centered that aims to recommend the most favorite items to users. These two tasks have strong dualities in terms of the recommendation space, selection probability and recommendation basis. Based on these dualities, we design three dual learning strategies to couple the whole recommendation process and realize the self-tuning and self-improvement of each task model, and finally optimize the whole recommendation model. Based on the dataset of Movielens and BookCrossing, we simulate data sparsity and cold start recommendation scenarios, the experimental results show that DLRA achieves substantial improvement when the labeled data is scare, and it outperforms other hybrid recommendation approaches and deep learning strategies with a smaller predictive error as well as better recommendation accuracy.

1. Introduction

With the explosive growth of Internet resources, particularly commodities and entertainment resources, more and more people turn to the Internet to search for items they need. Having an efficient and accurate recommender system (RS) has become an important requirement for satisfying user's personalized needs and experience. For example, 80% of movies watched on Netflix came from recommendations [1], and 60% of video clicks came from home page recommendation in YouTube [2]. RS provides the possibility of returning items which satisfy users' personalized needs [3]. Most recommender systems (RSs) heavily depend on history experience of active users or other users' evaluations to generate recommendations. Content-based filtering (CBF), collaborative filtering (CF) and hybrid filtering (HF) are the common recommendation methods for filtering items. CBF recommends items which are similar to the items that users like in the past [4]. One major issue of CBF approach is that it relies on a substantial number of item features and user's history files. CF recommender system aims to recommend items according to some other users who are similar to active users. User–item rating matrix is the basic criterion for calculating user similarity or item similarity for both CBF and CF methods [5]. It is obvious that these two approaches suffer from cold start and data sparsity problems. HF recommendation approach utilizes user's history information or context information to complete the rating matrix to reduce the risks of data sparsity [6–8]. However, some fields suffer from extreme lack of available labeled data due to the shortage of prior data, difficulties in data collection or low reliability of data [9]. Hence, the side information and user history files are not always available in RSs.

Deep learning-based approaches have been proved promising in extracting content features of items users' social relationships to optimize recommendation strategies [10–13]. However, deep learning-based RSs often face with some common plights. Firstly, the process of training deep learning-based method is kind of black-box, hence, it lacks interpretability and modifiability, and

* Corresponding author.
E-mail address: zniu@bit.edu.cn (Z. Niu).

https://doi.org/10.1016/j.knosys.2022.109551
0950-7051/

it weakens the possibility of using some inherent features of recommendation scenarios. Secondly, deep learning is heavily dependent on big data and labeled data, which limits its application on some RSs. Thirdly, deep learning has high requirements for hardware and it usually takes long training time. Fourthly, it may take a lot of time to deploy and adjust a deep learning model, and the specific effect of a deep learning model cannot be guaranteed. Therefore, the question of how to implement an efficient and effective RS under the situation of few labeled data is still a hot topic, which motivates the development of non-deep learning recommendation techniques.

In this study, we propose a dual learning-based recommendation approach (DLRA) which is expected to show less dependence on labeled data. Dual learning is a new learning framework that leverages the symmetric structure of tasks to obtain effective feedback or regularization signals to enhance the learning process. The premise of applying dual learning theory is that the task itself shows duality, that is, the input of one task is (or can be converted to) the output of another task, and vice versa. Two dual tasks can construct a natural closed loop and form a learning mechanism through effective feedback. Therefore, with this closed-loop structure, the learning task can be realized with less labeled data.

The importance of duality has been proved and magnified in many fields, such as translation from one language to another versus its opposite direction, speech recognition versus speech synthetization, image classification versus image generation, etc [14,15], as well as vehicle re-identification and vessel-bridge collisions [16,17].

Several reasons why dual learning can be applied to RSs are explained as follows:

(1) RSs show symmetric structure. Since RSs mainly focus on the process of matching users and items, a RS can be divided into two tasks. The primal task is a CBF based process, and it is a task of matching items with users who can rate items high. The dual task is a CF based process, and it is a task of matching users with their favorite items. Obviously, the output of primal task provides the possibility of calculating user similarity and item similarity, and these similarities can be used as the input of dual task. The output of dual task provides users' access records and preferences for items, and these similarities can be used as the input of primal task.

(2) The structural duality of RSs implies the strong duality connections between primal and dual tasks. Specifically, three kinds of duality are considered in this study.

- The duality of recommendation space. In primal and dual task of RSs, item-based recommendation space and user clustering-based recommendation space will be generated. These two spaces should be consistent and they can be verified each other. The gap of recommendation spaces is an important feedback signal to realize the closed loop.
- The duality of selection probability. The primal task and dual task in RSs share the same items and users. Therefore, the probability of mutual selection between items and users should be consistent. We take the selection probability as one of the duality characteristics of RSs. Probabilistic nature can strengthen the dual learning process through structural regularization and improve the accuracy of the recommendation model.
- The duality of recommendation basis completion. The user history access data in the primal task and the user rating matrix in the dual task are the recommendation bases for RSs. In fact, these two bases also show duality, that is, the potential rating of users on the items can be obtained from users' history access to items, and the possibility of users' access to the project can be predicted from the user rating matrix. Therefore, even in the face of zero or very few labeled data, the two tasks learn from each other to realize data completion and further complete the recommendation.

The aforementioned duality strategies are implemented through the mutual interactive and continuous trial and error feedback mechanisms in the primal and dual tasks. Consequently, the whole recommendation model achieves the potential of self-improvement and self-tuning. Thus, RSs can work effectively even when there is only few or zero history data. Accordingly, the adaptability and effectiveness of RSs are bound to be boosted.

The main contributions of the paper include, (1) Introducing a new recommendation model DLRA. DLRA is capable of reducing the dependence on labeled data in RSs by forming a closed loop between primal and dual tasks. Hence, DLRA shows promising ability to leverage unlabeled data and structural duality to address big-data challenge faced by CBF and CF approach. (2) Simulating users and items in RSs as entities that can actively initiate search tasks. The task of items searching users and the task of users searching items are exactly presented in dual forms. This provides a solid foundation for applying dual learning theory to such bi-directional and simultaneous dual tasks. (3) Designing the duality strategies of recommendation space, selection probability and recommendation basis. These strategies ensure the validity of the recommendation results and guarantee that the dual learning-based recommendation approach can effectively reduce the dependence on labeled data.

The remainder of the paper is organized as follows: Section 2 describes some studies related to hybrid recommendation approach and dual learning theory. Section 3 specifies the proposed dual learning-based recommendation approach. Section 4 presents the experiment datasets, comparison approach and the evaluation methods. Section 5 gives the experimental results for verifying the performance of the proposed approach. Finally, Section 6 concludes the study and introduces the future work.

2. Related works

This section presents a review of existing literatures which focus on how to alleviate the effect of data sparsity and cold start problems. Then, the advantages of dual learning theory are detailed through comparison with HF and deep learning-based approaches.

2.1. Hybrid recommendation approach

Hybrid recommendation techniques combine two or more recommendation techniques to improve the recommendation performance. The motivation for constructing hybrid recommender systems is that different types of RSs have different strengths and weaknesses, i.e. some work more effectively in cold start scenarios, whereas others work more effectively when sufficient data are available [18].

CBF recommender systems show the potentiality of addressing data sparsity issues, hence, they excel CF recommender systems in scenarios where there are few users or difficulties to obtain the item rating or comments [19–21]. For example, many CBF recommender systems have been applied on e-learning recommendation field because learners hate to evaluate learning resources or they are reluctant to post their feedback online due to limited time or individual personality [22,23]. Compared with CBF recommendation approach, CF is more widely applied because it can utilize other users' profiles to help active users, and the recommendations are more personalized [24]. A variety of learning techniques such as neural networks, dimensionality reduction, Bayesian networks and matrix and tensor factorization, are used to get more accurate explicit or implicit user correlation in CF [25–27]. The other effective way to suppress data sparsity is context-aware-based recommendation technique, which tries to find more valuable user relationships by extending user model or user auxiliary information, such as user personality or user background [28–30].

Hybrid recommender systems attempt to leverage the complementary strengths of source systems to create a system with greater robustness [31]. Apart from using both side information and social network, Zhou et al. tried to obtain user profiles through an additional interview process [32]. Wang and Blei proposed collaborative topic modeling (CTR) which applied topic model and latent Dirichlet allocation (LDA) to learn item content feature [33]. Zhang et al. introduced the concepts of popular items and frequent rater to identify the rating sources for recommendations [9].

In terms of that couple of algorithms are presented and combined in different way, Bobadilla et al. divided hybrid recommender systems into weighted, switching, mixed, feature combination, cascade, and feature augmentation etc. [8].

It is obvious that hybrid recommendation methods can be summarized into two categories: one is based on feature fusion from both CBF and CF, and the other is based on the combination of different algorithms. Although HF approaches have made great progress in addressing data sparsity and cold-start problems, the dual characteristic of recommendation scenarios has not been considered yet by HF recommendation approaches.

The distinctions between DLRA and other common hybrid approaches include, (1) The fusion mechanism of most HF methods is not suitable to define DLRA fusion. DLRA neither optimizes recommendation results of source algorithms nor switches/cascades the source algorithms. It also does not simply use one source algorithm to enhance the features of another. DLRA deeply couples different algorithms to optimize the recommendation strategies of each algorithm, and consequently achieves more accurate recommendation. (2) The initial conditions of the traditional hybrid filtering often use one recommendation approach to make up for the data sparsity or cold start of another recommendation approach, but the strategy must rely on a certain amount of available data. Dual learning mechanism makes it possible to realize recommendations when the data is extremely sparse or absent. (3) In terms of the control strategy of hybrid filtering, the traditional HF methods mostly present the characteristics of open-loop while DLRA is a whole closed-loop recommendation approach, and the algorithm parameters will experience a process from large-scale optimization to fine-tuning optimization. Therefore, the proposal of DLRA shows novelty in HF and it also provides a guiding research direction for hybrid filtering.

2.2. Deep learning-based recommendation approach

Currently, deep learning is widely used in RSs because it can learn a hierarchy of features from low level to high level. With the advantage of modeling different types of data, deep learning-based recommender systems can better understand users' demands, hence, they are able to improve the quality of recommendations [10]. Various deep learning methods have been applied to recommender systems, such as Restricted Boltzmann Machines (RBM), Neural Collaborative Filtering and Deep Collaborative Neural Networks [34].

R et al. described a deep learning-based HF recommender system which applied embeddings to represent users and items to learn non-linear latent factors [35]. The solution alleviates the cold start problem by integrating side information about users and items into a very deep neural network. Wang et al. proposed a deep belief network and probabilistic graphical model to simultaneously learn features from audio content and make personalized recommendations, and they proved that RBM-based recommendation model outperforms Matrix Factorization [36]. Cheng et al. proposed a wide and deep learning model for recommender systems [37], in which the wide component learns the correlation among features of items and users by utilizing history data. The deep component learns the unseen (latent) features among user–item interactions. This method works very well when there are rich feature sets available for users and items. Zhu et al. proposed a course recommender system by utilizing graph neural networks to describe students, courses, and other entities. Some side information are applied in this recommender system, such as students' rating, commentary text, grading and interpersonal relations [38].

Attempts have been made to solve the cold start problem by using deep learning. Jia et al. proposed an collaborative RBM method to make social event recommendation through pictures and posts from friends on social networks [39]. In [40], the preference relations of items and the side information are integrated into an collaborative model. Meanwhile, the second order and higher order user–item interactions can be captured with the proposed method. Yang et al. proposed a cross-modal video recommendation algorithm based on multi-modal deep learning [41]. RBM was used to model the relationship between each modality feature on high-level semantic features, so as to obtain a joint shared representation layer. As Wei et al. concluded, several recent deep learning-based studies attempted to enrich user profiles with information from other channels, such as social trust network, tagging system and interview process [42]. But such information is not always available, moreover, it is usually difficult to acquire personal information of new users because of privacy issue. In order to alleviate the information scarcity of cold start items, most research efforts so far have been devoted to profiling new items with additional information (e.g., collecting item attributes). However, it is always time-consuming and costly to gather fine-grained attributes like tags, keywords and categories.

2.3. Dual learning theory and its application

He et al. focused on the fact that human labeling in neural machine translation is very costly, and then they proposed dual learning method to tackle the training data bottleneck [43,44]. Specifically, the dual learning mechanism is inspired by the observation that a machine translation task has a dual task, that is, the translations between two languages from opposite direction are presented in dual forms. The primal and dual tasks can form a closed-loop and generate informative feedback signals to train the translation models, even without the involvement of a human labeler. These models can teach each other through feedback signals generated during the reinforcement learning process.

The high-level idea of dual learning is very intuitive: if we map an x from one domain to another and then map it back, we should recover the original x. This is the basic idea behind dual learning. In dual learning mechanism, one agent is designed to represent the model of primal task and the other agent is used to represent the model of the dual task, then duality strategies are employed to complete self-tuning and self-improvement of these two tasks through a reinforcement learning process. Based on the feedback information generated during this process (e.g., the language model likelihood of the output of a model, and the reconstruction error of the original sentence after the primal and dual translations), two models can be iteratively updated until convergence (e.g., using the policy gradient methods). Qin et al. provided the specification and implementation of the dual learning theory. Specifically, they presented theory work based on the joint probability principle [45]. Zhao et al. conducted a theoretical study to prove why and when dual learning can improve a mapping function [14].

Machine translation is the most popular and mature field of applying dual learning theory. The two-way directions of translation are classical dual tasks which can be trained jointly with mutual feedback. In [46], authors transferred the knowledge contained in the dual model to boost the training of the primal model. Dual learning also shows promising performance on the fields of image generation and image classification, because image generation and image classification also show the characteristic of duality [47,48]. Xia et al. made further research by leveraging the duality task into the inference stage of artificial intelligence [49]. The authors proposed a general framework of dual inference which can take advantage of both existing models from the two dual tasks to conduct inference for individual tasks.

Luo et al. applied dual learning method on semantic image segmentation to reduce labeling efforts [50]. The semantic image segmentation is modeled as two complementary learning tasks that are jointly solved. One predicts label maps and tags from images, and the other reconstructs the images using predicted label maps. The method is proved to outperform other existing best-performing baselines by extensive experiments. To reduce the risk of unlabeled instances in machine learning approaches, Gan et al. proposed dual learning-based safe semi-supervised learning method [51]. First, the method utilized a primal model to classify each unlabeled instance, and then used a dual model to reconstruct the unlabeled instances according to the obtained classification results. The risk can be measured by analyzing the reconstruction error and predictions of the original and reconstructed unlabeled instances.

So far, only a few duality characteristics are studied and applied to recommender systems. Zhang et al. employed dual mechanism to alleviate the cold start problem in RSs by selecting elite and representative items [52]. The authors identified the main problem as user selection and item selection, and they assumed that users and items can be projected into a shared subspace with categories, then a dual discriminative selection framework for rating elicitation was proposed. They used category labels as guidance to indicate user preferences and item attributes, and they mainly aimed to address the cold start problem. Hence, the rating matrix was assumed to be sufficient. Moreover, the dual mechanism was used to verify that the categories that user prefers are consistent with the categories of items in the shared space. Wang et al. jointly performed deep representation learning for the content information and collaborative filtering for the ratings (feedback) matrix [34]. On one hand, rating information can guide the feature learning process. On the other hand, the extracted features can further improve the predictive power of the CF models through the interaction. The tightly coupled methods can automatically learn features from the side information and naturally balance the influence of ratings and side information. However, this model only works on implicit rating prediction problem, and it cannot learn latent representation effectively. Moreover, the duality of RSs is not involved in this model.

Sun et al. designed a unified dual framework for explainable recommender systems [53]. In which, the tasks of user preference prediction and review generation were presented in dual forms, and the authors modeled the probabilistic correlation between these two dual tasks to achieve accurate users' assessment of items. To tackle the problem caused by unavailable preference and review in the test stage, the authors proposed a transfer learning-based model to simulate the review representation of each user–item pair, and the simulation results could be approximated for preference prediction and review generation. The dual task in this study was used to optimize the user model for recommendation scenarios which have comment setting. When the extant information is unavailable, the recommendation performance depends heavily on the simulation method of review generation, which reduces the generality of the method. Li et al. proposed an approach to cross-domain recommendations based on the dual learning mechanism between two related domains in an iterative manner until the learning process stabilizes [54]. The authors developed a latent orthogonal mapping to extract user preferences over multiple domains while preserving relations between users across different latent spaces. This study is limited to cross-domain scenarios, and the issues of data sparsity and cold start are not considered. In [53,54], only probability/similarity feedback in the recommendation process is considered, the dual characteristics of recommendation basis and recommendation result are not applied to optimize the recommendation model. Hence, application of dual theory in RSs needs further research.

Another unsupervised learning framework which is similar to dual learning is generative adversarial networks (GANs). GANs contain two components which compete each other to generate realistic looking outputs. GANs have been applied on image generation [55], bibliographic citation recommendation [56] and question answering [15,57]. When GANs are used to generate images, one component (a generator) is trained to generate images, while another (a discriminator) is trained to distinguish real versus generated images. Thus, the generated images are trained to look realistic in the sense that they are indistinguishable from those in the dataset. The two components of GANs network compete to get the same task. Sufficient dataset is necessary to guarantee the effectiveness of GANs model, and neural network mechanism is often combined to build such models [58]. The big difference between GANs and dual learning model is that, GANs aim to handle the situation which has only one task, and dual learning can be used to address the problem which has two tasks, and these two tasks are dependent. Hence, dual learning method is more appropriate to satisfy our original intention of using duality to solve cold start and data sparsity in RSs.

3. Dual learning-based recommendation strategy

By analyzing the duality characteristics of two independent tasks in RSs, we propose a dual learning-based recommendation approach — DLRA. The basic idea, framework and implementation details of the DLRA are introduced in this section.

3.1. Basic idea of DLRA

Fig. 1 is a brief illustration of DLRA. As a new hybrid filtering recommendation approach based on dual learning theory, DLRA can carry out dual learning-based recommendation task based on zero or very few training data.

We take the situation of zero labeled data as an example to explain the implementation of DLRA. Zero labeled data means that there is no user's history access data in the primal CBF task, and there is no user's rating data on items in the dual CF task.

For the primal task, without the user history data, the selection probability of an item matching a user is initialized. R refers to the recommended item list which is achieved by CBF model f. User clusters generated based on R are denoted as U'. Since there are no labeled data, the quality of the recommended results cannot be evaluated. However, based on U', dual task can generate user's favorite item set R'. If the recommendation models f and g perform well, R and R' should show high similarity. Similarly, for dual task, even without rating data, the selection probability of a user selecting an item can be initialized and then user cluster U can be obtained based on CF model g. The item recommendation R' is obtained consequently. With the recommendation of R', the primal task will generate new cluster U'. Hence, by comparing

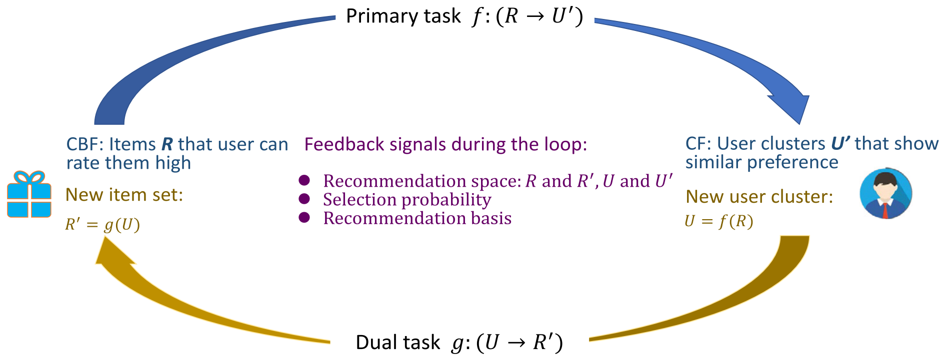

Fig. 1. Basic idea of DLRA.

the similarity of U and U', model f and model g can be trained and optimized.

Obviously, the primal task and dual task construct a closed-loop recommendation process. To be more specific, the mapping functions of f and g should be cycle-consistent to reduce the recommendation gap. As shown in Fig. 1, for the primal task, the recommendation cycle should be able to return an item r back to the same user u, that is, $r \rightarrow f(r) \rightarrow g(f(r)) \rightarrow r$. This cycle is called forward cycle consistency. It means if an item r is recommended to a user u in primal task, and the dual task g will recommend r to user u based on the user clusters. Similarly, $u \rightarrow g(u) \rightarrow f(g(u)) \rightarrow u$ represents a backward cycle consistency.

3.2. Framework of recommendation strategy

The framework in Fig. 2 describes the implementation process of two dual tasks in RSs and the connections between them. The top part of Fig. 2 is the schematic of the primal task and the lower part is the schematic of dual task. The primal task is to recommend users with appropriate items, and the task is modeled as $f(R, U'|UF; \theta_{ru'})$. UF refers to user preference. $\theta_{ru'}$ includes the parameters for model f. The recommendation process is to classify the candidate items according to user's preferences. The dual task is to recommend items to users, and this task is modeled as $g(U, R'|RU; \theta_{ur'})$. RU means the rating matrix of items. $\theta_{ur'}$ includes the parameters for model g. $R0$ is the candidate items and $U0$ is the user set.

The duality characteristics between primal task and dual task ensure that the recommender system can actively and effectively give recommendations by reducing the dependence on the labeled information. Finally, the recommendations can be achieved through the cooperation of these two tasks.

3.3. DLRA model

In this section, we elaborate how to design dual learning-based strategies by using the interaction between primary and dual tasks. Specifically, the duality strategies are based on recommendation space, selection probability and recommendation basis. The implications of three duality strategies of DLRA are shown in Fig. 3. These duality strategies cover the most important components in a recommender system, that is, what does a RS generate, how does a RS work and what is the available information in a RS. These three strategies can influence and verify each other, and finally form a closed-loop feedback. This interactive behavior is the basis to ensure that the recommender system can generate recommendations even when the available information is scarce.

The primal task aims to achieve recommendations through the space transition of $R0 \rightarrow R$ and $R \rightarrow U'$, and the space transition for dual task is $U0 \rightarrow U$ and $U \rightarrow R'$.

The details of the three duality strategies in Fig. 3 are described as follows.

(1) Duality of recommendation space. Space mapping between two tasks is the first important duality strategy of DLRA. Specifically, the space mapping includes $R \rightarrow R'$ and $U \rightarrow U'$. Such space mapping is not only the evaluation function of RSs, but also the main basis for generating feedback of user preference and rating matrix in two tasks. Space matching of recommendations can be regarded as a duality strategy from the perspective of recommendation results.

(2) Duality of selection probability. The second duality strategy of DLRA is based on the duality of selection probability of items and users in the two tasks. In the ideal case, the selection probability distribution in these two tasks should be the same. Probability matching can be regarded as a duality strategy from the perspective of recommendation process.

(3) Duality of recommendation basis. The third duality strategy is the completion of the recommendation basis in the two tasks. Recommendation basis refers to the user preference in the primary task and the rating information in the dual task. When a RS works, primal and dual tasks can gradually complete their recommendation basis through the feedback information from the interaction of two tasks. In this way, the recommendation quality of each task can be improved. The completion of recommendation information can be regarded as a duality strategy from the perspective of recommendation basis.

Based on the idea of maximizing the reward between primary and dual models, the mathematical theory of dual learning-based recommendation approach can be modeled as a parameter optimization of the recommendation approach.

For a primary task f, we denote R as the middle output. This middle output has an immediate reward $re_1 = g(R)$ based on the existing user history file, indicating how the output is consistent with the rating matrix RU in g. Then we use the log probability of recommendations recovered from R as the reward of the communication between two tasks. Mathematically, reward $re_2 = log(g(U, R'|R; \theta_{ur'}))$. We adopt a linear combination of these two rewards, $re = \alpha re_1 + (1 - \alpha)re_2$, where α is a hyper-parameter. Since the reward can be considered as a function of U, R, U', R', $\theta_{ur'}$ and $\theta_{ru'}$, we optimize the parameters in the models through policy gradient methods to maximize the reward [59,60].

The basic concept of policy gradient is that a feedback/reward is obtained when a certain action is taken. If the feedback is positive, we will adjust the model to increase the probability of taking the same action next time. If the feedback is negative, we need to update the model to reduce the probability of taking the same action.

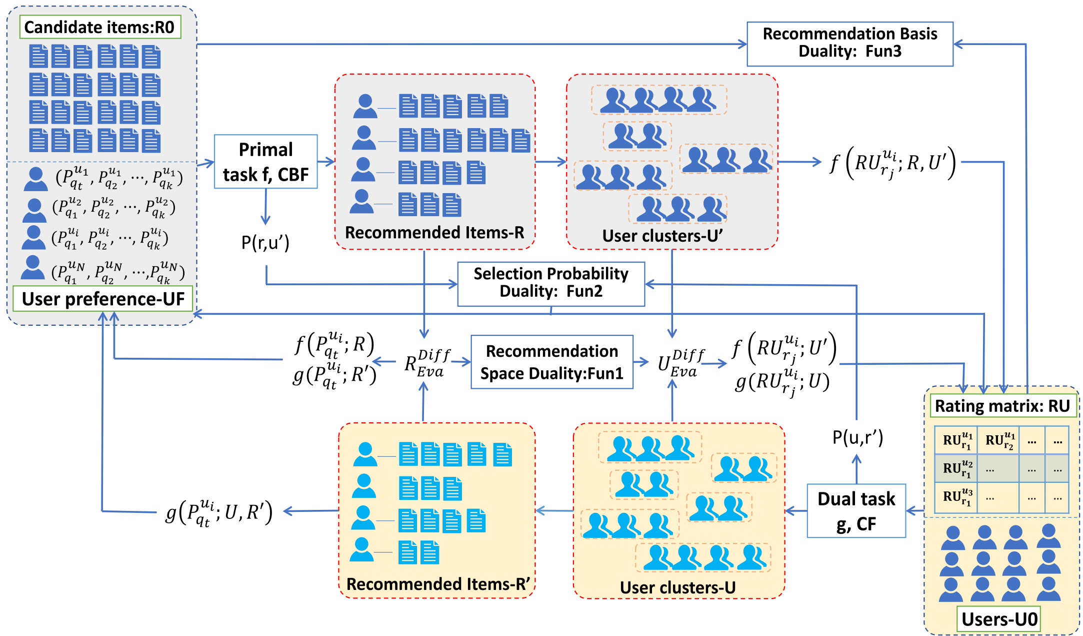

Fig. 2. Framework of DLRA.

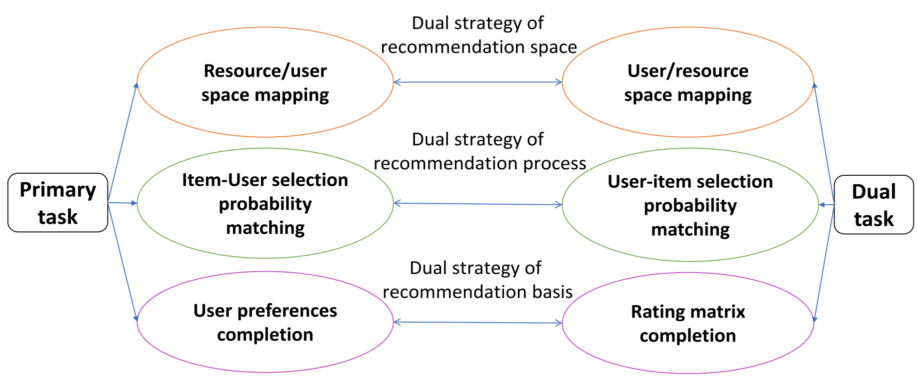

Fig. 3. Duality strategies of DLRA.

We sample R from the primal model f. Then we compute the gradient of the expected reward $E[re]$ with respect to parameters $\theta_{ur'}$ and $\theta_{ru'}$, it is easy to verify that,

$$\nabla_{\theta_{ru'}} E[re] = E[re \nabla_{\theta_{ru'}} log(f(R|R0;\theta_{ru'}))] \tag{1}$$

$$\nabla_{\theta_{ur'}} E[re] = E[(1-\alpha)\nabla_{\theta_{ur'}} log(g(R'|R;\theta_{ur'}))] \tag{2}$$

Similarly, if we sample U from the dual task g, the expected reward $E'[re]$ can be computed. And we have the following equations,

$$\nabla_{\theta_{ur'}} E'[re] = E[re \nabla_{\theta_{ur'}} log(g(U|U0;\theta_{ur'}))] \tag{3}$$

$$\nabla_{\theta_{ru'}} E'[re] = E[(1-\alpha)\nabla_{\theta_{ru'}} log(f(U'|U;\theta_{ru'}))] \tag{4}$$

To achieve the expected gradient, it is very important to analyze the reward functions between the two tasks. We design three duality strategies to represent the feedback signals which benefit the optimization of both f and g. The duality strategies include the recommendation space, selection probability of item and user recommendation space and the recommendation basis. Ideally, the three feedback signals between the two recommendation tasks are completely consistent and they interact in real time to ensure the model optimization through continuous feedback of comparison-trial-error. Hence, (1)∼ (4) can be regarded as a function of the loss function between f and g, and the loss function is obtained by analyzing the three duality strategies.

$$\begin{aligned}\nabla_{\theta_{ru'}\theta_{ur'}}(E[re]E'[re]) &= fun(f(R,U'|UF;\theta_{ru'}), g(U,R'|RU;\theta_{ur'})) + \\ &= fun(Fun1, Fun2, Fun3)\end{aligned} \tag{5}$$

In which, $Fun1$, $Fun2$ and $Fun3$ refer to the loss functions to recommendation space, selection probability and recommendation basis respectively.

The implementation of these duality strategies will be introduced in detail in Section 3.4.

3.4. Three duality strategies in DLRA

3.4.1. Duality strategy to minimize recommendation space

The first dual learning strategy of space mapping can be used as an objective function, which aims to minimize the matching differences of generated item spaces and user spaces in these two tasks. The difference between $R0$ and R is denoted as R_{Eva}^{Diff}, similarity, the difference between $U0$ and U is denoted as U_{Eva}^{Diff}. To be specific, the function is given as follows.

$$Fun1 = minimize(loss(R_{Eva}^{Diff}, U_{Eva}^{Diff})) \tag{6}$$

In which,

$$R_{Eva}^{Diff} = diff(\sum_{i}(Coverage(R_{u_i}, R'_{u_i}) + Top_y(R_{u_i}, R'_{u_i}) + Rank(R_{u_i}, R'_{u_i}) + Sim_R(R_{u_i}, R'_{u_i}))) \quad (7)$$

Where, $i \in [1, N]$, N is the number of the users. R_{u_i} and R'_{u_i} refers to the item sets which are recommended to user u_i in R and R' respectively, $R_{u_i}, R'_{u_i} \in R$. $Coverage(R_{u_i}, R'_{u_i})$ is used to evaluate the same items in R_{u_i} and R'_{u_i} for a same user. $Top_y(R_{u_i}, R'_{u_i})$ is used to compare the similarity of the top-y items recommended in these two tasks for a user, y is a parameter. $Rank(R_{u_i}, R'_{u_i})$ refers to the evaluation of item's sequence in R_{u_i} and R'_{u_i}. $Sim_R(R_{u_i}, R'_{u_i})$ measures the similarity of R_{u_i} and R'_{u_i} from the perspective of item features. The feature vector of items is the basis of similarity calculation.

U_{Eva}^{Diff} is defined as follows:

$$U_{Eva}^{Diff} = diff(\sum_{i}(Coverage_{clu}(U, U') + Relationship_{clu}(U, U') + Sim_{UF}(UC_{u_i}, UC'_{u_i}))) \quad (8)$$

In which, $Coverage_{clu}(U, U')$ is used to measure the similarity of clusters in U and U' by analyzing the composition of users in the clusters. $Relationship_{clu}(U, U')$ aims to compare the relationship strength between users in clusters. Relationship strength can be calculated according to user similarity in U' and ratings matrix in U firstly, then these two similarity results are normalized and compared. $Sim_{UF}(UC_{u_i}, UC'_{u_i})$ is designed to calculate the similarity of clusters in U and U' according to user preference UF. The calculation of similarity, relationship and coverage etc. refers to previous study [61].

3.4.2. Duality strategy to minimize the selection probability

Generally, the structural symmetry between primal task and dual task could be analyzed in a probabilistic view. If x and y represent the domain of primal and dual task respectively, the joint probability could be decomposed as $P(x, y) = P(x)P(y|x; f) = P(y)P(x|y; g)$. In which, $P(x)P(y|x; f)$ shows the primal view and $P(y)P(x|y; g)$ shows the dual view. Such decomposition implies the strong probabilistic connections between two tasks. The process of dual learning is to update model f and g, and finally maximize $P(y|x; f)$ and $P(x|y; g)$. Since both model f and model g aim to provide users with their favorite items, these two models should be trained simultaneously. Hence, a regularization term is added to minimize the gap between $P(y|x; f)$ and $P(x|y; g)$.

Based on the above mathematical analysis of dual learning, the second duality strategy is designed to minimize the difference of selection probability in these two tasks. The objective function is shown as follows:

$$Fun2 = minimize(loss(P(r, u'), P(u, r'))), \forall(r, u) \in R, U, \forall(r', u') \in R', U' \quad (9)$$

For primal task, selection probability $P(r, u')$ refers to the probability of an item matching a user based on $\theta_{ru'}$.

$$P(r, u') = P(r)P(u'|r; \theta_{ru'}) = P(u')P(r|u'; \theta_{u'r}) \quad (10)$$

$\theta_{ru'}$ and $\theta_{u'r}$ are parameters related to current user preference UF, UF gradient changing, similarity of items and the difference evaluation of U' and U.

Selection probability $P(u, r')$ refers to the probability of a user selecting an item based on $\theta_{ur'}$.

$$P(u, r') = P(u)P(r'|u; \theta_{ur'}) = P(r')P(u|r'; \theta_{r'u}) \quad (11)$$

$\theta_{ur'}$ and $\theta_{r'u}$ are parameters related to rating matrix, RU gradient changing, the current user clusters and the difference evaluation of R' and R.

Selection probability can be initialized randomly. During the process of RSs, the selection probability is updated in real time according to the comparison of recommendation results of the two tasks, which ensures that RSs can work effectively even if there are no labeled data for RSs. Therefore, the duality of selection probability provides the possibility of RSs being applied in the context of data sparsity or cold start scenarios.

3.4.3. Duality strategy to complete user preference and rating matrix

The third strategy is put forward to complete the user preference and rating matrix. User preference is specified as user's preference on a feature of items. Rating matrix is detailed as user's rating of items. We first explain the process of how to complete user preference and rating matrix, and then give the objective function, $Fun3$.

(1) User preference completion.

User preference can be represented as a feature vector, which is usually obtained by analyzing user's history access data.

Assumed the preference of a user u_i for an item is represented as $UF^{u_i} = (P_{q_1}^{u_i}, P_{q_2}^{u_i}, \ldots, P_{q_t}^{u_i}, \ldots, P_{q_k}^{u_i})$. q_t is the tth feature of the item, $P_{q_t}^{u_i}$ means the probability that user u_i prefers q_t.

Based on the policy gradient, the update equation of user preference in the primal task is given as (12). Firstly, the update of the user's preference for item features is determined by evaluating the feedback of the primal task and dual task in the item recommendation space. As well, the update of user preference also depends on the rating feedback of the user clusters generated in the dual task.

With the feedback of R and U from f and g respectively, $P_{q_j}^{u_i}$ is updated as follows:

$$P_{q_t}^{u_i\prime} = P_{q_t}^{u_i} + \alpha \cdot (\triangledown(f(P_{q_t}^{u_i}; R), g(P_{q_t}^{u_i}; R'))) + \theta \cdot \triangledown g(P_{q_t}^{u_i}; U, R') \quad (12)$$

The gradient of models is simply represented as $\triangledown$.

(13) gives the feedback evaluation definition of user preferences in item recommendation space generated from model f and model g.

$$f(P_{q_t}^{u_i}; R), g(P_{q_t}^{u_i}; R') \triangleq Con_{Quantity,Rank,Top_y}(R_{u_i}, R'_{u_i}, U_{q_t}; R, R') \quad (13)$$

In which, R_{u_i} and R'_{u_i} are the items that are recommended to u_i in two tasks respectively. U_{q_t} is the user preference for feature q_t. $Quantity$ means the number of q_t in the recommended items for u_i. $Rank$ is the sequence of q_t in u_i's preference. Top_y refers to the top y preferences of u_i. $Con()$ represents confidence evaluation on user's preference. For instance, if the recommendations for a user in both R and R' show high proportion of feature q_t, q_t will be assigned with high confidence for u_i, then the selection probability of u_i selecting q_t is enhanced.

The third part on the right side of (12) means that user preference in primal task can be updated according to the preference of neighbors in user clusters generated by dual task. If a kind of preference of a user is also highly acknowledged as important for his/her neighbors, the preference will be given a large weight for u_i, otherwise, the weight of this preference will be reduced.

Because of the strong duality of two tasks, the feedback from dual task can facilitate the user preference update in primal task. Assuming Z users are in the cluster that u_i belongs to, and the user u_i has L features in UF, the preference of u_i for q_t in $g(P_{q_t}^{i}; U)$ can be calculated as follows.

$$P_{q_t}^{u_i\prime} = \begin{cases} P_{q_t}^{u_i} + \omega \cdot \sum_{j=1}^{Z} P_{q_t}^{u_j}/Z & \text{if} \sum_{j=1}^{Z} P_{q_t}^{u_j}/Z \geq P_{q_a}^{u_i} \\ P_{q_t}^{u_i} - \omega \cdot \sum_{j=1}^{Z} P_{q_t}^{u_j}/Z & \text{if} \sum_{j=1}^{Z} P_{q_t}^{u_j}/Z < P_{q_a}^{u_i} \end{cases} \quad (14)$$

The similarity calculation of items in CBF recommendation approach is crucial. The duality characteristics between f and g

being considered, the similarity of items can be calculated by the consistency of user preference changes in the two tasks. That is, if the user's preference for some items is enhanced in both tasks, the similarity of these items is further improved. If two items $r1$ and $r2$ are described as: $r1 = (q_1^1, q_2^1, \ldots, q_k^1)$, $r2 = (q_1^2, q_2^2, \ldots, q_k^2)$, the similarity of $r1$ and $r2$ is computed as follows:

$$Sim(r1, r2) = \omega_t \cdot Sim(q_t^1, q_t^2), t \in [1, k] \tag{15}$$

$$\omega_t = (P_{q_t}^{u_i{}'}/P_{q_t}^{u_i})/\sum_{j=1}^{t}(P_{q_t}^{u_j{}'}/P_{q_t}^{u_j})$$

This equation utilizes the change of user preference obtained from two tasks to react on the similarity calculation of items. From (15), it also indicates that the similarity of items is different for different users.

(2) Rating matrix completion.

Similarly, as for the dual task, the rating matrix RU is expected to be updated through the feedback of user clusters and item recommendations from both two tasks. The value that a user u_i rates an item r_j, $RU_{r_j}^{u_i}$, is updated as follows:

$$RU_{r_j}^{u_i{}'} = RU_{r_j}^{u_i} + \alpha \cdot (\triangledown(g(RU_{r_j}^{u_i}; U), f(RU_{r_j}^{u_i}; U'))) + \theta \cdot \triangledown f(RU_{r_j}^{u_i}; R, U') \tag{16}$$

In which, both $g(RU_{r_j}^{u_i}; U)$ and $f(RU_{r_j}^{u_i}; U')$ aim to find the most valuable neighbors of u_i to complete and update u_i's ratings for items, and this part is defined as follows:

$$g(RU_{r_j}^{u_i}; U), f(RU_{r_j}^{u_i}; U') \triangleq argmax(UM_{U,U'}(u_i, u_x)) \tag{17}$$

In which, $u_i, u_x \in U$, $u_i, u_x \in U'$. $UM()$ refers to user relationship matrix. A most similar and useful neighbor set of u_i can be obtained through the following function:

$$U_{u_i,u_x} = function(u_i, u_x), u_x \in UC'_{u_i} \cup (UC'_{u_i} \cap UC_{u_i}) \cup UC'_{UC_{u_iTopm}} \tag{18}$$

UC and UC' refer to user clusters in dual task and primal task respectively. $UC \in U$, $UC' \in U'$. UC'_{u_i} and UC_{u_i} refer to the neighbors of u_i generated in these two task correspondingly. $UC'_{u_i} \cap UC_{u_i}$ is the intersection of user sets UC'_{u_i} and UC_{u_i}, which represents the users who belongs to u_i's cluster in both U and U'. $UC'_{UC_{u_iTopm}}$ refers to the user set in u' which has the same top m closest neighbors as u_i in U. u_x is a candidate neighbor among the union of the above three sets. This function ensures the that $RU_{r_j}^{u_i}$ in dual task can be obtained referring to the neighbors from user space mapping.

$f(RU_{r_j}^{u_i}; R, U')$ returns the item rating based on R from primal task. Assuming there are N items recommended for a user, $RU_{r_j}^{u_i}$ in f can be updated as follows:

$$RU_{r_j}^{u_i{}'} = RU_{r_j}^{u_i} + \alpha \cdot (\triangledown f(Rank_{r_j}^{u_i}/N, Rank_{r_k}^{u_i}/N, Sim_{r_j,r_k}, Rank_{r_j}^{u_k}/N)) \tag{19}$$

In which, u_i and u_k belong to a same cluster in U'. r_j and r_k belong to the recommendations to u_i in primal task. $Rank_{r_j}^{u_i}$ means the rank of r_j in the recommendation list to u_i. $Rank_{r_k}^{u_i}$ has the same meaning for r_k. Sim_{r_j,r_k} refers to the similarity of r_j and r_k based on feature vectors. $Rank_{r_j}^{u_k}/N$ indicates the rank of r_j in the recommended list for the neighbors of u_i in U'. (19) ensures the RU in dual task could be updated according to the recommendations in R from primal task.

Ideally, the completion of user preference and rating matrix should be consistent. If a user's preference for certain features increases, he/she will give a higher score to the items containing these features, and vice versa. Hence, the objective function of recommendation bases completion can be concluded as follows:

$$Fun3 = minimize(loss(P_{q_t}^{u_i{}'}, RU_{r_j}^{u_i{}'})) \tag{20}$$

In which, q_t is one of the features of item r_j.

3.5. Algorithm description

The algorithm of DLRA is described as Algorithm 1.

To address the process of DLRA, we apply CBF approach and CF approach on primal task and dual task respectively. In algorithm 1, item-based KNN (K-Nearest Neighbor) approach is used to obtain recommendations — R to users in primal task, the item similarity is acquired by computing the information of item's feature other than the ratings from users. User cluster U' is obtained by K-Means approach based on R set. As for dual task, K-Means approach is used to get U clusters, and R' is further obtained based on the user–item rating matrix among user clusters. That is, R' is the top N items assorted by the ratings of user clusters which the active user belongs to.

The reasons why KNN and K-Means are selected for the two tasks in Algorithm 1 are listed here. Firstly, these two algorithms are classic algorithms based on nearest neighbor recommendation and user clustering recommendation respectively. Both KNN and K-means are suitable for common recommendation scenarios. Secondly, this paper focuses on the effectiveness of dual characteristic in recommender system. The recommendation mechanisms of KNN and K-Means fit three duality strategies well. Thirdly, in order to evaluate the performance of DLRA, the hybrid recommendation methods based on KNN and K-Means are selected as comparative experimental methods in Section 4, so the feasibility and effectiveness of the experiment are guaranteed.

4. Experiment setup

In this section, we first introduce the computing environment which includes the hardware and software tools in Section 4.1. Then, we list the benchmark datasets and our design of the dataset in Section 4.2. Several comparison algorithms are described in Section 4.3. In Section 4.4, we design evaluation matrices to test the performance of these algorithms.

4.1. Computing environment

The experiments in this study were performed in a desktop computer which has 8 GB RAM, Intel Core i5-8250 CPU, Nvidia RTX 2070 GPUs, and only the deep learning-based program is run on GPUs. The operating system is Windows 10, 64-bit, and the software we used is Python 3.6.

Algorithm 1 DLRA algorithm

Input: User history files UH, candidate item set $R0$, historic item rating RH, user set $U0$.
Output: Recommendations, R_{rec}, to users.

1: Extract user preference UF if UH is not empty. Initialized $P(r, u^{'})$ with random probability. Extract item rating matrix RU and user relationship matrix UM if RH is not empty. Initialized $P(u, r^{'})$ with random probability.
2: Implement primal and dual models to generate the recommendations U, $U^{'}$, R and $R^{'}$.
3: Compute $\triangledown_{\theta_{ru'}, \theta_{ur'}}(E[re]E^{'}[re])$. If f and g converge or the termination condition is meet, jump to 4, else, do,
  3.1 Update $P(r, u^{'})$ according to R and $R^{'}$. Update $P(u, r^{'})$ according to U and $U^{'}$.
  3.2 Update UF according to (12).
  3.3 Update RU according to (16).
  3.4 Update $\theta_{ru'}$ in f and $\theta_{ur'}$ in g.
  3.5 Generate recommendations based on f and g.
4: Output R_{rec}.

Table 1
Description of the datasets.

Dataset	Users	Items	Ratings	Rating scale	Density
ML-100K	610	9742	100,836	[1,5]	6.3
ML-1M	6040	3706	1,000,209	[1,5]	4.47
ML-20M	138,493	27,278	20,000,263	[0.5,5]	0.53
BC	10,339	6708	361,349	[1,10]	0.005

Table 2
Sparsity simulation of experiment data.

No	Side information	Usage proportion of ratings
I1	Yes	100%
I2	No	
II1	Yes	50%
II2	No	
III1	Yes	10%
III2	No	

4.2. Datasets

The recommendation approaches are evaluated on four widely used datasets in recommendation fields which are MovieLens 100K (ML), Movielens 1M (ML-1M) and Movielens 20M (ML-20M) from movie rating community,[1] and BookCrossing (BC) dataset from book reading website.[2] In which, ML and ML-1M contain side information of users, like age, occupation, gender, and zip code. ML-1M and ML-20M contain information about movies, such as 19 different genres. BC also contains the side information, such as the user age, and the author, the published year and the category of books.

To evaluate the performance of our proposed recommendation approaches on dealing with data sparsity and cold start problems, we design different experiment schemes based on Table 1. First, we set different usage proportion of side information in this experiment. The experiment data design is listed in Table 2. The first column means the design strategies of the experimental dataset. These strategies can be applied to selecting appropriate experimental datasets from Movielens and BC. For example, the row II2 can be explained as, the side information of users or items is not considered in the experiments. If one dataset has 100 groups of scoring data for items, 50% of the rating data are applied to the experiment. If 10-fold cross validation is adopted, 45 sets of data are used for training set, 5 sets of data are used for basic test set to get basic algorithm performance index.

Cold start is a common issue in RSs. In this experiment, we simulate the cold start problem to test performance of recommendation models. User cold start means that a user has not rated any of the items, and item cold start means that an item has not been rated by any of users. Since the datasets in Table 1 have been preprocessed, there is no cold start problem here. That is, only those users with at least 20 interactions, and the items with at least 5 interactions are retained in Table 1. In BC, the users who have rated less than or equal to 5 ratings are removed [62]. To build a cold start environment, we delete specific ratings to simulate user cold start and item cold start problems. The experiment data are shown in Table 3. The first four rows are user cold start simulation and the last four rows are item cold start dataset. For instance, UCY2 means that the side information of users is considered when the recommendation approach is implemented, and 30% users in the dataset have not rated any items. ICN1 means the side information of items is not considered when the recommendation approach is implemented, and 10% items in the dataset have not received any ratings.

[1] http://www.grouplens.org/

[2] www2.informatik.uni-freiburg.de/ ∽ cziegler/BX/

Through the designs of Tables 1–3, we can evaluate the performance of the algorithms under different degree of data sparsity and cold start with different scale datasets.

We use 10-fold cross validation to evaluate the performance of recommendation algorithms.

4.3. Comparison algorithms

To the best of our knowledge, there is no research that treats a RS as two closed-loop interactive tasks and utilizes the duality characteristic of two dual tasks to achieve recommendations. In our study, we use traditional recommendation approaches to construct the two tasks of a RS. The primal task is implemented by item-based KNN algorithm, and the dual task is user-based KNN and item-based KNN algorithms respectively. All the recommendation methods in this study are listed in Table 4.

The DLRA proposed in this paper is a kind of HF recommendation approach. Hence, we use several HF recommendation strategies as comparison methods. As mentioned in Section 2, most of the HF recommendation approaches are based on feature extraction. In order to achieve better performance, these CF approaches use extra information, context-based or social relationship-based, to complete the rating matrix. In this study, we do not focus on mining implicit information of users or items to facilitate the recommendation process. Hence, we use switching HF and cascade algorithms as comparisons, which are labeled as S-HF and C-HF respectively. The item-based KNN CBF algorithm and user (item)-based K-Means CF algorithm are designed as S-HF and C-HF approaches.

As a kind of machine learning approach, singular vector decomposition (SVD) is a popular matrix factorization technique for that it can generate a lower rank approximation of the ratings matrix, and the lower rank matrix is used in recommender systems to acquire the latent relationships users and items, then the low-dimensional representation is used to compute the neighborhood of items. We take SVD as a representative of model-based CF approach, and use it as a comparison approach.

Since deep learning methods have seen huge success in RSs, we include deep learning-based recommendation approach as comparison in this study. The DNNRec and Non-deep NN (N-Deep) methods that are introduced in recent studies are applied here [35]. DNNRec integrates side information into a 3 hidden layer neural network. Non-deep NN only has one hidden layer.

4.4. Performance evaluation metrics

The proposed DLRA and the comparison approaches are evaluated based on the following criteria. Root Mean Squared Error (RMSE) is used to measure the accuracy of the recommendation approaches. The equation is listed as follows:

$$RMSE = \sqrt{(1/m)\sum_{i=1}^{m}(\hat{y}_i - y_i)^2} \tag{21}$$

In which, m is the total number of predications. $\hat{y}_i$ is the predicted rating and y_i is the actual rating.

R-squared is used to evaluate the fit performance of these recommendation models. It is defined as follows:

$$R^2 = 1 - \sum_{i=1}^{m}(\hat{y}_i - y_i)^2 / \sum_{i=1}^{m}(\bar{y}_i - y_i)^2 \tag{22}$$

Where, $\bar{y}_i$ is the average of the actual ratings.

If the recommendation process of a RS is regarded as a classification problem, we use F1 to measure the comprehensive index of the recommendation models. First, we assume that if the user's

Table 3
Cold start problem simulation of experiment data.

No	Side information	Side information included	None rating proportion
UCY1	Users	Yes	20%
UCN1	Users	No	
UCY2	Users	Yes	50%
UCN2	Users	No	
ICY1	Items	Yes	20%
ICN1	Items	No	
ICY2	Items	Yes	50%
ICN2	Items	No	

Table 4
Recommendation methods descriptions.

Name	Classification	Comments
DLRA	HF	integrated with item-based KNN CBF and K-Means CF
K-Means	Memory-based CF	item-based
SVD	Model-based CF	Biased based SVD
C-HF	HF	item-based KNN CB first and follows K-Means CF
S-HF	HF	switch between item-based KNN CB and K-Means CF
Non-deep NN	Deep learning	1 hidden layer
DNNRec	Deep learning	3 hidden layers

rating on an item is higher than or equal to the average of user's ratings on all the items, it is considered that the user likes the item, otherwise, it is explained that the user does not like the item. Similarly, through the statistics of the prediction ratings, we can acquire the accuracy rate and recall rate of the test dataset. Then, we can compute the precision, recall and F-measure to evaluate the algorithms based on the following equation.

$$P = TP/(TP + FP) \tag{23}$$

$$R = TP/(TP + FN) \tag{24}$$

$$F1 = 2 * P * R/(P + R) \tag{25}$$

Where, *TP* represents the number of positive cases which are predicted to be positive cases. *FN* represents the number of positive cases which are predicted as negative cases. *FP* represents the number of negative cases which are predicted as positive cases. *TN* represents the number of negative cases which are predicted as negative cases.

Besides, the training/running time is recorded to compare the efficiency of these recommendation approaches.

5. Results and discussions

This section mainly introduces the experimental process, experimental results and discussions for addressing data sparsity and cold start problems of recommender systems.

5.1. Comparisons on data sparsity problem

Table 5 shows the experimental results of the four basic datasets, in which, all the side information of users or items is considered. From this table, we can get a big picture of the performance of these recommendation methods. The performance difference of RMSE among K-Means, SVD, C-HF and S-HF is not very significant.

The proposed DLRA outperforms K-Means, SVD, C-HF, S-HF and N-Deep on most the datasets obviously. The performance of SVD in the first two datasets is better than K-Means, C-HF and S-HF. But in the last two datasets, the performance of SVD is lower than other algorithms on ML-20M and BC dataset which have higher data sparsity. In ML-20M dataset, DNNRec slightly outperforms DLRA on RMSE, however, DLRA is 4.88% better than

Table 5
Performance comparison of accuracy and model fitting.

Dataset	Method	RMSE	R^2	F1	Run time (ms)
Ml-100K	DLRA	**0.763**	**0.401**	0.43	2,394
	K-Means	0.917	0.238	0.36	2,183
	SVD	0.808	0.216	**0.48**	1,761
	C-HF	0.886	0.253	0.36	2,769
	S-HF	0.895	0.276	0.38	3,128
	N-Deep	0.821	0.299	0.41	1,265
	DNNRec	0.785	0.374	0.45	1,443
Ml-1M	DLRA	**0.807**	**0.424**	0.38	15,347
	K-Means	0.893	0.378	0.29	13.824
	SVD	0.827	0.362	0.31	9.324
	C-HF	0.905	0.335	**0.44**	20,454
	S-HF	0.911	0.341	0.27	23.193
	N-Deep	0.904	0.328	0.29	16,371
	DNNRec	0.856	0.399	0.35	14,290
ML-20M	DLRA	0.822	**0.268**	0.41	12,592
	K-Means	0.913	0.238	0.26	10,381
	SVD	0.884	0.216	0.36	11,825
	C-HF	0.916	0.253	0.34	13,621
	S-HF	0.905	0.246	0.29	12,476
	N-Deep	0.893	0.138	0.34	14,785
	DNNRec	**0.802**	0.216	**0.43**	13,443
BC	DLRA	**1.637**	0.132	**0.35**	21,425
	K-Means	1.789	**0.026**	0.21	18,765
	SVD	1.892	0.008	0.18	19,244
	C-HF	1.801	0.087	0.24	18,236
	S-HF	1.847	0.066	0.27	9,128
	N-Deep	1.894	0.072	0.20	22,832
	DNNRec	1.721	0.119	0.27	4,690

DNNRec on BC dataset. Through the analysis on R^2, it is found that DLRA does not show better performance compared with other methods in small dataset, ML-100K. However, with the sparsity increases, DLRA shows obvious advantages. The values of R^2 of DLRA are 6.27%, 5.93% and 10.9% higher than the next best methods on dataset ML-1M, ML-20M and BC. As for F1, the ranking of DLRA has always been in the top three ones, and in the BC dataset, DLRA exceeds other approaches greatly. When it comes to the comparison results of running time, DNNRec shows the shortest running time because of its GPU configuration. DLRA is far faster than N-Deep, which indicates that as a kind of HF recommendation approach, DLRA holds the advantage of lower running/training time than deep-learning based recommendation methods. DLRA also has the close running time with K-Means, S-HF and C-HF.

Table 6
Performance comparison of algorithms on user cold start problem.

Dataset	Method	RMSE	R^2	F1
ML-1M UCY1	DLRA	**0.803**	**0.421**	**0.38**
	SVD	0.899	0.386	0.29
	S-HF	0.879	0.334	0.25
	N-Deep	0.913	0.327	0.28
	DNNRec	0.824	0.385	0.27
ML-1M UCN1	DLRA	0.851	**0.405**	**0.34**
	SVD	0.917	0.351	0.27
	S-HF	0.923	0.311	0.23
	N-Deep	**0.831**	0.387	0.22
	DNNRec	0.877	0.363	0.23
ML-1M UCY2	DLRA	**0.827**	**0.204**	**0.31**
	SVD	0.914	0.154	0.18
	S-HF	0.945	0.112	0.24
	N-Deep	0.911	0.108	0.21
	DNNRec	0.908	0.176	0.22
ML-1M UCN2	DLRA	**0.819**	**0.138**	**0.25**
	SVD	0.912	−0.007	0.13
	S-HF	0.953	0.011	0.17
	N-Deep	0.928	0.021	0.16
	DNNRec	0.887	0.117	0.18

Table 7
Performance comparison of algorithms on item cold start problem.

Dataset	Method	RMSE	R^2	F1
BC ICY1	DLRA	**0.874**	**0.393**	**0.35**
	SVD	0.936	0.348	0.25
	S-HF	0.981	0.319	0.22
	N-Deep	0.952	0.307	0.26
	DNNRec	0.928	0.352	0.29
BC ICN1	DLRA	**0.891**	**0.386**	**0.31**
	SVD	1.132	0.342	0.21
	S-HF	1.195	0.306	0.19
	N-Deep	1.228	0.311	0.22
	DNNRec	1.103	0.337	0.24
BC ICY2	DLRA	**1.342**	**0.246**	**0.27**
	SVD	1.674	0.148	0.21
	S-HF	1.789	0.103	0.17
	N-Deep	1.591	0.093	0.13
	DNNRec	1.874	0.152	0.19
BC ICN2	DLRA	**1.693**	**0.183**	**0.21**
	SVD	1.874	−0.013	0.11
	S-HF	1.994	0.008	0.13
	N-Deep	1.9301	0.015	0.11
	DNNRec	1.904	0.076	0.14

From the preceding analysis, some conclusions and discussions can be drawn, (1) DLRA obviously outperforms other compared methods on RMSE and R^2 in all the datasets, expect for a small gap of 2.4% that DLRA is lower than DNNRec in RMSE. This shows that DLRA algorithm can obtain better performance than both traditional hybrid filtering algorithms and basic deep learning algorithms when certain labeled data exist. It owes to the enhanced learning mechanism brought by the duality strategies in DLRA. (2) With the decrease of data density as shown in Table 1, DLRA shows more obvious advantages compared with other algorithms. Taking R^2 as an example, for the four datasets listed in the first column in Table 5, the values of R^2 of DLRA are 7.2%, 6.3%, 5.9% and 10.9% higher than the next best algorithm respectively, and are 85.6%, 29.2%, 94.2%, 1550% higher than the worst algorithm respectively. The performance of other algorithms is affected when the data density decreases, while DLRA algorithm can still maintain good model fitting effect and high recommendation accuracy. (3) The F1 index of DLRA algorithm shows the best results on BC datasets and it is in the second or third position in other datasets. It also shows that when the labeled data is extremely sparse, the stability of DLRA algorithm is better. In other datasets, the F1 index of DLRA is not optimal, which may due to the initialized selection probability strategy affects the stability of the model in relatively smaller datasets.

5.2. Comparisons on cold start problem

To evaluate the performance of algorithms in addressing cold start problem, we use dataset selection criteria in Table 3 and take ML-1M and BC as the experiment datasets.

Table 6 shows the experimental results of user cold start problems and Table 7 shows the results on dataset BC. In both of these two tables, UCY1 and UCY2 mean the side information of users are considered, and UCN1 and UCN2 mean the side information of users is excluded. The same settings of item side information are shown in Table 7. Considering the similar performance of S-HF and C-HF, and the fact that DLRA outperforms K-Means markedly, we remove K-Means and C-HF methods from this experiment.

By comparing the experiment results of user and item cold start problems in Tables 6 and 7, it is noticed that the values of RMSE, R^2 and F1 in Table 7 are lower than those in Table 6 on a whole. It proves the fact that facing with the dataset with obvious data sparsity problem, the performance of algorithms in addressing the cold start problem will decline to some extent. If we focus on Table 6, the RMSE values of DLRA are 2.55%, 8.92% and 7.67% higher than the next best technique on the first, second and fourth datasets respectively. The R^2 values of DLRA are 9.06%, 4.65%, 15.91% and 17.95% better than the next best technique respectively. As for F1, the outperformance is 31.03%, 25.93%, 29.17% and 38.89% respectively. It is obvious that DLRA outperforms other techniques on all the datasets. Moreover, in the case of high data sparsity and serious cold start problem, the proposed DLRA performs apparently better than other algorithms.

By analyzing the experiment results in Tables 6 and 7, the following conclusions can be drawn, (1) In the experiment on ML-1M, DLRA shows obvious advantages in various indexes, except that the RMSE under UCN1 is lower than the N-deep algorithm. In the experiment of BC dataset, DLRA shows absolute advantages, and each index exceeds other algorithms. The results indicate that DLRA has strong robustness in the face of different data sparsity and different cold start problems. (2) Compared with the results in Table 5, DLRA shows more prominent stability in Tables 6 and 7. One reason is that we have constructed the data sparse conditions for the experimental data in Tables 6 and 7. Particularly, because the density of BC is relatively low, the scenario of a RS with extremely sparse data is generated in Table 7. The excellent performance of DLRA further proves that the recommendation efficiency of DLRA based on dual learning is very promising.

5.3. Performance evaluation of DLRA and DNNRec

It is acknowledged that, the more valid rating data available, the better the performance of recommendation system. But the problem of data sparsity is hard to avoid. Hence, we compare the performance of recommendation approaches by using a same dataset with different data sparsity.

The results in Table 8 show that DLRA maintains a relatively stable accuracy when the rated data drops rapidly, which shows that DLRA guarantees the effectiveness of the recommendation model by self-tuning mechanism through mutual feedback of two bidirectional dual tasks. However, the performance of DNNRec drops sharply. The main reason lies in that DNNRec method cannot guarantee effective recommendations because it can learn less latent influence factors when there is barely few of available data.

To have a specified comparison of these methods, we investigate how RMSE changes with the training iterations. The

Table 8
RMSE on ML-20M and BC.

Dataset	Dataset number	DLRA	DNNRec
ML-20M	II1	0.867	0.938
	II2	0.864	0.957
	III1	0.885	0.912
	III2	0.891	0.987
BC	II1	1.134	1.536
	II2	1.227	1.719
	III1	1.211	1.874
	III2	1.301	1.953

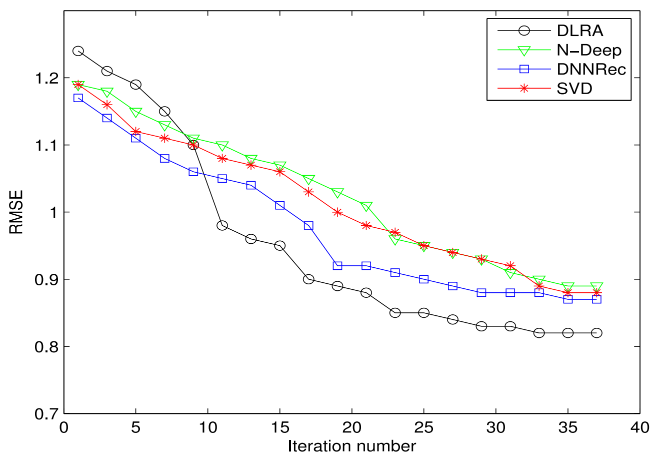

Fig. 4. RMSE Comparison.

representative training curves on ML-20M are presented in Fig. 4. In which, DLRA finally shows the lowest RMSE value. Moreover, its curve shows that the accuracy of recommendation algorithm is not very high at the beginning, but it quickly declines in the 10th generation or so, and then it maintains a stable downward trend. The results in Table 8 show that the recommendation performance of DLRA improves rapidly and maintains good stability after the parameters are adjusted in the initial stage. This curve also proves that DLRA has strong learning ability and model adjustment ability. It can quickly learn knowledge from the recommendation process and better adapt to the recommendation scenario with a few labeled data. SVD, N-Deep and DNNRec algorithms show a relatively stable downward trend too, but their performance is obvious inferior to DLRA. Particularly, the deep learning algorithm has some limitations in the case of highly sparse data.

5.4. Performance evaluation of duality strategies

In order to evaluate the significance of the three strategies in DLRA, we designed ablation experiments. The last three algorithms in the first row of Table 9 correspond to the absence of each dual strategy respectively. DLRA_23 indicates that the duality of the recommendation space is not applied. DLRA_13 means that the duality of the selection probability is not applied. DLRA_12 refers to the situation that the duality of the recommendation basis is not applied. We consider the recommendation scenarios of data sparsity and cold start using the ML-1M and ML-20M respectively. The results of RMSE are shown in Table 9. DLRA is used as a benchmark comparison algorithm. Compared with DLRA, if the RMSE difference between an algorithm and DLRA is larger, it indicates that the algorithm is more affected by the duality strategy that the algorithm excludes. That is, this absent duality strategy is more important to the implementation of the recommender system.

Table 9
RMSE on ablation experiments of DLRA.

Dataset	Dataset number	DLRA	DLRA_23	DLRA_13	DLRA_12
ML-1M	I1	0.867	0.841	1.128	1.214
	I2	0.864	0.877	1.191	1.225
	III1	0.885	1.012	1.026	0.873
	III2	0.891	1.127	1.321	0.916
ML-20M	ICY1	0.831	0.992	1.027	1.121
	ICN1	0.853	1.024	1.145	0.911
	ICY2	1.198	1.342	1.512	1.507
	ICN2	1.402	1.318	1.428	1.465

Through the comparison results between DLRA and other algorithms, it can be concluded that the performance of DLRA_23, DLRA_13 and DLRA_12 is slightly inferior to DLRA as a whole. In the experiment of data sparsity scenario, DLRA_23 performs better when the data is extremely sparse, and DLRA_12 performs better when the data is not significantly sparse. The dual strategy of recommendation space helps to guide the direction of recommendation process when available data is sparse, and it uses positive reward to improve the accuracy of recommendation. The recommended basis completion strategy is suitable for cases where data sparsity is not extremely obvious, indicating that this strategy will become more and more effective as the available user preferences or scoring matrix is more sufficient.

In the experiment of cold start scenario, the data of the last three columns show that the three strategies all play a significant role. In particular, the duality of recommendation space still shows good performance in the face of data sparsity. Moreover, the results on DLRA_13 column show that the duality strategy of selection probability plays an important and stable role in data sparse and cold start scenarios.

The results of ablation experiments show that the three strategies play an irreplaceable role in DLRA. The duality strategy of recommendation space can better address the recommendation scenario with sparse data. The duality recommendation strategy of probability selection ensures the implementation and convergence of the recommendation process, and it shows stable performance in all scenarios. The duality strategy based on recommendation basis completion has better performance when there is a certain amount of available data. With the progress of recommendation, the role of the three strategies is becoming more and more significant.

6. Conclusions and future work

A recommender system takes users' history information or the user–item rating matrix as the input data, and outputs the items that users may like best. To provide users with items that best meet their needs and preferences, the quality of both user history files and existing rating matrix play crucial roles. But cold start and data sparsity are usually unavoidable, which hinders the application of recommender systems. Even though some hybrid recommendation approaches use side information to complete the rating matrix, such information is not always available. Deep learning is also widely used in RSs, but its performance heavily depends on big data, and this limits its application in many scenarios.

In this study, we apply dual learning theory on recommender systems. Recommendation task is decomposed into a primal task and a dual task. The inherent duality characteristics between the two tasks in RSs are leveraged to design a novel recommendation strategy. Based on this analysis, we propose a dual learning-based recommendation method to address data sparsity and cold start problems. Specifically, three duality-based recommendation techniques are designed to achieve recommendations, which are implemented by minimizing the difference of user/item space, minimizing the selection probability of users and items, as well as completing user preference and rating matrix. DLRA has the following characteristics:

(1) The dual-learning mechanisms which ensure the self-tuning and self-improvement of the two tasks accordingly relieves the dependence on user history data or rating matrix.

(2) User preference and rating matrix are updated through real-time feedback from the interaction between the two tasks. This closed-loop feedback guarantees the accuracy and effectiveness of the whole recommendation task.

(3) The dual theory has a solid mathematical foundation, and the recommendation method proposed in this paper falls under white-box theory. Hence, DLRA can be interpreted and optimized.

We design experiments to evaluate the recommendation approaches considering the situation of data sparsity, cold start, and the availability of side information. Comparing with the traditional hybrid recommendation algorithms and deep learning algorithms, it was found that DLRA shows relatively stable and excellent recommendation accuracy in the case of extremely sparse data and lack of side information. DLRA also has significantly faster training time than deep learning algorithms. DLRA provides strong methodological guidance for unsupervised, semi-supervised and supervised recommendation approaches. Meanwhile, the dual-learning based approaches can be applied to many other fields if the duality exists or can be simulated.

As a recommendation method based on dual learning, DLRA has the ability to generate recommendations based on zero or a few labeled data. However, when there are zero or a few labeled data, the trial and error feedback mechanism of dual learning causes relatively low performance at the initial stage, hence, how to design a dual learning-based recommendation algorithm for real-time RSs needs further research. In addition, how to utilize duality characteristic to optimize the coupling of recommendation strategies together in RSs is also our research topic in the future.

CRediT authorship contribution statement

Shanshan Wan: Conceptualization, Methodology, Formal analysis, Writing – original draft. **Ying Liu:** Data curation, Funding acquisition, Software. **Dongwei Qiu:** Methodolgy, Mathematical model modification, Revised submission proofreading. **James Chambua:** Writing – original draft, Writing – review & editing, Validation. **Zhendong Niu:** Supervision, Methodology.

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

The authors wish to acknowledge the support of the National Natural Science Foundation of China (No. 61902016, No. 61871020), the high level innovation team construction project of Beijing Municipal Colleges and universities (No. IDHT20190506), the National Key R&D Program of China (No. 2019YFB1406302, No. 2018YFC0807806-2), the postgraduate education and teaching quality improvement project of Beijing University of Civil Engineering and Architecture, China (No. J2022005).

References

[1] C.A. Gomez-Uribe, N. Hunt, The netflix recommender system: Algorithms, business value, and innovation, ACM Trans. Manag. Inform. Syst. (TMIS) 6 (4) (2016) 13.
[2] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, et al., The YouTube video recommendation system, in: Proceedings of the Fourth ACM Conference on Recommender Systems, ACM, 2010, pp. 293–296.
[3] 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.
[4] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey, Knowl.-Based Syst. 46 (2013) 109–132.
[5] J. Bobadilla, A. Hernando, F. Ortega, J. Bernal, A framework for collaborative filtering recommender systems, Expert Syst. Appl. 38 (12) (2011) 14609–14623.
[6] W. Chen, Z. Niu, X. Zhao, Y. Li, A hybrid recommendation algorithm adapted in e-learning environments, World Wide Web 17 (2) (2014) 271–284.
[7] J.K. Tarus, Z.D. Niu, G. Mustafa, Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning, Artif. Intell. Rev. 50 (1) (2018) 21–48.
[8] R. Burke, Hybrid recommender systems: Survey and experiments, User Model. User-Adapt. Interact. 12 (4) (2002) 331–370.
[9] D. Zhang, C.H. Hsu, M. Chen, Q. Chen, N. Xiong, J. Lloret, Cold-start recommendation using Bi-clustering and fusion for large-scale social recommender systems, IEEE Trans. Emerg. Top. Comput. (2014) http://dx.doi.org/10.1109/TETC.2013.2283233.
[10] B.T. Betru, C.A. Onana, B. Batchakui, Deep learning methods on recommender system: a survey of state-of-the-art, Int. J. Comput. Appl. 162 (10) (2017) 17–22.
[11] F. Strub, R. Gaudel, J. Mary, Hybrid recommender system based on autoencoders, in: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, ACM, 2016, pp. 11–16.
[12] S. Zhang, L. Yao, A. Sun, Y. Tay, Deep learning based recommender system: A survey and new perspectives, ACM Comput. Surv. 52 (1) (2019) 5.
[13] R. Mu, A survey of recommender systems based on deep learning, IEEE Access 6 (2018) 69009–69022.
[14] Z. Zhao, Y. Xia, T. Qin, T.-Y. Liu, Dual learning: Theoretical study and algorithmic extensions, 2018.
[15] D. Tang, N. Duan, T. Qin, Z. Yan, M. Zhou, Question answering and question generation as dual tasks, 2017, arXiv preprint arXiv:1706.02027.
[16] Y. Huang, B. Liang, W. Xie, Y. Liao, X. Ding, Dual domain multi-task model for vehicle re-identification, IEEE Trans. Intell. Transp. Syst. PP (99) (2020) 1–9.
[17] B. Zhang, Z. Xu, J. Zhang, G. Wu, A warning framework for avoiding vessel-bridge and vessel-vessel collisions based on generative adversarial and dual-task networks, Comput.-Aided Civ. Infrastruct. Eng. (7) (2021).
[18] D. Mukherjee, S. Banerjee, S. Bhattacharya, P. Misra, Method and System for Context-Aware Recommendation, Google Patents, US Patent 9,510,050, 2016.
[19] M.J. Pazzani, D. Billsus, Content-based recommendation systems, in: The Adaptive Web, Springer, 2007, pp. 325–341.
[20] 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.
[21] F. Isinkaye, Y. Folajimi, B. Ojokoh, Recommendation systems: Principles, methods and evaluation, Egypt. Inform. J. 16 (3) (2015) 261–273.
[22] S.B. Aher, L. Lobo, Combination of machine learning algorithms for recommendation of courses in E-learning system based on historical data, Knowl.-Based Syst. 51 (2013) 1–14.
[23] W. Shanshan, N. Zhendong, An E-learning recommendation approach based on the self-organization of learning resource, Knowl.-Based Syst. 160 (2018) 71–87.
[24] J.B. Schafer, D. Frankowski, J. Herlocker, S. Sen, Collaborative filtering recommender systems, in: The Adaptive Web, Springer, 2007, pp. 291–324.
[25] L.M. De Campos, J.M. Fernández-Luna, J.F. Huete, M.A. Rueda-Morales, Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks, Internat. J. Approx. Reason. 51 (7) (2010) 785–799.
[26] M. Nilashi, O. Ibrahim, K. Bagherifard, A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques, Expert Syst. Appl. 92 (2018) 507–520, http://dx.doi.org/10.1016/j.eswa.2017.09.058.
[27] X. Zheng, W. Ding, Z. Lin, C. Chen, Topic tensor factorization for recommender system, Inform. Sci. 372 (2016) 276–293, http://dx.doi.org/10.1016/j.ins.2016.08.042.

[28] J.K. Tarus, Z. Niu, D. Kalui, A hybrid recommender system for e-learning based on context awareness and sequential pattern mining, Soft Comput. 22 (8) (2018) 2449–2461.
[29] F.M. Almutairi, N.D. Sidiropoulos, G. Karypis, Context-aware recommendation-based learning analytics using tensor and coupled matrix factorization, IEEE J. Sel. Top. Sign. Proces. 11 (5) (2017) 729–741, http://dx.doi.org/10.1109/JSTSP.2017.2705581.
[30] E. Ashley-Dejo, S. Ngwira, T. Zuva, A survey of context-aware recommender system and services, in: 2015 International Conference on Computing, Communication and Security, ICCCS 2015, 2016, http://dx.doi.org/10.1109/CCCS.2015.7374144.
[31] P. Kosmides, C. Remoundou, K. Demestichas, I. Loumiotis, E. Adamopoulou, M. Theologou, A location recommender system for location-based social networks, in: 2014 International Conference on Mathematics and Computers in Sciences and in Industry, IEEE, 2014, pp. 277–280.
[32] K. Zhou, S.H. Yang, H. Zha, Functional matrix factorizations for cold-start recommendation, in: SIGIR'11 - Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2011, http://dx.doi.org/10.1145/2009916.2009961.
[33] C. Wang, D.M. Blei, Collaborative topic modeling for recommending scientific articles, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2011, http://dx.doi.org/10.1145/2020408.2020480.
[34] H. Wang, N. Wang, D.-Y. Yeung, Collaborative deep learning for recommender systems, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 1235–1244.
[35] K. R., P. Kumar, B. Bhasker, DNNRec: A novel deep learning based hybrid recommender system, Expert Syst. Appl. 144 (2020) 113054, http://dx.doi.org/10.1016/j.eswa.2019.113054, URL http://www.sciencedirect.com/science/article/pii/S0957417419307717.
[36] X. Wang, W. Ye, Improving content-based and hybrid music recommendation using deep learning, in: The ACM International Conference, 2014.
[37] H.T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, R. Anil, Z. Haque, L. Hong, V. Jain, X. Liu, H. Shah, Wide & deep learning for recommender systems, in: ACM International Conference Proceeding Series, 2016, http://dx.doi.org/10.1145/2988450.2988454.
[38] Y. Zhu, H. Lu, P. Qiu, K. Shi, J. Chambua, Z. Niu, Heterogeneous teaching evaluation network based offline course recommendation with graph learning and tensor factorization, Neurocomputing (2020) http://dx.doi.org/10.1016/j.neucom.2020.07.064.
[39] X. Jia, X. Li, L. Kang, V. Gopalakrishnan, G. Xun, A. Zhang, Collaborative restricted Boltzmann machine for social event recommendation, 2016.
[40] A. Pujahari, D.S. Sisodia, Modeling side information in preference relation based restricted Boltzmann machine for recommender systems, Inform. Sci. 490 (2019) 126–145, http://dx.doi.org/10.1016/j.ins.2019.03.064.
[41] F. Yang, H. Xie, H. Li, Video associated cross-modal recommendation algorithm based on deep learning, Appl. Soft Comput. 82 (2019) 105597, http://dx.doi.org/10.1016/j.asoc.2019.105597.
[42] 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, http://dx.doi.org/10.1016/j.eswa.2016.09.040.
[43] D. He, Y. Xia, T. Qin, L. Wang, N. Yu, T.-Y. Liu, W.-Y. Ma, Dual learning for machine translation, in: Advances in Neural Information Processing Systems, 2016, pp. 820–828.
[44] Y. Xia, X. Tan, F. Tian, T. Qin, N. Yu, T.-Y. Liu, Model-level dual learning, in: International Conference on Machine Learning, 2018, pp. 5379–5388.
[45] T. Qin, Dual Learning, Springer Berlin Heid, Springer, 2020.
[46] Y. Wang, Y. Xia, L. Zhao, J. Bian, T. Qin, G. Liu, T.-Y. Liu, Dual transfer learning for neural machine translation with marginal distribution regularization, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
[47] T. Kim, M. Cha, H. Kim, J.K. Lee, J. Kim, Learning to discover cross-domain relations with generative adversarial networks, in: Proceedings of the 34th International Conference on Machine Learning-Vol. 70, JMLR. org, 2017, pp. 1857–1865.
[48] J.-Y. Zhu, T. Park, P. Isola, A.A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2223–2232.
[49] Y. Xia, J. Bian, T. Qin, N. Yu, T.-Y. Liu, Dual inference for machine learning., in: IJCAI, 2017, pp. 3112–3118.
[50] P. Luo, G. Wang, L. Lin, X. Wang, Deep dual learning for semantic image segmentation, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, http://dx.doi.org/10.1109/ICCV.2017.296.
[51] H. Gan, Z. Li, Y. Fan, Z. Luo, Dual learning-based safe semi-supervised learning, IEEE Access (2017) http://dx.doi.org/10.1109/ACCESS.2017.2784406.
[52] X. Zhang, J. Cheng, S. Qiu, G. Zhu, H. Lu, Dualds: A dual discriminative rating elicitation framework for cold start recommendation, Knowl.-Based Syst. (2015) http://dx.doi.org/10.1016/j.knosys.2014.09.015.
[53] P. Sun, L. Wu, K. Zhang, Y. Fu, R. Hong, M. Wang, Dual learning for explainable recommendation: Towards unifying user preference prediction and review generation, in: The Web Conference 2020 - Proceedings of the World Wide Web Conference, WWW 2020, 2020, http://dx.doi.org/10.1145/3366423.3380164.
[54] P. Li, A. Tuzhilin, DDTCDR: Deep dual transfer cross domain recommendation, in: WSDM 2020 - Proceedings of the 13th International Conference on Web Search and Data Mining, 2020, http://dx.doi.org/10.1145/3336191.3371793, arXiv:1910.05189.
[55] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems, 2014, http://dx.doi.org/10.3156/jsoft.29.5_177_2, arXiv:1406.2661.
[56] X. Cai, J. Han, L. Yang, Generative adversarial network based heterogeneous bibliographic network representation for personalized citation recommendation, in: 32nd AAAI Conference on Artificial Intelligence, AAAI 2018, 2018.
[57] Z. Yang, J. Hu, R. Salakhutdinov, W.W. Cohen, Semi-supervised QA with generative domain-adaptive nets, in: ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers), 2017, http://dx.doi.org/10.18653/v1/P17-1096, arXiv:1702.02206.
[58] H. Bharadhwaj, H. Park, B.Y. Lim, Recgan: Recurrent generative adversarial networks for recommendation systems, in: Proceedings of the 12th ACM Conference on Recommender Systems, ACM, 2018, pp. 372–376.
[59] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, Human-level control through deep reinforcement learning, Nature 518 (7540) (2015) 529–533, http://dx.doi.org/10.1038/nature14236.
[60] S. Li, M. Li, J. Su, S. Chen, Z. Yuan, Q. Ye, PP-PG: Combining parameter perturbation with policy gradient methods for effective and efficient explorations in deep reinforcement learning, ACM Trans. Intell. Syst. Technol. (TIST) 12 (3) (2021) 1–21, http://dx.doi.org/10.1145/3452008.
[61] W. Shanshan, N. Zhendong, A hybrid E-learning recommendation approach based on learners' influence propagation, IEEE Trans. Knowl. Data Eng. 32 (5) (2020) 827–840.
[62] C.N.C.N. Ziegler, S.M.S.M. Mcnee, J.A.J.A. Konstan, G. Lausen, Improving Recommendation Lists Through Topic Diversification, Promontory Press, 2009.