Postprint: An Implicit Deep Collaborative Recommendation Model Integrating Propositional Logic and Neural Networks

Authors: Chen Bang, Wu Maonian, Zhu Shaojun, Zheng Bo, Peng Wei

Date: 2022-05-10T11:22:57Z

Abstract

With the goal of enhancing the cognitive reasoning capability of recommendation algorithm models and overcoming the current limitation where traditional recommendation algorithms heavily rely on data quality, resulting in constrained performance, we propose an implicit deep collaborative recommendation model that integrates propositional logic with neural networks. First, we construct an implicit logical representation module to assist in bridging the gap between complex variables in practical problems and logical variables, and transform the recommendation problem into a logical expression. Subsequently, we utilize neural networks to fit logical symbols, solve the logical expression, and complete the recommendation. Experiments on three classical datasets with different characteristics—MovieLens, Book-Crossing, and Amazon-E—demonstrate that the proposed method exhibits superior recommendation performance.

Full Text

Preamble

Vol. 39 No. 9
Application Research of Computers
ChinaXiv Partner Journal

Implicit Deep Collaborative Recommendation Model Based on Propositional Logic and Neural Networks

CHEN Bang[1,2], WU Maonian[1,2]†, ZHU Shaojun[1,2], ZHENG Bo[1,2], PENG Wei[1,2]
(1. School of Information Engineering, Huzhou University, Huzhou Zhejiang 313000, China;

2. Zhejiang Province Key Laboratory of Smart Management & Application of Modern Agricultural Resources, Huzhou Zhejiang 313000, China)

Abstract: To enhance the cognitive reasoning capability of recommendation algorithm models and overcome the performance limitations of traditional recommendation algorithms that are highly dependent on data quality, this paper proposes an implicit deep collaborative recommendation model that fuses propositional logic with neural networks. First, an implicit logic representation module is constructed to bridge the gap between complex variables in real-world problems and logical variables, transforming the recommendation problem into a logical expression. Subsequently, neural networks are used to fit logical symbols to solve the logical expression and complete recommendations. Experiments on three classical datasets with different characteristics—MovieLens, Book-Crossing, and Amazon-E—demonstrate that the proposed method achieves superior recommendation performance.

Keywords: recommendation system; collaborative filtering; neural network; cognitive reasoning

---

0 Introduction

In recent years, the rapid development of the Internet and the continuous upgrading of mobile devices have brought diverse lifestyles to society. The further advancement of e-commerce and new retail has provided people with numerous lifestyle and work choices. However, the explosive growth of information and data has also made it difficult for users to identify potentially interesting products from millions or even hundreds of millions of items. Recommendation systems can quickly match users with the most relevant products, effectively saving purchase time and improving user experience, making them a research hotspot in related fields [?]. Current recommendation methods mainly include collaborative filtering, content-based recommendation, and association rule-based recommendation. Most mainstream recommendation methods today are based on and optimized from the collaborative filtering concept, which predicts the future based on users' historical interactions [?]. Classical collaborative filtering methods are represented by matrix factorization [?]. With the rapid development and widespread application of deep learning, using deep models to learn from large-scale data to fit matching functions has become a popular research direction [?].

However, an increasing number of scholars have realized that recommendation problems cannot be perfectly solved merely through statistical induction of interaction history [?]. For example, a user who recently purchased a computer will not need recommendations for similar products but rather peripherals such as keyboards and mice. In general, users are often recommended similar products immediately after purchase, even though their actual need has disappeared. Therefore, relying solely on statistical induction of user interaction history to infer future intentions is difficult to achieve ideal results. Second, statistical matching approaches struggle to learn sufficient user-item feature pairs when facing extremely sparse datasets. This is primarily because the possible combinations of items and users increase exponentially with dataset growth, while classical neural network models lack the reasoning ability to draw inferences, making their performance heavily dependent on dataset quality. Correspondingly, although traditional logic systems excel at solving reasoning problems, hard rule-based logical methods often lack generalization, making it difficult to accommodate internal conflicts when solving real-world problems. For instance, users with identical historical interactions may make completely different decisions, meaning the same logical expression may have different solutions. It is thus evident that fusing logical reasoning with neural networks to build models with both reasoning capability and good generalization will inevitably become a focus of future research [?].

Recently, Shi et al. [?] proposed a novel idea of transforming the recommendation problem into a logical expression for solving, constructing a recommendation prediction model with reasoning capability. Similar works include those by Chen et al. [?], [?] and Fan et al. [?]. However, the logical expressions in these methods only consider first-order relationships between variables, and serial solving causes information loss for earlier units. Therefore, based on transforming the recommendation problem into a logical expression for solving, this paper adds an implicit logic representation module between item embedding and logical solving, proposing an implicit deep collaborative reasoning model.

Compared with existing work, the innovations of this model are: (a) This paper treats variables in the interaction logic expression as a sequence and uses multi-layer self-attention mechanisms to mine implicit interaction relationships among variables, alleviating the problem of insufficient representation information caused by considering only first-order logical relationships. (b) A gated recurrent network module is used to improve information loss for earlier logical units during the serial solving process of logical expressions. Experiments on MovieLens, Book-Crossing, and Amazon-E datasets show that the proposed model achieves better performance on NDCG and HIT metrics.

1.1 Collaborative Filtering Recommendation Based on Deep Models

Collaborative filtering is a common and mainstream recommendation method [?]. Traditional collaborative filtering methods are based on matching concepts, achieving recommendations by learning matching functions. The earliest matrix factorization technology decomposes the user-item interaction matrix into the product of two or more matrices to represent the relationship between users and items [?]. Currently, researchers optimize collaborative filtering models mainly through two approaches. One focuses on optimizing embeddings. For example, Gediminas et al. [?] modeled user features through context-aware pre-filtering, Alexandros et al. [?] used higher-dimensional tensors instead of factorization matrices to enrich embedding information through dimensionality enhancement, and Yehuda [?] incorporated dynamic time information and user behavior into embeddings for learning. Meanwhile, some scholars integrate richer information structures into representations. For instance, Zhang et al. [?] jointly represented user embeddings using images, ratings, and other information, He et al. [?] directly extracted visual features from product images as an independent indicator affecting model decisions, and Ai et al. [?] used knowledge graphs to assist in building recommendation models and optimizing embedding learning.

The other optimization direction is building collaborative filtering recommendation models by finding better matching functions. For example, Hsieh et al. [?] used vector transformation instead of inner product and measured user preferences in a joint space. With deep learning demonstrating powerful effects in image and language domains, an increasing number of scholars tend to use complex neural networks to learn matching. Works such as Ruslan et al. [?], who first proposed an RBM model combining deep learning with collaborative filtering and achieved good results on movie datasets, He [?]' s Neural Collaborative Filtering (NCF) model, and Travis et al. [?]' s Collaborative Memory Network (CMN) model all use richer prior knowledge or more complex kernel models to improve recommendation effects. However, when facing tasks requiring reasoning capability, induction-based statistical methods struggle to bridge the gap from perception to cognition. Therefore, it is necessary to fuse inductive statistics with deductive reasoning to address complex recommendation tasks requiring certain reasoning abilities.

1.2 Neural-Symbolic Integration

As the most classical paradigm in artificial intelligence, symbolic AI uses symbols as a medium for setting hard rules to give AI cognitive intelligence. Neural-symbolic integration is a field that combines classical symbolic knowledge with neural networks, enabling models to provide both good computational capability and logical reasoning ability. Deep learning is regarded as a promising way to overcome the gap between symbols and subsymbols [?]. In recent years, many scholars have attempted to solve logical problems using deep learning methods. For example, Johnson et al. [?] and Yi et al. [?] designed deep network models to generate programs and perform visual reasoning. Yang et al. [?] proposed a first-order logic-based neural logical reasoning system for knowledge bases. Dong et al. [?] built a neural logic model for relational reasoning and decision-making. However, these works all preset a single architecture to handle different logical inputs. While they show good performance on specific problems, they lack flexibility when facing complex datasets requiring generalization capability.

To enable reasoning models to maintain better generalization performance, Shi et al. [?] proposed the Logic-Integrated Neural Network (LINN), which treats basic logical symbols as neural networks for fitting and learning, serving as a good medium between neural networks and symbolic AI. However, this approach neglects the implicit relationships between variables after different types of tasks are converted into logical problems. Additionally, constructing model computation graphs based on logical expressions causes information loss for front-end logical units. This paper targets recommendation systems and designs an implicit logic representation module combining self-attention mechanisms and gated recurrent networks to address problems arising from the fusion of propositional logic and neural networks.

2.1 Method Overview

Enhancing the cognitive reasoning ability of models is an effective means to solve complex recommendation problems. This paper transforms the recommendation problem into a logical expression for solving and proposes an Implicit Deep Collaborative Reasoning (IDCR) algorithm that fuses propositional logic with neural networks. The model mainly consists of an implicit logic representation module based on self-attention and gated recurrent units, and a deep reasoning module based on the integration of logical symbols and BP neural networks. First, the implicit logic representation module mines implicit logical information among items from user historical interactions and generates implicit logical variables. Subsequently, the deep reasoning module solves the logical expression composed of implicit logical variables and makes recommendation predictions.

The model consists of four main components: the input layer, implicit logic representation module, deep reasoning layer, and output layer. The following sections elaborate on each module in detail.

2.2 Input Layer

The input layer converts data into the structure required by the model. The original dataset consists of a series of user-item interaction ratings, including user ID set $U = \{u_1, u_2, u_3, ..., u_i\}$, item ID set $V = \{v_1, v_2, v_3, ..., v_j\}$, and rating matrix $Y_{i\times j}$. Since the model transforms the recommendation problem into a logical expression for solving, it is necessary to convert ratings to 0 and 1. Taking a 5-point scale as an example, ratings greater than or equal to 4 are treated as 1, representing positive interactions between users and items, while ratings less than 4 are treated as 0, representing negative interactions.

Subsequently, all user-item interactions are sorted by time and formed into sequences. Each sequence contains user u_i, item v_j, user rating y_{ij} for the item, and the m interaction item groups $H = \{h_1, h_2, h_3, ..., h_m\}$ before this interaction. Finally, to prevent integer encoding from creating false ordinal relationships between items and users, an EmbeddingLayer is used to map all integer encodings into a high-dimensional space to obtain feature vectors.

2.3 Deep Reasoning Layer

To better introduce the implicit logic representation module, this section first presents the deep reasoning layer for solving logical expressions. Its input is a logical expression composed of logical variables and three basic logical symbols $(\wedge, \vee, \neg)$, and its output is a solution vector. The basic logical symbols are treated as three independent neural networks for training.

For ease of understanding, assume that data processed by the input layer directly enters the deep reasoning layer. The input includes target user u, target item v_t, interaction rating y_t, and three previously interacted items v_1, v_2, v_3 (assuming $m = 3$). Thus, when user u gives positive ratings to items v_1, v_2, v_t and a negative rating to item v_3, it can be converted into a logical expression:

$$(v_1 \wedge v_t) \vee (v_2 \wedge v_t) \vee (\neg v_3 \wedge v_t) = 1$$

Equation (1) represents that the reason for positive interaction between user u and item v_t may be related to v_1, v_2, or v_3. Each conjunctive subformula is independent, which encourages the model to filter out different feature solutions and significantly reduces model coupling.

After converting historical interactions into logical expressions, the recommendation problem becomes a logical expression solving problem. To combine the advantages of symbolic logic and neural networks, this paper treats the three basic logical symbols as three learnable neural networks: $AND(\cdot,\cdot)$, $OR(\cdot,\cdot)$, and $NOT(\cdot)$. This approach benefits from the model' s ability to spontaneously learn and understand logical reasoning rules from data. For example, in one-dimensional space, 0 and 1 represent positive and negative, meaning $NOT(1) = 0$; in two-dimensional space, for any vector v, there exists only one vector $\neg v$ with similarity 0 to represent $NOT(v)$; when the dimension reaches three or higher, for any vector, vectors with similarity 0 are no longer unique. In other words, using classical logic systems alone makes it difficult to express logical relationships between high-dimensional vectors, necessitating the introduction of neural networks to fit complex logical relationships in different spaces.

Assuming all vectors have dimension d, the inputs of $AND(\cdot,\cdot)$ and $OR(\cdot,\cdot)$ networks are two d-dimensional vectors, and their output is a d-dimensional vector. The $NOT(\cdot)$ network takes one d-dimensional vector as input and outputs a d-dimensional vector. All three networks in this paper use a fully connected neural network structure with a single hidden layer. Taking the $NOT(\cdot)$ network as an example, its formula is:

$$NOT(v_t) = \sigma(W_n \cdot v_t + b_n)$$

where σ is the activation function, and ReLU is used throughout this paper.

For example, the output of $AND(v_1, v_t)$ represents the result of $v_1 \wedge v_t$, while $NOT(v_3)$ represents $\neg v_3$. Therefore, the solving process of Equation (1) in the model is shown in Figure 1.

Notably, the model only captures first-order relationships between variables because higher-order relationships grow exponentially with the number of variables, bringing massive computational demands and model overfitting problems.

Finally, the model' s output vector is compared with a randomly initialized base vector T for similarity to obtain the expression' s solution. The similarity formula is:

$$Sim(o, T) = \text{sigmoid}(\varphi \cdot o^T \cdot T)$$

where φ is a hyperparameter that scales the similarity to adapt the model to datasets from different domains. Similarity closer to 1 indicates that the user' s expected rating for the item tends to be more positive.

2.4 Implicit Logic Representation

The deep reasoning model encounters two problems in practical applications. First, the process of solving the disjunctive normal form of logical expressions is serial, causing continuous information loss for earlier logical units while inevitably giving later logical units greater influence on the final result. Second, although retaining only first-order logical relationships between variables can streamline the model, it also causes some loss of implicit information among logical variables. When facing practical problems like recommendation tasks, different variables always have complex implicit relationships—for example, different movies may have sequel relationships, and different products may belong to the same brand or category. These implicit relationships affect recommendation results. However, in the deep reasoning module, each product is represented as an independent logical variable. Using the assumption in Section 2.2 as an example, the model neglects the second-order and third-order relationships among historical interaction items v_1, v_2, v_3:

In short, when higher-order relationships between variables are not considered, the implicit information among v_1, v_2, v_3 is actively discarded, which obviously impacts model performance.

Therefore, it is necessary to construct an implicit logic representation module to address these two problems in the deep reasoning model. First, self-attention mechanisms are used to mine implicit relationships among different logical variables, solving the problem caused by ignoring higher-order relationships. Second, a Gated Recurrent Unit (GRU) layer is added to address the information loss problem caused by serial execution of disjunctive normal forms. The specific structure of the implicit logic representation module is shown in Figure 2.

First, the Attention module treats logical variables representing items as an input sequence and mines implicit interaction information among different items by learning correlation weights among logical variables. This process mainly consists of three components: query, key, and value. Linear projection is used to obtain three vectors Q, K, and V. The similarity matrix is obtained by dot product of Q and K, with the similarity formula:

$$f(Q,K) = \frac{QK^T}{\sqrt{d_k}}$$

After normalizing the similarity matrix, the weight distribution for vector V is obtained, and the weighted value of vector V is calculated by dot product:

$$Attention(Q,K,V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

To mine complex relationships among logical variables in different subspaces, multi-head attention mechanisms are introduced to capture different interaction information in multiple different projection spaces and effectively prevent the model from falling into local optimal traps. The formula is as follows, where W represents learnable parameter matrices:

$$MultiHead(Q,K,V) = \mathrm{concat}(head_1, head_2, ..., head_h)W^O$$

where $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$.

Through the self-attention module, implicit logical variables $\alpha_1, \alpha_2, \alpha_3$ containing high-order implicit interaction information among variables are obtained. Subsequently, a gated recurrent network is used to further optimize the model and solve the information loss problem caused by serial computation in the deep reasoning model.

The internal structure of the gated recurrent network is shown at the top of Figure 3, mainly consisting of two important units: the update gate z_t and the reset gate r_t. The update gate controls the degree to which state information from previous logical variables is brought into current information, while the reset gate controls how much information from the previous state is written into the current candidate hidden state $\tilde{h}_t$. The calculation formulas are as follows, where $[\cdot]$ represents vector concatenation:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

$$\tilde{h}_t = \tanh(W_h \cdot [r_t \odot h_{t-1}, x_t])$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$$

Through the gated recurrent network, the model can spontaneously learn and control the selection of variable information at different positions in the expression, effectively improving the problem of information loss for front-end variables in the deep reasoning model. Finally, the implicit logical variables obtained through the implicit logic representation module are treated as a logical expression and used as input for the deep reasoning model.

2.5 Output Layer

After the deep reasoning layer computes based on the input logical expression, it outputs a d-dimensional solution vector. This vector is calculated with the base vector T using Equation (3) to obtain a value between 0 and 1, representing the user' s expected preference for the target item. The closer this value is to 1, the more positive the user' s expected rating for the item. In recommendation tasks, items with more positive expected ratings rank higher in the recommendation list.

Algorithm 1 presents the specific implementation of the IDCR model. The algorithm description shows that: First, N items interacted with by the same user are sorted by interaction time to form sequence $V\{v_1, v_2, ..., v_n\}$, with the $(N+1)$-th item serving as the prediction item v_t. Then, each variable in sequence V is embedded to generate high-dimensional vectors, and this set of vectors is input into the self-attention module and gated recurrent network to generate a set of implicit logical variables. Next, this set of implicit logical variables is treated as a logical expression and solved via the logical symbol network to obtain a solution vector. Finally, this solution vector and a randomly generated base vector T are processed through Equation (3) to obtain the solution value of the logical expression, which serves as the predicted rating of the user for the target item.

3.1 Experimental Data

This paper uses three representative public datasets—MovieLens, Book-Crossing, and Amazon-E—to evaluate the recommendation effectiveness of the proposed IDCR model.

a) MovieLens [?]**:** A movie recommendation dataset established and maintained by Grouplens, commonly used in the recommendation field. It has relatively high data density, containing over 100,000 rating data from 900+ users on 1,000+ movies.

b) Book-Crossing [?]**:** A book recommendation dataset composed of ratings from 240,000 readers in the Book-Crossing community. It is very sparse, posing a significant challenge for deep learning methods based on statistical features.

c) Amazon-E [?]**:** An e-commerce recommendation dataset composed of online shopping ratings from Amazon users. It has moderate density and a very large data volume.

The basic information of the datasets is shown in Table 1.

3.2 Evaluation Metrics

This paper adopts the top-K recommendation method to evaluate the proposed model. For each item that a user has rated positively, 100 items that the user rated negatively or never interacted with are randomly selected to generate a test sequence, and the 101 items are ranked for recommendation. Normalized Discounted Cumulative Gain (NDCG) and Hit Ratio (HR) are used to measure recommendation effectiveness:

NDCG@K: Higher rankings have greater impact on the final result. A larger value indicates that the target item appears closer to the top of the TOP-K list, defined as:

$$NDCG@K = \frac{1}{|S|} \sum_{i=1}^{|S|} \frac{DCG_i@K}{IDCG_i@K}$$

where $DCG_i@K = \sum_{j=1}^{K} \frac{2^{rel_{ij}}-1}{\log_2(j+1)}$.

HR@K: Hit indicates whether the target item appears in TOP-K (1 if yes, 0 otherwise). HR@K represents the probability of hitting the target item across all test sets S, defined as:

$$HR@K = \frac{\text{number of hits}}{|S|}$$

To avoid randomness, this paper selects the average of eight metrics: HR@K (K=1,5,10,20,50) and NDCG@K (K=5,10,20) as comparison parameters.

3.3 Comparison Models and Parameter Settings

The proposed model is compared with six classical models (two based on matrix factorization, three based on neural networks, and LINN based on neural network and logical reasoning combination):

a) SVD++ [?]: An improved matrix factorization method that integrates explicit and implicit feedback.
b) BiasedMF [?]: A representative pairwise learning matrix factorization model focusing on item ranking over rating.
c) STAMP [?]: A model based on short-term attention and memory priority, using attention mechanisms to capture long-term and short-term user preferences.
d) GRU4Rec [?]: A gated recurrent network model for recommendation problems.
e) NARM [?]: A model combining attention mechanisms with gated recurrent networks, a popular sequential recommendation model.
f) LINN [?]: A model combining neural networks with logical reasoning. The proposed model is also based on neural network and logical reasoning combination, adding an implicit logic representation module to further improve performance on recommendation problems.

All models are implemented using PyTorch. For each model, the test results from the epoch with the best performance on the validation set are taken as the model' s optimal performance, and the average performance across multiple random seeds is taken as the model' s average performance metric. The hyperparameter settings for neural network baseline models are shown in Table 2.

3.4 Results and Analysis

The average performance of the seven models on the three datasets is shown in Table 3. The results demonstrate that the proposed model outperforms the six classical models on all three datasets, indicating that the method can effectively improve recommendation accuracy.

On the high-density and high-quality MovieLens and Amazon-E datasets, neural network-based methods significantly outperform matrix factorization methods, likely because neural network methods provide non-linear learning capabilities that can utilize more interaction information when facing feature-rich datasets. On the Book-Crossing dataset, neural network models other than the proposed method perform worse than matrix factorization methods, while matrix factorization models maintain their recommendation performance on the other two datasets. The analysis suggests that overly sparse datasets make it difficult for neural networks to capture sufficient features, leading to poor convergence.

The proposed method achieves the best performance on all three different datasets, primarily due to the added implicit logic representation module, which preserves the model' s reasoning capability while compensating for LINN' s difficulty in capturing high-order implicit information among variables and the information loss problem caused by serial computation graphs.

From the perspective of stability under sparse conditions, matrix factorization-based models show similar performance across different datasets, likely because they are less dependent on dataset quality. The performance of the other five neural network models declines significantly on the Book-Crossing dataset, but the proposed method' s decline is substantially lower than other neural network models. The analysis indicates that the added implicit logic representation module enhances model robustness, enabling good performance even when facing sparse data by mining implicit information among variables.

More detailed experimental results are shown in Tables 4-6. When different K values are selected, the model ranking shows slight variations. For example, on Book-Crossing dataset' s HR@1 and HR@5 metrics, SVD++ achieves excellent performance, while on Amazon-E dataset' s NDCG@5 and HR@1 metrics, LINN performs better. This may be because the proposed method' s consideration of implicit interactions among different items makes similar items closer in ranking, causing slight reductions in recommendation precision within very small ranges. Future work will consider introducing additional information to optimize embeddings and improve small-range recommendation precision.

3.5 Parameter Sensitivity Experiments

Parameter sensitivity tests are conducted on the proposed model' s embedding dimensions and the number of self-attention layers in the implicit logic representation module. Embedding dimensions are set to 64, 72, 84, 96, 108, and the number of self-attention layers is set to 1, 2, 3 for testing on the three datasets. The results in Figures 3-5 show that model performance improves significantly when the number of self-attention layers is 2, indicating that a 2-layer self-attention network can mine implicit high-order relationships among variables. Meanwhile, model performance also improves with increasing embedding dimensions. The best-performing embedding dimensions on the three datasets are 84, 96, and 96 respectively, suggesting that the richness of hidden interaction information varies slightly across datasets with different structures.

---

4 Conclusion

Based on the idea of transforming recommendation problems into logical expressions for solving, this paper proposes an implicit deep collaborative reasoning algorithm that fuses propositional logic with neural networks. The algorithm treats a set of logical variables in the logical expression as a sequence and uses multi-layer self-attention and gated recurrent networks to construct an implicit logic encoder that mines implicit interaction information among variables and generates implicit logical variables. This enables the model to utilize high-order interaction relationships and information among different variables during solving, improving its ability to solve complex real-world problems. Meanwhile, the memory mechanism of gated recurrent networks also alleviates the information loss problem for earlier units caused by serial solving. Experimental results on three public datasets—MovieLens, Book-Crossing, and Amazon-E—demonstrate that the proposed method significantly outperforms baseline methods.

The proposed method has certain limitations. For example, it does not consider the influence of prior knowledge from basic logical rules on symbolic modules, nor does it fully utilize rich external auxiliary information. Instead, it mines implicit interaction information among variables through model training. Future work will consider adding more auxiliary information about users and items to enrich features, while attempting to use basic logical rules to strengthen model reasoning capability.

References

[1] Cui Zhihua, Xu Xianghua, Xue Fei, et al. Personalized recommendation system based on collaborative filtering for IoT scenarios [J]. IEEE Transactions on Services Computing, 2020, 13 (4): 685-695.

[2] Liu Hualing, Ma Jun, Zhang Guoxiang. Summary of research on content recommendation algorithm based on deep learning [J]. Computer Engineering, 2021, 47 (7): 1-12.

[3] He Xiangnan, Zhang Hanwang, Kan M Y, et al. Fast matrix factorization for online recommendation with implicit feedback [C]// Proc of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. 2016: 549-558.

[4] Cheng H T, Koc L, Harmsen J, et al. Wide & deep learning for recommender systems [C]// Proc of the 1st workshop on deep learning for recommender systems. 2016: 7-10.

[5] Dacrema M F, Cremonesi P, Jannach D. Are we really making much progress? A worrying analysis of recent neural recommendation approaches [C]// Proc of the 13th ACM Conference on Recommender Systems. 2019: 101-109.

[6] Rendle S, Krichene W, Zhang L, et al. Neural collaborative filtering vs. matrix factorization revisited [C]// Fourteenth ACM Conference on Recommender Systems. 2020: 240-248.

[7] Bengio Y. From system 1 deep learning to system 2 deep learning [C]// Thirty-third Conference on Neural Information Processing Systems. 2019.

[8] Shi Shaoyun, Chen Hanxiong, Ma Weizhi, et al. Neural logic reasoning [C]// Proc of the 29th ACM International Conference on Information & Knowledge Management. 2020: 1365-1374.

[9] Chen Hanxiong, Li Yunqi, Shi Shaoyun, et al. Graph Collaborative Reasoning [C]// Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining. 2022: 75-84.

[10] Chen Hanxiong, Shi Shaoyun, Li Yunqi, et al. Neural collaborative reasoning [C]// Proc of the Web Conference 2021. 2021: 1516-1527.

[11] Fan Yujia, Zhang Yongfeng. Neural Logic Analogy Learning [J]. arXiv preprint arXiv: 2202.02436, 2022.

[12] Huang Zhenya, Liu Qi, Zhai Chengxiang, et al. Exploring multi-objective exercise recommendations in online education systems [C]// Proc of the 28th ACM International Conference on Information and Knowledge Management. 2019: 1261-1270.

[13] Koren Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model [C]// Proc of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. 2008: 426-434.

[14] Adomavicius G. Context-aware recommender systems [M]// Recommender systems handbook. Springer, Boston, MA, 2011: 217-253.

[15] Karatzoglou A, Amatriain X, Baltrunas L, et al. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering [C]// Proc of the fourth ACM conference on Recommender systems. 2010: 79-86.

[16] Koren Y. Collaborative filtering with temporal dynamics [C]// Proc of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. 2009: 447-456.

[17] Zhang Yongfeng, Ai Qingyao, Chen Xu, et al. Joint representation learning for top-n recommendation with heterogeneous information sources [C]// Proc of the 2017 ACM on Conference on Information and Knowledge Management. 2017: 1449-1458.

[18] He Ruining, McAuley J. VBPR: visual bayesian personalized ranking from implicit feedback [C]// Proc of the AAAI Conference on Artificial Intelligence. 2016, 30 (1).

[19] Ai Qingyao, Azizi V, Chen Xu, et al. Learning heterogeneous knowledge base embeddings for explainable recommendation [J]. Algorithms, 2018, 11 (9): 137.

[20] Hsieh C K, Yang Longqi, Cui Yin, et al. Collaborative metric learning [C]// Proc of the 26th international conference on world wide web. 2017: 193-200.

[21] Salakhutdinov R, Mnih A, Hinton G. Restricted Boltzmann machines for collaborative filtering [C]// Proc of the 24th international conference on Machine learning. 2007: 791-798.

[22] He Xiangnan, Liao Lizi, Zhang Hanwang, et al. Neural collaborative filtering [C]// Proc of the 26th international conference on world wide web. 2017: 173-182.

[23] Ebesu T, Shen Bin, Fang Yi. Collaborative memory network for recommendation systems [C]// The 41st international ACM SIGIR conference on research & development in information retrieval. 2018: 515-524.

[24] Jiang Zhuxi, Zheng Yin, Tan Huachun, et al. Variational deep embedding: an unsupervised and generative approach to clustering [C]// Proc of the 26th International Joint Conference on Artificial Intelligence. 2017: 1965-1972.

[25] Hohenecker P, Lukasiewicz T. Ontology reasoning with deep neural networks [J]. Journal of Artificial Intelligence Research, 2020, 68: 503-540.

[26] Makni B, Hendler J. Deep learning for noise-tolerant RDFS reasoning [J]. Semantic Web, 2019, 10 (5): 823-862.

[27] Johnson J, Hariharan B, Van Der Maaten L, et al. Inferring and executing programs for visual reasoning [C]// Proc of the IEEE International Conference on Computer Vision. 2017: 2989-2998.

[28] Yi Kexin, Wu Jiajun, Gan Chuang, et al. Neural-symbolic VQA: disentangling reasoning from vision and language understanding [C]// Proc of the 32nd International Conference on Neural Information Processing Systems. 2018: 1039-1050.

[29] Yang Fan, Yang Zhilin, Cohen W W. Differentiable learning of logical rules for knowledge base reasoning [C]// Proc of the 31st International Conference on Neural Information Processing Systems. 2017: 2316-2325.

[30] Dong Honghua, Mao Jiayuan, Lin Tian, et al. Neural logic machines [C]// International Conference on Learning Representations. 2018.

[31] Harper F M, Konstan J A. The movielens datasets: history and context [J]. ACM Transactions on Interactive Intelligent Systems (TIIS), 2015, 5 (4): 1-19.

[32] Ziegler C N, McNee S M, Konstan J A, et al. Improving recommendation lists through topic diversification [C]// Proc of the 14th international conference on World Wide Web. 2005: 22-32.

[33] He Ruining, McAuley J. Ups and downs: modeling the visual evolution of fashion trends with one-class collaborative filtering [C]// Proc of the 25th international conference on world wide web. 2016: 507-517.

[34] Rendle S, Freudenthaler C, Gantner Z, et al. BPR: Bayesian personalized ranking from implicit feedback [C]// Proc of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. 2009: 452-461.

[35] Liu Qiao, Zeng Yifu, Mokhosi R, et al. STAMP: short-term attention/memory priority model for session-based recommendation [C]// Proc of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2018: 1831-1839.

[36] Hidasi B, Karatzoglou A, Baltrunas L, et al. Session-based recommendations with recurrent neural networks [J]// International Conference on Learning Representations (2016).

[37] Li Jing, Ren Pengjie, Chen Zhumin, et al. Neural attentive session-based recommendation [C]// Proc of the 2017 ACM on Conference on Information and Knowledge Management. 2017: 1419-1428.