A method for improving the diversity of movie recommendation with knowledge graph

MA Wen-kai , WEN Yuan* , HOU Xia

Computer School, Beijing Information Science and Technology University, Beijing 100101, China
E-mail: yuanwen@bistu.edu.cn

This paper proposes a diversity enhancement method with knowledge graph for improving the diversity of recommendation lists in movie recommendation systems. Firstly, this method uses the IMDb movie dataset to build a knowledge graph of movie-character and movie-company relationships, and expands the fine-grained connectivity between movie entities through 20 relationships. Secondly, this paper proposes an aggregation algorithm to improve the diversity feature expression of movies in the knowledge graph module, and the outputs are embedded into the recommendation module as auxiliary information. Finally, the multi-layer perceptron is used to mine the deep expression of user and movie features. Considering the important impact of movie category on the diversity of recommendation list, this paper improved the calculation method of diversity. The experimental results show that on the MovieLens and Netflix dataset, the proposed method greatly improves the diversity while ensuring the accuracy.

Keywords: Movie Recommendation, Diversity, Knowledge Graph

1. INTRODUCTION

With the rapid development of information technology (IT) and the Internet, watching movies through the Internet has become a habit for many people. In the past two years, as COVID-19 has become a norm, offline cinema viewing has been restricted. Many people choose to watch movies online, and some films are gradually adapted to the current situation and screened online. When people are faced with massive movie resources, they cannot quickly get their favorite movie content. It makes the information overload of movies worse [1, 2]. The recommendation system can effectively alleviate the above problems. However, the traditional movie recommendation system pays too much attention to the accuracy of users' prediction of movie ratings. The movie list recommended for users is too relevant, and the freshness and satisfaction of users will decline over time. Accuracy alone is not enough to fully reflect the potential interests of users; hence it is important to improve the diversity of movie recommendation list [3]. Different users have some differences in fondness degrees for various types of movies. Some people like romantic narratives, others like sci-fi adventures, and different types of movies can increase diversity to the recommendation list. But the recommendation list diversity extends beyond the genre. Different languages, styles of directors, and cast members may affect the effect of the recommendation list diversity. Existing studies on movie recommendation mostly use users' explicit historical rating records to match with users' interest preferences. No more information about the movie was introduced. Accordingly, this paper considers the knowledge graph in film recommendation system. Through the rich entity relations of the knowledge graph, the fine-grained representation of movie entities can be obtained to improve the recommendation list diversity.

2. RELATED WORKS

2.1. Recommendation Diversity

Diversity is not the only purpose of the movie recommendation system. The purpose of the system is to display diversified results for users on the premise of ensuring the accuracy of the recommendation, so that the demands of users can be met as much as possible. For the movie recommendation system, a single-purpose recommendation list will not only reduce the user's interest in watching movies, but also have a certain impact on the audience rating of other movies and TV works. There are currently two ways to increase the diversity of recommendation lists. One is to introduce some diversity factors by improving the recommendation model before rating prediction; the other one is to reorder the recommended candidate set after the rating prediction with a new rule to improve the diversity of reordered recommendations.

Reference [4] proposed an adaptive attribute-based diversity approach. Based on rating data, simulating each user's propensity for diversity in different item attributes (such as film genres, directors, and actors). The item attributes are then assigned a weight that defines the similarity measure for the item (attributes that show a greater diversity of preferences will be given higher weight). This approach can consider users' diverse demands and customize the diversity degree of top-N

recommendation list. Reference [5] increases recommendation diversity by considering the influence of users with similar interest and different interests. Item prediction for target users is calculated by linear combinations of positive ratings for users with similar interests and negative ratings for users with different interests. Reference [6] relaxes the binary optimization problem of similarity and diversity into a trust domain problem. A quantitative strategy is proposed to solve the optimization problem to maximize the diversity of recommendation lists. But as long as similarity is the main selection criterion, the probability of recommending new items is greatly reduced. Reference [7] introduced a regularization framework with fairness awareness and proposed RankALS algorithm mechanism to balance the accuracy and the coverage ratio based on the related work of diversity optimization. The long tail coverage and ranking accuracy was modeled as a multi-objective optimization problem. The maximum marginal correlation (MMR) proposed in reference [8] is one of the earliest greedy reordering diversification methods. Reference [9] uses a heuristic algorithm based on classification similarity to increase the diversity in the recommendation list. A weighting factor is defined to control contributions from both sets. Therein, a collection contains items that are similar to the user's preference profile for basic attributes; the other collection contains items in reverse order of similarity to the user profile. Reference [10] proposed a diversity method based on the potential characteristics of matrix decomposition model. Maximizing the width of the itemset (that is, the distance perpendicular to the user vector) to achieve diversity. And minimizing the height of the item set (that is, the distance parallel to the user vector) to ensure the accuracy of the recommendation. Finally, the greedy algorithm similar to reference [9] is utilized to generate recommendation results.

2.2. Recommendation Based on Knowledge Graph

Knowledge Graph (KG) is a semantic network graph composed of nodes and edges, which contains rich semantic knowledge. It is applied to recommendation systems characterized by accuracy, diversity, and interpretability [11]. Recommendations based on the KG is generally divided into three categories. 1) Connection-based; 2) Embedding-based; 3) Mix-based.

The connection-based recommendation regards the KG as a heterogeneous information network graph and realizes the recommendation by calculating node similarity through the connection relation between different entities. The Hete-MF method proposed in reference [12] extracted *L* different meta-paths and calculated item similarity in each path. The user-item scoring matrix is decomposed and graph regularization is used to constrain the distance of potential vectors of similar items connected by meta paths. The FMG method proposed in reference [13] introduced meta-graph instead of meta-path. Because meta-graph contains richer connectivity information than meta-path, and the FMG method can capture the similarity between entities more accurately. The SemRec method proposed in reference [14] considers the interaction between items that users liked and disliked in the past. The model uses weighted HIN and weighted meta-path to integrate attribute values in links. By modeling positive and negative preference patterns, these paths can be used to describe more accurate item relationships and similarity of users' interest in movies, thereby spreading real user preferences.

The idea of recommendation based on embedding technique is to map entities and relationships in KG in low dimension and introduce entity and relational eigenvector into recommendation model. Reference [15] can directly model user preferences by introducing users into the graph to establish a user item graph. Reference [16] proposed the RKGE (Recurrent Knowledge Graph Embedding) method to learn explicit embedding of relative paths connecting users, which allows to model the user item relationship directly, automatically mining the path relationship between the user u_i and the item v_j without having to manually define the meta-path. The MKR method proposed in reference [17] consists of recommendation module and the KGE (Knowledge Graph Embedding) module. The former learns the potential representation of users and items, while the latter uses semantically matched the KGE to learn the representation of item-associated entities. The two parts are connected by a cross-compression unit to transfer knowledge and share the regularization of items in the recommendation module and entities in the KG.

The idea of recommendation based on mixing is to combine connection and embedding, to acquire users' preferences and interests through connection, and to learn users' preference characteristics through embedding. The RippleNet method proposed in reference [18] is used to simulate interest propagation of users. For each user, the RippleNet method views their historical interests as a seed set in the KG. Then, user interest is continuously expanded outward along the connection of the KG to form the distribution of user interest on the KG. Reference [19] proposed that the final representation of candidate *vj* could be modeled by embedding the entity from the far neighbor of candidate *vj* to *vj* itself into the KG. The AKUPM (Attention-Enhanced Knowledge-Aware User Preference Model) proposed in reference [20] firstly applies TransR to entity representation. In each propagation process, the AKUPM learns the relationship between entities with self-attention layer and propagates user preference to different entities with bias. Finally, the self-attention mechanism is used to aggregate the different order neighborhood embeddings of interaction items to obtain the final user representation.

3. RECOMMENDATION DIVERSITY METHOD

In this paper, the recommendation diversity method is divided into two modules (the knowledge graph and the recommendation), using sequential learning method to connect them. **Figs. 1** shows the specific details. Firstly,

knowledge graph module extracts relevant movies from the Internet Movie Database (IMDb) to construct a sub-KG and vectorizes it by TransE algorithm. Then we propose an aggregation algorithm for entity feature matrix, and the output of this module is connected to the aggregating layer of the recommendation module.

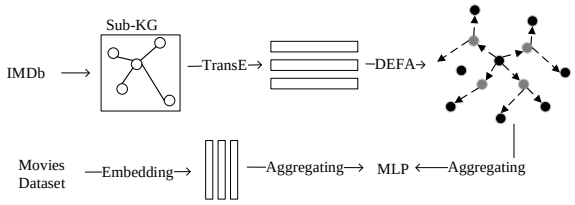


Fig. 1 Module process.

3.1. Formal Definition of Recommendation Task

Existing user set $U=\{u_1,u_2,...,u_m\}$ and movie set $V=\{v_1,v_2,...,v_n\}$, user-movie explicit rating record $Y=\{y_{uv} \in\{1,5\} | u \in U, v \in V\}$ and knowledge graph $G=(E,R)$. As shown in Eq. (1), the movie recommendation problem is a prediction task. Using the user-movie interaction matrix Y and the knowledge graph G to predict potential interest of u in v, where θ is all parameters of the function f.

$$\hat{y}_{uv} = f(u, v \mid Y, G; \theta) \tag{1}$$

In this paper, $G=(E, R)$ is represented the knowledge graph, where E is the set of entities corresponding to the nodes in the graph, and R is the set of relationships between the entities corresponding to the edges. Any edge in the knowledge graph represented by a triple (h, r, t), where h, r, t represented the head entity, relationship and tail entity of the triple, i.e. G={(h, r, t) |h∈E, r∈R, t∈E}.

3.2. Knowledge Graph Construction and Vectorization

This paper uses the IMDb dataset to construct a sub-knowledge graph (sub-KG).

The IMDb is an online database containing numerous movies, and it has included about 2.8 million movies so far. This paper uses the IMDb subset to construct the movie knowledge graph, and the constructed triples contain all movies information with IMDb numbers from *tt0017150* to *tt9916754*. The head entity of the triple is all movie entity, and the tail entity is character or company entity (i.e. movie-character relationship, movie-company relationship). And there are 20 kinds of relationships in total. The relation triple makes the tail entity also a meaningful body with an independent page, and the movie entity no longer has attribute relationships such as genre or release year. On the one hand, the movie dataset selected in this paper already contains the basic description files of movies, which can model movie-related attribute relationships; on the other hand, attribute triples cannot mine more movie diversity features. This paper hopes to explore deep-level diversity connectivity between movie entities, rather than meaningless attribute relationships such as year or movie genre.

The method of constructing the sub-KG is as follows:

We treat movie-character or movie-company as 1 hop, and the entity required in this paper is that the movies in MovieLens/Netflix can be connected in 4 hops through 20 relations. Constructing these relationship triples into a movie knowledge graph. The constructed triples are learned by the TransE algorithm for embedding and training of the recommendation module [21]. As shown in Eq. (2), the idea of TransE is to make $h+r$ equal to t as much as possible by continuously adjusting h, r and t, and the function is to translate triples into embedding word vectors.

$$\overline{h} + \overline{r} \approx \overline{t} \tag{2}$$

3.3. Aggregation Algorithm for Entity

Entity vector matrix E and relation vector matrix R are obtained after knowledge graph vectorization. In this section, a diverse entity feature aggregation algorithm (DEFA) is proposed. The movie entity features are aggregated in diversity as the output of the knowledge graph module. The purpose is to enhance movie features expressive. As shown in **Figs. 2**, *Ai* represents a movie entity, and *Bi* represents a character or company entity. *(A1,r1,B1)* and *(A2,r2,B1)* are two triples, that means movie *A1* and *A2* can be connected by two hops through relations *r1* and *r2*, and *B1* is an intermediate entity (a character or a company entity). The relation *r1* and *r2* may be the same or different. Such as *B1* is the director of *A1* and *A2*, or *B1* is the director of *A1* but the writer of *A2*, because a character or a company entity can have different functions. Regardless of whether *r1* and *r2* are the same, we believe that the two movies connected by the same character or company are likely to be similar, and it is difficult to express more diversity. Therefore, the idea of the DEFA is reachable and diverse. Taking movie *A3* as an example, find the movie set *Q1*(*Ai* ∈*Q1*) that is connected with *A3* after two hops and the movie set *Q2*(*Aj* ∈*Q2*) that is connected to *A3* after four hops but cannot be connected by two hops.

By connecting two different company or character entities, the diversity between two movie entities can be increased in the distance of four hops. And the two movie entities cannot be connected by two hops, which further reduces the correlation between the two entity while ensuring the connection.

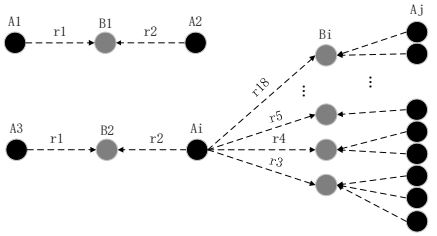


Fig. 2 DEFA algorithm process.

The specific process of the DEFA is as follows, where *D(A1, A2; r1, r2)* indicates that the movie entities *A1* and *A2* can be connected by two hops through the relation *r1* and *r2*, and *D(A1, A2)* indicates that *A1* and *A2* cannot be connected by two hops.

Algorithm DEFA

Input: Movie entity set *A*, character or company entity set *B*, relationship set *R*, movie set *M* in dataset.
Output: Movie entity set *Q1*, *Q2*.
Begin：
(1) For *m0* ∈*M*
(2) For *a0* ∈*A*
(3) If *D(m0,a0;ri,rj)*
(4) *Q1.append(a0)*
(5) For *a1* ∈*A(a1* ≠*a0)*
If *D(a0,a1;ri,rj) && D(m0,A1)*
(6) *Q2.append(a1)*
(7) Output *Q1,Q2*

Assuming that the lengths of *Q1* and *Q2* are m and n respectively for movie *m0*, so the final output of the knowledge graph module is expressed as the weighted sum of m0, *Q1*, and *Q2*, as shown in Eq. (3):

$$0.7m0 + 0.2\sum_{i=1}^{i=m} Q1 + 0.1\sum_{i=1}^{i=n} Q2 \tag{3}$$

3.4. Recommended Modules

The recommendation module completes the training of user and movie feature vectors, and outputs the user top-N recommendation list.

The input of this module is the MovieLens/Netflix dataset, which includes a user-movie rating matrix and basic description information for users and movies. The movie title is a text feature processed by a convolutional neural network (CNN), and the rest of the discrete features are normalized.

Embedding layer, after the text feature obtains the embedding vector of each word corresponding to the movie title from the embedding matrix, convolution and maximum pooling are performed on the embedding layer using convolution kernels of different sizes. And the final output text feature is as shown in Eq. (4), where T_j represents the matrix representation of movie *j* after the embedding layer, assuming there are *k* convolution kernels, and *e* represents the feature map after max pooling.

$$m^{title} = \mathrm{CNN}(W, T_j) = \tanh(W_2(\tanh(W_1 e + b_1)) + b_2) \tag{4}$$

The rest of the non-text feature inputs form a k dimensional embedding vector (different features have different k value). After a fully-connected layer for spatial transformation, each features of different distributions can be projected in the same vector space. And finally the output ui and vj pass through a fully-connected layer to ensure that the user feature vector and movie feature vector have the same dimension.

$$\begin{aligned} u_i &= concatenate(u_{id}^{16}, u_{age}^{8}, u_{job}^{8}, u_{gender}^{8}) \\ v_j &= concatenate(m_{id}^{16}, m_{genres}^{8}, m_{title}^{32}) \end{aligned} \tag{5}$$

As shown in Eq. (6), the aggregating layer adds the movie entity feature vector q_j of the knowledge graph and the movie feature vector v_j of the embedding layer bit by bit (the two feature vectors maintain the same dimension). The result is spliced with the user feature vector u_i as the output of this layer, the input of the multi-layer perceptron.

$$L^{input} = \mathrm{concatenate}(u_i, (q_j \oplus v_j)) \tag{6}$$

$$\begin{aligned} L_1 &= a_1(W_1^T x + b_1) \\ L_2 &= a_2(W_2^T L_1 + b_2) \\ &\cdots \\ L_n &= a_n(W_n^T L_{n-1} + b_n) \end{aligned} \tag{7}$$

After the multi-layer perceptron shown in Eq . (7), the prediction is sorted in descending order to output the user's top-N recommendation list.

4. EXPERIMENTS

4.1. Datasets

This paper uses the MovieLens dataset, which is provided by the Group Lens research group of Minnesota State University, and Netflix dataset, which is a small subset of the data provided as part of the Netflix Prize. The specific information is shown in **Table 1**. Each dataset contains the user's explicit historical rating data. The user's implicit feedback data has limited meaning. 0 and 1 can only feedback a weak user preference. For example, if you click on a product on Taobao, it does not mean that you will like it. And the user's rating of the movie is a strongly correlated preference representation. The rating between 1-5 expresses the user's interest of the movie.

In this experiment, the dataset has no new users. So there is no cold start problem (each user has rated at least 20 movies). And 80% of the dataset is randomly selected as the training set, and the remaining 20% is used as the test set. When constructing the knowledge graph, it is necessary to consider that some movie cannot be queried in IMDb. The MovieLens-1M dataset actually contains 3837 movies, the MovieLens-100K dataset actually contains 1664 movies, and the Netflix dataset actually contains 1801 movies. Finally, the knowledge graph contains 38,392 entities and 20 kinds of relationships, forming 130,733 triples.

Table. 1 Dataset Statistics.

Dataset	User	Movie	Interaction	sparsity
MovieLens-1M	6040	3886	1000209	4.26%
MovieLens-100K	943	1682	100000	6.30%
Netflix	28978	1821	227773	0.43%

4.2. Evaluation Indicators

In order to measure the accuracy and diversity of the results, this paper adopts Hit Radio (HR), Normalized Discounted Cumulative Gain (NDCG) and Movie Intra-List Diversity (MILD).

(1) HR is used to measure the recall rate of the movie recommendation list.

$$HR@N = (NumberOfHits@N) / GT \quad (8)$$

(2) NDCG is composed of cumulative gain (CG) and discounted cumulative gain (DCG), which is used to measure the accuracy of movie recommendation list sorting. CG only considers the degree of correlation and does not consider the location factor, so DCG is to divide the result of each CG by a discount value.

$$NDCG@N = \frac{DCG@N}{IDCG@N} \quad (9)$$

The intra-list diversity (ILD) is used to measure the intra-user diversity, and it is defined as Eq. (10). This paper proposes a method for measuring the intra-user diversity of movies (MILD) with the basis of ILD, which adds the diversity impact of movie categories in the recommendation list. As shown in Eq. (11), it is known that there are 18 genres of movies in the MovieLens/Netflix dataset, and *G(Lu)* represents the number of movie categories in user's recommendation list.

$$\text{ILD@N} = \frac{2}{N(N-1|)} \sum_{i \in L(u),\, j \in L(u), i \neq j} dist(i, j) \quad (10)$$

$$MILD@N = \frac{G(L_u)}{18} \times ILD@N \quad (11)$$

4.3. Experimental Parameters

This experimental environment uses TensorFlow2.8.0, Python3.9, and uses 4 NVIDIA Tesla P100 16GB GPUs for parallel training. The recommendation module uses the Root Mean Square Error (RMSE), and the initialization parameter is Gaussian random distribution. The initialized user feature and movie feature dimension are 32. Adam is selected for gradient descent optimization, so that each element in the independent variable of the algorithm objective function has its own learning rate. The multi-layer perceptron chooses ReLU as activation function. After experimental verification, when the learning rate is 0.001, the training batch is 128, and the multi-layer perceptron hidden layer is 3 layers, the training effect is the best, and the RMSE is 0.8875.

4.4. Comparative Experiment

The diversity enhancement method proposed in this paper (DEFA-KG) will be compared with the following four experiments:

(1) ItemCF-DP: a top-N recommendation algorithm based on user diversity preferences [22].

(2) MLSI: A recommendation diversity optimization algorithm that incorporates users' long-term and short-term interests [23].

(3) UBCF: User-based collaborative filtering algorithm.

(4) UCB: Collaborative filtering algorithm based on user clustering [24].

The experimental results are shown in **Figs. 3**. On the MovieLens-100K dataset, DEFA-KG has a small improvement in HR and MILD, which proves the feasibility of the experiment. The small decrease in NDCG is because the ranking of movies in the list has changed, but the movie that user expects is still in the recommended list.

On the MovieLens-1M dataset, when N=10 or N=15, HR and NDCG of DEFA-KG are slightly lower than MLSI, but MILD has a certain improvement. Because the movie that user expects has been removed from the top-N recommendation list, resulting in a drop in hit rate. If the movie that user expects is not removed from the recommendation list, then the ranking of different movies will only affect the sorting accuracy without affecting recommendation accuracy. Although HR is slightly reduced, it is generally higher than the two collaborative filtering algorithms UBCF and UCB, in exchange for the improvement of recommendation diversity.

On the Netflix dataset, HR and NDCG of DEFA-KG are both slightly lower than MLSI, indicating that the movies previously at the top of the recommendation list are removed from the recommendation list. But MILD has been significantly improved, which is in line with the expected results of the experiment.

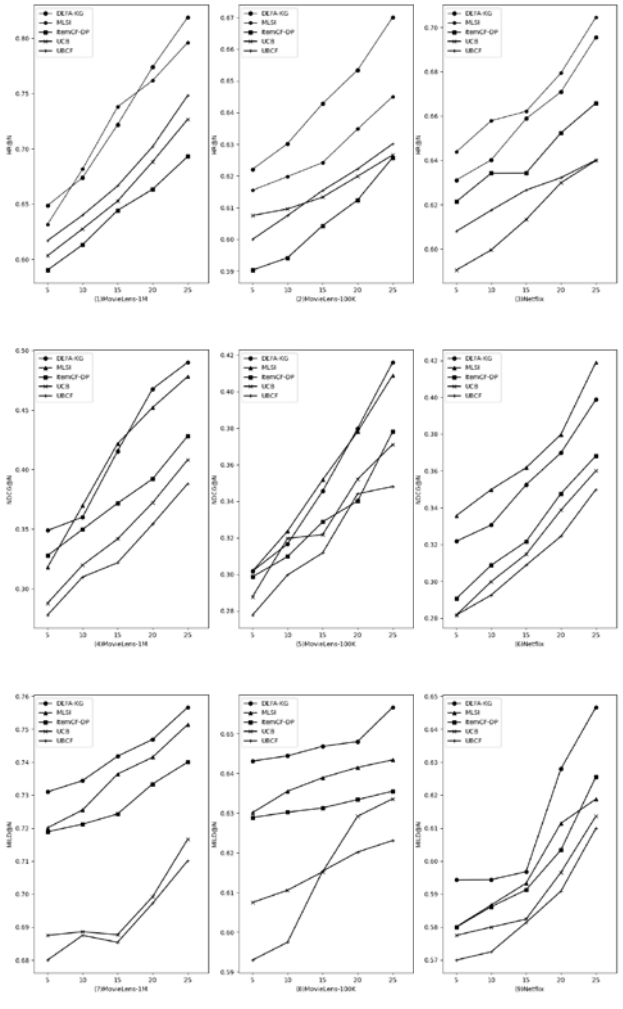


Fig. 3 Results of different datasets.

4.5. Ablation Study

To prove the positive effect of knowledge graph, this paper adds an ablation study. As shown in **Figs. 1**, non-KG means the aggregating layer does nothing, and the output dimension of non-kg is the same as that of KG.

As shown in **Table 2**, the length of recommendation list N=10. The results show that the diversity of the top-10 recommendation list increases by 5.1%, 1.4% and 4.8% respectively with the knowledge graph, but the HR of the recommendation list decreases by 0.5%, 0.8% and 1.7% respectively with the knowledge graph. Like the above comparative experiments, it is proved to be feasible to exchange a small loss of recommendation accuracy for a large increase in recommendation diversity.

Table. 2 Results of the Ablation Study.

	Dataset	HR	NDCG	MILD
KG	MovieLens-1M	0.692	0.401	0.433
	MovieLens-100K	0.524	0.327	0.301
	Netflix	0.640	0.331	0.594
non-KG	MovieLens-1M	0.697	0.385	0.392
	MovieLens-100K	0.532	0.311	0.287
	Netflix	0.657	0.304	0.546

5. CONCLUSION

The recommendation system effectively alleviates the problem of movies overload and improves the user experience, so it is widely used in well-known movie and video websites at home and abroad and brings huge commercial value.

This paper explores the diverse features of movies by building a relationship-based knowledge graph of character and company and proposes an aggregation algorithm to obtain the movies' diversity feature expression. Then it is embedded into the recommendation module for training. We improve the calculation of diversity and add the category influence in movies. The results indicate the effectiveness of the DEFA for recommending diversity movies.

As future works, we plan to use multi-modal dataset to extend our knowledge graph and get richer feature expressions.

Acknowledgements

This study is sponsored by the promoting the promotion of classified development of colleges and universities—the postgraduate education reform project of the School of Computer Science (project code: 5112211037).

REFERENCES:

[1] Wang Ji-zhe,Huang Pi-pei,Zhao Huan,et al. Billio n-scale Commodity Embedding for E-commerce Reco mmendation in Alibaba[C]// Knowledge Discovery & Data Mining,2018:839-848.

[2] Ricci F, Rokach L, Shapira B. Introduction to re commender systems handbook[M]//Recommender syste ms handbook. Springer, Boston, MA, 2011: 1-35.

[3] Kaminskas M, Bridge D. Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems[J]. ACM Transactions on Interactive Intelligent Systems (TiiS), 2016, 7(1): 1-42.

[4] Di Noia T, Ostuni V C, Rosati J, et al. An analysis of users' propensity toward diversity in recommendations[C]//Proceedings of the 8th ACM Conference on Recommender Systems. 2014: 285-288.

[5] Zeng W, Shang M S, Zhang Q M, et al. Can dissimilar users contribute to accuracy and diversity of personalized recommendation?[J]. International Journal of Modern Physics C, 2010, 21(10): 1217-1227.

[6] Zhang Mi, Hurley N.Avoiding monotony: improvi ng the diversity of recommendation lists[C]//Proceedin gs of the 2008 ACM conference on Recommender s ystems. 2008: 123-130.

[7] Abdollahpouri H,Burke R,Mobasher B.Controlling popularity bias in learning-to-rank recommendation[C] //Proceedings of the eleventh ACM conference on rec ommender systems. 2017: 42-46.

[8] Carbonell J, Goldstein J. The use of MMR, diversity-based reranking for reordering documents and producing summaries[C]//Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. 1998: 335-336.

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

[10] Willemsen M C, Graus M P, Knijnenburg B P. Understanding the role of latent feature diversification on choice difficulty and satisfaction[J]. User Modeling and User-Adapted Interaction, 2016, 26(4): 347-389.

[11] Ji S X, Pan S R, Cambria E, et al. A Survey on Knowledge Graphs: Representation, Acquisition, and Applications[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021. DOI: 10.1109/TNNLS.2021.3070843.

[12] Yu X, Ren X, Gu Q Q, et al. Collaborative Filtering with Entity Similarity Regularization in Heterogeneous Information Networks [C]//Proceedings of IJCAI-13 HINA Workshop. 2013: 27.

[13] Zhao H, Yao Q M, Li J D, et al. Meta-Graph Based Recommendation Fusion over Heterogeneous Information Networks[C]//Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2017: 635-644.

[14] Shi C, Zhang Z, Luo P, et al. Semantic path based personalized recommendation on weighted heterogeneous information networks[C]//Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. 2015: 453-462.

[15] Palumbo E, Rizzo G, Troncy R. Entity2Rec: Learning User-Item Relatedness from Knowledge Graphs for Top-N Item Recommendation[C]//Proceedings of the 11th ACM Conference on Recommender Systems. 2017: 32-36.

[16] Sun Z, Yang J, Zhang J, et al. Recurrent Knowledge Graph Embedding for Effective Recommendation[C]//Proceedings of the 12th ACM Conference on Recommender Systems. 2018: 297-305.

[17] Wang H, Zhang F, Zhao M, et al. Multi-task feature learning for knowledge graph enhanced recommendation[C]//The world wide web conference. 2019: 2000-2010.

[18] Wang H W, Zhang F Z, Wang J L, et al. RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems[C]//Proceedings of the 27th ACM International Conference on Information and Knowledge Management. 2018: 417-426.

[19] Wang H, Zhao M, Xie X, et al. Knowledge graph convolutional networks for recommender systems[C]//The world wide web conference. 2019: 3307-3313.

[20] Tang X, Wang T, Yang H, et al. AKUPM: Attention-enhanced knowledge-aware user preference model for recommendation[C]//Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019: 1891-1899.

[21] Bordes A, Usunier N, Garcia-Duran A, et al. Translating embeddings for modeling multi-relational data[J]. Advances in neural information processing systems, 2013, 26.

[22] Liu Li. A top-N recommendation algorithm based on user diversity preferences[J]. Computer Engineering and Applications, 2021, 57(17): 116-121.

[23] Xia Ruiling, Li Guoping, Wang Guozhong. Diversity optimization of recommendation algorithms incorporating long-term and short-term interests of users[J]. Computer Age, 2021(10):44-48.

[24] Zha Wenqin, Liang Changyong, Cao Lei. Collaborative filtering recommendation method based on user clustering[J]. Computer Technology and Development, 2009,19(06):69-71+75.