Top-N Collaborative Filtering Recommendation Algorithm Based on Knowledge Graph Embedding

Ming Zhu, De-sheng Zhen(✉), Ran Tao, You-qun Shi, Xiang-yang Feng, and Qian Wang

School of Computer Science and Technology, Donghua University, Shanghai 201620, China
{zhuming, taoran, yqshi, fengxy}@dhu.edu.cn, 2397646126@qq.com, 2171834@mail.dhu.edu.cn

Abstract. The traditional collaborative filtering recommendation algorithm only uses the item-user rating matrix without considering the semantic information of the item itself, resulting in a problem that the recommendation accuracy is not high. This paper proposes a Top-N collaborative filtering recommendation algorithm based on knowledge graph embedding. The knowledge graph embedding is used to learn a low-dimensional vector for each entity and relationship in the knowledge graph, while maintaining the structure and semantic information of the original graph in the vector. By calculating the semantic similarity between items, the semantic information of the item itself is incorporated into the collaborative filtering recommendation. The algorithm makes up for the defect that the collaborative filtering recommendation algorithm does not consider the knowledge information of the item itself, and enhances the effect of collaborative filtering recommendation on the semantic level. The experimental results on the MovieLens dataset show that the algorithm can get higher values on precision, recall and F1 measure.

Keywords: Collaborative filtering · Knowledge graph embedding · Recommendation algorithm · Semantic similarity

1 Introduction

Collaborative filtering recommendation algorithm is one of the most widely used algorithms in recommender system, and has been successfully applied in e-commerce [1], news media [2] and other fields, and has been receiving much attention in the academic community. The collaborative filtering algorithm discovers the user's preferences by mining the user's historical behavior data, and classifies the users based on different preferences and recommends similar products. Collaborative filtering algorithms fall into two categories, namely user-based collaborating algorithm [3] and item-based collaborative filtering [4]. Literature [5] proposes a collaborative filtering algorithm based on user characteristics and expert opinions. The algorithm analyzes user characteristics, compares the similarity between users and experts, and then calculates the similarity matrix to reduce the sparseness of the data set and improve the prediction accuracy. In literature [6], an improved cluster-based collaborative filtering algorithm is proposed, which uses clustering algorithm to cluster users and projects respectively, and then uses improved similarity measure to find the nearest neighbors of users and clusters. Finally the candidate set recommended by the project. The algorithm effectively solves the problem of new users, and can also reflect the user's interest changes through multi-dimensional user images. Literature [7] proposes a project-based collaborative filtering algorithm with low time and space complexity. Then interactive iterations are introduced to reflect the user's latest preferences and improve user satisfaction. Compared with the traditional project-based CF algorithm, it has better recall and precision. The existing various collaborative filtering algorithms basically use information such as user explicit feedback datas (such as user ratings) and user implicit feedback datas (such as browsing logs) to recommend the items required by the user to the other party. These algorithms mainly use the items-users rating matrix information, and do not fully consider the semantic information of the items itself.

The knowledge graph contains rich semantic associations between entities, providing a potential source of auxiliary information for the recommendation system. A generic feature-based method was represented by LibFM [8]. This type of method uniformly uses the attributes of users and items as input to the recommendation algorithm. It is not specifically designed for knowledge graph, so it is not possible to efficiently use all the information of the knowledge graph. Path-based methods were represented by PER [9] and MetaGraph [10]. These type of methods treat the knowledge graph as a heterogeneous information network and then constructs meta-path or meta-graph based features between items. The advantage of these type of methods are that they fully and intuitively utilizes the network structure of the knowledge graph. The disadvantage is that manual design of meta-path or meta-graph is required, which is difficult to achieve optimally in practice. This paper proposes a Top-N collaborative filtering algorithm based on knowledge graph embedding. Through the knowledge graph embedding, a low-dimensional vector is obtained for each entity and relationship learning, and the structure and semantic information of the original graph are maintained in the vector. By calculating the semantic similarity between items, the semantic information of the items is incorporated into the recommendation process, and the user is better recommended.

2 Definition

The collaborative filtering recommendation based on knowledge graph establishes a rating matrix according to the user's rating information of the item, calculates the similarity between the items, and then fuses the semantic similarities learned by the knowledge graph embedding to generate an item similarity matrix. Then each user is calculated a predicted rating for the item for which the behavior has not occurred. Based on these predicted ratings, a list of recommended items is generated for user, and finally an appropriate evaluation indicator is selected to measure the accuracy of the recommendation algorithm. The following are concepts related to the algorithm proposed in this paper.

Definition 1 Knowledge graph: Describing the various entities or concepts and their relationships that exist in the real world. It constitutes a huge semantic network graph. Nodes represent entities or concepts, while edges are composed of attributes or relationships. Based on the triplet, which is a general representation of the knowledge graph [11], denoted as (h, r, t).

Definition 2 User: Let sets U be the collection of all users. Where the number of user sets u are expressed as m.

Definition 3 Item: Let sets I be a collection of all items that can be recommended to all users. Where the number of item sets I are expressed as n.

Definition 4 Rating matrix: Each user may generate a rating for each item, and its value constitutes the users-items rating matrix $R_{m \times n}$. $R_{ij}(1 \leq i \leq m, 1 \leq j \leq n)$ is the rating of the item I_j by the user U_i, and the rating represents the degree of preference of the user i for the item j.

Definition 5 Similarity measure: The item-based collaborative filtering algorithm is generally composed of two parts, first calculating the similarity between items, and then generating a recommendation list for the user according to the similarity of the items and the historical preference of the user. The core of this process is the calculation of item similarity. This paper uses Pearson correlation to calculate the similarity between items. The Pearson correlation formula is as follows:

$$PC(I_i, I_j) = \frac{\sum_{e \in I_{i,j}} \left(R_{i,e} - \overline{R}_i\right)\left(R_{j,e} - \overline{R}_j\right)}{\sqrt{\sum_{e \in I_{i,j}} \left(R_{i,e} - \overline{R}_i\right)^2} \sqrt{\sum_{e \in I_{i,j}} \left(R_{i,e} - \overline{R}_j\right)^2}} \tag{1}$$

Where $R_{i,e}$ and $R_{j,e}$ represent the ratings of the user u_i and the user u_j for the item i_e, respectively; $\overline{R}_i$ and $\overline{R}_j$ represent the average ratings of the items by the users u_i and u_j, respectively.

Definition 6 Evaluation indicators: This paper uses the recall, precision, and F1 to measure the prediction accuracy of the recommendation algorithm. The recall represents how many of the test sets are in the user's recommendation list. The precision represents the proportion of items that are recommended to the user and belongs to the test set. The F1 is the weighting of the accuracy rate and the recall rate, which evenly reflects the recommended effect.

3 Top-N Collaborative Filtering Recommendation Algorithm Based on Knowledge Graph Embedding

This paper proposes a Top-N collaborative filtering recommendation algorithm based on knowledge graph embedding (denoted as CTransR-CF algorithm), which combines knowledge graph and collaborative filtering algorithm. By introducing the rich content information of the item in the collaborative filtering algorithm, the defect of the collaborative filtering algorithm ignoring the content information of the item itself is effectively compensated, thereby alleviating the problem of data sparsity. The specific idea of the CTransR-CF algorithm is to use the knowledge graph embedding algorithm to embed the items into a low-dimensional space, then calculate the semantic similarity between the items, generate a semantic similarity matrix, and finally obtain the semantic neighbors of the items. At the same time, by adjusting the fusion ratio, the similarity of the items obtained by the semantic neighbor and the item-based collaborative filtering algorithm are proportionally combined to obtain the final recommendation set. The framework process is shown in Fig. 1.

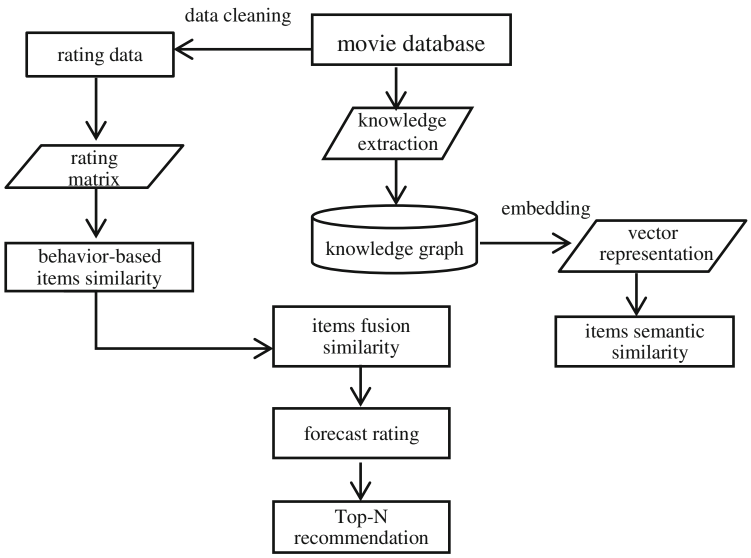

Fig. 1. CTransR-CF algorithm flow

3.1 Construction of Knowledge Graph

Since the constructed knowledge graph is a follow-up recommendation service, and the recommendation algorithm proposed in this paper uses the data set in the movie field, this section builds a knowledge graph in the movie field. Building a knowledge graph in the field of movie requires relevant data support in the movie field. Considering the integrity of the required movie data and the difficulty of data acquisition, this article uses the data provided by the "The Movie Database (TMDb)" website [13], the extracted data. Stored in the form of a triple. Through the analysis and research of the relevant knowledge in the movie field, several elements of the knowledge graph in the movie field are determined. The knowledge of the movie field includes the basic elements of the movie, the basic information of the actors, and the basic information of the director and the screenwriter. Where movie information provides the type, name, introduction, date of film release, language of the movie, and rating information for the movie. The director's elements, scriptwriters, and actor elements provide the names, profiles, births and deaths, and birthplace information of the respective characters.

The ontology [12] belongs to the conceptual layer in the knowledge graph. The ontology establishes an abstract model of the real world by describing concepts in the knowledge graph. The ontology library of the film knowledge graph is constructed in a top-down manner, abstracting the three concepts of "movie", "person" and "type", and defining the classes of related concepts in the domain, the attributes of the entities and their value ranges. Create an instance of the ontology. Through the above steps, the ontology library construction in the movie field is completed. Then the ontology is instantiated, the data layer in the knowledge graph is filled, and the knowledge graph is finally constructed. This paper builds a knowledge map of the movie field, and the knowledge of entities and their attributes is extracted from the structured data.

3.2 Vector Representation of Items Based on CTransR Algorithm

In order to obtain the embedded vector of the entity and relation, in particular the vector of the corresponding item entity in the recommendation system. Entity and relationship embedding are established by translating models like TransE [14] and TransH [15] as de novo entities to tail entities. Entities and relationships are embedded in the same space, thus enhancing the user rating matrix in the collaborative recommendation algorithm. Semantic information about the item. The models mentioned above, including TransR, learn the individual vectors of each relationship. Under this relationship, this low expressiveness may not apply to all pairs of entities, because these relationships are usually diverse. In order to better model these relationships, this paper uses the idea of piecewise linear regression combined with TransR.

The basic idea is that for a triple (h, r, t) in the knowledge graph S, the input instances are first divided into groups. Formally, for a particular relationship r, all pairs of entities (h, t) are clustered into groups, and the pairs of entities in each group exhibit a similar relationship r. All entity pairs are clustered by their vector difference $(\mathrm{h} - \mathrm{t})$. A separate relationship vector $\mathrm{r_c}$ is then learned for each cluster, and the matrix Mr is learned for each relationship. The projection vectors defining the entities are $\mathrm{h_{r,c}} = \mathrm{hMr}$ and $\mathrm{t_{r,c}} = \mathrm{tMr}$. The formula of the rating function is as follows:

$$f_r(\mathrm{h,t}) = \left\| h_{\mathrm{r,c}} + r_c - t_{\mathrm{r,c}} \right\|_2^2 + \alpha \left\| r_c - \mathrm{r}_2^2 \right\| \tag{2}$$

Where closer the head entity vector is to the tail entity vector after adding the relation vector, the more accurate the embedding, $\left\| \mathrm{r_c} - \mathrm{r}_2^2 \right\|$ ensures that the clustering vector $\mathrm{r_c}$ of a particular relationship is not too far from the original relation vector r, and α plays a role in controlling this limitation. For the rating function of the single triple above, we define a marginal-based rating function as the goal to train, the cost function is as follows:

$$\mathrm{L} = \sum\nolimits_{(\mathrm{h,r,t}) \in S} \sum\nolimits_{(h',\mathrm{r},t') \in S'} \max(0, f_r(\mathrm{h,t}) + \gamma - f_r(h', t')) \tag{3}$$

Where S is a positive triplet set, we believe that all triples in the knowledge graph are correct. S' represents the generated negative triplet set. The negative triplet needs to be constructed by itself. When facing a one-to-many, many-to-one or many-to-many relationship triplet, replace the head and tail entities with different probabilities. Max(x, y) represents the maximum value in x and y, and the cost function is such that each accumulated child value is not less than zero. γ represents the size of the spacing and is used to control the distance between positive and negative samples, typically set to 1. In the learning process, the cost function is optimized by stochastic gradient descent, and the final training result is that the semantically similar entities in the knowledge graph are mapped to the corresponding positions in the vector space. And can make these vectors retain the semantic information of the item entity. Model training algorithm 1 is as follows:

```
Algorithm 1:The Learning Algorithm Of CTransR
Input:  An scoring function $f_r(h,t)$,training set S={(h,r,t)} ,entity set ε(h or t) and
relation set R, margin $\boldsymbol{\gamma}$,learning rate λ,embedding dimensions k, batch quantity m.
Output:  All the enbeddings of e and r,where e∈ε and r∈R
Initizlize parameter $\boldsymbol{\theta}$;
Loop
e←e/||e|| for each entity
  $\mathbf{S_{batch}}$←sample(S,m)
    $\mathbf{T_{batch}}$←∅     //initialize set of pairs
    foreach (h,r,t) ∈ $\boldsymbol{T_{batch}}$ do
        $(h',r,t')$←negSample ∈ S //sample negative triplet with "unif"
        $\mathbf{T_{batch}}$←$\mathbf{T_{batch}}$ ∪((h,r,t),(h',r,t'))
        (h,r',t) ←negSample ∈ S //sample negative triplet by corrupting relation
$\mathbf{T_{batch}}$←$\mathbf{T_{batch}}$ ∪((h,r,t),(h,r',t))
    Update parameter $\boldsymbol{\theta} \leftarrow \frac{1}{m}\sum_1^m \frac{\partial f}{\partial \theta}$
    Update embedding w.r.t $L=\sum_{((h,r,t),(h',r,t'))\in T_{batch}} \nabla[f_r(h,t)+\gamma-f_r(h',t')]_+$
End loop
```

3.3 Semantic Similarity Measure

Through the vectorized representation of the items given in the previous section, the vector of the item in the high-dimensional space is finally obtained. It is therefore possible to calculate the semantic similarity of an item based on the embedded vector of the item. We map the item entities and relationships in the knowledge graph to the k-dimensional space. The item I_i is embedded as a k-dimensional vector with the following formula:

$$I_i = (E_{1i}, E_{2i}, \ldots, E_{ki})^T \tag{4}$$

Where E_{ki} represents the value of the embedded vector of the item I_i in the k-dimensional. The rating function used is calculated based on the Euclidean distance when embedding entities and relationships in the knowledge graph. Therefore, this paper uses the Euclidean distance to measure the similarity of the item entities. The final semantic similarity measure is as follows:

$$sim_{kg}(I_i, I_j) = \frac{1}{1 + \sqrt{\sum_{d=1}^{k} (E_{di} - E_{dj})^2}} \tag{5}$$

It can be seen from the above formula that the closer the result of $\mathrm{sim_{kg}}(\mathrm{I_i}, \mathrm{I_j})$ is to 1, the more similar the two entity vectors are, that is, the closer the relationship between the two is in the knowledge graph. When this value is close to zero, we think that the two entities are almost completely different. Therefore, using this semantic neighbor, the collaborative filtering neighbor set based on collaborative filtering recommendation is replaced by semantic fusion, and finally the required recommendation list is obtained.

3.4 Similarity Fusion

Based on the embedded vector of the items in the knowledge graph, the similarity of the items based on the knowledge graph $\mathrm{sim_{kg}}(\mathrm{I_i}, \mathrm{I_j})$ is obtained. Based on the user's rating matrix for the item, the similarity of the item based on the user behavior $\mathrm{sim_{ub}}(\mathrm{I_i}, \mathrm{I_j})$ is obtained by using Eq. 1. Assume that I is a collection of items that a collaborative filtering algorithm will recommend to the user. The forecast ratings of each object $i \in I$ are sorted to generate a corresponding ordered sequence L. Similarly, an ordered sequence K to be replaced is generated for the semantic neighbor set T. Assuming that the length of the sequence L is ρ and the fusion ratio is p:q, the number of substitutions is $n = \rho \times \frac{p}{p+q}$, and the fusion process is as shown in Algorithm 2:

```
Algorithm 2:Similarity Fusion Algorithm
Input:   Collaborative filtering of neighbor sets I, semantic neighbor set
Output:  Recommended set C
Begin
    Set L = list(I),K = list(T)
    for $L_i \in \{L_{1-n}, L_{1-n+1}, \ldots, L_1\}$ do;
        for $K_j \in \{K_1, K_2, \ldots, K_n\}$ do;
                  $\mathbf{L_i = K_j}$
        end do
        end do
    Set C = set(L)
End
```

3.5 Forecast Rating

This paper uses p_{ui} to represent the forecasted user u's rating on the item i. The calculation formula of p_{ui} is as follows:

$$p_{ui} = \frac{\sum_{j \in N(u) \cap S(i,k)} \left(S_{ij} \times R_{uj}\right)}{\sum_{j \in N(u) \cap S(i,k)} S_{ij}} \tag{6}$$

Where S_{ij} represents the similarity of the item I_i and the item I_j, R_{uj} represents the rating of the item u in the existing rating matrix, and N(u) represents the set of items that the user u has scored, and S(i, k) represents the collection of the most similar k items of items i. The meaning of this prediction formula is that the set of items that the user u scored and the set of k items most similar to the item i are first as an intersection, and a reference set is generated, and the rating of the item i is referred to the set. In the reference item set each item generates a weight, and the item weight indicates the proportion of the similarity of the item in the similarity of all items. The final user's forecast rating on the item is expressed as a weighted sum of the user's ratings on all items in the reference items set, such as Algorithm 3 shows:

```
Algorithm 3: Forecast Rating Algorithm
Input:   Rating matrix R， items similarity matrix S
Output:   User u's predicted rating for I
Begin
      $I_1$ = GetTopSimItem(S,i,k); //Find the set of k items most similar to item
I
      $I_2$ = GetRatingItem(N,u); //Find the sets of items that user u has scored
      I = $I_1 \cap I_2$; //Refer to the sets
      P= PredictRatingItem(I,j,u,S,R);
Output p;
End
```

3.6 Recommendation List Generation

In the TOP-N recommendation system, we generate a list of recommendations for each user that contains N items. We recommend the user based on the following assumptions: The higher the user's forecast rating for the item, the more interesting the user is to the item, the more the recommendation system should recommend the item to the user. To put it simply, for each user in the recommendation system, first forecast its rating of the item by Eq. 6, then sort the items in descending order according to their forecast ratings, and use the top N items of the sorted sequence as the item recommendation list is recommended to the user. The specific process is shown in Algorithm 4:

```
Algorithm 4: Top-N Recommend List Algorithm
Input: Initial recommendation list topNu for user u, recommended number n
Output:   User u final recommendation list topNu_New
Begin
for all i∈topNu do
            if i ∈all items
                  topNu_New =   ForecastRatingDescend(topNu);
              else
                  P = ForecastRatingItem(I,j,u,S,R);
                  topNu_New = topNu_New ∪ N;
            end do
End
```

4 Experimental Settings and Results

4.1 Dataset

This section mainly validates and evaluates the effectiveness of the proposed CTransR-CF algorithm. The experimental data set uses MovieLens-1M, which contains 1000209 ratings, involving 6040 users and 3900 movies. The MovieLens dataset has been widely used in academia. At the same time, in order to match the movie in the MovieLens-1M data set and the movie entity extracted from the TMDb movie database, this paper uses the entity link method to map each movie in the MovieLens-1M data set to the knowledge graph. After mapping, it finally matches 3634. In the movie entity data, 93% of Movies in MovieLens-1M can be accurately mapped into the knowledge graph, which is enough for subsequent work. This paper divides the rating data set, with 70% as the train set and 30% as the test set.

4.2 Experimental Results

(1) Determination of mixing ratio

The recommendation algorithm proposed in this paper combines the similarity of items based on knowledge graph and the similarity of items based on user behavior, and finds n neighboring movies for each user. When the fusion ratio is different, the recommended effects are different. The number of Top-N neighbors selected by the experiment is 50. In order to determine the optimal fusion ratio of CTransR-CF, we select the fusion ratio of similarity in the interval [0, 1].When p:q is 0:10, the whole recommendation algorithm is an item-based collaborative filtering algorithm; when p:q is 10:0, the whole algorithm is a recommendation algorithm based on the content of the knowledge map item. The following is the experimental results of the algorithm in the test set with different degrees of convergence given in Fig. 2. It can be seen from the figure that as the proportion of fusion increases, the accuracy, recall rate and F value increase, and reach a peak at a ratio of 5:5. For the case of N = 50, the fusion ratio is 5:5, which works best.

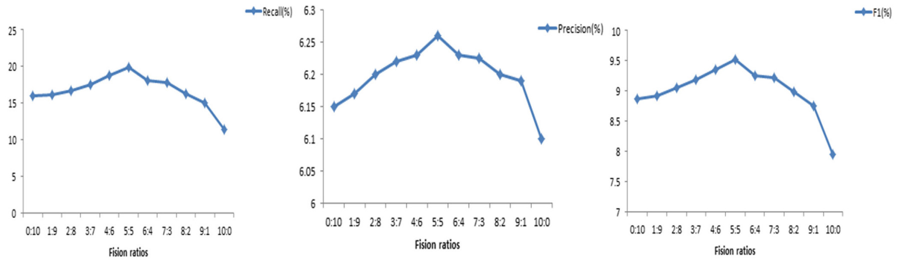

Fig. 2 Recall, precision, F1 under different fusion ratios

(2) Embedded dimension

Knowledge graph embedding embeds the knowledge graph into a low-latitude space, and different recommendation effects for different dimensions are selected for the dimensions embedded in the project entity. Experiments were selected from 100 to 500 dimensions, and it is concluded from Fig. 3 that the best is achieved when 200-dimensional is selected.

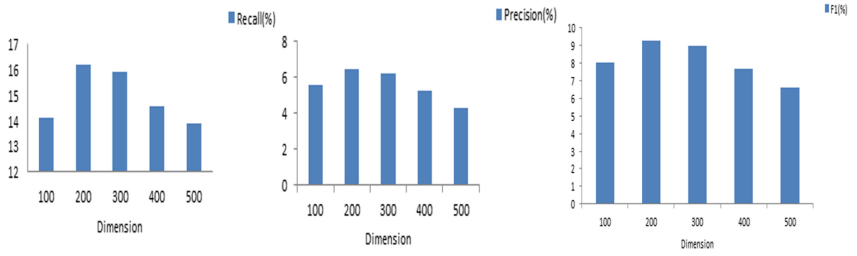

Fig. 3. Recall, precision, F1 under different embedding dimensions

(3) Algorithm performance

Here, the fusion ratio with the highest F1 is selected, and the embedded dimension is 200, and the neighbor number is 30–110. Figure 4 shows that the CTransR-CF algorithm proposed in this paper has a recall of 13%–25%, an precision of 3%–7%, and an F1 of 6%–9%.

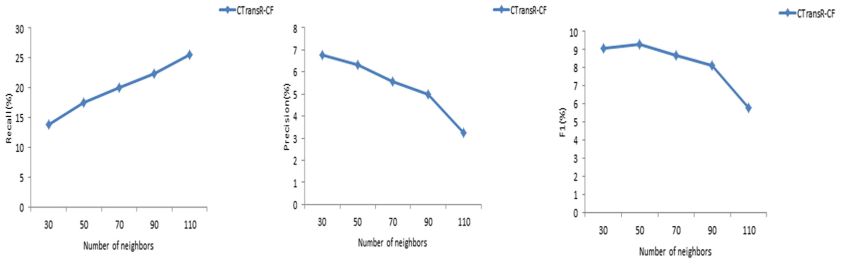

Fig. 4. Recall, precision, F1 under different neighbors

5 Discussion

In this paper, the traditional collaborative filtering recommendation algorithm does not consider the semantic information of the items itself, which leads to the problem that the recommendation accuracy is not high. The article in the knowledge graph is vectorized and introduced into the item-based collaborative filtering recommendation algorithm, and other Compared with the algorithm, CTransR-CF algorithm proposed in this paper has the following advantages:

(a) Using the knowledge graph as auxiliary information, the item semantic information is fully utilized, and the traditional item-based collaborative filtering algorithm is not semantically compensated for by the items. As can be seen in Fig. 5, the collaborative filtering algorithm based on pearson correlation has the worst recommendation effect, and the knowledge graph is introduced into the collaborative filtering algorithm, and the recommendation performance is significantly improved.
(b) The semantic similarity matrix of the article based on the knowledge graph is merged with the object similarity matrix based on the user behavior, and the final similarity of the article is obtained by adjusting the dimension and fusion ratio of the knowledge graph embedding.
(c) The clustering-based TransR (CTransR) knowledge graph item vectorization model is integrated into the collaborative filtering algorithm. In the experiment, it is compared with the TransE and TransH models, because CTransR groups the clusters by different head-to-tail entities. Learning the relationship vector of each group can better model the relationship. By comparing several algorithms, CTransR-CF algorithm has improved recall, precision and F1, which is superior to other algorithms.

This paper only considers the connotation knowledge of the item itself. The user's potential relationship information is also an important potential factor in the recommendation system. Therefore, the next research focus is to further improve and integrate the user's potential information in the proposed algorithm. In addition, it will also be devoted to the research of text representation and semantic entity analysis, and apply this algorithm to other fields for recommendation and further optimize recommendation performance.

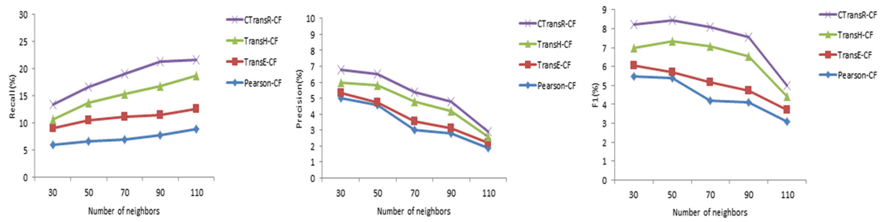

Fig. 5. Performance comparison with other algorithms

6 Conclusion

This paper proposes a Top-N collaborative filtering recommendation algorithm that integrates knowledge graph embedding, namely CTransR-CF algorithm. Compared with the traditional recommendation algorithm that only uses the item-user rating information, the CTransR-CF algorithm can fully utilize the semantic information inherent in the item itself, and uses the external rating matrix of the item to reflect the attributes of the item more comprehensively. The algorithm uses knowledge graph embedding to embed the entity into the low-dimensional space, computes the semantic similarity between the entities, and integrates it into the item-based collaborative filtering recommendation algorithm. To some extent, the effect of collaborative filtering recommendation is enhanced, and the problem that collaborative filtering recommendation does not consider semantics is solved. At the same time, using semantic information, the problem of data sparsity is alleviated to some extent. The experimental results verify that the CTransR-CF algorithm proposed in this paper effectively improves the accuracy of the recommended effect.

7 Acknowledgement

This research is supported by the Dongguan professional town innovation service platform construction project, Dongguan Humen garment collaborative innovation center, and the Guangdong provincial cooperation innovation and platform environment construction special funds (Grant No. 2014B090908004).

References

1. Wang, C.D., Deng, Z.H., Lai, J.H., et al.: Serendipitous recommendation in e-commerce using innovator-based collaborative filtering. IEEE Trans. Cybern. **PP**(99), 1–15 (2018)
2. Huang, X.Y., Xiong, L.Y., Qin-Dong, L.I.: Personalized news recommendation technology based on improved collaborative filtering algorithm. J. Sichuan Univ. **55**(01), 49–55 (2018)
3. Wu, Q., Huang, M., Mu, Y.: A collaborative filtering algorithm based on user similarity and trust. In: IEEE Web Information Systems and Applications Conference, pp. 263–266 (2018)
4. Li, D., Chen, C., Lv, Q., et al.: An algorithm for efficient privacy-preserving item-based collaborative filtering. Futur. Gener. Comput. Syst. **55**(C), 311–320 (2016)
5. Gao, F.Z., Huang, M.X., Zhang, T.T.: Collaborative filtering recommendation algorithm based on user characteristics and expert opinions. Computer Science (2017)
6. Liu, X.: An improved clustering-based collaborative filtering recommendation algorithm. Clust. Comput. **20**(2), 1–8 (2017)
7. Ji, Z., Zhang, Z., Zhou, C., Wang, H.: A fast interactive item-based collaborative filtering algorithm. In: Du, D., Li, L., Zhu, E., He, K. (eds.) NCTCS 2017. CCIS, vol. 768, pp. 248–257. Springer, Singapore (2017). https://doi.org/10.1007/978-981-10-6893-5_19
8. Rendle, S.: Factorization machines with libFM. ACM Trans. Intell. Syst. Technol. **3**(3), 1–22 (2012)
9. Yu, X., Ren, X., Sun, Y., et al.: Personalized entity recommendation: a heterogeneous information network approach. In: Proceedings of the 7th ACM International Conference on Web Search and Data mining. ACM (2014)
10. Zhao, H., Yao, Q., Li, J., et al.: Meta-graph based recommendation fusion over heterogeneous information networks. In: ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM (2017)
11. Zeng-Lin, X.U., Yong-Pan, S., Li-Rong, H.E., et al.: Review on knowledge graph techniques. J. Univ. Electron. Sci. Technol. China **45**, 589–606 (2016)
12. Chah, N.: OK Google, What Is Your Ontology? Or: Exploring Freebase Classification to Understand Google's Knowledge Graph. University of Toronto, Faculty of Information (2018)

13. The Movie Database. The Movie Database [EB/OL], 27 October 2017. https://www.themoviedb.org/
14. Feng, J., Zhou, M., Hao, Y., et al.: Knowledge graph embedding by flexible translation, **2015**, 557–560 (2015)
15. Wang, Z., Zhang, J., Feng, J., et al.: Knowledge graph embedding by translating on hyperplanes. In: Twenty-Eighth AAAI Conference on Artificial Intelligence, pp. 1112–1119. AAAI Press (2014)