Matrix factorization recommendation algorithms based on knowledge map representation learning

Xuejian Huang1, Lu Min2, Gensheng Wang1, Tian Gan1, Zhipeng Li1

1Computer Practice Teaching Center, Jiangxi University of Finance and Economics, Nanchang 330013, China
2School of Humanities, Jiangxi University of Finance and Economics, Nanchang 330013, China

Corresponding author and e-mail: Xuejian Huang, 761780598@qq.com

Abstract. The matrix factorization recommendation algorithm does not consider characteristics of the recommendation object itself, resulting in poor recommendation results. Therefore, a matrix factorization recommendation algorithm based on knowledge map representation learning is proposed. Firstly, the recommendation object is represented as a low dimensional semantic vector by using the knowledge map distributed representation learning algorithm. Then the semantic similarity between objects is calculated, and the semantic similarity is incorporated into the objective optimization function of matrix factorization, so that the feature vectors obtained by matrix factorization can also contain semantic knowledge, which makes up for the shortcoming of matrix factorization recommendation algorithm that does not consider characteristics of the recommendation object itself from the semantic perspective. The experimental results show that the improved algorithm has higher accuracy, recall and coverage than the traditional matrix factorization recommendation algorithm.

1. Introduction

Recommendation algorithm is a typical technology to solve the problem of network information overload, which has been widely used in network media, e-commerce, advertising and other commercial fields. At present, recommendation algorithms are divided into three categories according to different recommendation engines: content-based filtering recommendation, collaborative filtering recommendation and hybrid recommendation [1]. Collaborative filtering recommendation algorithm is based on user's historical behavior data and has no domain restrictions. It is the most widely used recommendation algorithm at present. It is mainly divided into User-based CF, Item-based CF, Model-based CF. User-based CF and Item-based CF faces sparse matrix data, it can't achieve good recommendation effect [2]. Model-based CF uses machine learning algorithm to model, which can solve the problem of matrix sparsity to a certain extent [3]. Matrix factorization is a typical algorithm based on model-based collaborative filtering [4].

The matrix factorization recommendation algorithm only uses the user project evaluation matrix without considering other factors, which results in low accuracy of recommendation results. Many experts put forward relevant improvement schemes for this problem. For example, reference [5] proposes a matrix factorization method based on attribute coupling, which combines the attribute information of the project into the matrix factorization model; reference [6] introduces the trust relationship between users to improve the performance of the matrix factorization recommendation algorithm; reference [7] adds additional implicit information (such as browsing, purchase and click History).

It is found that the main method to improve the matrix factorization recommendation algorithm is to add additional relevant information of users or projects. With the development of knowledge map technology, there are a lot of open semantic knowledge data, such as Freebase, OpenKN, DBpedia, BMKG, etc. The knowledge map representation learning algorithm can embed the recommended object into a low-dimensional semantic vector space, so this paper proposes a matrix factorization recommendation

1Fund: NSFC funded project (Project No.: 71461012); Science and technology project of Jiangxi Provincial Department of Education (GJJ181550)
algorithm based on knowledge map representation learning to make up for the shortcomings of the matrix factorization recommendation algorithm which does not consider the characteristics of the recommended object itself.

2. Correlation theory

2.1. Matrix factorization recommendation algorithm

Matrix factorization recommendation algorithm (FunkSVD) decomposes two low dimensional user characteristic matrix and item characteristic matrix through the score matrix, and uses these two decomposed matrices to fit the user's score on the project. Matrix factorization is shown in Eq. (1):

$$ R = U^T V $$

Where $R$ is the user item scoring matrix, $U \in \mathbb{R}^{m \times d}$ represents the decomposed user characteristic matrix, $V \in \mathbb{R}^{n \times d}$ represents the decomposed project characteristic matrix, $m$ and $n$ represent the number of users and items, and $d$ is the user and item characteristic dimension. The scoring calculation of user $i$ for item $j$ is shown in Eq. (2):

$$ R_{ij} = U_i^T V_j $$

In order to make Eq. (1) fit the real score data, use the idea of linear regression to establish the objective optimization function, as shown in Eq. (3):

$$ J = \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij} (r_{ij} - U_i^T V_j)^2 + \lambda (\|U_i\| + \|V_j\|) $$

Where $I_{ij}$ indicates whether user $i$ has scored project $j$, if so $I_{ij}$ is 1 otherwise 0, $r_{ij}$ is the actual score of user $i$ for project $j$, the regularization parameter $\lambda$ to prevent over fitting, and $\|U_i\|$ and $\|V_j\|$ are the norms of $U_i$ and $V_j$ respectively. Based on FunkSVD algorithm, an improved Bised MF algorithm is proposed in reference [8]. Bised MF introduces global average sub item, user bias item (difference between user evaluation average score and global average score) and project bias item (difference between project average score and global average score) into objective optimization function Eq. (3). The final objective optimization function is shown in Eq. (4):

$$ J = \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij} (r_{ij} - U_i^T V_j - \mu - \alpha_i - \beta_j)^2 + \lambda (\|U_i\| + \|V_j\| + \|\mu\| + \|\alpha_i\| + \|\beta_j\|) $$

Where $\mu$ is the global average item, $\alpha_i$ is the use $i$ offset item, $\beta_j$ is project $j$ offset item. The scoring calculation of user $i$ for item $j$ is shown in Eq. (5):

$$ R_{ij} = U_i^T V_j + \mu + \alpha_i + \beta_j $$

2.2. Knowledge map distributed representation learning

The knowledge map uses the triple of "entity-relationship-entity" to describe the relationship between entities and forms a network of knowledge structures through the relationship. Knowledge map distributed representation learning can get semantic vector distributed representation of entities and relationships [9]. Due to its simple parameters and low computational complexity, Transe is the mainstream knowledge map distributed representation learning model, because of its simple parameters, low computational complexity, and remarkable performance on large-scale knowledge map. For each triple $(h,r,t)$, $h$ and $t$ represent the head entity and the tail entity respectively, and $r$ is the relationship between the head entity and the tail entity, Transe expresses $h$, $t$ and $r$ as embedding vectors $v_h$, $v_t$ and $v_r$. The $v_t$ is the translation between $v_h$ and $v_t$, also known as translation. The relationship among them is shown in Eq. (6):

$$ v_h + v_r = v_t $$

The closer the two sides of the equation are, the more likely there is a relationship $r$ between the two entities. The loss function of the Transe model is shown in Eq. (7):

$$ f(v_h, v_r, v_t) = \|v_h + v_r - v_t\|^2 $$

The total cost function is shown in Eq. (8):

$$ L = \sum_{(h,r,t) \in \mathcal{S}} \sum_{(h',r',t') \not\in \mathcal{S}} \max(0, f(v_h, v_r, v_t) - f(v_{h'}, v_{r'}, v_{t'}) + \gamma) $$

Where $\mathcal{S}$ is the set of all triples, which is called a positive sample, and $\mathcal{S}'$ is the negative sample of set $\mathcal{S}$, that is, to randomly replace the head entity or tail entity of each existing triplet in $\mathcal{S}$, a new triplet is obtained, and the new triplet does not belong to $\mathcal{S}$, and $\gamma$ is the distance between positive and negative samples.

TransE doesn't distinguish entities under different relationships, and there are shortcomings in dealing with knowledge map of complex relationships. To solve this problem, reference [10] proposes TransR model,
which embeds entities and relationships into different spaces, and implements entity representation in corresponding relationship spaces. Its loss function is shown in Eq. (9):

$$f(v_h, v_r, v_f) = [v_h M_r + v_r - v_f M_r]_v$$

Where $M_r$ is the matrix of relation $r$, $v_h$ is the projection of entity vector $v_h$ into relation $r$ space.

3. Fusion recommendation algorithm

The matrix factorization recommendation algorithm only uses the user project evaluation matrix, and does not consider the characteristics of the project itself, resulting in poor recommendation results. This paper proposes a matrix factorization recommendation algorithm based on knowledge map representation learning. The algorithm integrates the semantic similarity between the recommended objects into the objective optimization function of matrix factorization to make up for the defect that the matrix factorization recommendation algorithm does not consider the characteristics of the recommended objects. The algorithm flow is shown in Fig.1:

3.1. Semantic vector representation

According to the knowledge map distributed representation learning algorithm, the vector representation of all entities and relationships in the domain of the recommended object is obtained, and the entity representation of the recommended object is screened out from the entity vector. The vector representation of the recommendation object integrates the entity knowledge related to it in the whole field, so the vector representation contains the context semantic knowledge of the recommendation object. The recommended object entity is represented as a $d$-dimensional semantic vector, as shown in Eq. (10):

$$I_i = (E_{i_1}, E_{i_2}, ..., E_{i_d})$$

3.2. Item semantic similarity

The similarity calculation mainly includes cosine similarity, Pearson similarity, Jaccard similarity, log likelihood similarity and Euclidean distance similarity. The loss function of knowledge map distributed representation algorithm is based on Euclidean distance. In order to maintain consistency, the similarity of project semantics is also measured by Euclidean distance, as shown in Eq. (11):

$$d(I_i, I_j) = \sqrt{\sum_{i=1}^{d} (E_{i} - E_{i})^2}$$

Adjust it to $[0, 1]$, and the specification calculation is shown in Eq. (12):

$$sim(i, j) = \frac{1}{1 + d(I_i, I_j)}$$
The larger the value of \( \text{sim}(i, j) \) is, the closer the semantics of item \( i \) and item \( j \) are.

3.3. Fusion matrix factorization

Based on the idea that the feature vectors of projects with similar semantics should also be similar, the fusion algorithm integrates the semantic similarity of projects into the objective optimization function Eq. (4) of the Bissed MF matrix factorization. The objective optimization function after fusion is shown in Eq. (13):

\[
J = \sum_{i \in U} \sum_{j \in I} \left( u_{ij} - \langle u_i, v_j \rangle - \mu - \alpha - \beta \right)^2 + \lambda_1 \left( \| u_i \|^2 + \| v_j \|^2 + \| R_{ij} \|^2 \right) + \lambda_2 \sum_{i \in I} \left( \cos(V_i, V_j) + \frac{1}{2} - \text{sim}(i, j) \right)^2
\]

Eq. (13) adds \( \lambda_2 \sum_{i \in I} \left( \cos(V_i, V_j) + \frac{1}{2} - \text{sim}(i, j) \right)^2 \) on the basis of Eq. (4), to control the consistency of semantic similarity and feature vector similarity. Where \( \cos(V_i, V_j) \) is the cosine value of the eigenvectors of items \( i \) and \( j \), \( \text{sim}(i, j) \) is the semantic similarity of items \( i \) and \( j \), \( \lambda_2 \) is the fusion coefficient. The calculation of \( \text{sim}(i, j) \) is shown in Eq. (12).

3.4. Recommended results

The fusion matrix decomposes two low-dimensional user feature matrix and project feature matrix, and uses Eq. (5) to calculate the prediction score. Based on the principle that the higher the prediction score, the more interested the user is, the project with the prediction score greater than a certain threshold is recommended to the user.

4. Experimental Verification and Result Analysis

4.1. Experimental data

In this experiment, movie recommendation is selected as the research object. The experimental data comes from Douban movie review data, which contains 210000 comments of 7800 users on 1500 movies. The number of stars reflects how much users like movies. The star rating is divided into 1-5 stars. In this experiment, 4-5 Stars are labeled as user's favorite movies, and 1-3 stars are user's dislike movies. This experiment uses the latest film knowledge map (BMKG) released by Tsinghua University, which contains more than 720000 movie related entities, 91 attributes and more than 13 million triples. The BMKG integrates the data of Douban film, LinkedMdb and other Chinese and British films. In order to reduce the training time of knowledge map distributed representation learning algorithm, only knowledge related to experimental data is extracted from BMKG.

4.2. Evaluating indicator

In this experiment, Precision, Recall and Coverage are used to measure the performance of the algorithm. Their calculation is shown in Eq. (14), (15) and (16) respectively.

\[
\text{Precision} = \frac{TP}{TP + FP} \quad (14)
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \quad (15)
\]

\[
\text{Coverage} = \frac{N_d}{N} \quad (16)
\]

(1) \( TP \), \( FP \), \( FN \) are the values in the confusion matrix, as shown in Table 1.

| Predict=like | Fact=like | Fact=dislike |
|--------------|-----------|--------------|
| TP           | FN        | TN           |

(2) In Eq. (16), \( N \) is the number of all films in the experiment, and \( N_d \) is the number of films given by the recommended algorithm. The higher the coverage, the better the diversity and novelty of the recommendation results.

4.3. Experimental result analysis

(1) Experimental comparison of different movie entity semantic vector dimensions

In the distributed representation of knowledge map, different dimensions of movie entity vector representation will have a certain impact on the experimental results. Therefore, four groups of comparative
experiments are set with dimensions 50, 100, 150 and 200. Other key parameters in the experimental process are shown in Table 2, and the experimental results are shown in Fig.2.

| parameter | value |
|-----------|-------|
| Regularization parameters $\lambda$ | 1 | 1e-3 |
| Fusion coefficient $\lambda_2$ | 1 |
| Gradient decline $\alpha$ | 1e-2 |
| Iterations of gradient descent $n$ | 300 |
| User and movie feature dimensions $d$ | 60 |

**Figure 2.** Comparison of experimental results in different dimensions of movie entity semantic vector

From Fig.2, we can see that the accuracy, recall and coverage rate of the algorithm are relatively good when the entity dimension of the distributed knowledge representation algorithm is 150.

(2) Experimental comparison of different users and movie features

The feature dimension $d$ of users and movies needs to be set during matrix factorization. In the experiment, 10 groups (10, 20, 30, 40, 50, 60, 70, 80, 90, 100) were set for comparison. The dimension of movie entity semantic vector was set to 150, and other parameters were consistent with Table 2. The experimental results are shown in Fig.3.

**Figure 3.** Comparison of experimental results in different user and movie feature dimensions

According to Fig.3, when the user and movie dimensions decomposed by the matrix are 60, the accuracy, recall and coverage of the algorithm are relatively good.

(3) Experimental comparison of different fusion coefficient values

The fusion coefficient $\lambda$ in Eq. (13) controls the proportion of semantic similarity in the whole algorithm. In this experiment, five kinds of $\lambda$ values (0, 0.5, 1, 1.5, 2) are set for experimental comparison. The semantic vector dimensions of movie entities are set to 150, the feature dimensions of users and movies are set to 60, and other parameters are consistent with Table 1. The experimental results are shown in Fig.4.
Figure 4. Comparison of experimental results with different fusion rates

When the fusion coefficient is 0, the algorithm in this paper degenerates into the Biased MF matrix factorization recommendation algorithm. When the fusion coefficient is not 0, the semantic similarity of the movie is fused in the Biased MF algorithm. The experimental results show that the fusion algorithm has higher accuracy, recall and coverage than the Biased MF algorithm, and the best result is when the fusion coefficient is 1.

5. Conclusions

The recommendation algorithm based on matrix factorization alleviates the problem of matrix sparsity in collaborative filtering algorithm, but the algorithm only uses the user project evaluation matrix, without considering the additional relevant information of the project, resulting in inaccurate recommendation results. Therefore, this paper proposes a matrix factorization recommendation algorithm based on knowledge map representation learning. The semantic similarity of the project is obtained by knowledge map distributed representation learning algorithm. The semantic similarity is integrated into the objective optimization function of matrix factorization, so that the feature vectors of semantic similar items are similar. The experimental results show that the algorithm is effective. However, the algorithm in this paper also has some shortcoming. On one hand, the algorithm depends on open source knowledge map, which leads to certain domain limitations of the algorithm; on the other hand, when faced with massive data, the efficiency of matrix factorization is low. These are next steps to study.

References

[1] Wu M L, Chang C H, Liu R Z. Integrating content-based filtering with collaborative filtering using co-clustering with augmented matrices [J]. Expert Systems with Applications, 2014, 41(6):2754-2761.

[2] Zhang L, Qin T, Teng P Q. An Improved Collaborative Filtering Algorithm Based on User Clustering [J]. Information Science, 2014, 32(10):24-27+32. (In Chinese)

[3] Huang L W, Jiang B T, Lv S Y, et al. Survey on Deep Learning Based Recommender Systems [J]. Chinese Journal of Computers, 2018, 41(7):1619-1647. (In Chinese)

[4] Bokde D, Girase S, Mukhopadhyay D. Matrix Factorization Model in Collaborative Filtering Algorithms: A Survey [J]. Procedia Computer Science, 2015, 49:136-146.

[5] Yu Y, Wang C, Wang H, et al. Attributes coupling based matrix factorization for item recommendation [J]. Applied Intelligence, 2017, 46(3):521-533.

[6] Forsati R, Mahdavi M, Shamsfard M, et al. Matrix Factorization with Explicit Trust and Distrust Side Information for Improved Social Recommendation[J]. ACM Transactions on Information Systems, 2014, 32(4):1-38.

[7] Shulong C, Yuxing P. Matrix Factorization for Recommendation with Explicit and Implicit Feedback [J]. Knowledge-Based Systems, 2018, 158:109-117.

[8] Bi H L, Zhou W, Lu F Q. Bias based matrix factorization recommender techniques [J]. Application Research of Computers, 2018, 35(10):2928-2931+2964. (In Chinese)

[9] Fan M, Zhou Q, Zheng T F, et al. Distributed representation learning for knowledge graphs with entity descriptions[J]. Pattern Recognition Letters, 2017, 93(01): 31-37.

[10] Lin Y , Liu Z , Sun M, et al. Learning Entity and Relation Embeddings for Knowledge Graph Completion[C]// Twenty-ninth Aaai Conference on Artificial Intelligence. AAAI Press, 2015.