Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 3, No. 3, 2023

Algorithm Improvement of Movie Recommendation System based on Hybrid Recommendation Algorithm

Ziyun Liu [1, a, *], Feiyu Ren [2, b]

[1] Xi'an Polytechnic University, China
[2] Xinji Fani Leather Industry Co., China
[a, *] 18134282382@163.com, [b] 1151333999@qq.com
[*] **Corresponding author**: Ziyun Liu (Email: 18134282382@163.com)

Abstract: In recent years, the Internet has developed rapidly, and in the face of thousands of data and information, it has become very critical for users to find the information that is of high value to them in the mass of information, and the recommendation system is one of the most effective ways to solve this information overload phenomenon. In this paper, the current movie recommendation algorithm is improved by using an item-based collaborative filtering algorithm for the similarity measure of items in the item-based recommendation process; In the recommendation process, two more applicable recommendation methods are considered: collaborative filtering content-based recommendation and matrix decomposition-based recommendation. It saves users time in searching, viewing and filtering, while discovering information about their potential movie preferences.

Keywords: Recommended System; Similarity Algorithm; Recommendation Algorithm based on Matrix Factorization; Alternating Least Squares.

1. Introduction

In the past two decades, Internet technology has developed rapidly, and the number of active users on the Internet has grown and the amount of data generated has increased. How to quickly find the information users want from such massive data becomes a challenge, i.e., the information overload problem [1]. For this reason, search engine and recommendation engine technologies were created. At present, the scale of China's Internet search engine users reached 687 million, and the scale of cell phone search applications also has as many as 680 million users. Search engine technology has solved the problem of how to filter useful information from large-scale data for users. But search engine technology is for all users, and the recommendation results are universal and not personalized enough.

Movie search also faces the above problem, and for this reason, movie recommendation system [2] was born. By studying users' personal information, item information and log records to filter and sort through the clutter of data, it emphasizes providing personalized services that help users find the information they want better and faster. At the same time, as he will track the user's log records and constantly modify the recommendation model according to the user's behavior, thus providing more real-time and accurate recommendations. In this paper, the current movie recommendation algorithm is improved by first using an item-based collaborative filtering algorithm for the item similarity measure in the item-based recommendation process. In the recommendation process, a matrix decomposition-based recommendation algorithm and a regularized singular value decomposition algorithm are used, and alternating least squares is used to train the data and find the objective function.

2. Collaborative Filtering Recommendation Algorithm

2.1. User-based Recommendation Algorithm

The main idea of the algorithm is to calculate the similarity between two users using some similarity measure based on user-item association data, such as the user's evaluation information about the item, and thus obtain a set of users containing similarity values. Its recommendation value is related to the similarity between users and the rating of recommended items by users' neighbors, and the recommendation results are ranked in descending order according to the recommendation value. User-based recommendations are highly community-based, time-sensitive and influenced by user size. but also suffer from high data sparsity [3] and poor scalability. Therefore, a recommendation system based on the nearest neighbor algorithm may not be able to make effective recommendations for users whose relevant data are too sparse.

2.2. Item-based Recommendation Algorithm

The algorithm is based on the assumption that most users' preferences are focused, that two items are similar if they have been expressed as preferences by more than one user at the same time, and that the closer the ratings given by the users who expressed their preferences for both items together, the more similar the two items are. Collaborative item-based filtering has the following three features: First, it does not need to calculate similar users and directly calculates similar items. Second, the recommendation results are more personalized, rich in long-tail items, and their recommendation results are more targeted. Third, collaborative filtering based on items needs to calculate the set of items similar to the user's favorite items, and its calculation is proportional to the square of the number of items, which is suitable for business scenarios with small

number of items and large user size.

The core algorithm in the item-based collaborative filtering recommendation algorithm is to calculate the similarity between items and then use the similarity between items to predict the user's rating system for candidate items. Therefore, the choice of the similarity measure plays a decisive role in the final result. Three main methods are included as follows:

(1) Euclidean distance calculation formula

In calculating user similarity, the Euclidean distance [4] is a more intuitive similarity algorithm. According to the common evaluation of Item as a dimension between users, a multidimensional space is established, and the location of this user in this multidimensional space can be located by the coordinate system X (s1, s2, ..., si) composed of the user's evaluation Score on a single dimension, then any two locations Distance (X, Y) can reflect the degree of similarity between two users or items. The calculation formula is shown in Equation (1):

$$\text{dist}(X, Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (1)$$

(2) Cosine similarity

Cosine similarity [5] is a measure of the similarity between two texts using the cosine of the angle between two vectors in the vector space, which focuses more on the difference in direction between two vectors than the distance measure. In general, after obtaining the vector representation of two users or items in the generated collaborative filtering matrix, the similarity between the two texts can be calculated using the cosine similarity, where the result is independent of the length of the vector and is only related to the direction in which the vector is pointing. The cosine between the vectors can be found by using the Euclidean dot product formula:

$$a \cdot b = ||a||||b|| \cos\theta \quad (2)$$

(3) Pearson correlation coefficient

Pearson correlation coefficient, also known as Pearson product-moment correlation coefficient [6], is a measure of the linear correlation between two variables X and Y, with a value between -1 and 1. The Pearson correlation coefficient between two variables is defined as the quotient of the covariance and standard deviation between the two variables and is calculated as shown in Equation (3):

$$\rho(X, Y) = \frac{\text{cov}(x,y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y} \quad (3)$$

The above equation defines the overall correlation coefficient, which is usually expressed using lowercase letters ρ. Estimating the covariance and standard deviation of the samples gives the Pearson correlation coefficient, which is often expressed in lowercase letters γ and calculated as shown in Equation (4):

$$\gamma = \frac{\sum_{i=1}^{n}(X_i-\bar{x})(Y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(X_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(Y_i-\bar{y})^2}} \quad (4)$$

2.3. Matrix Decomposition-based Recommendation Algorithm

The recommendation algorithm based on Singular Value Decomposition [7] (SVD) is a relatively novel algorithm in collaborative filtering recommendation at present. It is an important technique introduced from linear algebra into machine learning. The SVD technique is used to complement the user-item rating matrix to predict user ratings of items, which requires us to find a complementary method that minimizes the perturbation of the original matrix. We generally use the difference in matrix eigenvalues to represent the difference between the complementary matrix and the original matrix.

3. Algorithm Improvement

3.1. Algorithmic Ideas

The goal of this paper is to implement a personalized movie recommendation algorithm based on the movie dataset of MovieLens website. According to the characteristics of movie recommendation, the similarity measure used in the collaborative filtering model is gradually improved, and the number of users who jointly evaluate two movies based on their ratings will be used to determine the degree of similarity between two movies more precisely. In a movie recommendation website, each user generally sees only a small fraction of all movies in the dataset, and the MovieLens dataset used in this paper has filtered out users with less than 20 reviews. The statistics reveal that each user has rated about 100 movies on average, and the sparsity of the data can be expressed using Equation (5). The sparsity in the MovieLens dataset is about 99.83%. Thus, the rating matrix of user movies is a very sparse matrix, and it is possible that only few users rate a movie. Also, when a new movie is added to a movie website, the sparsity of data is further increased because there is no rating record about this movie, also known as the cold start problem (lack of data information to carry out personalized recommendation service).

$$Sparsity = 1 - \frac{Number\ of\ ratings}{Number\ of\ users \times Number\ of\ films} \quad (5)$$

To solve the problem of sparse data, content-based recommendation is introduced to describe a movie using its attributes, which include movie title, genre, and other movie information. Finally, a combination of content-based recommendations and matrix decomposition-based recommendations are used to make hybrid recommendations.

3.2. Similarity Algorithm Improvement

The similarity algorithm of this system uses the cosine similarity algorithm, considering that each data record in the data set keeps a time stamp, and according to the characteristics of human memory, the interests of each user will change accordingly over time, and the longer the interval of data is, the more unreliable the recommendation of this system is, therefore, the time factor of the rating is taken into account to improve the cosine similarity calculation method to achieve the expected purpose. In this paper, we use the time interval between users' ratings of items to optimize the data. When the time interval between users' ratings of two items is larger, the probability of users' interests changing is larger, and the error after using the original rating data for similarity calculation recommendation is larger. On the contrary, the more recent the user rating data timestamp is from now, the more informative this record is and the more reliable the recommended results are.

Based on the above description, this paper introduces a temporal factor [4] to reflect the changes of users' interests over time. Assuming that the user's rating time for movie x is Tx and the user's rating time for movie y is Ty, the formula for the time factor in this paper is shown in Equation (6). The smaller the time interval, the smaller the time factor, and the minimum case is the time factor result of 1 when the interval time is 0.

$$\alpha = \ln(e+(|T_x - T_y|)) \quad (6)$$

The formula for calculating the cosine similarity after adding the time factor is shown in Equation (7).

$$similarity = \cos\theta = \frac{A \cdot B}{||A|||B||} = \frac{\sum_{i=1}^{n} A_i \times B_i \times \frac{1}{\alpha}}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}} \quad (7)$$

3.3. Recommendation Algorithm Improvement

3.3.1. User-based Recommendation Algorithm

It is assumed that users with similar interest preferences may show some degree of interest preference for the same items among themselves. The basic principle of user-based recommendation algorithm is to find users with similar preferences for the target object based on the evaluation data recorded in the system, and then generate recommendation results for the target object based on the preference information possessed by the near-neighbor users. The main steps are as follows:

(1) Establishing a scoring model

This step is the basic step of the algorithm, which lays the foundation for the subsequent steps such as similarity calculation. A two-dimensional matrix is used in the algorithm to represent the scoring model, so the similarity calculation of different objects can be transformed into the similarity calculation between different vectors in this matrix. A value between 0 and 5 is used in this system to indicate the user's liking of the movie, each score represents a different rating or liking level, and a higher score means a higher liking.

(2) Calculate the similarity between users

Calculating the similarity between users is the basic step of establishing the set of nearest neighbors, and the accuracy of similarity calculation will directly affect the selection of nearest neighbor users and finally affect the accuracy of prediction results. The similarity algorithm used in this system is the cosine similarity algorithm, which is a common method to calculate the similarity of user rating vectors.

(3) Establishing the set of similar nearest neighbors

There are two main methods commonly used to select nearest neighbor users: 1) set threshold method, i.e., a threshold value δ is given in the system, and the similarity calculated between two users can be selected as nearest neighbor if it exceeds the threshold value δ. The number of nearest neighbor users obtained by the set threshold method is predetermined, but the similarity between the obtained nearest neighbor users and the target users generally does not show a large deviation. 2) K nearest neighbor method, firstly, the similarity between the target object and other users is arranged in the order from largest to smallest, and the top K users are taken to form the set of nearest neighbors.

In the set threshold method, if the threshold δ is set high, it will select users with high similarity to the target user's interest preferences, but at the same time, it may result in fewer or even no users satisfying the threshold, so that it is impossible to find the set of the target user's near-neighbor users. However, the threshold δ, if set lower, will lead to the existence of users with lower similarity among the similar nearest neighbors satisfying the threshold. Although the number of similar nearest neighbor users satisfying the lower threshold is higher, the similarity between the set of such nearest neighbor users and the target users cannot be guaranteed, and it increases the computation and error for the subsequent calculation.

Similarly, the setting of the K value in the K-nearest neighbor method affects the quality of the nearest neighbor user selection and thus the subsequent calculation. The effect of a larger K value is similar to the effect of a smaller threshold in the set threshold method, but since the K value is fixed in the K nearest neighbor method, there is no uncontrolled introduction of a large number of nearest neighbor users with low similarity as in the set threshold method. The effect of setting a small K value is similar to the effect of setting a large threshold in the threshold method, but because the K value is fixed in the K nearest neighbor method, there will not be too few or even no nearest neighbor users as in the threshold method.

This system calculates a unique one-dimensional value based on the cosine similarity, and it is more appropriate to use the threshold method. Based on certain experiments, it is judged that when the threshold value is 0.8, there are both new recommendations for users with few ratings and the recommended movies meet the needs of users, so this department adopts the set threshold method with a threshold value of 0.8.

(4) Predictive scoring

There are two ways to generate recommendation results for the target user based on the set of similar nearest neighbors obtained in the previous step, one is the prediction of the target user's rating for any item, and the other is to generate a recommendation list. Both ways of generating recommendation results require the prediction of the target user's rating for an item first.

The common calculation method used to perform predictive scoring is the ordinary weighted average, where similarity is used as a weight for weighted averaging. Under this calculation, the more similar nearest neighbor users to the target user contribute to the final predicted rating, the more the predicted rating is eventually dominated by some nearest neighbor users with high similarity, and due to this feature, the system can abstract a specific number of users to make their ratings for movies representative in order to reduce the huge amount of computation due to the large number of movies and users.

3.3.2. Matrix Decomposition-based Recommendation Algorithm

Matrix decomposition is the decomposition of a high-dimensional matrix into a representation of a low-dimensional matrix. This approach reduces the dimensionality of the target matrix and allows the program to run at a faster speed. The most used decomposition is the singular value [6] (SVD) decomposition. When the number of user and item ratings is very large, the target matrix is also very large and a sparse matrix, which is very unfriendly for the operation of the algorithm, and after reducing the dimensionality of the target matrix using SVD technique, the potential properties in the original matrix can be inferred and the potential correlations between users and items can be found.

Assuming that the system has m users and n items, the collaborative filtering matrix R is constructed by the rating relationship between items and items, and then the complete rating matrix R needs to be constructed when recommending information to users, so the corresponding prediction filling is needed for the true information of the current matrix. Initially, the mean values of the corresponding user ratings can be filled in uniformly where the data are missing. In addition, the data can also be normalized, i.e., the entire data is subtracted from the mean, which in a geometric sense behaves as data centered at the origin, and thus yields the final rating matrix R. The SVD decomposition of the matrix R yields (8):

$$R \approx U \cdot S \cdot V^t \quad (8)$$

Where U and V denote two orthogonal matrices of

dimension m×r and r×n, respectively, used to represent the potential factors of users and items, respectively. S is a diagonal matrix of order r×r, consisting of the singular values of the original matrix, similar to the orthogonalized decomposition of a matrix. After getting the r singular values, the selection of singular values can be carried out, and the first five larger singular values are selected in this paper, at which time the diagonal matrix S becomes S_k, and U and V become k-dimensional accordingly, and the sum is obtained, which in turn achieves the dimensionality reduction needed in this paper. Next, the three matrices are dotted and multiplied to obtain the desired reconstruction matrix (9):

$$R_k \approx U_k \cdot S_k \cdot V_k^t \tag{9}$$

Using the decomposed matrix, the corresponding prediction can be made for the user's unrated items by dotting the i-th row and the j-th column, and adding the mean value of the ratings of the previous user rating data, i.e., after decentering, to obtain the predicted rating value of user i for item j, as shown in (10):

$$r_{i,j} = \overline{r_i} + U_k \cdot \sqrt{S_k^T}(i) \cdot \sqrt{S_k} \cdot V_k^T(j) \tag{10}$$

The SVD matrix decomposition algorithm has great applications in recommendation systems, improving the performance of the recommendation system and realizing real-time recommendations even in the case of extremely sparse matrices, laying a good foundation for more matrix decomposition recommendation algorithms in the future.

3.3.3. Regularized Singular Value Decomposition Algorithm

The SVD algorithm introduced in Section 3.3.2 needs to calculate the eigenvectors and eigenvalues of the matrix during the computation, and the matrix cannot have null values during the computation, which is not friendly to the program and the computation process is complicated. The Regularized Singular Value Decomposition (RSVD) algorithm is a further improvement of the SVD algorithm, which is a very popular recommendation system algorithm nowadays. In the RSVD model, the initial collaborative filtering matrix R is decomposed into two low-rank matrices U and V with matrix dimensions m×f and n×f, respectively, where f<<min(m,n), The mathematical representation of this algorithm is given in (11):

$$R \approx U \cdot V^T \tag{11}$$

Where, matrix U is the user potential factor matrix, which indicates the liking of each user for the item feature factors, and matrix V is the item potential factor matrix, which indicates the feature factor composition of each item. Where each row of the U matrix indicates the popularity of each user for different item feature attributes, and each row of the V matrix indicates the weight of each item's feature attributes. Each column of the potential factor matrix represents a potential factor, and these potential factors form a vector space, which in turn represents the characteristics possessed by a user or an item. In other words, the matrix decomposition model extracts f potential factors from the original matrix, which in turn indicates the degree of preference of this user for the item. The preference r of user i for item j can be obtained by equation (12):

$$\hat{R}_{i,j} = U_i \cdot V_j^T \tag{12}$$

Where and denote the i-th row of the potential factor prototype matrix U and the j-th row of V, respectively, and the objective function L can be determined to calculate the matrices U and V, i.e., Equation (13):

$$lossfunction = \sum_{r(i,j)\neq 0}(R_{i,j} - \hat{R}_{i,j})^2 + \lambda \sum U_i^2 + \lambda \sum V_j^2 \tag{13}$$

Where $r(i,j)$ denotes the original rating of the user, $\hat{r}_{i,j}$ denotes the predicted value of the user's rating using this model, and λ is a regularization parameter that varies continuously with the operation.

The regularization terms of U and V are added to the objective function in Eq. (13), which in turn controls the complexity of the project. λ indicates the ability of this regularization term to constrain the model. If λ is too small, this constraint is not sufficient to reduce the complexity, and if λ is too large, it may indicate the loss of some important parameters, which in turn leads to a decrease in model accuracy. In the actual operation process, it is necessary to balance the degree of fit and accuracy of the model by constantly grid operations to find the appropriate λ.

3.4. Alternating Least Squares Method

In the recommendation system of matrix decomposition used, there are usually Stochastic Gradient Descent (SGD) [4] and Alternating Least Squares (ALS) [6] for the solution of the loss function (e.g., Equation (13)), and the alternating least squares (ALS) method is used in this paper to optimize the loss function, and the solution process of the alternating least squares method is described below. The alternating least squares method uses the idea of constantly performing further updates to solve the objective function. When updating a parameter to be updated, the control variables method is used to treat the other parameters as fixed values, so that the objective function is reduced to a quadratic equation, which in turn leads to an optimal solution.

Taking Equation (13) as an example, the potential factor matrix V is first randomly generated, and then the bias derivative is found for the user potential factor matrix, and the result is as follows Equation (14):

$$\frac{\partial L}{U_i} = -2V_j + 2\lambda U_i \tag{14}$$

According to the symmetry, the potential factor matrix U is fixed and the partial derivatives are obtained for the potential factor vectors of the users as in Equation (15):

$$\frac{\partial L}{V_i} = -2U_i + 2\lambda V_j \tag{15}$$

Let both equation (14) and equation (15) be 0. Iterative computation is performed by calculating the extreme values of the binary function to find the suitable and thus the desired objective function. Each complete traversal of the entire data set is called an iteration. There are generally three ways to end an iteration: the first is to set a threshold value and end the training when the desired objective function value is below the threshold, which is very simple and different from the operation, but is usually used rarely; The second method is: set a threshold value, calculate the absolute value of the difference between the objective function value of this iteration and the objective function value of the next iteration, and terminate the iteration if it is less than the threshold value; The third method is: fixing the number of iterations. In the usual experimental procedure, the iterations are usually performed by combining the second and third methods.

4. Summary and Outlook

In this paper, two recommendation techniques based on content and matrix-based decomposition are studied and analyzed for the personalized movie recommendation problem, and a hybrid recommendation scheme introducing content-based recommendations is proposed for the classical collaborative filtering algorithm data sparsity and item cold

start problems. Since the number of users in movie recommendation is generally much larger than the number of movies, the item-based collaborative filtering recommendation method is chosen. Also considering some characteristics of movie recommendation, several similarity measures used by the item-based collaborative filtering algorithm are improved in three aspects, namely, user rating characteristics, the number of users who jointly rate two movies and the time factor of user rating, in turn. The improvement of the recommendation algorithm in this paper is specific to the MovieLens dataset, but the idea of the improvement is somewhat general. When designing personalized recommendation systems for other items, we can also improve the similarity measure used by the collaborative filtering algorithm according to the characteristics of the domain.

In this paper, we propose a scheme to improve the item similarity measure according to the characteristics of the domain to which the recommended items belong in the study of recommendation systems, and we also propose the use of item attributes to solve the problems of data sparsity and item cold start in collaborative item filtering. However, the optimization of recommendation methods is endless, we can build user models and design user feedback mechanisms to discover users' interest preferences, or we can further use deep learning methods to train our recommendation models.

References

[1] Hua-fu Duan. Application of matrix decomposition techniques considering time effects in recommender systems[J]. Microcomputer applications, 2013, 29(03): 53-55+64.

[2] Liang-jun Liu, Liu Yang. Design and implementation of movie recommendation system[J]. Internet of things technology, 2021, 11(03): 86-88+92.

[3] Zong-yan Chen. Collaborative filtering recommendation algorithm for sparse data optimization[D]. Nanjing University of Posts and Telecommunications, 2017.

[4] Deuk Hee Park, Hyea Kyeong Kim, Il Young Choi, Jae Kyeong Kim. A literature review and classification of recommender systems research[J]. Expert Systems with Applications, 2012, 39(11).

[5] Yao-ning Fang, Yun-fei Guo, Xue-tao Ding, Ju-long Lan. An improved recommendation algorithm for singular value decomposition based on local structure[J]. Journal of Electronics and Information, 2013, 35(06): 1284-1289.

[6] Shi-chao Dai. Research on parallelization of matrix decomposition based on graph computation model [D]. Zhejiang Sci-Tech University, 2016.

[7] Li H., Liu Y., Qian Y. et al. HHMF: hidden hierarchical matrix factorization for recommender systems. Data Min Knowl Disc 33, 1548–1582 (2019).

[8] Dong-ting Sun. Research on Cold Start Problem in Collaborative Filtering Recommender System[D]. National University of Defense Technology, 2011.