A New Collaborative Filtering Algorithm based on Modified Matrix Factorization

Hanmin Ye
College of Information Science and Engineering
Guilin University of Technology
48177428@qq.com

Qiuling Zhang
College of Information Science and Engineering
Guilin University of Technolog
309867230@qq.com

Xue Bai
College of InformationScience and Engineering
Guilin University of Technology
824238570@qq.com

***Abstract*—Recommendation algorithm based on matrix factorization has a global presented objective function by optimization technology, in which singular value decomposition is a typical representative, but there are still some bottleneck restricting the further development of problems such as high-dimensional sparse problem. Concerning that problem, we propose an alternating least square based on singular value decomposition algorithm. Firstly, we fill the user-item rating matrix with each item's mean score. Secondly we use singular value decomposition to identify the best potential factor dimension and initialize the potential factor matrix of users and items. Thirdly we use alternating least squares to get the final potential factor matrix of users and items. Finally, we use the final potential factor matrix of users and items to recommend. The results on the Movielens datasets show that the proposed algorithm can effectively improve the recommendation accuracy so as to ease the high-dimensional data sparsity.**

Keywords—Alternating least square; SVD; Sparsity; Recommender algorithm; Matrix factorization.

Ⅰ. Introduction

Recommendation algorithm is an important tool to help users find the interested items and solve the problem of information overloaded. In recommendation algorithm based on matrix decomposition [1], the matrix of user behavior is decomposed into hidden factor space users, matrix of item's characteristic. Thus the algorithm has many advantages such as high accuracy, good extensibility, etc. But the traditional matrix decomposition algorithm still exists the problem of recommending effect not being ideal in high-dimensional sparse data [2].

In the study of recommendation algorithm, the number of users and projects are often tens of thousands or even millions of, the obtained user---each row of the project evaluation matrix represents a user, and each column represents an item. This matrix could be very sparse, namely the proportion of known scores is very low. We usually adopt sparse degree to measure the sparsity of data. The sparsity is defined as the percentage [3-4] of not scoring entities in the user evaluation matrix. For example, Movielens dataset [5] contains 943 users and 1682 items, but the scoring records are only 100000. If we adopt traditional measurement methods such as Pearson similarity, then the similarity of any two users is approximate to 0. That is to say, we can't distinguish them. The sparse degree is 22166*296277/922267≈0.006.

Traditional methods to solve the high-dimensional sparsity of data are dimensionality reduction and methods based on clustering. Literature [6] adopts the method of Principal Component Analysis (PCA) to conduct the process of dimensionality reduction on user-item score matrix. This can improve the sparsity of the data inputted. Literature [7] proposes a collaborative filtering recommendation algorithm based on clustering smooth joint to reduce the ill effects of data sparseness. But dimension reduction method loses part of user rating data [8] for the project, and the method based on clustering cannot reflect the interest difference between users [9], therefore recommend result accuracy and has not been significantly improved.

Alternating Least Squares (Alternating further Squares, ALS) [10] is proposed by Zhou Y, etc. are put forward in 2008, this method is often used in the recommendation algorithm based on matrix decomposition. For example, the matrix of user scoring the goods is decomposed into two matrices; one is the user preferences for goods implied characteristics of matrix, the other one is characteristics matrix implicitly contained in goods. In the process of decomposition of matrix, the missing of grading has been filled, that is to say, it can be based on the ratings to fill the optimal products recommended to the user.

Because the rating data has a large number of missing items, traditional matrix decomposition algorithm such as Singular Value Decomposition (SVD) exist serious data fitting problem in the treatment of data sparseness. While ALS can solve this problem very well. Therefore this paper proposes an algorithm called IMSA (Item Mean SVD Alternating Least Squares) which is a fusion of item mean filling and SVD decomposition algorithm of ALS algorithm to ease the ill effects caused by high-dimensional sparse nature of the data.

Ⅱ. The Related work

A. The Baseline Prediction

Some of the observed score data is the effect of the factors that is independent of the user or item, namely, that is part of the factors which has nothing to do with the use but only depends on the user or the characteristics of item itself [11]. For example, in the film, some users rate high for all movies, and the user may use 4 points to be a benchmark, and give 3 points to the dislike, and give score of 5 points to the favorite movie; Other users' score for all film may be low, these users may use a benchmark, and give only three points score to their favorite movies. Also, some films prefer to get a higher score than others (probably the movie plot, director, and actor). In this paper, the factors independent of the user or factors independent of objects are called Bias information [12]. A score joined prediction algorithm for the Bias is shown in formula (1):

$$\hat{R}_{ij} = \mu + bu(i) + bi(j) \tag{1}$$

There is the Baseline prediction, which represents the forecast score of user *i* to project *j*. μ is the overall bias information of datasetss. *bu*(*i*) represents biased information of the user *i*. *bi*(*j*) represents biased information of project *j* .

For example: to predict the scores of user User1 to film Movie1 , we assume that bias μ which is the overall movie datasets is Section IV, film Movie1's reputation 0.7 points lower than other films, Namely, the bi(Movie1) equals -0.7. On the other hand, the user User1 is a positive user, generally prefers to 0.4 points higher to the film, namely the bu (User1) = 0.4.Then the user predicted ratings of film Movie1 by User1 is 4.3 - 0.7 + 0.4 = 4 points.

In order to get the offset information, in practical applications, we often adopt the method of formula (2) to solve the value of bi and bu.

$$bi = \frac{\sum_{u \in R(i)} \left(R_{ui} - \mu\right)}{\lambda_1 + |R(i)|}$$
$$bu = \frac{\sum_{i \in R(u)} \left(R_{ui} - \mu - bi\right)}{\lambda_2 + |R(u)|} \tag{2}$$

Where u is a particular user, and i is a project. R(i) is user set having evaluated a project i ,R (u) represents project collection being evaluated by user u. λ1 and λ2 are the compression coefficient, needing to determine by experiment.

B. The Singular Value Decomposition

Singular value decomposition [13] is a kind of matrix decomposition technique; it profoundly reveals the internal structure of the matrix, proposed in 1990 by ScottDeerwester et al proposed as a method of matrix decomposition. Complementing methods of sparse matrix are various, but to judge whether a matrix completing method is good is to see if the disturbance of the method to matrix is minimal. It is generally observed that if the eigenvalues of the matrix after completion and eigenvalues of original matrix are similar, even if the disturbance is small, and the SVD method in the mathematics just can make the smaller disturbance of complemented matrix.

For a two-dimensional sparse matrix RS, if using the singular value decomposition algorithm to forecast the missing elements, firstly, you need to use a certain method to fill the new matrix RS, and we obtained a new matrix Rf. Then we use the SVD algorithm to decompose Rf into three matrices, as shown in formula (3):

$$R_f \approx U_{mm} S_{mn} V_{mm}^T \tag{3}$$

Rf is a matrix with m rows and n columns. Umm is left singular orthogonal matrix and Vnn is right singular orthogonal matrix. Smn is a diagonal matrix, every element on the diagonal ranges from big to small, called the singular values. Singular values can represent the proximity of the given matrix to the matrix whose rank is lower, and then take the Top-k dimensions of the diagonal matrix of Smn to form matrix S(k), calculating the sum and finally forecast according to missing values of the original matrix.

C. The Singular Value Decomposition

Singular value decomposition [13] is a kind of matrix decomposition technique; it profoundly reveals the internal structure of the matrix, proposed in 1990 by ScottDeerwester et al proposed as a method of matrix decomposition. Complementing methods of sparse matrix are various, but to judge whether a matrix completing method is good is to see if the disturbance of the method to matrix is minimal. It is generally observed that if the eigenvalues of the matrix after completion and eigenvalues of original matrix are similar, even if the disturbance is small, and the SVD method in the mathematics just can make the smaller disturbance of complemented matrix.

For a two-dimensional sparse matrix RS, if using the singular value decomposition algorithm to forecast the missing elements, firstly, you need to use a certain method to fill the new matrix RS, and we obtained a new matrix Rf. Then we use the SVD algorithm to decompose Rf into three matrices, as shown in formula (4):

$$R_f \approx U_{mm} S_{mn} V_{mm}^T \tag{4}$$

Rf is a matrix with m rows and n columns. Umm is left singular orthogonal matrix and Vnn is right singular orthogonal matrix. Smn is a diagonal matrix, every element on the diagonal ranges from big to small, called the singular values. Singular values can represent the proximity of the given matrix to the matrix whose rank is lower, and then take the Top-k dimensions of the diagonal matrix of Smn to form matrix S(k), calculating the sum and finally forecast according to missing values of the original matrix.

Ⅲ. Item Mean SVD Alternating Least Squares algorithms

In recommendation algorithm based on matrix decomposition, for user-item scoring matrix Rmn, the algorithm is designed to find two low dimensional matrices Ukm and Vkn, to approximate Rmn, as shown in formula (5) :

$$R_{mn} \propto U_{km}^T * V_{kn} \tag{5}$$

The Rmn represents the scoring matrix of user to goods, and Ukm represents the user preference matrix. Vkn represents the matrix whose entities are implicit characteristics of goods, with k << min (m, n). k is the dimension of implicit characteristic .

In order to make inner products of the sum of the low rank matrix be as much as possible close to the original matrix

Rmn, we need to minimize the squared error loss function, as shown in formula (6):

$$L(U,V)=\sum_{u,i}\left(R_{ui}-U_u^T*V_i\right) \tag{6}$$

Loss function usually needs to add the regularization item to avoid over fitting, and generally use the L2 regularization [15], as shown in formula (7):

$$L(U,V)=\sum_{u,i}\left(R_{ui}-U_u^T*V_i\right)+\lambda*\left(\left|U_u^T\right|^2+\left|V_i\right|^2\right) \tag{7}$$

The Uu is a characteristic vector of u, and the Vi is the characteristic vector the project i. λ is the regularization coefficient. Then the recommendation algorithm is transformed into an optimization problem, because the sum of variables coupling together.

Algorithm 1. IMSA algorithm

Input: we input user-items evaluation matrix R, indicating matrix R_L, potential factor dimension vector k vector, possible value vector lambda Vector of the regularization coefficient. Via SVD we get the potential factors of user's matrix U, potential factors of project matrix V, and error threshold RMSE. We also get the maximum iteration.

Output: The outputs are the dimensionality k of the optimal potential factor, the optimal regularization coefficient λ, the final user potential factors matrix U and project potential factor matrix V.

The basic steps of IMSA are as follows:

Step 1: calculate the indicating matrix R_L = R > 0 ;

Step 2: set score values sum = 0 and scoring record number num = 0;

Step 3: fill the original user - project scoring matrix R with itemMean;

for traverse the elements in Rmn according to columns do;

num is the number of nonzero elements of the column;

sum is the sum of nonzero elements of the column;

so the corresponding project scoring average of elements of the column is 1.0 * sum/num;

end for

Step 4:decompose the filled matrix R into U,V and k;

for traverse each element in k Vector kEach do

Define whether the indication of flag = false is a maximum number of iterations or root mean square error threshold, and the default is maximum number of iterations;

for loop iterates until maxIter do

[U,S,V] = svd(R,kEach);

select the submatrices corresponding to the first few singular values the size of which is kEach, and express the submatrices in S(kEach);

RMSE = norm((U*-R).*R_L,'fro')/sqrt(sum(R_L));

if the RMSE of the last time iteration and the next iteration are both bigger than RMSEThreshold

continue；

else

flag = true；

k= k；

break；

end if

if the number of iteration is maximum maxIter&&flag==false

k= k-1

end if

end for

Step 5: run ordinary ALS algorithm;

for traverse each element lambda of lambdaVector do

flag = false ;

for loop iterates until maxIter do fix v, and take partial derivative of L(U,V) with respect to v, making the derivative be equal to 0, and we get formula(8):

$$U_u=\left(V^T*V+\lambda I\right)^{-1}V^TR_u \tag{8}$$

λ as the regularization coefficient need to be determined by experiment.

fix v, and take partial derivative of L(U,V) with respect to u, making the derivative be equal to 0, and we get formula(9):

$$V_i=\left(U^T*U+\lambda I\right)^{-1}U^TR_i \tag{9}$$

Ru is the uth line of Rmn, and Ri is the ith column of Rmn, and I is a unit matrix with the size of k*k;

if the RMSE of the last time iteration and the next iteration are both bigger than RMSE threshold

continue

else

λ = lambda

flag = true

break

end if

end for

if the number of iteration is maximum maxIter&&flag==false

λ = lambda

end if

end for

End

Ⅳ. Experimental Analysis

In this paper we test the recommended quality of the proposed algorithm, and at the same time compare IMSA algorithm proposed in this paper with the classic itemMean algorithm, Baseline prediction algorithm, ALS algorithm, SVD algorithm.

itemMean: predict the unknown scores by using the score average of the score of each item respectively.

Baseline: predict the score according to the offset information.

ALS: predict the score by calculating alternately for the user and target eigenvector.

Iterative SVD: loop iteration SVD decomposition is to get the U and V, choosing the inner product of U and V to predict the score, when the minimum Root Mean Square Error (RMSE) reaches the minimum.

A. Experimental Datasets

In this paper, we use the famous movie scoring dataset MovieLens provided by the GroupLens research project team as test dataset. Use an integer ranging from 1 to 5 to represent user scoring value. The bigger the number is, the greater the user is interested in. we choose 100 000 scoring data as the initial dataset including 943 users and 1682 item scoring.

B. Experimental Environment Configuration

The experimental environment of the software and hardware as follows:

1) Hardware environment configuration: Intel core i3-4150 processor, dominant frequency 3.5 GHz, memory 4G, hard disk 500G.

2) software environment configuration: development tools MatlabR2014a, operation system win7 ultimate.

C. The experimental evaluation standard

RMSE as a kind of statistical precision index is by far the most commonly used kind of recommendation quality metrics. Because the deviation is processed by absolute value, it won't appear the situation of the positive and negative offset. We show the accuracy of prediction by calculating the RMS of error sum of squares between prediction user scoring and actual user scoring. The smaller RMSE is, the better the recommendation quality is. As shown in formula (10):

$$RMSE = \sqrt{\frac{\sum_{u,i}(\hat{R}_{ui} - R_{ui})^2}{N}} \tag{10}$$

$\hat{R}$ ui for prediction score, Rui for the actual score of test set, N for the number of data item in test set.

D. The experimental results and analysis

Below we use IMSA algorithm proposed in this paper to predict results in the MovieLens datasets, at the same time, we draw the RMSE three-dimensional plots under the random initialization output parameters of ALS algorithm, finally compared with classical algorithm, to verify the optimization results of different algorithms.

1) Baseline prediction

Use Baseline prediction result in the MovieLens datasets, the three-dimensional graph is shown Fig.1:

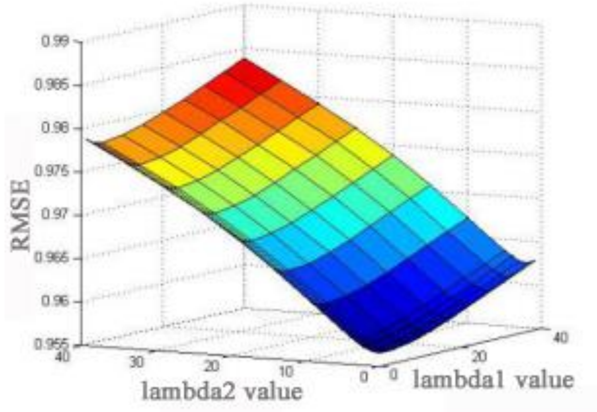

Fig.1 Baseline Prediction of Movielens Dataset

From Fig.1, we can conclude that as the change of λ 1 and λ 2, RMSE value presents a kind of parabolic rule, by the calculation, when λ 1 =1 and λ 2 =5, RMSE reaches optimum, and the minimum value is 0.9559.

2) Random initialization ALS

As shown in Fig. 2, it shows the change of RMSE along with the change of regularization factor λ and potential factor,

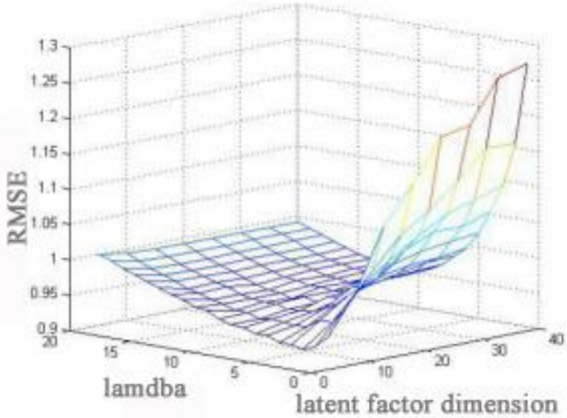

Fig.2 RMSE of Random initialization ALS on Movielens Dataset

We can see from the graph, with the regularization factor λ and potential factor dimension increasing, RMSE value overall is increased at first and then is reduced. By calculating, when λ =3, potential factor is 5, RMSE reaches optimum 0.91628.

3) Iterative SVD algorithm:

As shown in Fig. 3, it shows the change curve of RMSE along with the change of the number of iterations, by using SVD algorithm in the Movielens datasets. Choose the potential factor dimension (value 5). We can see that the iterative SVD begins overfitting from the third training, and the minimum value of RMSE is 0.9158.

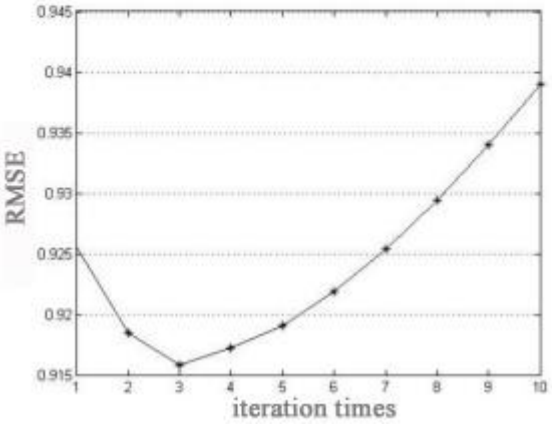

Fig.3 RMSE of Iterative SVD on Movielens Dataset

4) The RMSE of different algorithms

In order to validate the RMSE optimization results of IMSA algorithm proposed in this paper, which compares with Baseline, ALS, SVD, we test the three algorithms in the MovieLens datasets. As shown in Fig. 4, we can see from the graph that the value after ALS optimal is improved more obviously than the Baseline value, IMSA algorithm compared with ALS, iterative SVD algorithm, further improve the precision, and the precision of IMSA algorithm compared with ALS algorithm increases by 1.88‰, 1.56‰ increased compared with the iterative optimization of iterative SVD.

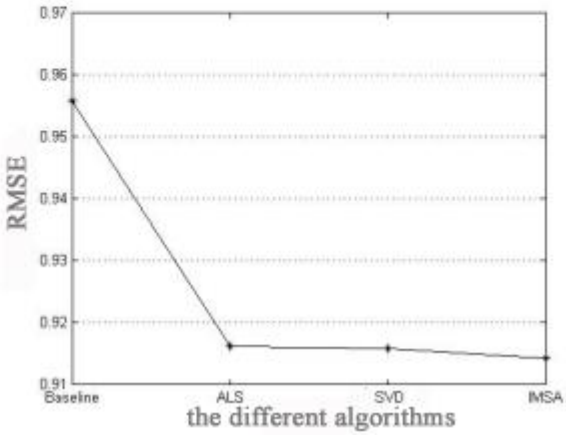

Fig.4 RMSE of Different Method on Movielens Dataset

Ⅴ. Conclusion

The paper focuses on the research of the effect of the high-dimensional sparsity of data on the quality of the recommendation in the personalized recommendation algorithm. Use the ALS algorithm optimized by SVD to predict score, decrease the sparse degree of the datasets, so as to improve the commendation quality of personalized recommendation algorithm. But how to make use of more information such as user social network information, the classification information of projects is worth to continue to study.

References

[1] Liang D, Charlin L, McInerney J, et al. Modeling User Exposure in Recommendation[J]. arXiv preprint arXiv:1510.07025, 2015.

[2] Abdillah O, Adriani M. Mining User Interests through Internet Review Forum for Building Recommendation System[C]// Advanced Information Networking and Applications Workshops (WAINA), 2015 IEEE 29th International Conference on. IEEE, 2015:564-569.

[3] Guan H, Li H, Xu C Z, et al. Semi-sparse algorithm based on multi-layer optimization for recommendation system.[J]. Journal of Supercomputing, 2012, 66(3):148-155.

[4] Xu K, Yuan C, Li F, et al. An Expected Item Bias based Hybrid Approach for Recommendation System[J]. International Journal of Database Theory & Application, 2014, 7.

[5] Sun W, Zhang X, Liang W, et al. High Dimensional Explicit Feature Biased Matrix Factorization Recommendation[M]// Trends and Applications in Knowledge Discovery and Data Mining. Springer International Publishing, 2015.

[6] Andriy Mnih and Ruslan Salakhutdinov, Probabilistic Matrix Factorization, Proc. of NIPS 2007, Vancouver, Canada, December, 2007, pp.1257-126

[7] Yin C X, Peng Q K. A careful assessment of recommendation algorithms related to dimension reduction techniques[J]. Knowledge-Based Systems, 2012, 27: 407-423.

[8] Shepitsen A, Gemmell J, Mobasher B, et al. Personalized recommendation in social tagging systems using hierarchical clustering[C]//Proceedings of the 2008 ACM conference on Recommender systems. ACM, 2008: 259-266.

[9] Zhou Y, Wilkinson D, Schreiber R, et al. Large-scale parallel collaborative filtering for the netflix prize[M]//Algorithmic Aspects in Information and Management. Springer Berlin Heidelberg, 2008: 337-348.

[10] Celma Ò, Cano P. From hits to niches?: or how popular artists can bias music recommendation and discovery[C]//Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition. ACM, 2008: 5.

[11] Deerwester S C, Dumais S T, Landauer T K, et al. Indexing by latent semantic analysis[J]. JAsIs, 1990, 41(6): 391-407.

[12] Liang D, Charlin L, McInerney J, et al. Modeling User Exposure in Recommendation[J]. arXiv preprint arXiv:1510.07025, 2015.