Universe Scientific Publishing

Improved SVD + + Recommendation Algorithm Based on Fusion Time Factor

Wentao Zhao, Ziheng Cui, Tingting Feng

College of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 454003, China.

Abstract: Collaborative filtering algorithm is widely used in recommendation system. Aiming at the problems of data sparsity and low recommendation accuracy in traditional collaborative filtering algorithm, an improved recommendation algorithm is proposed PT _ SVD++. Firstly, the attribute information of users and the implicit feedback information of items are introduced to improve the SVD++ algorithm, which solves the insufficient utilization of information and alleviates the problem of sparse data; Secondly the time effect model is established to further improve the accuracy of the prediction results. The experimental results on MovieLens dataset show that compared with other algorithms, the average absolute error and root mean square error of this algorithm are lower, and its recommendation accuracy is higher.

Key words: SVD + + algorithm; Time effect; Recommendation algorithm

1. Introduction

In the process of Internet and information technology rapid development, from the beginning information blocks seriously, now entered the era of impact is the Internet a lot of news every day, every day many users in the face of vast amounts of information, usually by people around or system is recommended to make a choice, but in the face of the endless commodity information and resources, the user will see the dazzling, Therefore, it is very difficult to select the best result. In order to facilitate users to make appropriate choices in this massive amount of information, recommendation system is launched to solve the problem of "information overload" [1-2]. In collaborative filtering, neighborhood based[3] method and cryptic model are better methods[4], but the cryptic model based on matrix decomposition has better accuracy and stability in prediction. It is based on matrix decomposition, and SVD is the representative of the implicit factor model.

2. SVD++ model

Classical SVD algorithm and some deformation of SVD algorithm are based on explicit feedback research, and SVD++ algorithm model which considers both explicit and implicit user feedback [5] is generated accordingly. Explicit feedback refers to that users actively or under certain prompts and guidance give scores, ratings or language evaluations on the subjective use or experience of a recommended object or service according to their own preferences, so as to clearly express user experience and preference. Implicit feedback refers to the data of all users' operation behaviors recorded by the system without users' knowledge, including observing whether users click on some recommended objects, the number of clicks, the depth of clicks, the browsing duration, and whether they are favorites. SVD++ adds the implicit feedback of users into the model, so the scoring prediction function of SVD++ algorithm is defined as:

$$r_{ui}' = \mu + b_u + b_i + (p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j) q_i^T \qquad \text{\\* MERGEFORMAT (1)}$$

Which p_u is the matrix, on behalf of the user characteristics; $N(u)$ representing with behavior items collection; $N(u)^{-\frac{1}{2}}$ is the canonical factor; $|N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j$ is a characteristic vector, on behalf of the implicit feedback; y_j on behalf of the user implicit preference for goods j , is a recessive factor dimension of the same dimension and vector, all component indicates the degree of hidden factors like commodity composition.

According to the scoring prediction function, the optimization objective function can be obtained as follows:

$$\min \sum_{(u,i)\in T} (r_{ui} - \mu - b_u - b_i - (p_u + |N(u)|^{\frac{1}{2}} \sum_{j\in N(u)} y_i) q_i^T)^2 + \lambda(b_u^2 + b_i^2 + p_u^2 + q_i^2 + y_j^2)$$

* MERGEFORMAT (2)

3. Collaborative filtering algorithm based on time effect and improved SVD++

3.1 Improved SVD++ algorithm

SVD++ is a collaborative filtering recommendation algorithm. Based on matrix decomposition and score prediction, it combines explicit feedback with implicit feedback to optimize the deficiency of only explicit feedback. Due to the particularity of implicit feedback, the value of missing value is not taken into account, and the influence of user's own attributes and the attributes of the recommended object's own characteristics is ignored. In actual evaluation, in addition to different preferences of users, different attributes of goods will also affect the score. Therefore, an improved SVD++ (P_SVD++) algorithm was proposed, which added user attribute information and implicit feedback information of the project on the basis of SVD++. The improved formula is as follows:

$$r_{ui}' = \mu + b_u + b_i + (p_u + |N(u)|^{-\frac{1}{2}} \sum_{j\in N(u)} y_j + \sum_{a\in A(u)} y_a)(q_i + |H(i)|^{-\frac{1}{2}} \sum_{j\in H(i)} g_j)^T$$

* MERGEFORMAT (3)

Where $A(u)$ is the attribute set of user u ; y_a is the user attribute vector; $H(i)$ is the implicit feedback set of the project, $|H(i)|^{-\frac{1}{2}} \sum_{j\in H(i)} g_j$ the implicit project feedback feature matrix, g_i is the project feature vector.

Consider using objective function optimization:

$$\min \sum_{(u,i)\in T} (r_{ui} - r_{ui}')^2 * \lambda(\mathrm{b}_u^2 + b_i^2 + p_u^2 + q_i^2 + y_j^2 + y_a^2 + g_j^2)$$

* MERGEFORMAT (4)

3.2 Time factor integration

The baseline prediction method mainly has two time effects: the time effect of user bias (scoring scale) and the time effect of item bias (product popularity). User bias means that users' tastes may change over time and their baseline rating will change. For example, a movie that a user once gave three stars might now be given four. Item bias refers to the idea that the popularity of an item changes over time, and changes in the popularity of a movie can be triggered by external events, such as actors (known or unknown) appearing in a new movie. So the time factor is introduced into the improved model. On day t_{ui} , user u scores item i as follows:

$$r_{ui} = \mu + b_u(t_{ui}) + b_i(t_{ui})$$

* MERGEFORMAT (5)

Where $b_u(t_{ui})$ is user bias and $b_i(t_{ui})$ is item bias, both are real functions that change over time.

For all users, the mean of scoring time is represented by t_u . At present, if users evaluate the movie on the t day, the time deviation related to scoring is defined as follows

$$dev_u(t) = sign(t - t_u) \bullet |t - t_u|^\beta$$

* MERGEFORMAT (6)

Where $|t - t_u|$ measures the time distance (for example, days) between dates t and t_u .

At some time in our lives, may be a sudden unexpected situations, including in the film score data set, the user evaluation movie, one day to basic comparison unified grading, the evaluation of such behavior occurs only on a certain day, reflect the user's mood that day, perhaps a user rating affected him, perhaps is it change the rating criteria, In order to deal with the influence of this short time, set parameters for all users and every day to respond to the change of the situation on a particular day. Set this parameter as $w_{u,t}$, and the definition of item deviation can be obtained:

$$b_i(t) = b_i + b_{i,bin(t)} + w_{u,t}$$

* MERGEFORMAT (7)

At the same time, a new parameter named α_u is introduced for each user, and the day t is associated with an integer $bin(t)$, so the definition of user deviation can be obtained:

$$b_u(t) = (b_u + \alpha_u \bullet dev_u(t))z(t_{ui})$$

* MERGEFORMAT (8)

$$z_i(t_{ui}) = z_i + z_{i,t}$$

* MERGEFORMAT (9)

Where $z_i(t_{ui})$ represents the extended feature related to time and user bias, z_i and $z_{(i,t)}$ represent the part that is stable and changes with time, respectively; $w_{u,t}$ 和 $z_i(t_{ui})$ represent the situation that the prediction is unstable due to the introduction of time factor.

The user and item bias are represented by $b_u(t)$ and $b_i(t)$ respectively, both of which include two parts that change with time and are stable. The stochastic gradient descent algorithm is used to obtain the squared error function value to complete the learning process and import the time effect. The recommendation model is established in the PS_SVD++ algorithm. Then user u evaluates item i and the final prediction score formula is as follows:

$$r_{ui}^{'} = \mu + b_u(t_{u,i}) + b_i(t_{u,i}) + (p_u(t) + |N(u)|^{-\frac{1}{2}} \sum_{j\in N(u)} y_j + \sum_{a\in A(u)} y_a)(q_i + |H(i)|^{-\frac{1}{2}} \sum_{j\in H(i)} g_j)^T$$

* MERGEFORMAT (10)

4. Experiment

4.1 Evaluation Indicators

In the field of recommendation system, literature [6-7] summarizes various evaluation indicators. Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) are selected in this paper to measure the accuracy of prediction. The formula is as follows:

$$MAE = \frac{1}{|Test|} \sum_{r_{ui}\in Test} |r_{u,i} - r_{u,i}'|$$ * MERGEFORMAT (11)

$$RMSE = \sqrt{\frac{1}{|Test|} \sum_{r_{ui}\in Test} (|r_{u,i} - r_{u,i}'|)^2}$$ * MERGEFORMAT (12)

Where $r_{u,i}$ 、 $r_{u,i}'$ represent the actual score and predicted score respectively, and Test represents the Test set. MAE is used to calculate the average difference between the predicted value and the real value in the test set. The smaller MAE value is, the more accurate the prediction of the algorithm is. RMSE is used to calculate the square root of the square sum of the mean of the prediction error in the test set. Similar to MAE, the smaller RMSE value is, the closer the prediction score is to the true bisection, and the higher the prediction accuracy of the algorithm is.

4.2 Experimental results and analysis

4.2.1 Analysis of prediction accuracy results

In this experiment, algorithms with different iterations were tested on data sets ML-100K and ML-1m respectively.

Regularization coefficient λ =0.003, factor dimension f =20, and learning rate parameters were set as γ_0 =0.05, γ_1 =1.2, γ_2 = 0.01. Figure 2 and Figure 3 respectively show the prediction accuracy results of each algorithm under mL-100K and ML-1m data sets. It can be seen from the figure that MAE and RMSE values of PT_SVD++ algorithm are smaller than those of the other three algorithms under different iterations. It can be seen from Figure 2 that in the MLA-100K data set, the MAE value of PT_SVD++ algorithm fluctuates around 0.69, and the RMSE value fluctuates around 0.86. When the number of iterations is 50, the MAE of PT_SVD++ algorithm model is the smallest, and the MAE value is about 0.6814. When the number of iterations is 30, the RMSE of PT_SVD++ algorithm model is the smallest, and its value is about 0.8607.

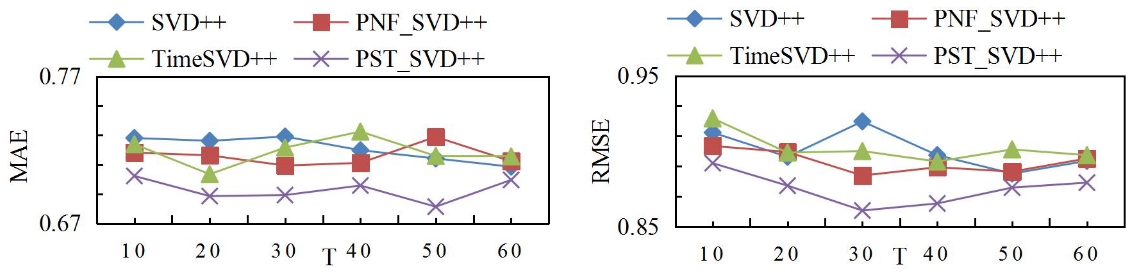

Fig.1 MAE and RMSE in ML-100K dataset

It can be seen from Figure 3 that the MAE value of PT_SVD++ algorithm fluctuates around 0.68 and the RMSE value fluctuates around 0.87 in the M1-1M dataset. When the number of iterations is 40, the MAE of PT_SVD++ algorithm model is the smallest, and the MAE value is about 0.6856. When the number of iterations is 20, the RMSE of PT_SVD++ algorithm model is the smallest, and its value is about 0.8743. It can be seen from FIG. 2 and 3 that compared with PT_SVD++, the values of the other three algorithms are unstable, and their MAE and RMSE values are large. Therefore, it can be shown that PT_SVD++ can effectively improve the prediction accuracy of the algorithm.

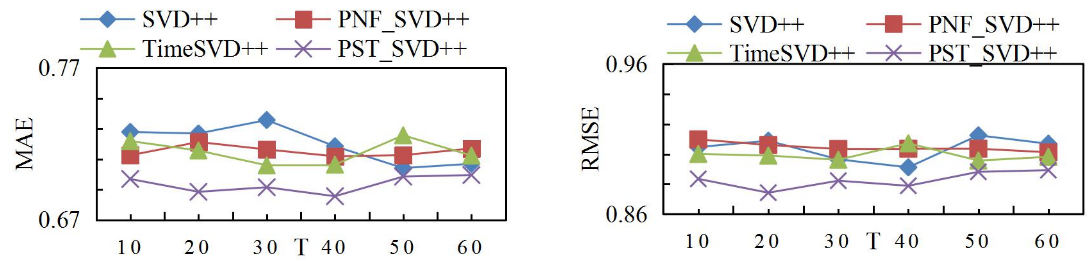

Fig.2 MAE and RMSE in ML-1M dataset

5. Summarizes

This paper proposes an improved recommendation algorithm, which is based on SVD + + model and adds implicit feedback and user attribute information to improve efficiency. Based on time effect modeling, the cost function is optimized by random gradient descent method to make the function value smaller to the maximum extent, and then improve the accuracy of prediction. Through experiments on movielens data sets, the sizes of these data sets are different. From the results, we can know that the method used in this paper can greatly improve the calculation efficiency, and will not affect the accuracy of prediction. At the same time, it can better improve the sparse problem of the scoring matrix.

References

[1] Ren YG, Zhang YP, Zhang ZP. Collaborative filtering recommendation algorithm based on rough set rule extraction[J]. Journal on Communications, 2020,41(1):76-83.

[2] Agrawal S, Jain P.An improved approach for movie recommendation system[C]// International Conference on

loT in Social, Mobile, Analytics and Cloud (I-SMAC), Palladam, 2017:336-342.
[3] Huang LW, Jiang BT, Lyu SY, et al. Survey on deep learning based recommender systems[J]. Chinese Journal of Computer, 2018, 41(7):1619-1647.
[4] Zhang KH, Liang JY, Zhao XW, et al. A collaborative filtering recommendation algorithm based on community expert information [J], Computer research and development, 2018, 55 (5): 968-976.
[5] Mei LX, Yu X. Representation learning method for implicit feedback recommendation system [J], Computer application research, 2020, 37(8): 2266- 2272.
[6] Zhu AY, Ren XJ. Regularized recommendation model based on user trust and score bias [J]. Modern computer, 2018, 615(15):30-35.
[7] Kumar R,Verma BK, Rastogi SS. Social Popularity based SVD++ Recommender System[J]. International Journal of Computer Applications, 2014, 87(14):33-37.